Lecture Notes in Electrical Engineering 1095

Gayatri Mehta
Nilmini Wickramasinghe
Deepti Kakkar *Editors*

Innovations in VLSI, Signal Processing and Computational Technologies

Select Proceedings of the 2nd International Conference, WREC 2023

Springer

Lecture Notes in Electrical Engineering

Volume 1095

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

Gayatri Mehta · Nilmini Wickramasinghe ·
Deepti Kakkar
Editors

Innovations in VLSI, Signal Processing and Computational Technologies

Select Proceedings of the 2nd International Conference, WREC 2023

Springer

Editors
Gayatri Mehta
Department of Electrical Engineering
University of North Texas
Denton, TX, USA

Nilmini Wickramasinghe
La Trobe Univeristy
Melbourne, Australia

Deepti Kakkar
NIT Jalandhar
Punjab, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-7076-6 ISBN 978-981-99-7077-3 (eBook)
https://doi.org/10.1007/978-981-99-7077-3

Contents

Effect of Lithium Dopant on Stanene Nanotube's Properties 1
Kanika Sharma, Gurleen Kaur Walia, and B. C. Choudhary

SOT-MRAM Memories for Energy Efficient Embedded and AI Applications 13
Inderjit Singh, Balwinder Raj, and Mamta Khosla

A Comprehensive Review to Investigate the Effect of Read Port Topology on the Performance of Different 7 T SRAM Cells 25
Bhawna Rawat and Poornima Mittal

A Comparative Performance Analysis of 10 T and 11 T SRAM Cells . . . 35
Puneet Yadav and Poornima Mittal

VLSI Floorplan Area Optimisation Technique 47
Mithilesh Kumar Lobiyal and Sukwinder Singh

NG-PON3-Based Bidirectional High-Speed PON with 1:32/64/128 Split Ratio 55
Meet Kumari

All-*p*-Type Digital Circuits Using Single Gate and Double Gate Organic Field Effect Transistors 61
Vidhi Goswami, Brijesh Kumar, and Richa Yadav

Rotor Unbalance Severity Detection Using Maximum Overlap Discrete Wavelet Transform 73
Sonalika Bhandari, Sachin Taran, and Varun Sangwan

Self-attention-Based Deep Learning Approach for Machine Translation of Low Resource Languages: A Case of Sanskrit-Hindi 85
Nandini Sethi, Amita Dev, and Poonam Bansal

Reinforcement Learning Method for Identifying Health Issues for People with Chronic Diseases 93
Sunita Dhote, Michaelraj Kingston Roberts, and K. P. Sridhar

Metrics Evaluation of Bell Pepper Disease Classification Using Deep Convolutional Neural Network (DCNN) 103
K. Sowmiya and M. Thenmozhi

Analysis of Heart Disease Prediction Using Various Machine Learning Algorithms 115
G. Sakthipriya, Y. Suresh, C. Varnisha, R. Sindhu, and R. Shivraj

Analysis of Agricultural Commodities Prices Using BART: A Machine Learning Technique 123
Eva Mishra and R. Murugesan

Deep Learning Model-Based Approach for Agricultural Crop Price Prediction in Indian Market 133
Eva Mishra, R. Murugesan, and Deba Prasad Dash

Enhanced Intracranial Tumor Strain Prediction and Detection Using Transfer and Multilevel Ensemble Learning 147
Premanand Ghadekar, Ajinkya Mahajan, Aditya Bodhankar, Diksha Prasad, Shivani Mahajan, and Riya Dhakalkar

Detection and Classification of Blood Cancer Using Deep Learning Framework 159
Manish Prajapati, Santos Kumar Baliarsingh, Jhalak Hota, Prabhu Prasad Dev, and Shuvam Das

Deep Learning-Based Multi-label Image Classification for Chest X-Rays 167
Maya Thapa and Ravreet Kaur

Fine-Tuning the Deep Learning Models Using Transfer Learning for the Classification of Lung Diseases from Chest Radiographs 175
H. Mary Shyni and E. Chitra

A Systematic Approach for Effective Apgar Score Assessment in 1 and 5 min Using Manifold Machine Learning Algorithms 183
S. Shaambhavi and G. Bhavani

Data Pre-processing Techniques for Brain Tumor Classification 195
Neha Bhardwaj, Meenakshi Sood, and S. S. Gill

A Novel Approach for Detection of Lumpy Virus 205
Veena Kumari, Sahil Sandip Thorat, Sree Pravallika Pulikallu, Nayansi, Anmol Kaur, Tanvi Kumari, and Amritpal Singh

Internet of Healthcare Things-Enabled Open-Source Non-invasive Wearable Sensor Architecture for Incessant Real-Time Pneumonia Patient Monitoring 217
K. M. Abubeker, S. Baskar, and Michaelraj Kingston Roberts

VANET Security Optimization with Blowfish Algorithm and Adversarial Transfer Learning 227
Richa Singh and Deepti Kakkar

LoRa-IoT-Based Smart Weather Data Acquisition and Prediction for PV Plant Using Machine Learning 239
S. Ramalingam, M. Nagabushanam, H. S. Gururaja, and K. Baskaran

Entity Perception Using Remotely Piloted Aerial Vehicle 253
J. Jeba Emilyn, S. Sri Chandrika, T. Susma, S. Vinisha, and S. V. Yesvantini

RF-MEMS SPDT Capacitive Switch: Accelerating the Performance in B5G Applications 263
Raj Kumari and Mahesh Angira

Investigation of Bulk, Electronic and Transport Properties of Armchair Silicene Nanoribbon as Liquefied Petroleum Gas Combustion Indicator: A DFT Study 273
Jaskaran Singh Phull, Harmandar Kaur, Paramjot Singh, Gurleen Kaur Walia, and Deep Kamal Kaur Randhawa

Recent Advancement of Artificial Intelligence in COVID-19: Prediction, Diagnosis, Monitoring, and Drug Development 289
Priya Rachel Bachan, Udit Narayan Bera, and Prakriti Kapoor

Analysis of Corner Truncated Rectangular Microstrip Patch Antenna for IoT Applications 303
Sonam Gour, Reena Sharma, and Gaurav Sharma

Dual-Polarized Textile Antenna for Full-Duplex Wearable Applications 311
Neha Dalakoti and Priyanka Jain

Low-Power High-Speed Modified Three-Stage Comparator with Low Kickback Noise 321
T. Thamaraimanalan, S. Ramalingam, S. Dhanasekaran, and K. Baskaran

Analysis of the Electronic and Optical Properties of MoS_2 and WSe_2 Using the First-Principle Density Functional Theory 333
Emarishi, Reshma Sinha, and Jasdeep Kaur

Harmony Search-Based Fractional-Order Chebyshev Filter 343
Ashu Soni, Namrata Pandey, and Maneesha Gupta

A Neural Network-Based Wavelet Thresholding Technique for De-Speckling of Ultrasound Images 351
Mayank kumar Singh, Indu Saini, and Neetu Sood

Artificial Neural Network (ANN)-Based Supervised Control of Single Wheel Robotic System (SWRS) 359
Ashwani Kharola, Ayush Krishali, Prateek Gurung, and Prince Kumar Jha

An Approach to the Classification of Ancient Indian Scripts Using the CNN Model 367
Muskan Agarwal, S. Indu, and N. Jayanthi

Addressing Vehicle Safety and Platooning Using Low-Cost Object Detection Algorithms 379
Prathmesh Sharma, Priti Gangwar, Ritik Gupta, and Poornima Mittal

Facial Appearance Discerning Using Convolutional Neural Networks 387
Kushal Kashyap and Vijay Kumar

Deep Learning-Based Multi-state Colorectal Cancer Histological Image Classification 395
Hardeep Kaur, Anil Kumar, and Varinder Kaur Attri

Transcription of American Sign Language (ASL) Using Convolutional Neural Networks (CNNs) 407
Nishu Chaudhary, Ridham Sheel, Ranjan Kumar, and Poornima Mittal

Analysis of Greedy, Semi-greedy, and Random Scheduling Heuristics with DVFS for Heterogeneous Fog Computing Platform 419
Savina Bansal, Rakesh K. Bansal, and Nikita Sehgal

The Effect of Ground Temperature on the Operating Temperature of Solar Photovoltaic Module Using FEM Analysis 431
Prakash Pratik and Kunal Ghosh

A Descriptive Analysis of Different Dual-Port and Single-Port 11T SRAM Cells for Low-Voltage Operations 439
Yogita Chopra and Poornima Mittal

ASIC Design of High-Performance MIPS Processor Using Aprisa 447
Naga Surya Manikanta Talla and Prakash Kodali

A Proficient Multi-level Data Analytic Suite for Ascertaining Preliminary Gestational Hazards Associated with Its Influences 457
G. Bhavani and C. Jeyalakshmi

CO-OFDM System with Self-Organizing Map (SOM) for Improving the Spectral Efficiency by the Compensation of Phase Noise 469
Gurpreet Kaur and Gurmeet Kaur

Intelligent Car Damage Detection System 477
Ved Mistry, Chirag Jagad, Umang Jhunjhunwala, and Prachi Tawde

An Efficient Malicious URL Detection Approach Using Machine Learning Techniques 485
Swati Chaudhari, Archana Thakur, and Alpana Rajan

A Cluster-Based Routing Protocol for Wireless Sensor Networks-Based IoT System 497
Irak Rigia, Rajdeep Chakraborty, Bhaskar Bhuyan, and Hiren Kumar Deva Sarma

A Novel Approach for Deep Learning Based Video Classification and Captioning using Keyframe 511
Premanand Ghadekar, Vithika Pungliya, Atharva Purohit, Roshita Bhonsle, Ankur Raut, and Samruddhi Pate

Classification of Image with Convolutional Neural Network and TensorFlow on CIFAR-10 Dataset 523
Gunjan Sharma and Vijay K. Jadon

A Comparative Inspection and Performance Evaluation of Distinct Image Fusion Techniques for Medical Imaging 537
Harmanpreet Kaur, Renu Vig, Naresh Kumar, Apoorav Sharma, Ayush Dogra, and Bhawna Goyal

Early Detection and Classification of Waterlogging Stress in Broccoli Plants Prior to Visual Symptom Appearance Through Electrophysiological Signal Analysis 551
Kavya Sai, Neetu Sood, and Indu Saini

Detection of Parkinson's Disease Based on Biological Features Using Deep Neural Models 563
Nikita Aggarwal, Barjinder Singh Saini, and Savita Gupta

EEG-based Binary Classification of Brain State of Activities Level Using a Single-Sensor Headset 573
Rakesh Kumar Rai and Dushyant Kumar Singh

Text Analysis for Information Retrieval Using NLP 585
Kuldeep Vayadande, Harshal Baru, Abhishek Kashid, Aniket Kulkarni, Prathamesh Londhe, and Atharv Vanjari

About the Editors

Gayatri Mehta is Professor in the Department of Electrical Engineering at the University of North Texas (UNT), USA. She received her Ph.D. in electrical and computer engineering from the University of Pittsburgh in 2009. Her research interests are broadly in electronic design automation, reconfigurable computing, low-power VLSI design, system-on-a-chip design, embedded systems and portable/wearable computing. Dr. Mehta is the director of the Reconfigurable Computing Lab at UNT. She has received the UNT College of Engineering Faculty Service Award in 2023, the UNT TAMS and Honors College Excellence in Undergraduate Mentoring Award in 2021 and the UNT College of Engineering Research Award in 2017. She has also received the IEEE-HKN C. Holmes MacDonald Outstanding Teaching Award in 2013. She has designed an interactive mapping game UNTANGLED to uncover human mapping strategies. UNTANGLED received the People's Choice Award in the Games and Apps category of the 2012 International Science and Engineering Visualization Challenge conducted by the Science and National Science Foundation.

Nilmini Wickramasinghe as of August 1, 2023, is Professor and Optus chair of Digital Health at La Trobe University within the School of Engineering. She also holds honorary research professor positions at the Peter MacCallum Cancer Centre, MCRI, Epworth HealthCare and Northern Health. After completing five degrees at the University of Melbourne, she completed Ph.D. studies at Case Western Reserve University and later executive education at Harvard Business School, USA, in Value-based HealthCare. For over 20 years, she has been actively researching and teaching within the health informatics/digital health domain with over 350 scholarly publications, a patent, 25 books, numerous posters and book chapters and a very successful grant funding portfolio. In 2020, she was awarded the prestigious Alexander von Humboldt award for her outstanding contribution to digital health.

Deepti Kakkar pursued her Bachelor of Technology in Electronics and Communication Engineering from Himachal Pradesh University, India, in 2003 and her Master of Engineering in Electronics Product Design and Technology from Punjab University,

India. Dr. Kakkar obtained her Ph.D. in spectrum sensing in cognitive radios from Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, India. She has a total academic experience of 18 years and is currently an associate professor in the Electronics and Communication Department with Dr. B. R. Ambedkar National Institute of Technology. She has guided more than 50 postgraduate engineering dissertations and several projects. She is currently guiding three Ph.D. scholars, and one Ph.D. has been submitted under her guidance. She has more than 50 papers in various international journals and conferences.

Effect of Lithium Dopant on Stanene Nanotube's Properties

Kanika Sharma, Gurleen Kaur Walia, and B. C. Choudhary

Abstract A two-probe system is created to compare pure and lithium-doped stanene nanotubes using non-equilibrium Green's function and density functional theory. In order to better understand the numerous transport phenomena occurring in these devices, a number of electronic parameters, including band structure, the density of states, I–V curve, differential conductance, and transmission spectra, are examined at discrete bias voltage. Stanene nanotubes are shown to be direct band semiconductors, with band gaps narrowing as the number of dopants increased, indicating a boost in conductivity. When the bias voltage is greater than 0.6 V, negative differential resistance (NDR) is seen in the devices with 2 and 3 dopant atoms. NDR is brought on by the transmission coefficient's variation with the bias voltage since the system's transport characteristics are sensitive to the applied bias voltage. It can be concluded that a device with three dopant atoms has better conduction behaviour. These inferences offer thorough information about the use of stanene nanotubes in upcoming nano-scale applications.

Keywords Stanene · Dopant · DFT · Lithium

1 Introduction

Numerous research communities recognized the advancements in science and technology following the discovery of graphene, a 2D mono-elemental carbon. With the discovery of graphene, a fascinating new area of study in the biological-dimensional world was created, and a new family of two-dimensional (2D) materials received considerable attention. Ultrahigh charge carrier mobility (Hu et al. 2014), bizarrely

K. Sharma · G. K. Walia (✉)
School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, Punjab, India
e-mail: gurleen.24800@lpu.co.in

B. C. Choudhary
National Institute of Technical Teachers Training and Research (NITTTR), Chandigarh, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_1

high strength and flexibility (Zhao et al. 2015), high optical transparency (Akturk et al. 2016), ultralow weight (Garcia et al. 2011), enormous specific surface area (Abbasi 2019a), and high thermal conductivity (Chen 2020) are just a few of graphene's remarkable characteristics. It can also be produced at scale (Abbasi 2019b). Due to its extraordinary properties, graphene is the first two-dimensional (2D) material to be of increasing importance in research (Sardroodi 2018a), and its use has been found in a variety of fields, including biomedicine (Ma et al. 2021; Yaghoobi 2018) and advanced electronics (Sardroodi 2018b). Graphene is a successful material, but its zero band gap restricts its use in the semiconductor sector (Ai et al. 2020). To make up for this deficiency, scientists have worked hard to investigate the characteristics of honeycomb-like structures made of elements in the I–V group like silicene (Si), germanene (Ge), and stanene (Sn) (Bhuvaneswari et al. 2018; Fadaie et al. 2018; Hongyana et al. 2020). Stanene has a higher conductivity at room temperature than silicene and germanene (Li et al. 2020; Sharma et al. 2021), which likely creates novel opportunities for modifying its electronic properties.

Stanene is a one-layer, or two-dimensional, material (DOE/SLAC National Accelerator Laboratory 2013; Liuet al. 2011; Xu et al. 2013). Similar to graphene, it is made up of tin atoms arranged in a single hexagonal layer. Its name combines the Latin word for tin, stannum, with the graphene suffix, -ene. Researchers at the Indian Institute of Technology Bombay were the first to describe the synthesis and examination of the optical properties of stanene (Gao et al. 2016). A second team reported the molecular beam epitaxy-based synthesis of stanene on a bismuth telluride substrate in 2015. (Gao et al. 2016). According to Modarresi et al., the electronic band structure of stanene can have a suitable energy gap opened up by applying external strain (Modarresi et al. 2015). Li et al. modified the two-dimensional stanene's electronic structure and magnetic properties through hydrogenation (Li and Zhang 2014). In doped materials, topological superconductivity is time-reversal invariant (Wu et al. 2014). Doped stanene may therefore be a promising material for a variety of uses. In order to learn more about the electronic properties, this work takes into account the band structure, density of states, transmission spectrum, differential conductivity, and current–voltage characteristics of pristine and doped armchair stanene nanotubes.

2 Modelling and Methodology

The geometry optimizations are performed with nanotubes rolled up from a graphitic single layer and labelled by (n, m). In this work, we did calculations for armchair (n, n). Pristine armchair stanene nanotube with $n = 5$ as prototype is formed with bond length of 2.890 Å. For the transport properties of stanene nanotube, the two-probe configuration is formed by positioning a central region (C) amid two electrodes, left (L) and right (R). We have considered the electrode length to be 5 Å. Later, doping of stanene nanotube is done by replacing one, two and three tin atoms by lithium in a chronological order as shown in Fig. 1.

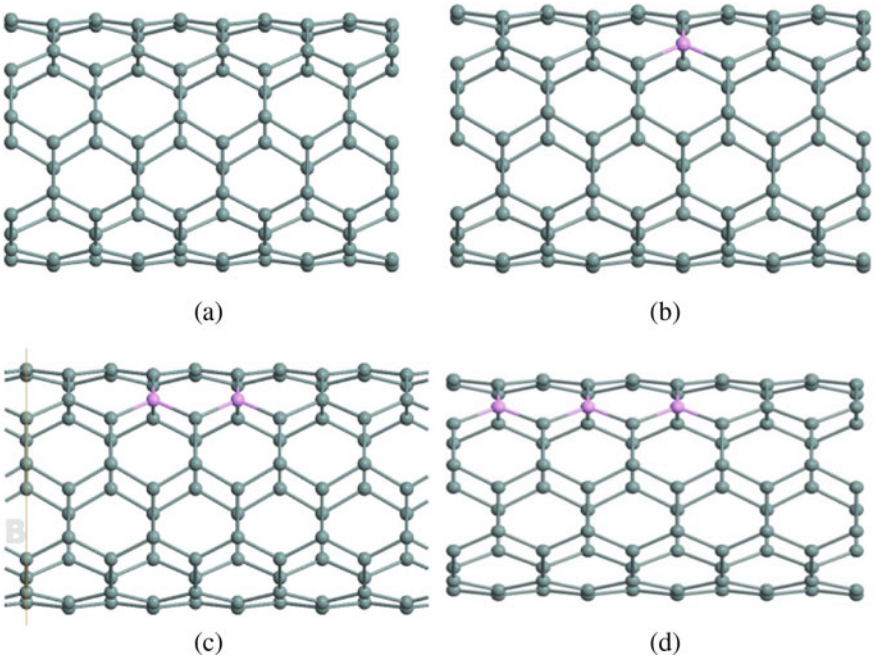


Fig. 1 **a** Structure of pristine stanene nanotube, **b** structure of doped stanene nanotube with 1 lithium atom, **c** structure of doped stanene nanotube with 2 lithium atoms, **d** structure of doped stanene nanotube with three lithium atoms. The green and purple spheres denote tin and lithium atoms, respectively

The analytical model of armchair stanene nanotube has been executed using density functional theory (DFT) within the generalized gradient approach (GGA) and NEGF method. This semi-empirical model of nanotube has been investigated and illustrated using Atomistix Toolkit-Virtual Nanolab (ATK-VNL) (Li and Zhang 2014). For all the optimizations and calculations, we employed double-zeta polarization (DZP) basis set and the gradient corrected Perdew–Burke–Ernzerhof (PBE) functional (Wu et al. 2014). The cut-off energy was chosen to be 500 eV, and sampling with a $1 \times 1 \times 15$ k-point grid was applied in the DFT calculations. All atoms were allowed to relax until the forces were smaller than 0.05 eV/A° at a constant temperature of 300 K. The most effective method at the moment for determining the electronic structure of matter is density functional theory (Stokbro et al. 2010). Atoms, molecules, and solids as well as nuclei, quantum, and classical fields are all applicable. The density of states (DOS) is calculated from the PBE-DFT parametrization. Using the Landauer–Buttiker formalism, the nano-scale current is calculated from the transmission coefficient:

$$I = \frac{e}{h} \int T(E) \left[f\left(\frac{E - E_F^L + eV_R}{k_b T_R} \right) - f\left(\frac{E - E_F^R + eV_L}{k_b T_L} \right) \right] dE, \qquad (1)$$

where f is the Fermi function, E_F^L is the Fermi energy, T_L and T_R are the electron temperatures of the left and right electrodes, respectively, and then the differential conductance is calculated from the transfer function by numerical differentiation of the above equation:

$$G = \frac{dI}{dV} = \frac{I(V_R, T_L, T_R) - I(V_L, T_L, T_R)}{V_R - V_L}. \tag{2}$$

The simulation process is repeated for all devices created at variable bias voltages with a step size of 0.2 V from −1 V to +1 V. The band structure, density of states, transmission spectrum, differential conductance, and current–voltage characteristics (using two-probe method) were calculated on optimized structure.

3 Results

3.1 Band Structure

In order to describe the electrical properties of solids, band structures are used as a representation of the acceptable electronic energy levels. The band structure representation identifies a material's direct or indirect band gaps as well as whether it is metallic, semi-metallic, or insulating.

In Fig. 2, the band structure of a pure stanene nanotube (SnNT) is depicted. Due to band crossings between the valence and conduction bands, SnNT exhibits metallic behaviour and has a band gap of 0.04813 eV. Table 1 shows that as the number of dopant atoms increases, the band gap shrinks. The reduction in the band gap leads to corresponding rise in the conductivity.

3.2 Density of States

The number of states that are available to be occupied at each energy level is indicated by a system's state density. As a result, a high DOS at a particular energy level denotes a large number of states that are available for occupation. For instance, a DOS of zero means that no states can be occupied at that energy level. Figure 3 shows the DOS at various bias voltages ranging from −1 to 1 V. From the analysis of DOS, it is easy to figure out the probability of given state that whether the Fermi energy is occupied or unoccupied. It can be easily analysed that whether the conduction is through HOMO or by LUMO. In the case of pristine stanene nanotube device, the number of peaks is less around the Fermi level. From the comparison of the four figures, it can be seen that the device with 3 doping atoms has the sharpest peaks, and the lowest unoccupied molecular orbital (LUMO) is responsible for the maximum

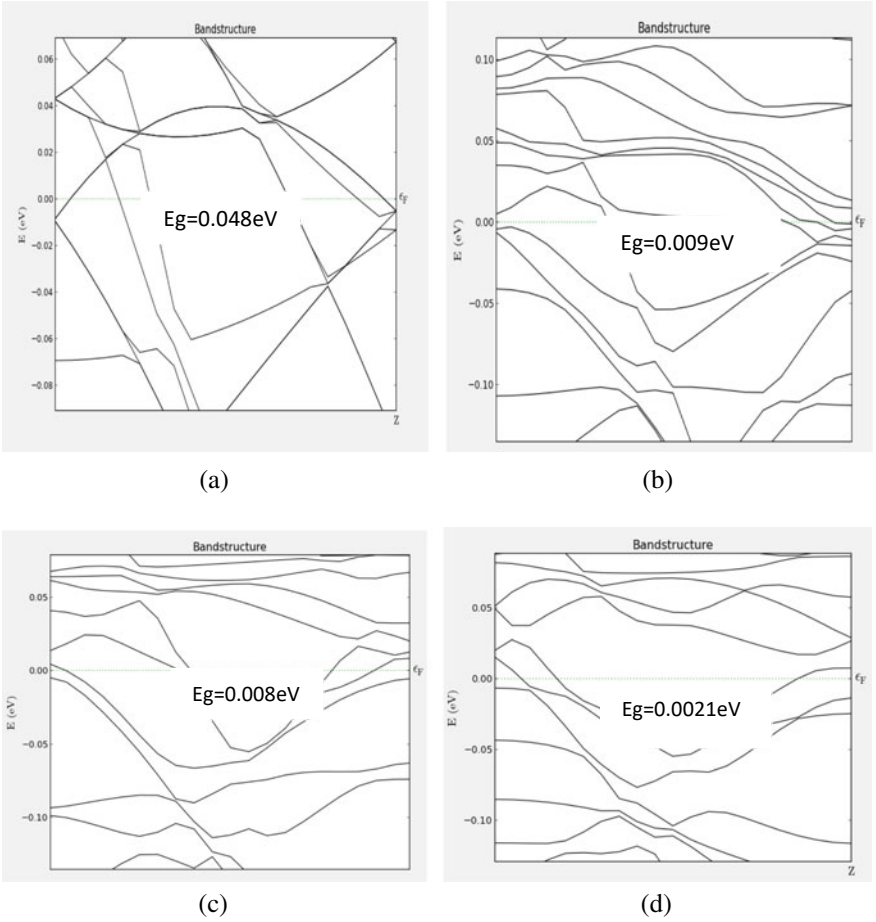


Fig. 2 Band structures of **a** pristine SnNT, **b** 1 dopant SnNT, **c** 2 dopants SnNT, **d** 3 dopants SnNT

Table 1 Band gaps for lithium-doped nanotubes

S. No	Nanotube	Band gap (eV)
1	SnNT (Pristine)	0.04813
2	SnNT (with 1 dopant)	0.00942
3	SnNT (with 2 dopants)	0.00889
4	SnNT (with 3 dopants)	0.00214

conductivity. As we progress from the original to the doped structures, the strength of these peaks gets stronger. The Fermi energy level was considered at zero bias voltage.

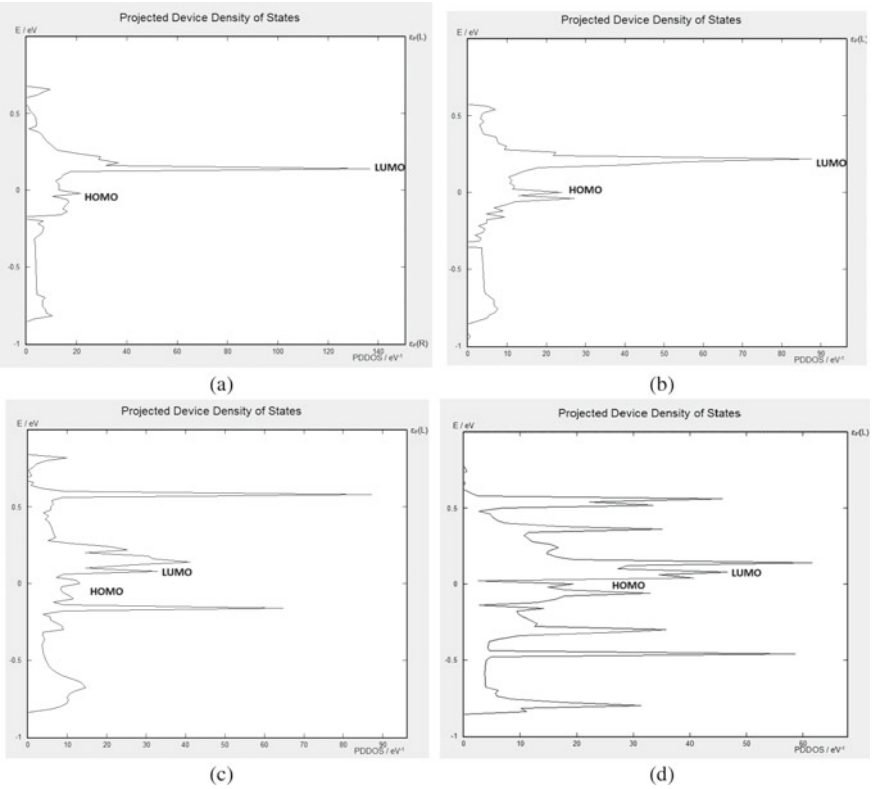


Fig. 3 **a** DOS for pristine SnNT **b** DOS for 1 dopant SnNT, **c** DOS for 2 dopants SnNT, **d** DOS for 3 dopants SnNT

3.3 Transmission Spectra

The transmission spectrum of the two-probe configuration device is used to examine the molecular orbital (MO) energies for applied bias voltages in order to identify the energy at which the electron transfer would be strongest. Figure 4 displays the transmission spectra for all the devices being evaluated at various voltages. The system is forced out of equilibrium by varying the bias voltage from −1 V to 1 V. To learn more about the charge flow resulting in transport under various bias voltages, the transmission spectra of every device are analysed. The higher the number of peaks, the higher will be the transmission. From these curves, it can be deduced that in the case of three dopant atoms, there are more peaks than in other devices. These results derived from the transmission spectra are consistent with the analysis of DOS.

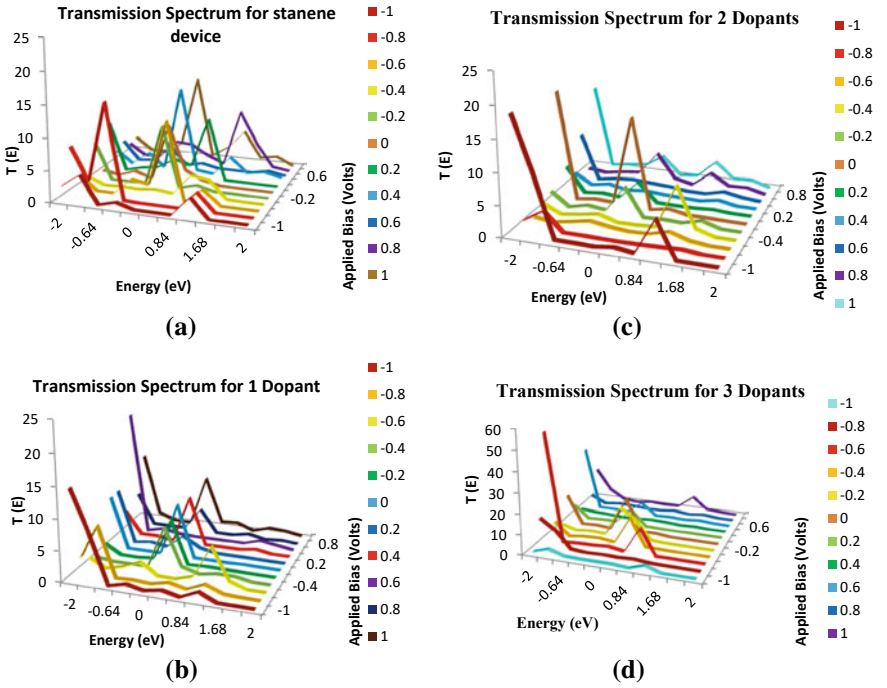


Fig. 4 **a** Transmission spectrum for pristine stanene nanotube, **b** transmission spectrum for 1 dopant stanene nanotube, **c** transmission spectrum for 2 dopants stanene nanotube, **d** transmission spectrum for 3 dopants stanene nanotube at various voltages

3.3.1 I–V Curve

The current–voltage (I–V) characteristic is the most important transport property which is plotted in Fig. 5. I–V curve is found to be approximately negligible within a bias range of −1 V to +1 V for the pristine SnNT. For SnNT device doped with 1 lithium atom, it displayed linear features and exhibited metal-like behaviour in the range (0.8 V to 1 V). For SnNT device doped with 3 lithium atoms, we found that the current increased slowly in the beginning and then increased exponentially in the range (0.2 to 0.4 V). The highest peak values in the transmission spectrum could provide justification for this. Comparing the transmission spectrum of the two-probe system under the bias 0.4 V and 0.6 V, it is obvious that the transmission peak is suppressed, which leads to the drop of the current (peak A to peak B) resulting in negative resistance in this region. This feature can be exploited further in tunnel diode applications. When the bias is increased, the transmission peak begins to rise and the current increases again. From the inset, we can observe that current decreases when the bias voltage is in the range (0.4 V to 0.6 V) for SnNT doped with 3 lithium atoms and (0.6 V to 0.8 V) for SnNT doped with 1 atom and 2 atoms, respectively. This indicates that NDR appears in the above bias ranges. The magnitude of current for

nanodevices is in the order of: 3-dopants > 2-dopants > 1-dopant > pristine device which is listed in Table 2.

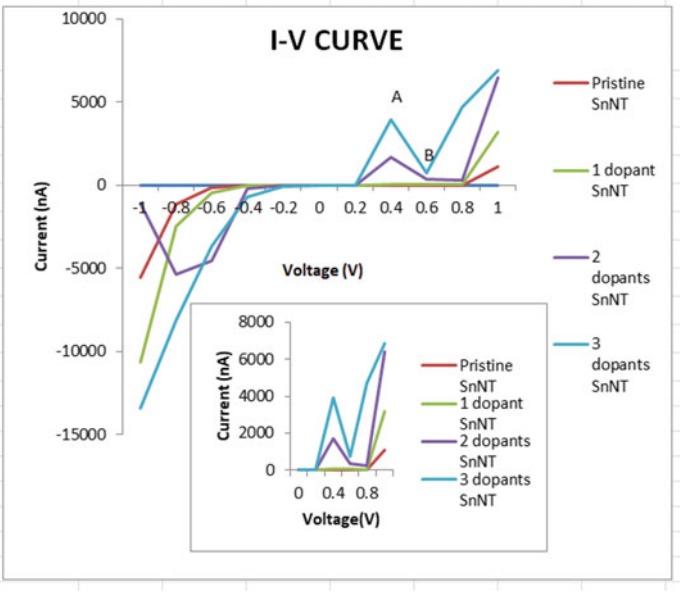


Fig. 5 I–V characteristics

Table 2 Values of current at various bias voltages for all the devices

Voltage (V)	Pristine SnNT (nA)	1 dopant SnNT (nA)	2 dopants SnNT (nA)	3 dopants SnNT (nA)
−1	−5568.38	−10,627	−1116.21	−13,396.1
−0.8	−1146.86	−2466.54	−5381.56	−8155.81
−0.6	−141.187	−483.584	−4544.03	−3648.32
−0.4	−4.85639	−2.22465	−196.736	−688.621
−0.2	−1.96538	−0.372876	−32.4212	−51.4924
0	0	0	0	0
0.2	0.0053307	0.0413948	0.24198	0.719692
0.4	1.69072	57.6171	1706.84	3925.74
0.6	4.20181	61.9481	369.85	764.545
0.8	6.63925	13.9911	273.085	4701.89
1	1108.94	3175.6	6425.2	6895.57

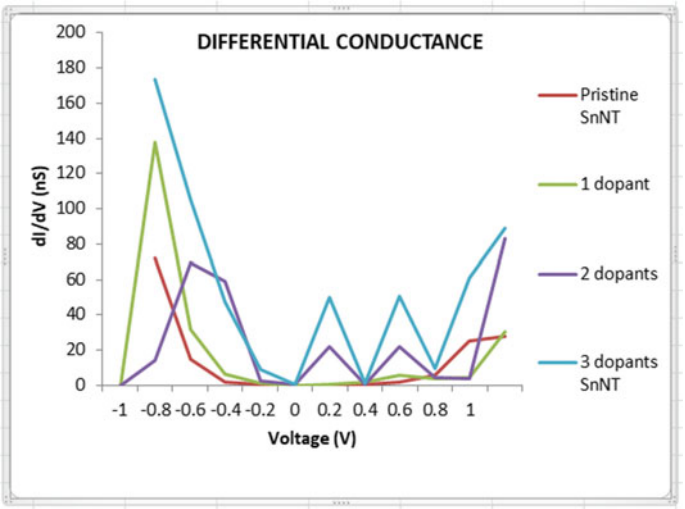


Fig. 6 Differential conductance

3.4 Differential Conductance

The differential conductance provides a clear indication of the bias potentials where electrons are most active or can be used to identify and forecast conditions that are likely to result in tunnelling. Differential conductance can be measured and utilized in different research areas. dI/dV is directly proportional to density of states and is the most effectual way of measurement. The comparative graph of differential conductance is given in Fig. 6. We calculated their conductance values precisely in terms of the fundamental quantum conductance unit, which has the value Go = G/T (EF) = 2e2/h = 77.27 μS, in order to obtain a clearer understanding of the conductance. The differential conductance values of the various devices at different applied voltages are compared in Table 3 using the basic quantum conductance unit. With increasing voltage, the conductance is reduced for negative values and at zero voltage the conductance raises, and the maximum conductance is recorded for SnNT with 3 dopant atoms.

Table 3 Different bias conductance expressed in terms of the fundamental quantum conductance unit

Voltage (V)	Pristine SnNT (μS)	1 dopant SnNT (μS)	2 dopants SnNT (μS)	3 dopants SnNT (μS)
−1	72	137.53	14.44	173.36
−0.8	14.84	32	69.64	105.54
−0.6	1.82	6.25	58.80	47.21
−0.4	0.62	1.42	2.54	9
−0.2	0.02	0.04	0.41	0.66
0	0.25	0.38	22.2	50.07
0.2	0.64	1.81	0.80	0.93
0.4	2.07	5.82	22.08	50.8
0.6	5.82	4.07	4.78	9.9
0.8	25.6	4.52	3.51	60.85
1	28.2	30.41	83.15	89.23

4 Conclusion

Pristine and doped stanene nanotubes have been studied using density functional theory and non-equilibrium Green's functions. In this work, the effects of doping are shown using two-probe configurations. The outcomes for lithium-doped and undoped nanotubes were examined and compared. All nanotubes are semiconductors, as shown by the band structure, where the band gaps range from 0.0483 to 0.002 eV. It has been shown that the device with 3 doping atoms has the best current characteristics, with a value even higher than the stanene nanotube-only device. The sequence of current properties is obtained as 3-dopants device >2-dopants device >1-dopant device >pristine SnNT. This conclusion can be drawn from the analysis of the density of states, transmission spectra, differential conductivity, and I–V curve. Nanotubes favour LUMO transmission. The conductivity increases with the number of doping atoms. These results could be useful for the application of stanene nanotubes in nano-electronic devices, as their properties change from a semiconductor to a near-metallic character. Stanene is undoubtedly a promising material for use in future quantum devices.

References

Abbasi A (2019a) Theoretical investigation of the interaction between noble metals (Ag, Au, Pd, Pt) and stanene nanosheets: a DFT stud. J Inorg Organomet Polym Mater 29:1895–1915

Abbasi A (2019b) Adsorption of phenol, hydrazine and thiophene on stanene monolayers: a computational investigation. Synth Met 247:26–36

Ai LY, Chang CR, Liu Y, Liu H (2020) Tuning the electronic properties and band gap engineering in stanene monolayers via codoping of Mn and Al/P/Ga/As atoms: a DFT study. Comput Theor Chem 112939

Akturk O, Akturk U, Ciraci S (2016) Effects of adatoms and physisorbed molecules on the physical properties of antimonene. Phys Rev B 93

Bhuvaneswari R, Nagarajan V, Chandiramouli R (2018) Adsorption studies of trimethyl amine and n-butyl amine vapors on stanene nanotube molecular device—a first-principles study. Chem Phys 501:78–85

Brandbyge M, Mozos J, Ordejo P, Taylor J, Stokbro K (2002) Density-functional method for nonequilibrium electron transport. Phys Rev B 65:165401

Chen J (2020) The effects of transition metal adatoms on the electronic properties of stanene. Physica E 114365:124

DOE/SLAC National Accelerator Laboratory (21 Nov, 2013) Will 2-D tin be the next super material? Science daily retrieved on June 25, 2022 from www.sciencedaily.com/releases/2013/11/131121135635.htm

Fadaie M, Shahtahmassebi N, Roknabad MR, Gulseren O (2018) First-principles investigation of armchair stanene nanoribbons. Phys Lett A 382(4):180–185

Gao J, Zhang G, Zhang YW (2016) Exploring Ag (111) Substrate for epitaxially growing monolayer stanene: a first-principles study. Sci Rep 6(1)

Garcia J, Lima C, Assali LVC, Justo JF (2011) Group IV graphene- and graphene-like nanosheets. J Phys Chem C 115(27):3242–13246

Hongyana L, Guixian T, Bina H, Dana L, Liu Z (2020) Efficient band gap opening in single-layer stanene via patterned Ga-As codoping: towards semiconducting nanoelectronic devices. Synth Met 116388:264

Hu W, Xia N, Wu X, Li Z, Yang J (2014) A first-principles study of gas adsorption on germanene. Phys Chem Chem Phys 16(41):22495–22498

Li SS, Zhang CW (2014) Tunable electronic structures and magnetic properties in two dimensional stanene with hydrogenation. Mater Chem Phys 16:1667–1671

Li H, An F, Xie J, Wang Y (2020) DFT study of the adsorption behavior of NO gas molecule on the Ga and As pair doped stanene nanosheets. Physica E 114348:124

Liu CC, Jiang H, Yao Y (2011) Low-energy effective Hamiltonian involving spin-orbit coupling in silicene and two-dimensional germanium and tin. Phys Rev B 84(19)

Ma K, Chen J, Dai X, Xiao J, Wang L, Xu L, Wang Z (2021) The potential of stanene with transition metal adsorbed as a promising gas sensor: a first-principles study. Results Phys 28:104617

Modarresi M, Kakoee A, Mogulkoc Y, Roknabadi MR (2015) Effect of external strain on electronic structure of stanene. Comput Mater Sci 101:164–167

Perdew JP, Burke K, Ernzerhof M (1996) Generalized gradient approximation made simple. Phys Rev Lett 77:3865

Sardroodi JJ (2018a) Interaction of sulfur trioxide molecules with armchair and zigzag stanene-based nanotubes: electronic properties exploration by DFT calculations. Adsorption 24(5):443–458

Sardroodi JJ (2018b) An Innovative method for the removal of toxic SO_x molecules from environment by TiO_2/stanene nanocompsites: a first principles study. J Inorg Organomet Polym Mater 28(5):1901–1913

Sharma N, Chi CH, Swaminathan N, Dabur D, Wu HF (2021) Introducing stanene oxyboride nanosheets as white light emitting probe for selectively identifying <5 μm microplastic pollutants. Sens Actuators: B 130617, 348

Stokbro K, Peterson DE, Smidstrup S, Blom A, Ipsen M, Kaasbjerg K (2010) Analysis of electronic structure of boron nitride nanotubes with different positions of intrinsic impurities. Phys Rev B 82:075420

Wu GSC, Shan B, Yan (2014) Two-dimensional time-reversal-invariant topological superconductivity in a doped quantum spin-hall insulator. Phys Rev B 90:054503

Xu Y, Yan B, Zhang HJ, Wang J, Xu G, Tang P, Duan W, Zhang SC (2013) Large-gap quantum spin Hall insulators in tin films. Phys Rev Lett 111(13)
Yaghoobi M (2018) Theoretical study of the structural and electronic properties of novel stanene-based buckled nanotubes and their adsorption behaviour. Appl Surf Sci 435:33–742
Zhao M, Zhang X, Li L (2015) Strain-driven band inversion and topological aspects in Antimonene. Sci Rep 5(1)

SOT-MRAM Memories for Energy Efficient Embedded and AI Applications

Inderjit Singh, Balwinder Raj, and Mamta Khosla

Abstract Modern data intensive and intelligent applications are slowly gearing to be a hardware deployable technology with the advancements in non-conventional computing architectures. With this comes the major challenge of holistic efficiency particularly low power and high-performance implementations in the light of memory system architectures. In this context, modern spintronics based non-volatile memory technologies (NVM) such as spin–orbit torque has significant advantages over existing SRAM with advantages like non-volatility, ultra-low power consumption, higher density, and scalability. In this article we present and demonstrate the simulation framework for the evaluation of energy-time performance of computing architectures running embedded and AI applications. The suitability of spintronic based SOT-MRAM for modern energy-efficient computations and relevant integration architectures in comparison to existing technology is established. Our investigations reveal that SOT-MRAM is ~7% faster, >30% energy efficient and ~40% better in EDP than SRAM technology and therefore can be better choice for modern computing architectures.

Keywords Non-volatile memory (NVM) · Spin orbit torque (SOT) · NVSIM · Gem5 · ARM · AI

I. Singh (✉) · B. Raj · M. Khosla
National Institute of Technology, Jalandhar, Punjab, India
e-mail: inderjits.ec.18@nitj.ac.in

B. Raj
e-mail: rajb@nitj.ac.in

M. Khosla
e-mail: khoslam@nitj.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_2

1 Introduction

The scaling of CMOS integrated circuits has been facing physical limits primarily due to short-channel effects (SHE), increased leakage currents, lithography issues, reduced control over thresholds and increased process variations. The research quest is to improve performance of modern computation systems despite scaling issues. Therefore there is a strong need to explore and use non-conventional devices aside or replacing CMOS devices, referred as 'More than Moore' concept (More Moore 2015; Weste and Harris 2010).

Modern systems-on-chip lay quite large amount of die area for on-chip memories. Today operating at advanced technology nodes, SRAMs are facing the greater challenge of high leakage leading to high static power consumption. Main memories have been implemented in charge storage based Dynamic RAM, primarily because of their excellent packaging density, but they suffer in operating speed and scalability at advanced technology nodes. Down the memory hierarchy, FLASH is used as secondary storage attributed to its non-volatility and high density as illustrated in Fig. 1. On the other hand, the data intensive applications are in higher demand with continuous improvements in GPU architectures, deep neural networks (DNNs) for scalable applications like image recognition, chip placement and object detection (Inci et al. 2022; Khalili 2022). This demand has also increased the memory capacity of last-level cache implementations in multi-core CPU/GPU architectures with current capacity as large as 8MB (Inci et al. 2022; Walden et al. 2021).

Past decade researchers have tried emerging non-conventional, non-volatile memory technologies (NVMs) to combat performance and power challenges (Wong and Salahuddin 2015). While being non-volatile, Magnetic Random Access Memory (MRAM) is scalable, has low leakage, low access time and high density that makes it suitable to become a universal memory (Senni et al. 2016). To understand the magnetic memories, their design, performance and optimum use, the remaining fraction of discussion is carried forward in the following sub-section.

New magnetic memories comprise of magnetic memory cells in conjunction with conventional CMOS periphery. In contrast to conventional mature memories,

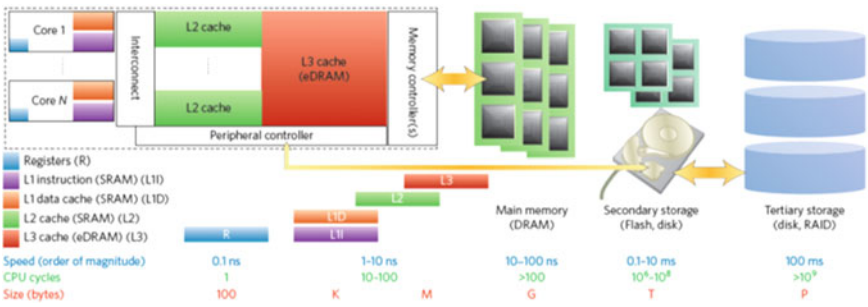


Fig. 1 Memory hierarchy in a typical computing system (Wong and Salahuddin 2015)

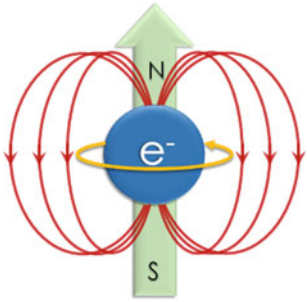


Fig. 2 Electron's spin magnetic moment

modern novel non-volatile memories differ in material and switching mechanism. The switching mechanisms can be one out of the ionic reactions, phase transition, quantum mechanical phenomenon, molecular reconfiguration, etc. (Kaushik et al. 2017; Tang and Lee 2010). A glimpse of principles and foundations of spintronics, the spintronic devices, the evolution of devices to their practicability etc. are discussed in the following section.

2 Background

2.1 Spintronics

Spintronics involves the electron transport phenomenon that is governed by magnetic force in addition to electric potential across device terminals. The spin magnetic moment of the spinning electron (Fig. 2) interacts with the external magnetic field changing the physical transport of electron. This changes the effective electrical resistance of the material and the phenomenon is called magnetoresistance effect (Tang and Lee 2010).

2.2 Magnetic Tunnel Junction–MTJ

The Magnetic Tunnel Junction (MTJ) structure is composed of two Ferromagnetic Material (Ferro-Magnet, FM) layers and a sandwiched Non-magnetic Material (Non-Magnet, NM). FM layers are made differently with ferromagnetic materials. One of the FM layers usually called as soft or free layer (FL) is made from softer magnetic material like CoFeB. The magnetization direction of FL can be altered electrically. The other FM layer called reference layer or the hard layer, the direction of magnetization of which is fixed by magnetic annealing process (Ralph and Stiles 2008; Tang and Lee 2010). Therefore, two FM layers exhibit parallel or antiparallel magnetizations at a given time, giving rise to different magnetoresistances. This very difference

in magnetoresistance exhibited by MTJ is deployed to store a bit. The bit information is read by passing electric current via access device through this layered structure (Khvalkovskiy et al. 2013).

2.3 STT–and SOT–MRAM Bit Cell

Spin Transfer Torque–Magnetic Random Access Memory (STT-MRAM) bit cell has 3 terminals as depicted in Fig. 3. The bit-line (BL) is connected to the free layer whereas source line (SL) is connected to reference layer via access transistor. The gate of access transistor is controlled by wordline (WL) signal. The MTJ current is compared to a reference $((I_{\text{Parallel}} + I_{\text{AntiParallel}})/2)$ current to establish the cell read logic. The write operation is non-symmetric and is achieved through bidirectional switching of two distinct resistive states, i.e., R_P and R_{AP} (Khvalkovskiy et al. 2013). STT-MRAM suffers with two limiting design problems. First, the R/W performance exhibit a tradeoff due to common read/write path. Second, is a lifetime reliability issue posed by large write voltage (current) which shortens the life-span of the dielectric tunnel junction (Zhao et al. 2012). This makes the design of STT-MRAM difficult where on one hand one would prefer lower junction resistance for larger write current, and higher magnetoresistance for read operation on the other hand (Gambardella and Miron 1948).

More recently a novel spin orbit torque MRAM (SOT MRAM) bit cell was proposed as shown in Fig. 4b. The SOT structure uses the Rashba (Gambardella and Miron 1948) rather primarily uses the Spin-Hall effect (SHE) (Liu et al. 2012) to change the magnetization direction of the free layer, summarized in Fig. 4a. In SOT-MRAM the read/write paths are separate, hence optimized independently Fig. 4b, but requires an extra write access transistor (Prenat et al. 2016).

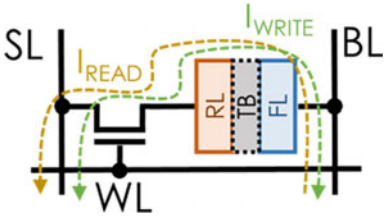
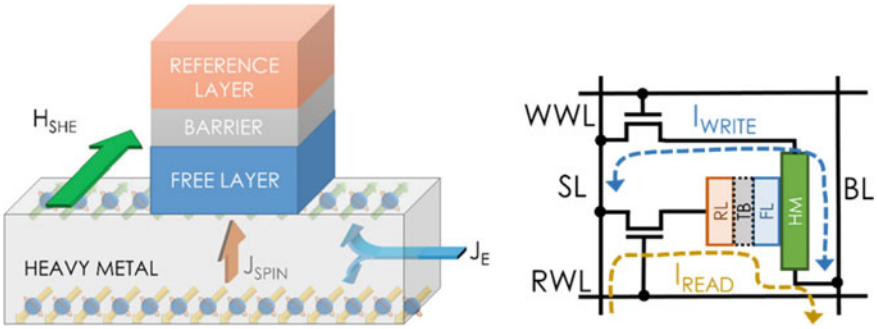


Fig. 3 1 bit STT-MRAM



(a) MTJ for SOT-MRAM (b) 1-bit SOT-MRAM cell

Fig. 4 Spin orbit torque–MTJ and 1-bit cell

3 Exploration and Simulation Framework

The setup uses cross layer platform (Fig. 5) for performance evaluation and benchmarking. The cell-level parameters of SRAM and STT- or SOT-MRAM can be obtained using circuit-level analysis tools like HSPICE in conjunction with compatible Verilog–A model of MRAM cells (Hu et al. 2019). These circuit-level memory cell parameters are fed to community standard tools like CACTI and/or NVSIM to investigate performance numbers at the memory array scale.

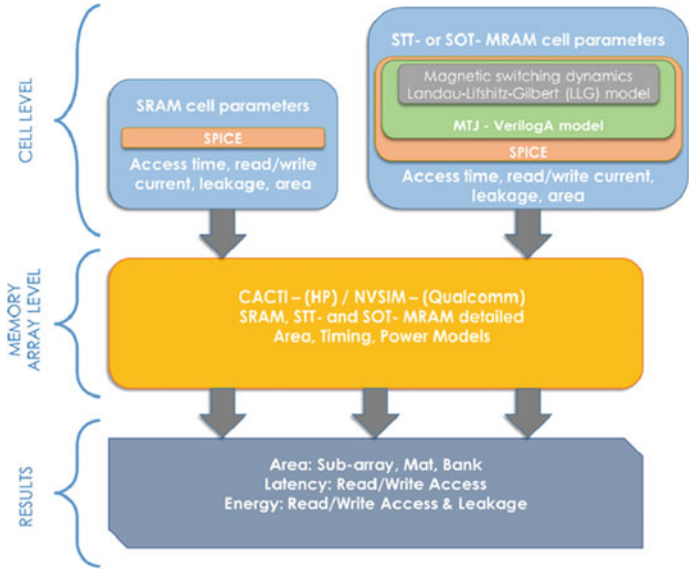


Fig. 5 Cross layer analysis platform

NVSim is an open-source analytical tool used for early estimation of area, performance, and energy of conventional and modern non-volatile memories at circuit-level. It is useful to predict and optimize the performance of different design combinations of memory array organizations (Dong et al. 2012). CACTI is a similar tool from HP labs (Kwak and Jeon 2010).

3.1 System Level Analysis

The results obtained at memory array level from NVSIM are used at system-level evaluations (see Table 1). Gem5 (Binkert et al. 2011), an open source full-system microarchitectural simulator uses the 64-bit ARM processor with 4-cores, with dedicated 32 KB 4-way L1 caches and 512 KB 8-way shared L2 cache running a Linux OS, as shown in Table 2. The gem5 full-system simulation is carried out to obtain system level performance numbers for L1 and L2 caches.

Table 1 Cache parameters

Technology	Level	Capacity	Assoc	Cache area		Latency (ns)		Energy (mJ)		
				Cell (F^2)	Total (mm^2)	Read	Write	Read	Write	Leakage
SRAM	L1	32KB	4	146	0.0212	2.18	1.39	0.01	0.002	18.635
SOT-MRAM				57.5	0.0362	1.78	1.20	0.12	0.012	0.182
SRAM	L2	512KB	8	146	0.3222	5.87	4.05	0.10	0.005	290.859
SOT-MRAM				57.5	0.3192	1.95	1.55	0.39	0.034	1.507

Table 2 System architecture details

Hierarchy level	Configuration
Operating system	Ubuntu 18.04 ARM64
Kernel	v4.14.0
Processor	4-Core, 2GHz, 64-bit RISC ARMv8
L1 I/D cache	Dedicated, 32KB, 4-way associative, 64B cache-line SRAM–5/3 (R/W) cycles SOT-MRAM–4/3 (R/W) cycles
L2 cache	Shared, 512KB, 8-way associative, 64B cache-line SRAM–12/9 (R/W) cycles SOT-MRAM–4/4 (R/W) cycles
Main memory	DRAM, 8GB, LPDDR3, 100-cycle latency

3.2 Benchmark Workloads

Further a mix of embedded and AI workload applications statically compiled for ARM architecture. AI applications include *Decision Tree* and *K-Nearest Neighbors* algorithm. *Decision Tree* (*DTree*) is a supervised typed machine learning algorithm used for a variety of classification problems. The *K-Nearest Neighbor* algorithm, also known as *KNN*, is also widely used non-parametric supervised learning classifier. *Rijndael* is an iterative block cipher, it encrypts (referred as *rj-c*) and decrypts (referred as *rj-d*) a block of data through iterative transformations. *Stringsearch* (referred as str) is an embedded string searching algorithm. *Susan* is an embedded image processing application that performs edge (*ss-edge*), corner (*ss-corner*) and smooth (*ss-smooth*) image enhancements. *GSM* is yet another encryption (*toast*) and decryption (*untoast*) algorithm for embedded applications.

3.3 Cache Memory Activity

Using the above-mentioned toolchain, the workload characteristics are initially obtained in the form of memory activity: cache miss-rate, instruction classification and read/write ratio in L1/L2.

This serves as a primary investigation to a more comprehensive analysis of forthcoming outcomes indicating the direct impact of SOT-MRAM based cache implementations.

It can be observed from Fig. 6 that the chosen benchmark applications have 20–40% of instructions that are classified as memory access. In MRAM the cache read/write energy consumption is different, hence is worth to analyze the read/write ratios for L1 and L2 caches. AI applications *Decision Tree* and *K-Nearest Neighbors* have comparatively higher writes in L2, whereas certain embedded applications like *rijndael* and *toast* exhibit opposite behavior.

The energy consumption in cache memories is a function of bulk accesses and the runtime of the applications. These performance metrics are deduced from gem5 simulations and are depicted in Fig. 7. It can be seen that AI-DecisionTree application has largest execution time Fig. 7a, at the same time largest number of accesses Fig. 7c, therefore bearing moderate accesses per unit time depicted in Fig. 7b. susan-smoothening algorithm has largest number of accesses per unit time among the benchmark applications.

3.4 Total Energy Consumption

Figure 8 depicts the total energy consumption in 8 dedicated L1 I/D caches and a shared L2 cache showing dynamic and static energy consumption details. Evidently

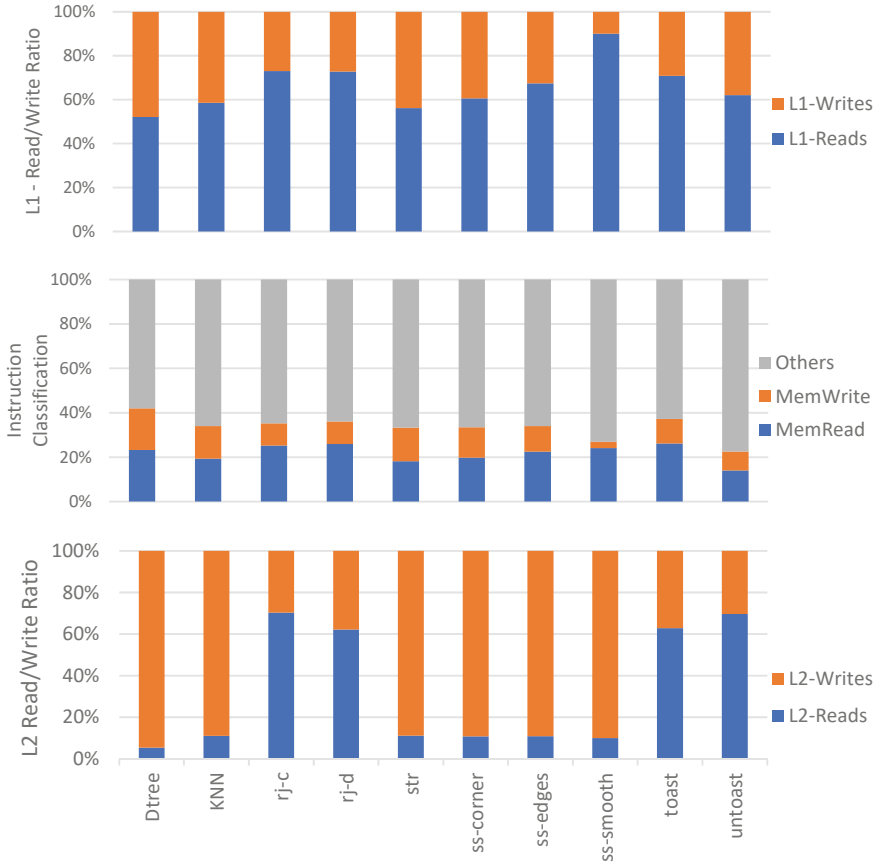


Fig. 6 Instruction classification, L1-read/write ratio, L2-read/

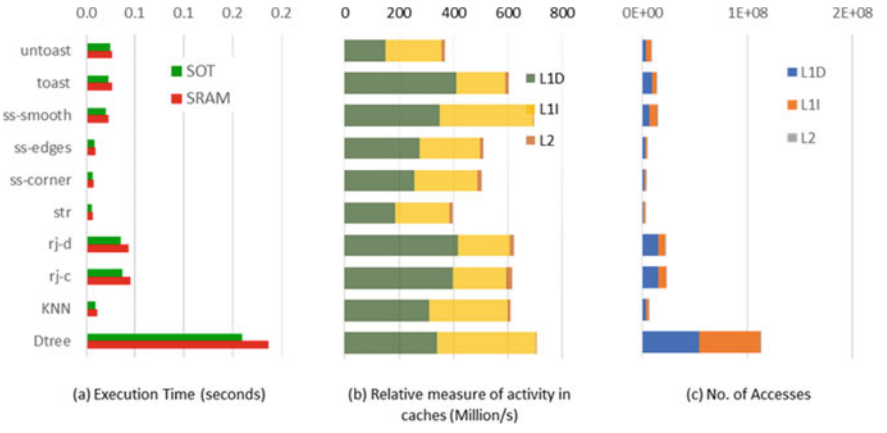


Fig. 7 Workload characteristics

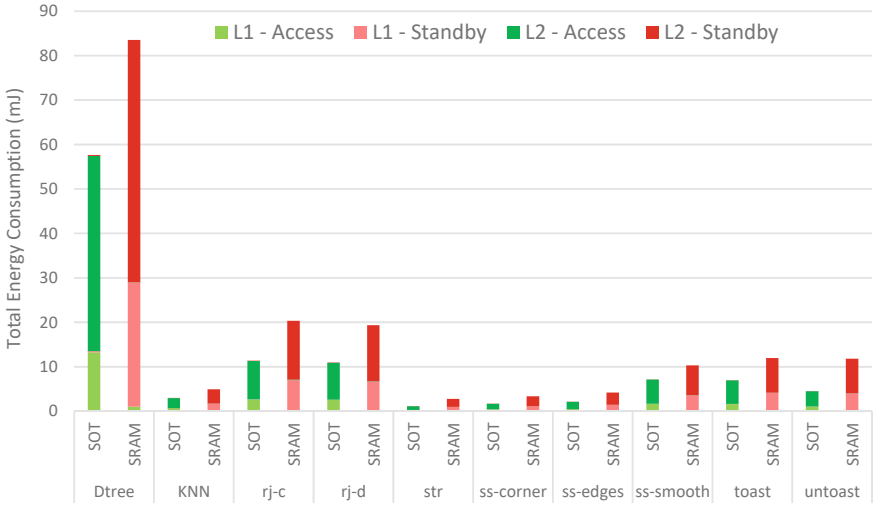


Fig. 8 Relative total energy consumption in SOT-MRAM versus SRAM cache organization

SRAM suffers from standby (static) consumption, while in SOT-MRAM caches dissipate more in the form of access (dynamic) energy. Averaging through the benchmark applications SOT-MRAM does 38.25% total energy savings over SRAM for both L1 and L2 implementations. Nevertheless, certain application with still higher number of accesses per unit time may make SOT-MRAM dissipate higher total energy, in this study a good mix of application cases has been shown.

3.5 Energy Delay Product

Though the energy savings is a considerable improvement but it cannot come at a compromising cost. Energy-Delay-Product (EDP) is an important metric to measure performance and energy savings at the same time. EDP is used to subjectively measure the tradeoff between total energy consumption and the access time (response time) of the memory technologies. Figure 9 shows the EDP metric for chosen applications. It can be observed that the EDP improvements are 61% and 33% for L1 and L2 SOT-MRAM implementations respectively.

Further a normalized average runtime, EDP and Energy comparison is made among SOT-MRAM and SRAM in Fig. 10. It can be seen clearly that the SOT is ~7% faster, >30% energy efficient and bears ~40% lower EDP than conventional SRAM counterpart.

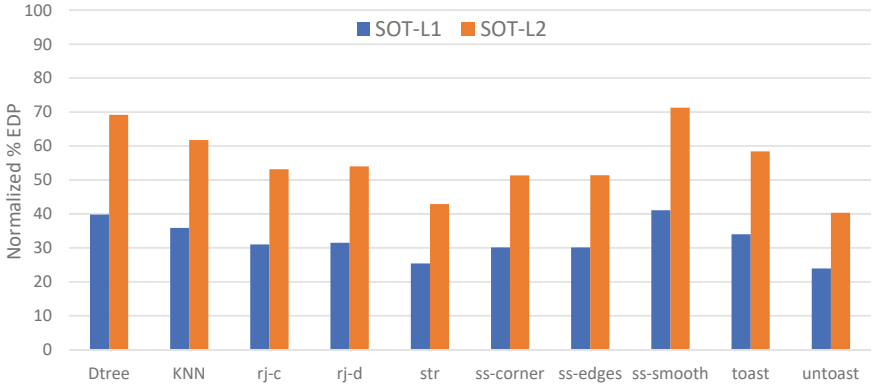
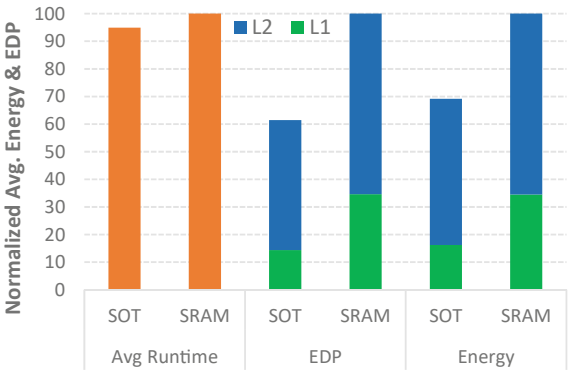


Fig. 9 Normalized SOT-MRAM energy delay product

Fig. 10 Normalized average energy and energy-delay-product



4 Discussion

It can be noted that SOT-MRAM is a strong enough candidate to replace SRAM technology at least for L2-level caches. It supersedes SRAM by exhibiting ultra-low leakage and better execution times. From the analysis it is evident that SOT-MRAM is ~7% faster, ~30% energy efficient than SRAM for L1 + L2 cache implementations for a variety of benchmark workloads. Importantly, SOT-MRAM based L1 is 70% area expensive, 58% and 28% energy efficient in L1 and L2 respectively and ~40% better EDP than counter SRAM cache implementation. The analysis is not independent of the choice of workloads chosen and can be treated as a fair indicator for the made choices. Further, the memory system architecture—cache sizes, operating CPU clock have meaningful impact over the inferences made.

5 Conclusion

SOT-MRAM is a purposeful candidate within the sought locus of More-than-Moore era for energy efficient computing architectures. SOT-MRAM inherits all advantages from Spin-Torque Transfer MRAM (STT-MRAM), while improving over almost all its shortcomings. Its promising features like ultra-low leakage, scalability, non-volatility, and high density makes it a considerable emerging candidate. A circuit-to system-level performance evaluation of an 4-core ARM computing system with dedicated L1 caches, shared L2 cache is presented that runs benchmark embedded and AI applications on gem5 full system (no emulation, no system calls) simulation to reveal the energy numbers. NVSIM is used for circuit-level analysis and calibration of the results thereof to perform system level analysis in gem5. The results indicate that SOT-MRAM outperforms SRAM for L2 implementations in all performance metrics. For smaller caches SRAM had been a sweet spot, SOT-MRAM consumes larger area than SRAM.

References

Binkert N, Sardashti S, Sen R, Sewell K, Shoaib M, Vaish N, Hill MD, Wood DA, Beckmann B, Black G, Reinhardt SK, Saidi A, Basu A, Hestness J, Hower DR, Krishna T (2011) The gem5 simulator. ACM Sigarch Comput Archit News 39(2):1

Dong X, Xu C, Jouppi N, Xie Y (2012) NVSim: a circuit-level performance, energy, and area model for emerging non-volatile memory. IEEE Trans Comput Des Integr Circ Syst 31(7):994–1007

Gambardella P, Miron IM (2011) Current-induced spin-orbit torques. Philos Trans R Soc Ser A Math Phys Eng Sci 369(1948):3175–3197

Hu X, Timm A, Brigner WH, Incorvia JAC, Friedman JS (2019) SPICE-only model for spin-transfer torque domain wall MTJ logic. IEEE Trans Electron Devices 66(6):2817–2821

Inci A, Isgenc MM, Marculescu D (2022) Efficient deep learning using non-volatile memory technology (Ml):1–28

Kaushik BK, Verma S, Kulkarni AA, Prajapati S (2017) Next generation spin torque memories, no. 9789811027192

Khalili P (Oct 2022) Advanced voltage-controlled MRAM devices for low-power memory and machine learning (conference presentation). Spintron, vol XV, p 55

Khvalkovskiy AV, Apalkov D, Watts S, Chepulskii R, Beach RS, Ong A, Tang X, Driskill-Smith A, Butler WH, Visscher PB, Lottis D, Chen E, Nikitin V, Krounbi M (2013) Basic principles of STT-MRAM cell operation in memory arrays. J Phys D Appl Phys 46(7):074001

Kwak JW, Jeon YT (2010) Compressed tag architecture for low-power embedded cache systems. J Syst Archit 56(9):419–428

Liu L, Pai C-F, Li Y, Tseng HW, Ralph DC, Buhrman RA (May 2012) Spin-torque switching with the giant spin hall effect of tantalum. Science (80) 336(6081):555–558

More Moore (2015) International technology roadmap semiconductor, pp 1–52

Prenat G, Jabeur K, Vanhauwaert P, Di Pendina G, Oboril F, Bishnoi R, Ebrahimi M, Lamard N, Boulle O, Garello K, Langer J, Ocker B, Cyrille MC, Gambardella P, Tahoori M, Gaudin G (2016) Ultra-fast and high-reliability SOT-MRAM: from cache replacement to normally-off computing. IEEE Trans Multi-Scale Comput Syst 2(1):49–60

Ralph DC, Stiles MD (2008) Spin transfer torques. J Magn Magn Mater 320(7):1190–1216

Senni S, Torres L, Sassatelli G, Gamatie A, Mussard B (2016) Exploring MRAM technologies for energy efficient systems-on-chip. IEEE J Emerg Sel Top Circ Syst 6(3):279–292

Tang DD, Lee Y-J (2010) Magnetic memory fundamentals and technology, 1st edn. Cambridge University Press

Walden C, Singh D, Jagasivamani M, Li S, Kang L, Asnaashari M, Dubois S, Jacob B, Yeung D (Dec 2021) Monolithically integrating non-volatile main memory over the last-level cache. ACM Trans Archit Code Optim 18(4)

Weste N, Harris D (2010) CMOS VLSI design: a circuits and systems perspective, 4th edn. Addison-Wesley Publishing Company, USA ©2010

Wong HSP, Salahuddin S (2015) Memory leads the way to better computing. Nat Nanotechnol 10(3):191–194

Zhao WS, Zhang Y, Devolder T, Klein JO, Ravelosona D, Chappert C, Mazoyer P (2012) Failure and reliability analysis of STT-MRAM. Microelectron Reliab 52(9–10):1848–1852

A Comprehensive Review to Investigate the Effect of Read Port Topology on the Performance of Different 7 T SRAM Cells

Bhawna Rawat and Poornima Mittal

Abstract SRAM-based cache memory is an important component for microprocessor-based circuit. The growing demand for portable devices with prolonged battery life has driven designers to report SRAM bit cells with varying topologies. The 7T cell is one of the most popular bit cell topologies. As a consequence, five different 7T cells with a common internal memory core configuration and different read port topologies are reviewed in this paper. The technology node for each cell is 32 nm, while the supply voltage is 0.8 V. The cell topologies are analyzed for static noise margin and time requirement for each operation. It is identified that the 7T4 and 7T5 cells have best read stability at 314 and 324 mV, respectively. In terms of read time requirement, the 7T4 and 7T5 cells are identified to be the best at 5 ps each.

Keywords Read port · SRAM · High integration density · Improved stability · Single port · Dual port

1 Introduction

The increasing popularity of hyper-personalized devices has led to the advent of a world governed by artificial intelligence and applications of Internet of things. The hardware elements essential for implementation of these applications are dependent on cache memories that are fast, have low area footprint, and consume minimal power. Microprocessor memory is composed of static random access memory (SRAM) cells, which occupy 90% of its surface area and consequently dominate the power appetite of the device (Divya and Mittal 2022; Rawat and Mittal 2022a). Thus, it is critical to keep cell area and power in check. The area occupied by an SRAM bit cell can be limited to a smaller value if less transistors are used in designing, whereas, to limit

B. Rawat · P. Mittal (✉)
Delhi Technological University, Delhi, India
e-mail: poornimamittal@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_3

power consumption by a bit cell, lowering supply voltage (V_{DD}) is the prevalent and conventional technique. The active power for a cell has a quadratic relation with V_{DD}. Therefore, a small reduction in V_{DD} has a large impact on its active power (Ahlawat et al. 2021). But, decreasing V_{DD} to near-threshold or subthreshold region has severe implications on cell performance (Bharti and Mittal 2021; Calhoun et al. 2005). Also, at lower V_{DD}, the circuit yield decreases due to the greater influence of variation in the threshold voltage (V_{TH}) (Oh et al. 2021). This is an implication of minimally sized transistors to ensure high integration density. A conventional mechanism to uplift the bit cell performance at lower V_{DD} is to increase the bit cell count for the circuit. Consequently, different architectures such as 8T (Giterman et al. 2018; Kushwah and Vishvakarma 2016), 9T (Cho et al. 2020; Shin et al. 2017), 10T (Jiang et al. 2019; Wen et al. 2019) have been reported by designers across the globe. A major challenge for SRAM cells with high transistor count is their lower integration density; as the transistor count for cell increases, its on-chip area also increases. SRAM is formed by an array of bit cells. Thus, a minimal increase in the layout footprint for a cell has drastic impact on the overall area of cache, thus threatening the memory's economic feasibility and commerciality. Consequently, it is essential to design a cell with the capability to demonstrate robust performance at lower V_{DD} and high integration density. Reduction in technology node has been a major contributor to density increment. But, at lower technology node, the conventional single port 6T cell is unable to perform reliably, so the 7T cell has become the next conventional alternative. In this paper, different 7T SRAM cells are reviewed for different performance parameters. This paper is categorized into five sections, including this introduction section. In Sect. 2, the 7T topologies pre-existing in literature, are elaborated upon and their working mechanisms are explained. The cell performance in terms of static margin analysis, time requirement, and their impact on different SRAM topologies is detailed in Sects. 3 and 4, respectively. The results obtained for the different cells are summarized in Sect. 5.

2 Pre-existing 7 T Bit Cell Topologies

Different 7T cell topologies reported in literature are reviewed in this section. The different 7T cells have common memory core design, while they are differentiated based on the read port configuration. The data core for each cell comprises of an inverter pair (P1-N1 and P2-N2). The inverter pair is mutually connected via a positive feedback connection dependent on transistor N3. It is controlled by signal W1, which is used to disable the feedback between the inverter pair. This converts the mutually coupled inverter configuration into a cascaded configuration. Additional to the aforementioned memory core, each cell consists of a read and write port to aid the respective operation. The latter operation for all the cells is single ended and requires the internal core of the memory to be converted into a cascaded inverter configuration. Thus, during write operation, the W1 signal is turned OFF. The nature and mechanism for read operation for each cell are explained individually.

The first 7T cell (7T1) described in the paper was reported by Aly and Bayoumi (Aly and Bayoumi 2007) in 2007. Its write operation is designed to be single ended and the read operation is differential ended. Thus, only bitline (BL) signal is used to perform write operation, whereas both BL and bitline bar (BLB) are employed for read. Its prime highlight is its ability to reduce power dissipation. The schematic for 7T1 is represented in Fig. 1a. Its single ended write topology reduces its activity factor to half (Rawat et al. 2018), thereby resulting in a major power reduction during write mode of the cell. During the read operation, both WL and RL signals are asserted and the cell is accessed via transistors N4 and N5. But, the extra N3 transistor in the read path for the cell introduces an asymmetry in its design causing a lower read stability. Also, the read power consumption is high due to its differential nature.

A second 7T cell (7T2) was described by Rawat and Mittal (Rawat and Mittal 2021a). The transistor-level representation for 7T2 cell is depicted in Fig. 1b. This cell reduces read power consumption by relying only on a single bitline. Thus, this cell has low power appetite for both access (read-write) operations. Its data node lies in the discharge current path for the read operation, making it highly vulnerable to noise. Also, as there are two NMOS transistors N1 and N5 in the read path, the time required to execute the read operation will be high. Thus, this cell suffers due to its noise vulnerability and greater delay for the read operation.

Another cell (7T3) designed by Yang et al. (2016) in 2016 attempts to resolve the challenges encountered by the 7T1 cell. The circuit schematic of 7T3 is presented in Fig. 1c. Its structure is similar to 7T1 with the exception of the N5 transistor. In 7T1 cell, the N5 transistor is used to form the differential read port for the cell whereas for

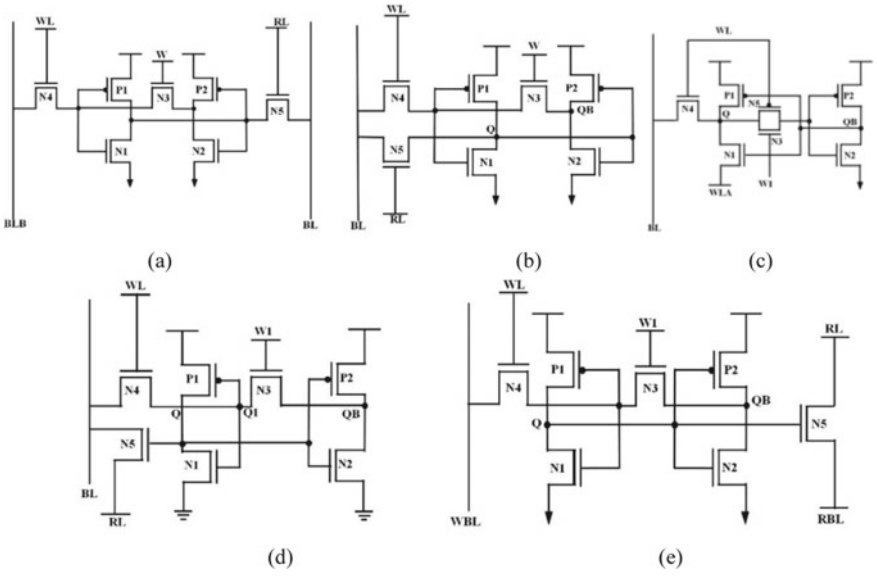


Fig. 1 Schematic diagram for **a** 7T1, **b** 7T2, **c** 7T3, **d** 7T4, and **e** 7T5 SRAM bit cells

7T3 cell, it is used to create a transmission gate parallel to N3 transistor. Also, unlike 7T1 and 7T2 cells, the N5 transistor here is of PMOS type. Another modification that is added to the cell topology is the use of additional control signal WLA at the source of the N1 transistor. Due to its single port topology and single ended configuration for operation, it observes a significant reduction in dynamic power consumption. But, the increase in control signal count increases the circuit complexity, decreases its integration density and also increases the peripheral circuitry required for the memory. Thus, the improvement in power consumption for the cell gets overshadowed by the other challenges.

A modification to mend the read noise stability for a cell was reported in Rawat and Mittal (2021b). This 7T (7T4) SRAM cell modifies the read port configuration for 7T2 to make it read SNM free. The transistor-level representation of 7T4 cell is shown in Fig. 1d. An SRAM cell that does not include data node in the read discharge current path is referred to as the read SNM free (Singh et al. 2013). The read port for 7T4 is functional only when Q = '1'. The read discharge current passes from BL to RL via N5 transistor, thereby maintaining the integrity of data stored at the node. This cell thus has better power performance along with improved read stability.

Another 7T cell (7T5) with read SNM-free topology is reported in Rawat and Mittal (2022b). This cell differs from the 7T4 cell in terms of its dual-port configuration. This cell isolates the read and write ports, thereby facilitating pipelining for the memory architecture. The transistor-level representation for the 7T5 is shown in Fig. 1e. The single ended nature for the read as well as write operations of the cell facilitates power reduction. The read SNM-free port topology boosts its read stability, and isolated read port for the cell makes its operation faster and pipelining enabled.

3 Static Margin Analysis

A cell irrespective of its topology has to work reliably under three different operational modes—hold, read, and write. A reliable cell should be able to perform each of these operations and withstand noise due to internal or external factors. The resilience of a cell toward noise is estimated in terms of static noise margin (SNM). It is the maximum noise fluctuation that a cell can withstand without resulting in an erroneous data flip for the cell (Kulkarni et al. 2007). It is measured as the side of the largest square that fits inside the smaller lobe of the butterfly curve (Kumar et al. 2014). It is measured individually for hold, read, and write operations in terms of HSNM, RSNM, and WSNM, respectively.

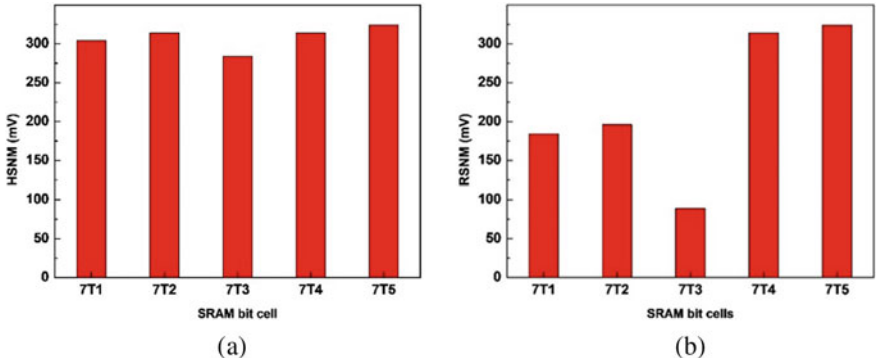


Fig. 2 **a** Hold and **b** read SNM values obtained for the different 7 T cells

3.1 HSNM

A cache memory is designed using a large array of cells, but at a given instant only a few cells are accessed. Therefore, an SRAM cell is usually maintained in hold state. Consequently, a cell is required to have high resilience to noise during hold state. The HSNM value obtained for all the 7T cell topologies explained in Sect. 2 is graphically compared in Fig. 2a. Maximum HSNM value is registered for the 7T5 SRAM bit cell at 324 mV. The 7T2 and 7T4 cells have slightly lower values at 314 mV each. They are followed by 7T1 cell which registers an HSNM of 304 mV, whereas the least SNM value for hold operation is recorded for the 7T3 cell at 284 mV.

3.2 RSNM

The read is an access operation for a cell. A crucial aspect of the read operation is maintaining the integrity of the information stored in the cell is to be maintained during the operation. This is largely dependent on the cell's read port configuration. If the read discharge current passes through the data storage nodes, the likeliness for a destructive read operation is high, whereas, if the discharge current path does not include the data node, the likelihood for a destructive read operation is eliminated. Cells that have such a read port configuration are called SNM-free read port. Among the different cells explained in Sect. 2, the only cells with read SNM-free port configuration are 7T4 and 7T5. In these cells, the read current flows through the N5 transistor controlled by node Q.

Therefore, read current will flow only for Q = '1'; for Q = '0', no read current is registered by the cell. As the 7T4 and 7T5 cells are read SNM free, their HSNM and RSNM values are identical. The other cells—7T1, 7T2, and 7T3 register an RSNM value of 184, 196, and 89 mV, respectively. The RSNM values obtained for the 7T cells are represented in Fig. 2b.

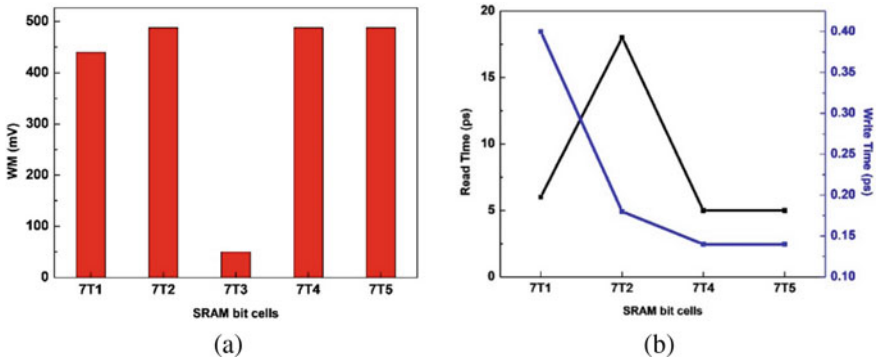


Fig. 3 **a** WSNM and **b** read and write time values obtained for different 7T cells

3.3 WSNM

Another access operation that a cell performs is the write operation. During its course, the information retained inside the cell is to be deliberately flipped. A cell's resilience to noise during the write operation is quantified as WSNM. It is calculated as the voltage difference between the wordline and the V_{DD} (Chaturvedi et al. 2021). The WSNM values obtained for all the cells are compared in Fig. 3a. As the write mechanism for all the bit cells is the same, similar values are obtained for all the cells.

4 Delay

During the SNM analysis, each operation for a cell is analyzed irrespective of the time required to complete the operation. But, in real-life application, an SRAM memory is a part of a microprocessor circuit. Consequently, the circuit is driven in keeping with the clock pulse for the setup. Therefore, when designing an SRAM memory, it is crucial to analyze the delay requirement for an individual bit cell. The two access operations (read and write) for a cell are time dependent. Therefore, each cell in the paper is analyzed for both read and write time in the following subsections.

4.1 Read Time

Timing constraint is crucial for the read operation as a longer time pulse may threaten a destructive read operation, whereas an extremely short pulse may result in an insufficient discharge resulting in erroneous read operation. Thus, for an optimal read operation, the read pulse has to be highly optimized. If the read cell topology is

of differential nature, the read time is the time needed to develop a 50 mV differential voltage on the bitline pair, whereas, for a single ended operation, it is the time required to reach the trip voltage for the inverter (Lin et al. 2010). The read time for the different bit cells is depicted in Fig. 3b. The fastest read operation for the 7T4 and 7T5 SRAM cells is at 5 ps each. These two cells have a single ended read SNM-free topology and therefore are faster than their other counterparts. The 7T1 SRAM bit cell also has a similar read time value of 6 ps. It has a differential read operation and therefore performs better than 7T2 cell which relies on a single ended pass transistor configuration for read operation. The timing analysis is not included for the 7T3 cell as it has a two-step process while the other cells have a single step process.

4.2 Write Time

The write operation implies that the data in the cell has to be altered. It is usually the most time-consuming process for a cell. Therefore, an SRAM cell is deemed more reliable and robust if it completes its write operation in least amount of time possible. Consequently, write time is defined as the least time needed to successfully change the values stored in the cell. The write time values recorded for all the 7T cell explained in Sect. 2 are presented in Fig. 3b. The 7T1 cell requires the largest delay for write operation completion of 0.4 ps. This is because the 7T1 cell uses all typical NMOS transistors for inverter implementation, and the cascaded inverter configuration thus makes it slow. But the 7T2, 7T4 and 7T5 SRAM bit cell have modified their inverter core to include a high-performance NMOS transistor to boost the operational speed for the circuit. Therefore, the three cells 7T2, 7T4, and 7T5 are comparatively faster than the 7T1 cell with write delay of only 0.18, 0.14, and 0.14 ps, respectively.

5 Conclusion

In this paper, five different 7T cell topologies are reviewed, and their results are presented. The 7T cells are designed for technology node of 32 nm and simulated at 0.8 V. The 7T cells pre-dominantly differ in implementation of their read port. Thus, the impact of different read port configurations on cell performance is analyzed. Each cell is evaluated for its resilience to noise during the three operations (hold, read, and write) in terms of the static noise margin. The 7T5 cell is most noise resilient during the read operation with RSNM value of 324 mV. Another cell with a slightly lower value for RSNM is 7T4 cell at 314 mV. These two cells also register the least read time requirement of 5 ps. The same is achieved due to the single ended read SNM-free topology of the read port for the bit cells.

References

Ahlawat S, Siddharth, Rawat B, Mittal P (2021) A comparative performance analysis of varied 10T SRAM cell topologies at 32 nm technology node. International conference on modeling, simulation and optimization

Aly RE, Bayoumi MA (2007) Low-power cache design using 7T SRAM cell. IEEE Trans Circ Syst II Express Briefs 54(4):318–322

Bharti R, Mittal P (2021) Comparative analysis of different types of inverters for low power at 45 nm. In: 3rd IEEE international conference on advances in computing, communication control and networking (ICAC3N-21)

Calhoun BH, Wang A, Chandrakasan A (2005) Modeling and sizing for minimum energy operation in subthreshold circuits. IEEE J Solid-State Circ 40(9):1778–1786

Chaturvedi M, Garg M, Rawat B, Mittal P (2021) 8T SRAM bit cell with improved read stability for 90 nm technology node. International conference on simulation, automation, & smart manufacturing (SASM)

Cho K, Park J, Oh TW, Jung SO (2020) One-sided Schmitt-trigger-based 9T SRAM cell for near-threshold operation. IEEE Trans Circ Syst I Regul Pap 67(5):1551–1561

Divya, Mittal P (April 2022) A low-power high-performance voltage sense amplifier for static RAM and comparison with existing current/voltage sense amplifiers. Int J Inf Technol (BJIT) 14:323–331

Giterman R, Vicentowski M, Levi I, Weizman Y, Keren O, Fish A (2018) Leakage power attack-resilient symmetrical 8T SRAM cell. IEEE Trans Very Large Scale Integr (VLSI) Syst 26(10):2180–2184

Jiang J, Xu Y, Zhu W, Xiao J, Zou S (2019) Quadruple cross-coupled latch-based 10T and 12T SRAM bit-cell designs for highly reliable terrestrial applications. IEEE Trans Circ Syst I Regul Pap 66(3):967–977

Kulkarni JP, Kim K, Roy K (2007) A 160 mV robust Schmitt trigger based subthreshold SRAM. IEEE J Solid-State Circ 42(10):2303–2313

Kumar B, Kaushik BK, Negi YS (2014) Design and analysis of noise margin, write ability and read stability of organic and hybrid 6-T SRAM cell. Microelectron Reliab 54(12):2801–2812

Kushwah CB, Vishvakarma SK (2016) A single-ended with dynamic feedback control 8T subthreshold SRAM cell. IEEE Trans Very Large Scale Integr (VLSI) Syst 24(1):373–377

Lin S, Kim YB, Lombardi F (2010) Design and analysis of a 32 nm PVT tolerant CMOS SRAM cell for low leakage and high stability. Integration 43(2):176–187

Oh JS, Park J, Cho K, Oh TW, Jung SO (2021) Differential read/write 7T SRAM with bit-interleaved structure for near-threshold operation. IEEE Access 9:64104–64115

Rawat B, Gupta K, Goel N (2018) Low voltage 7T SRAM bit cell in 32 nm CMOS technology node. 2018 international conference of computing, power and communications technologies (GUCON)

Rawat B, Mittal P (2021a) Single bit line accessed high high performance ultra-low voltage operating 7T SRAM bit cell with improved read stability. Int J Circ Theory Appl 49(5):1435–1449

Rawat B, Mittal P (2021b) A 32 nm single ended single port 7T SRAM for low power utilization. Semicond Sci Technol 36(9):095006–095022

Rawat B, Mittal P (2022a) A reliable and temperature variation tolerant 7T SRAM cell with single bitline configuration for low voltage application. Circ Syst Sign Proces 41(1):2779–2801

Rawat B, Mittal P (2022b) A comprehensive analysis of different 7T SRAM topologies to design a 1R1W bit interleaving enabled and half select free cell for 32 nm technology node. Proc R Soc a: Math, Phys, Eng Sci 478(2259):20210745–20210771

Shin K, Choi W, Park J (2017) Half-select free and bit-line sharing 9T SRAM for reliable supply voltage scaling. IEEE Trans Circ Syst I Regul Pap 64(8):2036–2048

Singh J, Mohanty SP, Pradhan DK (2013) Robust SRAM designs and analysis. Springer, New York, pp 95–96

Wen L, Zhang Y, Zeng X (2019) Column-selection-enabled 10T SRAM utilizing shared diff-VDD write and dropped-VDD read for power reduction. IEEE Trans Very Large Scale Integr (VLSI) Syst 27(6):1470–1474

Yang Y, Jeong H, Song SC, Wang J, Yeap G, Jung SO (2016) Single bit-line 7T SRAM cell for near-threshold voltage operation with enhanced performance and energy in 14 nm FinFET technology. IEEE Trans Circ Syst I Regul Pap 63(7):1023–1032

A Comparative Performance Analysis of 10 T and 11 T SRAM Cells

Puneet Yadav and Poornima Mittal

Abstract Memory is already known to be one of the most crucial parts of any electronic system. However, a class of memory called the cache memory is even more crucial among the type of memories since it is the one working closely in synchronization with the central processing unit. There are millions of SRAM cells inside cache memory. SRAM cells must therefore possess a few essential attributes for cache memory to be reliable, including low dynamic and static power consumption, high data stability, and low read latency. To successfully incorporate these qualities, a comparative analysis of different 10 T and 11 T SRAM cells has been performed. The performance of the conventional TG10T and 11 T SRAM models is compared to the 10 T SRAM to showcase enhancements obtained. TG10T SRAM cell deploys two transmission gates instead of two NMOS access transistors to strengthen writing ability. It also employs two additional buffer transistors so that read stability can be enhanced. All simulations are carried out using LTSPICE software operating at 0.5 V in 32 nm CMOS process technology. The TG10T SRAM cell is proven to be more enhanced in almost every aspect, but it consumes more power. The read SNM and write SNM are found to be the largest in the TG10T SRAM cell. The power dissipated by the TG10T cell (i.e., 233.69 nW) is approximately two times as compared to the 10 T SRAM cell (i.e., 108.65 nW) and 11 T SRAM cell (i.e., 88.491 nW). The analysis also portrays that both read and write delay is minimum in TG10T SRAM cells. The read delay is 343.3 psec, and the write delay is 494 psec, respectively.

Keywords 10 T SRAM · 11 T SRAM · Stability · Butterfly curve · Transmission gate · Static noise margin (SNM)

P. Yadav (✉) · P. Mittal (✉)
Department of Electronics and Communication Engineering, Delhi Technological University, New Delhi, India
e-mail: puneetyadav_2k21vls21@dtu.ac.in

P. Mittal
e-mail: poornimamittal@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_4

1 Introduction

Memory usage has increased to an unprecedented level in the technology sector today. Memory-intensive processes are carried out across a variety of devices, including computers and mobile phones. Hence, memory is one of the most crucial parts of every system, from personal computers and workstations to microcontrollers and SoCs. Cache memory plays a critical role in the memory hierarchy for the CPU's rapid data processing. They serve as spare memory for the CPU by momentarily storing data that the CPU often needs. Static RAM, a type of random-access memory, constitutes most of this cache memory. So, it is crucial to guarantee that the SRAMs being employed are of the best caliber and dependability. The increasing appeal of IoT has risen the computational load on System on Chip (SoC). The primary components of SoC are cache memory, which happens to be generated by an array of SRAM bit cells. The fundamental characteristics of processors are multi-tasking and speed. Each processor relies on an internal cache for faster applications. This cache is a high-speed memory specifically designed to support processor performance. The cache is a collection of bit cells from SRAM cells. The SRAM cell is operated by peripheral circuitry which consists of bit lines, sense amplifiers, and decoders. The SRAM cells consume around 30% of the power in embedded systems. An SRAM bit cell consists of two key components for power usage, i.e., static and dynamic. The relationship of the supply voltage (VDD) is linear with static power consumption whereas quadratic with dynamic power usage.

The industry has been dragged to shift toward a nanoscale feature size owing to the increasing demand for streamlined designs with longer battery lives to meet consumers' demands. Moreover, in line with Moore's law in nanotechnology, the feature size of highly integrated nanoelectronics devices is continuously shrinking. This can be ensured by qualities like strong data retention stability (also known as resilience), low static as well as dynamic power consumption, and short read delay. However, for a variety of reasons, the features are still at risk. First, voltages have been substantially scaled down to diminish the size of transistors and devices, which impairs the stability of data contained in the cell. Moreover, a significant proportion of power consumption, in particular leakage power consumption, occurs since SRAM cells are memory components.

To highlight the improvements made, the performance of the TG10T and 11 T SRAM models is compared to that of the 10 T SRAM in this paper. To facilitate better comprehension, the paper is divided into four sections. A brief description of several SRAM cells, including their functional mechanics, is discussed in Sect. 2. The implementation of the model will also be covered. The results of the simulation and associated analysis will be presented in Sect. 3. Part 4, which discusses the conclusion and potential future directions, will wrap up the paper.

2 Overview of SRAM Cells

2.1 10T SRAM Cell with Source-Biased Inverter

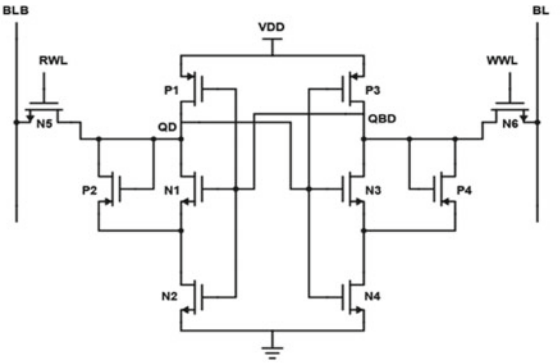
An SRAM cell is a component of a circuit that is capable of storing a single data bit which may either be 0 or 1. Also, it must be capable to read or write data, while there is a power supply. The 10 T cell with a source-biased inverter, which consists of 10 transistors, will be utilized as a baseline for comparison with the other two models. A back-to-back CMOS inverter is used for storing data, and two access transistors are placed on either side of the inverter to facilitate reading from and writing to the cell. Figure 1a illustrates the circuit.

Even before turning on access transistors N5 and N6, the word line is triggered to '1.' The bit line is set the same as the data bit. The bitline_bar is set the same as the complement of the data bit. This pulls data onto the two different nodes of in cross-coupled inverter. Here, by utilizing a precharge circuit, BL and BLB are precharged '1' to successfully read data from the SRAM cell. Following this operation, for turning on access transistors N5 and N6, the word line is triggered to 1 according to the data on the nodes, so that either the BL or BLB is discharged to the GND. During hold mode, WWL and RWL are grounded to turn OFF the access transistor. Hence, the core of the cell is separated from bit lines. So, no read or write operation is performed.

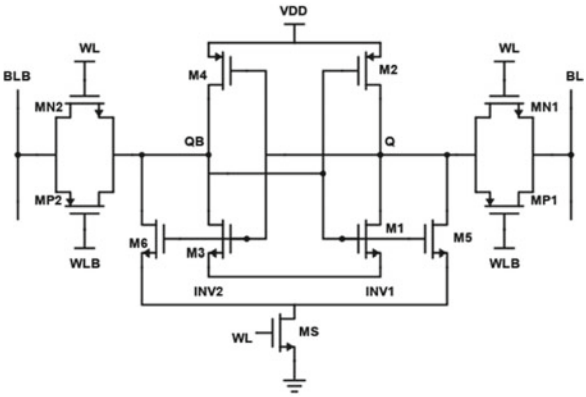
2.2 Transmission Gate-Based 10 T SRAM Cell

Figure 1b displays a TG10T SRAM Cell. The right inverter, which consists of M1 and M2, is constricted by QB, while the left inverter, which consists of M3 and M4, is by Q. Both Q and QB are storage nodes. A latch is created by the back-to-back inverters for holding the data gathered in the cell. In addition, the TG10T cell uses two transmission gates, TG 1 (MN1-MP1) and TG 2 (MN2-MP2), controlled by both word lines to the write capability. The read stability is improved by two buffer transistors, M6 and M5, which are gated by Q and QB (Abbasian and Gholipour 2022). The MS transistor which is driven by the word line connects the source of the above transistor to GND. The nodes Q = 0 and QB = 1. When WL is 1 and WLB is 0, the bit lines can be accessed by switching the transmission gates. The MS transistor is also ON. Here, BL is at VDD and BLB at GND, which writes a '0'/'1' to QB/Q. The write operation is quicker because both strong logic '0' and strong '1' values are passed through the transmission gates. Through M1 and M5, node Q discharges, whereas MS is initially brought to GND for a small duration of time. During this process, QB discharges via M3 and M6, where MS is '0.' The required data is thus successfully written to the storage nodes.

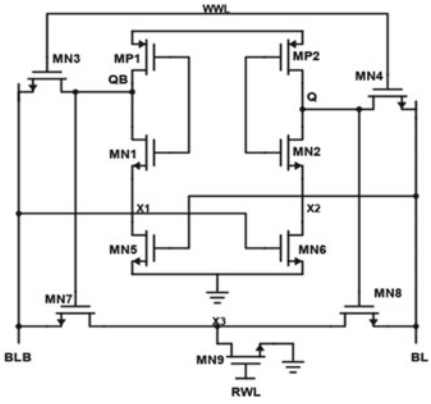
The BL and BLB are precharged to VDD before the read operation. Next, WLB is set to GND, and WL is set to VDD. This turns on the MS and transmission gates and



(a)



(b)



(c)

Fig. 1 Schematic of **a** 10 T SRAM cell with source-biased inverter, **b** TG10T SRAM cell, and **c** fully differential 11 T SRAM cell

discharges either of the bit lines to GND. The other remains at a precharged value of '1' initially in storage nodes QB and Q depending on the data stored (Krishna and Duraiswamy 2021). Assume that node QB stores '1' and node Q stores '0.' The additional pull-down path makes the read operation faster, thus improving the RSNM, which helps discharge the voltage at the Q node. During hold mode, WLB is VDD, and WL is GND to turn off the MS transistors and transmission gates. Thus, the cell core is disconnected from the bit lines, and no reading or writing takes place.

2.3 Fully Differential 11 T SRAM Cell

The 11 T cell demonstrated in Fig. 1c is made up of the smallest-sized devices for reducing area overhead up to a maximum extent. Here, the read buffer of the cell (MN7-MN9) consists of oversized transistors to achieve minimum read latency as well as maximum stability in the read operation which are driven by bit lines. The bit lines drive the gate electrodes of transistors MN 5 and MN 6 which are serially connected. The node X1 is high during the write operation. If 1 is to be written at QB, BLB is high, and BL is GND. The transistor MN 5 turns OFF. So, the pull-down path of the left inverter is disconnected, and node QB attains a higher value due to quick charging. For a stack of two devices, if the bottom device is OFF and the top device is ON, a voltage is built at the internal node. Hence, node X1 is raised. The VX1 (positive) offers body effect and stacking which helps in diminishing the delay of the write signal during a write operation on rapidly charging node QB. The write operation mainly consists of a write driver, pull-up devices, and access devices, while RWL is deactivated and WWL is activated. Furthermore, BL turns OFF transistor MN 5 disconnecting the pull-down path while 1 is written at QB. This improves the write-ability of the cell by making the write operation faster.

Before the read operation, bit lines are precharged, transistor MN 1 is ON, QB stores a 0 and Q stores a 1 (let us assume). The bitline_bar drops through transistor MN 1 and MN 3, when WL is activated. Before the read operation, QB and Q are separated by disconnecting WWL, 0 in QB turns transistor MN 7 OFF, and 1 in Q turns transistor MN8 ON, which increases the internal node voltage X3 to a value larger than 0. When the word line for read is activated node X3 offers a body effect to transistor MN 8.

Due to positive VX3, the body-to-source voltage of transistor MN 8 becomes negative. This results in an increase in V_{th} of MN 8 which causes the drive strength to decrease. So, the time taken to discharge BL is more. Hence, the body effect offered to transistor MN 3 is much smaller comparably. During hold mode, RWL and WWL are grounded to switch OFF the access transistor. Hence, the core of the cell is deactivated from bit lines, and no read or write operation occurs.

3 Simulation Results and Analysis

The LTSPICE tool was used to perform simulations by constructing schematics of all three SRAM cells for comparison. Tables 1, 2, and 3 show the transistor sizes used in the 10 T SRAM cell, TG10T SRAM cell, and 11 T SRAM cell.

Since every other transistor is the smallest size feasible, the overall area of the circuit is significantly smaller.

3.1 Transient Analysis

To ensure that the hold, read, and write operations worked correctly, transient analysis was performed on all three models. Similar read/write signals, identical input data signals, and precharge signals for read and write bit lines were employed to simulate the models. For all models, identical sense amplifiers, precharge circuits, and write

Table 1 Dimensions for 10 T cell

	Width (W)	Length (L)
P1, P3 (Pull-up transistors)	36n	32n
N1, N2, N3, N4 (Pull-down transistors)	72n	32n
N5, N6 (Access transistors)	36n	32n
P2, P4 (Buffer transistors)	72n	32n

Table 2 Dimensions for TG10T cell

	Width (W)	Length (L)
M2, M4 (Pull-up transistors)	128n	32n
M1, M3 (Pull-down transistors)	128n	32n
MN1-MP1, MN2-MP2 (Access transistors)	36n	32n
M5, M6 (Buffer transistors)	320n	32n

Table 3 Dimensions for 11 T cell

	Width (W)	Length (L)
MP1, MP2 (Pull-up transistors)	32n	32n
MN1, MN2, MN5, MN6 (Pull-down transistors)	32n	32n
MN3, MN4 (Access transistors)	32n	32n
MN7, MN8 (Buffer transistors)	50n	32n
MN9 (Buffer transistor)	64n	32n

drivers were adopted to ensure the reliability of performance comparison. The results of the simulation are demonstrated in Fig. 2.

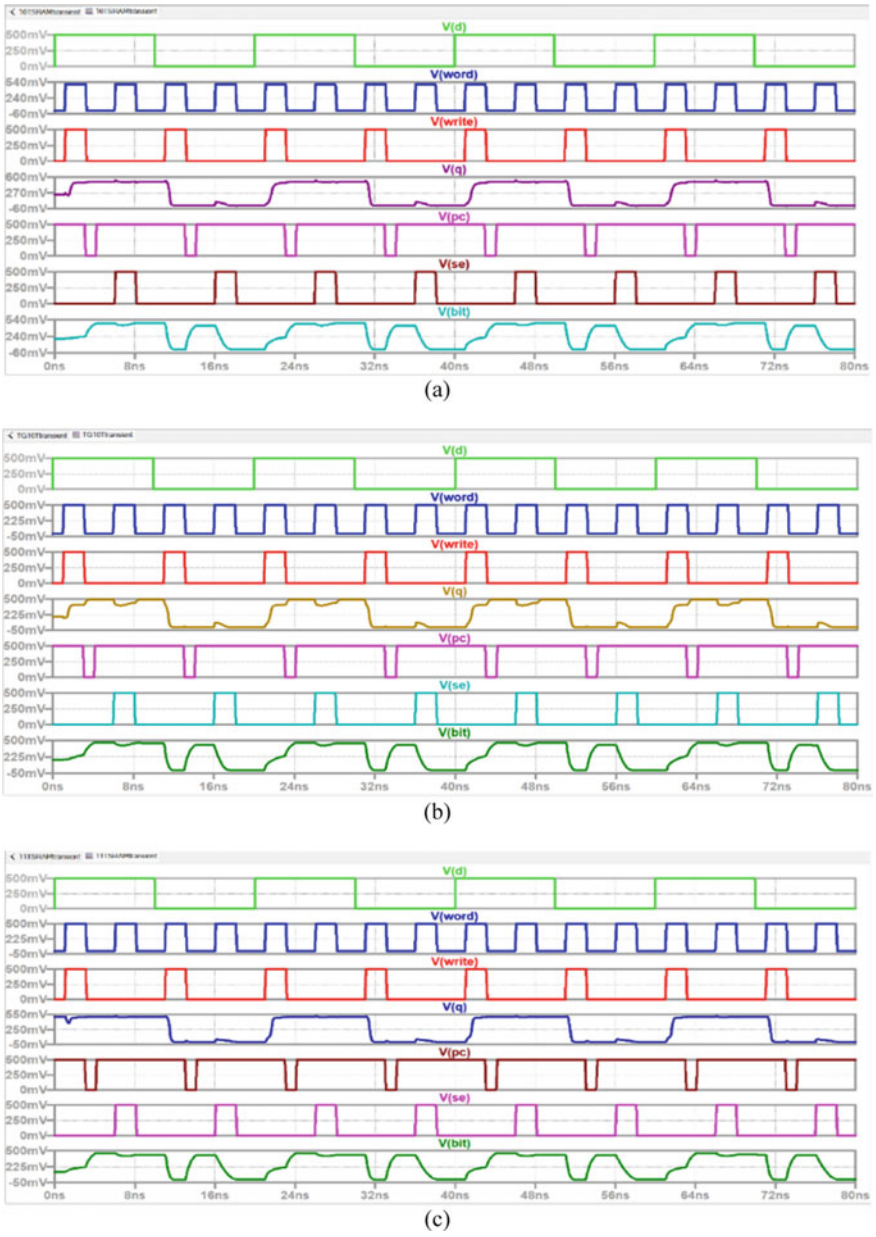


Fig. 2 Transient analysis of **a** 10 T SRAM cell, **b** TG10T SRAM cell, and **c** 11 T SRAM cell

According to the outcomes of the transient analysis, hold, read, and write operations are performed satisfactorily in all three models. Here, we observe when both word and write are high, it writes onto the cell, i.e., whatever data is there in D is transferred to Q. When both word and SE are high, it reads from the cell. So, whatever data is there in Q is reflected in the bit line.

3.2 Data Stability

The static noise margin is calculated for all three operating modes. It is used to evaluate the SRAM cell's data stability during the operations like hold, read, and write. SNM, which is pertinent to all three operating modes, is defined as noise that applies to storage nodes before the stored state gets reversed. The specifics of measuring SNM using butterfly curves are described in this study (Fig. 3).

3.3 Power Consumption

The analysis confers power consumption for different variants of SRAM cells. The analysis associated depicts that the power consumed is maximum in TG10T SRAM cell, whereas it is minimum in the case of 11 T SRAM cell. The power consumed by the TG10T cell (i.e., 233.69 nW) is approximately two times as compared to the 10 T SRAM cell (i.e., 108.65 nW) and 11 T SRAM cell (i.e., 88.491nW). As the technology node used in the analysis is 32 nm, the power consumption is very low, i.e., in nanowatts. Figure 7a depicts a graphical comparison of the three models in terms of power usage.

3.4 Read and Write Delay

The TG10T model features a lower read and write access delay than the other two models, making it possible to read and write data to it more quickly. The read delay and write delay comparison for different variants of the SRAM cell is shown in Fig. 7b. The analysis associated depicts that the read delay is maximum in 11 T SRAM (i.e., 568.5 psec) due to a greater number of transistors in it, whereas the write delay is maximum in case of 10 T SRAM cell with source-biased inverter (i.e., 525 psec). The analysis also portrays that both read and write delay is minimum in TG10T SRAM cells. The read delay is 343.3 psec and the write delay is 495 psec, respectively (Fig. 4).

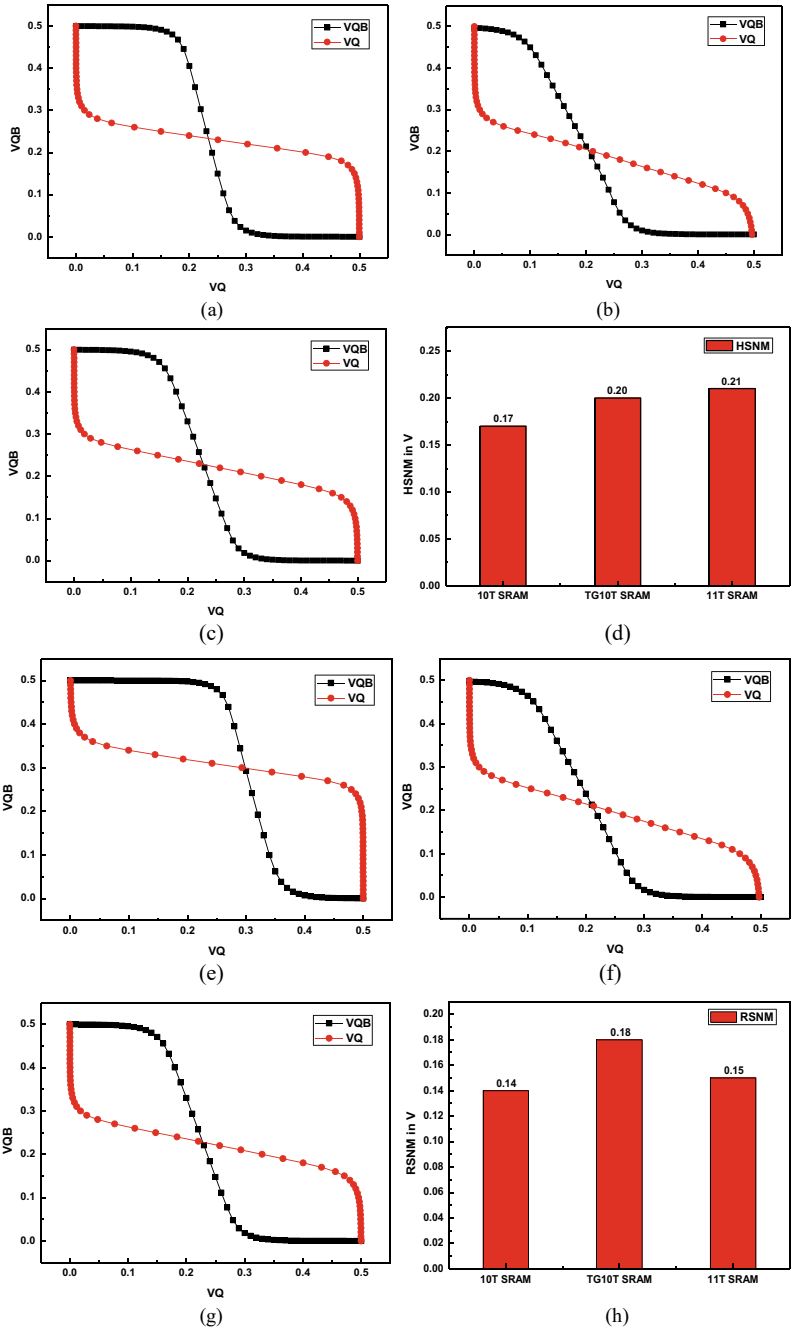


Fig. 3 Butterfly curves for HSNM **a** 10 T SRAM, **b** TG10T SRAM, **c**11 T SRAM, **d** HSNM comparison; butterfly curves for RSNM **e** 10 T SRAM, **f** TG10T SRAM, **g** 11 T SRAM, **h** RSNM comparison; butterfly curves for WSNM **i** 10 T SRAM, **j** TG10T SRAM, **k** 11 T SRAM, **l** WSNM comparison

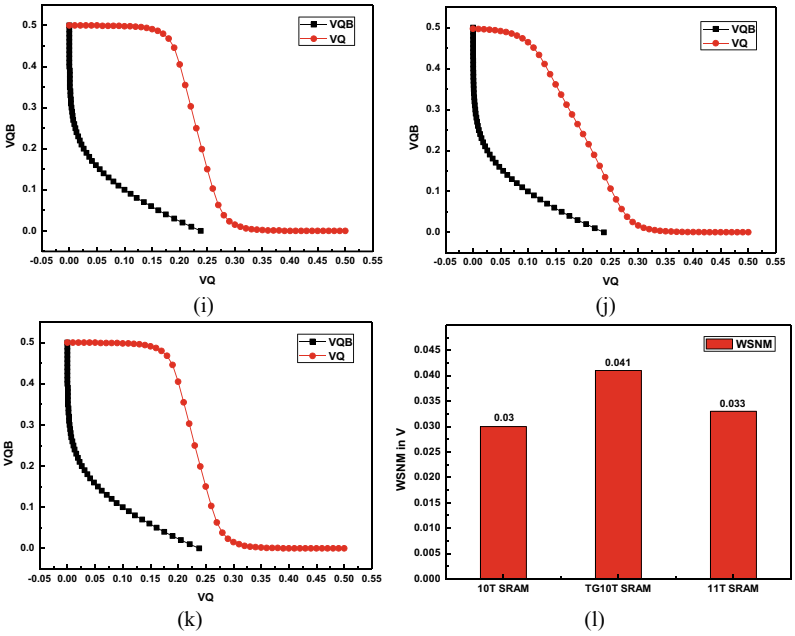


Fig. 3 (continued)

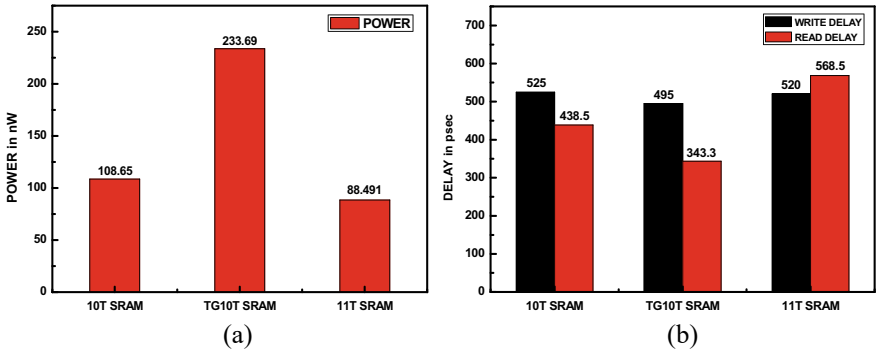


Fig. 4 Comparison of **a** power consumption and **b** delay

4 Conclusion

The TG10T SRAM cell is proven to be more enhanced in almost every aspect, but it consumes more power. Firstly, the TG10T SRAM cell considerably improves the stability of the data stored. This is primarily due to storage nodes being entirely cut from read bit lines. Secondly, compared to 10 T SRAM, the latency is minimized for both TG10T SRAM and 11 T SRAM. Thirdly, read stability may have significantly

increased as the TG10T SRAM cell has multiple bit line discharging paths. It boosts writability by replacing NMOS access transistors with transmission gates. Moreover, the TG10T SRAM cell has increased read and write access speeds. It is plausible to conclude that the 11 T model is much more compact than the 10 T model despite using one more transistor due to the deployment of small-sized transistors, demonstrated by Tables 1, 2, and 3 even though the comparison of the occupied area is not reproduced. The TG10T SRAM cell is proven to be more enhanced in almost every aspect but it consumes more power. The static noise margin for read and write operation is found to be highest in the TG10T SRAM cell. The power consumed by the TG10T cell (i.e., 233.69 nW) is approximately two times as compared to the 10 T SRAM cell (i.e., 108.65 nW) and 11 T SRAM cell (i.e., 88.491 nW). The analysis also portrays that both read and write delay is minimum in TG10T SRAM cells. The read delay is 343.3 psec and the write delay is 494 psec, respectively. Future developments could include adding extra PMOS transistors to the TG10T SRAM cell to improve performance and minimize leakage power usage.

References

Abbasian E, Izadinasab F, Gholipour M (2022) A reliable low standby power 10T SRAM cell with expanded static noise margins. IEEE Trans Circuits Syst I Regul Pap 69(4):1606–1616

Abbasian E, Gholipour M (2022) Robust transmission gate-based 10T subthreshold SRAM for internet-of-things applications. Semicond Sci Technol

Dutt D, Mittal P, Rawat B, Kumar B (2022) Design and performance analysis of high-performance low power voltage mode sense amplifier for static RAM. Adv Electr Electron Eng 20(3):285–293

Eslami N, Ebrahimi B, Shakouri E, Najafi D (2020) A single-ended low leakage and low voltage 10T SRAM cell with high yield. Analog Integr Circ Sign Process 105(2):263–274

Gavaskar K, Ragupathy US (2019) Low power self-controllable voltage level and low swing logic based 11T SRAM cell for high-speed CMOS circuits. Analog Integr Circ Sig Process 100(1):61–77

Islam A, Hasan M (2012) Variability aware low leakage reliable SRAM cell design technique. Microelectron Reliab 52(6):1247–1252

Jose AA, Balan NC (2016) Static noise margin analysis of 6T SRAM cell. In: Artificial intelligence and evolutionary computations in engineering systems. Springer, New Delhi, pp 249–258

Krishna R, Duraiswamy P (2021) Low leakage 10T SRAM cell with improved data stability in deep sub-micron technologies. Analog Integr Circ Sig Process 109(1):153–163

Mansore SR, Gamad RS, Mishra DK (2020) A 32 nm read disturb-free 11T SRAM cell with improved write ability. J Circ Syst Comput

Mittal P, Kumar N (2020) Comparative analysis of 90 nm MOSFET and 18 nm FinFET based different multiplexers for low power digital circuits. Int J Adv Sci Technol 29(8s):4089–4096

Nayak D, Acharya DP, Rout PK, Nanda U (2018) A high stable 8T-SRAM with bit interleaving capability for minimization of soft error rate. Microelectron J

Pourbakhsh SA, Chen X, Chen D, Wang X, Gong N, Wang J (2016) Sizing-priority based low-power embedded memory for mobile video applications. 17th international symposium on quality electronic design

Rawat B, Mittal P (2021c) Analysis of varied architectural configuration for 7T SRAM bit cell. In: Recent trends in communication and electronics. CRC Press, pp 204–209

Rawat B, Mittal P (2023) A low power single bit-line configuration dependent 7T SRAM bit cell with process-variation-tolerant enhanced read performance. Analog Integr Circ Sign Process 1–16

Rawat B, Mittal P (2021a) A 32 nm single-ended single-port 7T static random-access memory for low power utilization. Semicond Sci Technol 36(9):095006

Rawat B, Mittal P (2021b) Single bit line accessed high-performance ultra-low-voltage operating 7T static random-access memory cell with improved read stability. Int J Circ Theory Appl 49(5):1435–1449

Rawat B, Mittal P (2022a) A comprehensive analysis of different 7T SRAM topologies to design a 1R1 W bit interleaving enabled and half select the free cell for 32 nm technology node. Proc Royal Soc A 478(2259):20210745

Rawat B, Mittal P (2022b) A reliable and temperature variation tolerant 7T SRAM cell with single bit line configuration for low voltage application. Circ Syst Sign Process 41(5):2779–2801

Sharma V, Vishvakarma S, Chouhan SS, Halonen K (2018) A write improved low power 12T SRAM cell for wearable wireless sensor nodes. Int J Circ Theory Appl 46(12):2314–2333

Weste NH, Harris D (2015) CMOS VLSI design: a circuits and systems perspective. Pearson Education India

VLSI Floorplan Area Optimisation Technique

Mithilesh Kumar Lobiyal and Sukwinder Singh

Abstract Floorplanning is a crucial phase in the physical design. It estimates the overall chip area, the interconnects, and the latency by determining the shape, size, and placements of the modules in a chip. The VLSI floorplan problem has been addressed by several researchers using a variety of heuristics and metaheuristic techniques. An essential component of the floorplan stage is the floorplan representation. The VLSI floorplanning problem is surveyed in this paper, along with a comparison of the various optimisation algorithms (like Simulated Annealing, Particle Swarm Optimisation, Differential Evolution, Ant Colony Optimisation as optimisation algorithms) and the representations involved in the VLSI floorplanning problem.

Keywords Floorplanning · Simulated annealing · Particle swarm optimisation · Polished notation · Ant colony optimisation

1 Introduction

Recent technical advances have dramatically increased design complexity, and a chip today has an unfathomably high number of transistors. Physical Design has therefore taken on a significant role in the VLSI design flow. Floorplanning is the first phase in Physical Design Flow. Optimising the whole chip area, including the area used by components and interconnects, is the goal of floorplanning. It chooses the position and size of the modules inside a chip in order to maximise the chip area, making sure that no module overlaps another. Another objective is to determine the optimal wirelength while considering the trade-off between wirelength and area optimisation.

An essential component of the floorplanning stage is the floorplan representation. A wide range of representations, including Bounded Slicing Grid Structure (BSG),

M. K. Lobiyal (✉) · S. Singh
Department of Electronic and Communication, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India
e-mail: mithileshkl.vl.21@nitj.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_5

Polish Notation, Corner Block List (CBL), O Tree, etc., have been proposed by several academics. The Compacted floorplan, such as B* Tree, is fast and has the best computational complexity, even though the Sequence Pair gives the most flexibility (Chang et al. 2000).

Floorplan is regarded as being NP-hard (Sowmya and Sunil 2013). There are various published heuristic solutions to the floorplanning problem. As a result, several researchers have employed various optimisation techniques, such as Wong and Liu's use of Simulated Annealing to maximise floorplan area (Kahng et al. 2011). Simulated Annealing, which is being used today and includes an inbuilt optimiser in many floorplanners like Parquet, might be considered the first metaheuristic method to floorplanning. Many other algorithm are used by researchers like Debarshi Chatterjee uses GA and Sequence Pair representation (Murata et al. 1996), Rebandengo and Reorda (1996) uses the Genetic Algorithm, B* Tree, and PSO were used by Sun et al. (2006), G. Chen using Discrete PSO and Integer Coding representation (Chen and Guo 2008), Angeline using PSO, D. Jackulinei Moni uses a hybrid of and the B* tree representation, and PSO (Moni and Arumugam 2009). The Differential Evolution Method was another suggestion made by D. Jackuline Moni for fixed outline floorplanning (Moni et al. 2007). This paper will provide a survey of the research these scholars have done to address the VLSI floorplanning issue.

2 Problem Definition

A chip's modules are organised using a VLSI layout. A collection of x rectangular modules, $B = b1, b2, b3, \ldots, bx$, serves as the input for the floorplan. Each module has a rectangular shape having wi, hi, and ai are the width, height, and area, respectively. The goal of floorplanning is to arrange the modules on a chip so that no two modules are positioned next to one another and all other performance metrics, such as area and wirelength, are at their best.

2.1 Types of Modules

A module can be of two different types:

Hard module (Kahng et al. 2011; Moni 2009): Hard module have fixed dimensions.

Soft module (Kahng et al. 2011; Moni 2009): Soft module have fixed aspect ratio but height and width may vary.

2.2 Types of Floorplan

A floorplan can be of two different types:

Slicing (Kahng et al. 2011; Moni 2009): These floorplans may be broken down into a smaller portion until it consists of just one module. The Slicing Tree, a binary tree that includes modules at the leaves and the inner nodes' cut types H and V which is horizontal and vertical cut respectively, can be used to represent them.

Non-Slicing or Mosaic (Kahng et al. 2011; Moni 2009): There is no way to separate these floorplans into separate sections. Because of this, we utilise the Constraint Graph Pair to describe them. Both the Horizontal Constraint Graph and the Vertical Constraint Graph, which define the horizontal and vertical interactions between the modules, are parts of this pair.

2.3 Floorplaning Cost

A cost function that combines area and wirelength and is commonly used in floorplanning is given by Eq. (1)

$$\text{Cost} = \alpha A/A_{\text{avg}} + (1-\alpha)\, W/W_{\text{avg}} \tag{1}$$

where α is weight factor, the floorplan area is A, and the total wirelength is W. The average wirelength is W_{avg}, while the average area is A_{avg}.

3 Different Types of Floorplan Representation

Floorplan representation is an important aspect in floorplanning. The complexity of the transition between a representation and its associated floorplan is determined by it, as well as the size of the search space. A floorplan must first be represented using an appropriate representation scheme before being taken into account for any optimisation. In this section we will discuss some of the representation scheme.

3.1 Sequence Pair

A sequence pair is an ordered pair of $P+$ and P, each of which represents a permutation of the names of a certain module. Using the six modules as an example, (abcdef, cbfade) can describe a floorplan. To establish a pair of sequences, the constraint graph pairs for the horizontal and vertical constraints must first be constructed.

3.2 Polish Notation

Slicing floorplans are modelled using Polish notation. Polish expressions can be used to record the Binary Slicing Tree. Thus, each number indicates a module, while the letters *H* and *V* stand for horizontal and vertical cuts, respectively, in the Slicing Floorplan. A binary tree's postfix ordering, which can be discovered by utilising post-order traversal, is the Polish expression (Rebaudengo and Reorda 1996). To transform a normalised Polish expression into its equivalent floorplan, we might use a bottom up approach based on Polish expression to recursively combine the sliced sub-floorplans.

3.3 Bounded Slicing Grid

Nakatake et al. (1996) developed the grid style for constrained slicing in 1996. A Bounded Slicing Grid is constructed by generating two-unit long, non-overlapping horizontal line segments in each row, with an offset of one unit long between consecutive rows. In the same way, vertical line segments are also created, and these lines are referred to as Bounded Slice Lines, or BS Lines (Nakatake et al. 1996). A room is the rectangular area bounded by four BS lines. By allocating the blocks to these rooms, a floorplan is depicted, and this is known as a Bounded Slicing-seed.

On the basis of the BSG structure, Kang and Dai (1997) created a new greedy algorithm that runs in linear time.

3.4 B* Tree

In order to depict condensed floorplans, B* Trees are built on ordered binary trees. No modules can be shifted to the floorplan's left or bottom in a compressed floorplan (Fig. 1).

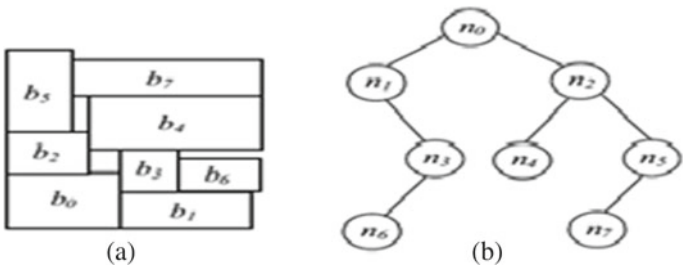


Fig. 1 Floorplan and with its B* tree by Sun et al. (2006)

The B*-tree's root is represented by block b0 in the placement's bottom left corner, which is indicated by the number n0. A B*-tree is constructed starting at the root, then the left subtree, and finally the right subtree. The lowest, unexplored block in the floorplan belongs to the node ni's left child. The ni's right child, whose x-coordinate is the same as bi's and whose y-coordinate is smaller than the top boundary of the module on the left and next to it, represents the lowest block. The solution space consists of all B*Tree combinations produced by perturbing the tree using the three operations Node: Movement operation (Lin and Hung 2011), Block rotation operation, and Node swap operation.

4 Floorplan Algorithm

As we've already mentioned, the primary purpose of a floorplan is to identify the placement of modules to reduce both the overall area and length of connection wires. The layout of any circuits was first designed using conventional methods, such as and cluster growth method (Kahng et al. 2011) and linear ordering, which were predicated on the notion of incrementally adding blocks along a path. Many optimising algorithms, such as Simulated Annealing, Genetic Algorithm, Ant Colony Optimisation, Particle Swarm Optimisation are now used by researchers. Some of the floorplan algorithm will be covered in this section.

4.1 Simulated Annealing (SA)

One of the earliest methods utilised in all phases of VLSI physical design is simulated annealing (Kahng et al. 2011). It is used in many floorplan problem with different floorplan representation like B*-tree, Polish notation, etc. SA is comparable to the annealing process used to cool molten metal. The mobility of the atoms in metal is unrestricted at higher temperatures but is constrained as the temperature drops. The atoms then gradually begin to arrange, leading to the formation of crystals. The main disadvantage of SA is its computational complexity and it give optimal solution but not stable.

4.2 Genetic Algorithms

A relatively simple approach called a genetic algorithm is utilised to handle floorplanning issues like area and wirelength optimisation. The whole population of chromosomes is taken into account initially in genetic algorithms (Chen and Zhu 2010; Chen et al. 2005). The chromosome that provides the best solution is chosen from

among them. A better solution is defined for the existing population. The fitness function is determined depending on a number of floorplan variables, including block height, the quantity of modules, etc. This holds true for both slicing and non-slicing floorplans. As the initial population, a sample of randomly selected people is first collected. Afterwards, each person's level of fitness is assessed, Using the supplied Polish notation, we can determine how many crossings there are. All of these crossings are considered to be genetic mutations since they modify the genetic structure. Using the mutation rate, a bit is selected for a population unit. A fully new unit is added to the population as a result of flipping the chosen bit. Each iteration's cost is calculated for each population. The algorithm halts and outputs the current population as the best outcome if the cost is the lowest.

4.3 Particle Swarm Optimisation (PSO)

Swarm intelligence serves as the basis for the optimisation technique known as PSO (Hoo et al. 2013). The population-based evolutionary algorithm PSO's starting population is made up of random solutions. The PSO method refers to the population as a swarm, and each individual inside it is referred to as a particle. PSO begins with random particles that, as they wander across the search space, individually recall their neighbours and own prior-best placements. The local best (Pbest) of each particle, which is the optimal value for each individual swarm particle, and used to update each particle's state at the end of each iteration.

4.4 Ant Colony Optimisation

Ant colony optimisation, is a successful technique for resolving difficult optimisation problems. The initialisation, the construction, and the feedback phases make up the three stages of the ant-based algorithm. Setting parameters like the number of colonies and ants takes place during the initialisation step, which is the first stage. The following stage, known as the construction stage, is building a route based on pheromone concentration and the last stage is known as the feedback stage, is concerned with extracting and reinforcing the ants' travelling knowledge acquired on the previous search journey.

5 Results and Discussion

According to the above table, Apte has nine modules, Xerox has ten, and Ami33 has Thirty Three modules and Ami49 has Forty Nine modules. We examined a number of floorplan formats and algorithms put forward by different academics (Table 1).

Table 1 Area optimisation result

	Apte	Xerox	Ami33	Ami49
PSO with B*tree	46.92	19.55	1.28	41.01
PSO with sequence pair	48.12	20.69	1.31	38.84
SA with B* tree	47.3	20.47	1.36	43.34
SA with sequence pair	48.12	20.69	1.31	38.84
ACO	46.9	19.8	1.20	37.2

We may infer from the above table that, for various benchmark circuits, the ACO method outperforms other techniques in terms of area optimisation. PSO algorithm delivers positive results after ACO algorithm.

6 Conclusion

A NP-hard physical design issue called the floorplan problem was examined in this paper. In order to minimise the chip's size and wirelength, modules are arranged and placed. In this work, we explore a several floorplan representations, including Bounded Slicing Grid, Sequence Pair, Polish Notation, and B* Tree. ACO Algorithm provides the best results in area optimisation, as seen in the table. Taking second place is the PSO Algorithm with B*-tree.

References

Chang Y-C, Chang Y-W, Wu G-M, Wu SW (2000) B*-trees: a new representation for non-slicing floorplans. In: Design automation conference, pp 458–463

Chen G, Guo W, Cheng H, Fen X, Fang X (2008) VLSI floorplanning based on particle swarm optimization. In: Proceedings of 2008 3rd international conference on intelligent system and knowledge engineering

Chen J, Zhu W (Oct 2010) A hybrid genetic algorithm for VLSI floor-planning. IEEE Trans Intell Comput Intell Syst 2:128–132, 29–31

Chen GL, Guo WZ, Tu XZ, Chen HW (2005) An improved genetic algorithm for multi-objective optimization. ISICA'2005: progress in intelligent computation and its applications, pp 204–210

Hoo C-S, Jeevan K, Ganapathy V, Ramiah H (July, 2013) Variable-order ant system for VLSI multiobjective floorplanning. Appl Soft Comput 13(7)

Kahng A, Leinig J, Markov IL, Hu J (May 2011) VLSI physical design: from graph partitioning to timing closure

Kang M, Dai WWM (1997) General floorplanning with Lshaped, T-shaped and soft blocks based on bounded slicing grid structure. Proceedings of design automation conference 1997 Asia and South Pacific. IEEE, pp. 265–270

Lin J-M, Hung Z-X (2011) SKB-tree: a fixed-outline driven representation for modern floorplanning problems. IEEE Trans VLSI Syst 20(3):473–484

Moni DJ (June 2009) Certain optimization techniques for floorplanning in VLSI physical design. Ph.D. thesis, faculty of information and communication engineering, Anna University

Moni DJ, Arumugam S (2009) VLSI floorplanning based on hybrid particle swarm optimization. Karunya J Res 1(1):111–121

Moni DJ, Arumugam S, Rani GN (2007) VLSI Floorplanning relying on differential evolution algorithm. ICGST Int J Artif Intell Mach Learn 7(1):62–67

Murata H, Fujiyoshi K, Kajitani Y (1996) VLSI module placement based on rectangle-packing by the sequence-pair. IEEE Trans Comput-Aided Des Integr Circ Syst 15

Nakatake S, Fujiyoshi K, Murata H, Kajitani Y (1996) Module placement on BSG-structure and IC layout applications. In: Proceedings of 1996 IEEE/ACM, international conference on computer aided design, pp 484–491

Rebaudengo M, Reorda MS (1996) GALLO: a genetic algorithm for floorplan area optimization. IEEE Trans Comput Aided Des Int Circ Syst 15(8):943–951

Sowmya B, Sunil MP (Aug 2013) Minimization of floorplanning area and wire length interconnection using particle swarm optimization. Int J Emerg Technol Adv Eng 3(8)

Sun T-Y, Hsieh S-T, Wang H-M, Lin C-W (2006) Floorplanning based on particle swarm optimization. In: Proceedings of the 2006 emerging VLSI technologies and architectures (ISVLSI'06)

NG-PON3-Based Bidirectional High-Speed PON with 1:32/64/128 Split Ratio

Meet Kumari

Abstract Next generation passive optical network (NG-PON) is extensively regarded as the highest demanding optical access network result taking into account it's high-speed and cost-effective services to individuals as well as enterprise users. In this paper, a NG-PON stage-3-based bidirectional PON with different split ratio has been proposed. The results indicate that the proposed PON offers high-speed data rate upto 80 Gbps over 10 km distance successfully for 1:32 split ratio. Also, the transmission distance can be extended upto 40 km at 10 Gbps data rate with split ration of 32. Moreover, the maximum signal to noise ratio of 18 dB can be obtained with 10 Gbps data rate over 10 km fiber.

Keywords Next generation passive optical network (NG-PON) · Bit error rate (BER) · Split ratio

1 Introduction

Future proof option for service operators is to minimize the birthright networks to different optical access networks. However, in some obtuse urban areas, the operators have previously developed access networks, primarily Ethernet and Gigabit-based passive optical networks (PONs), where maximum 2.5 Gbps data rate is shared by various tens of end users using time division multiplexing (TDM). Passive optical network (PON) is a future proof access network which provide enormous merits if realized in fiber to the home scenarios. The merits consist of a high quality, cost-effective, point-to-multipoint triple, and high-speed access network effectively (Arya et al. 2022; Chaudhary et al. 2018; Kumari and Arya 2022). Also, to further enhance the capacity of access network per consumer, operators can either minimize the sharing factor or shift toward next generation PON stage 2 (NG-PON2) which provide maximum data rate of 40 Gbps over 40 km distance. But for the future access network,

M. Kumari (✉)
Chandigarh University, Mohali, Punjab, India
e-mail: meetkumari08@yahoo.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_6

NG-PON stage 3 (NG-PON3) with ≥40 Gbps data rate over long-reach is required (Kumari et al. 22021a, 2021c; Mas Machuca 2015; Singh et al. 2021).

Even though the attempts to accommodate these schemes to handle the NG-PON2 requirements, issues such as high capacity, minimizing the cost, power saving, expanding the reach, and need to be implemented further. Also, recently, in several previous work based on NG-PON (Chaudhary et al. 2018; Kumari and Arya 2022; Kumari et al. 2021b; Veen et al. 2019), limited data rate with restricted transmission range has been received. Thus in this paper, a NG-PON3-based high-speed bidirectional PON with different split ratio has been proposed.

Paper organization: Sect. 2 describes the proposed work of PON. Results and analysis in Sect. 3. Then conclusion is drawn in Sect. 4.

2 Proposed System

The block diagram of NG-PON3-based bidirectional PON is shown in Fig. 1.

The design of proposed bidirectional PON with distinct split ratio consists of different downstream and upstream wavelengths, i.e., 1596–1598.4 nm and 1527–1529.4 nm, respectively, having 0.8 nm spacing. Here, each transmitter includes random sequence with non-return to zero data format signals at 10–80 Gbps data rate to be modulated by Mach Zehnder modulator with input wavelength signals from laser sources. Also, receiver section consists of photodector, filter, and analyzer to

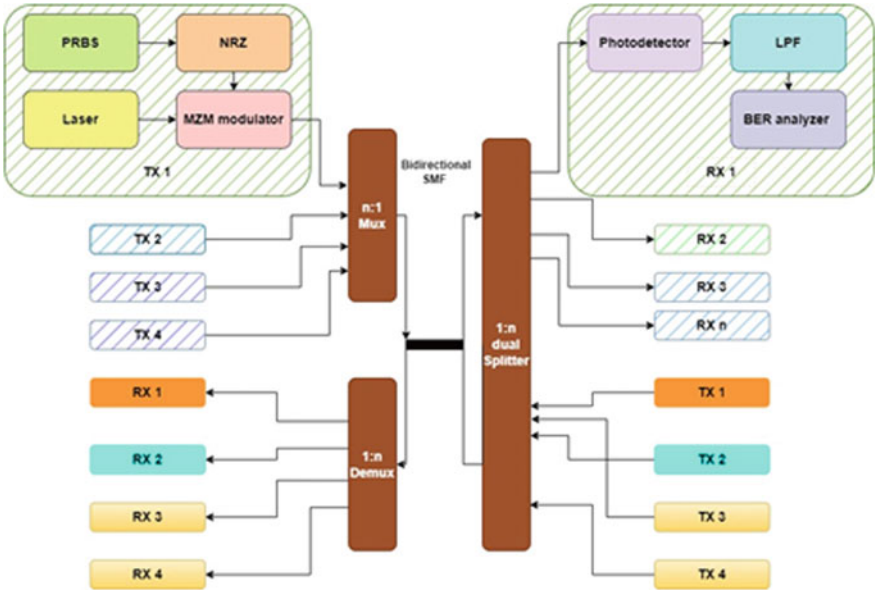


Fig. 1 Block diagram of NG-PON3-based bidirectional PON with different split ratio

Table 1 Parameters used

S.No.	Parameter	Unit
1	Input power	7 dBm
2	Traffic rate	10 Gbps
3	Distance	50 m
4	MZM extinction ratio	30 dB
5	Dark current	9nA
6	Thermal noise	10^{-23} W/Hz
7	Responsitivity	0.8 A/W

provide electrical signal analysis in terms of bit error rate (BER) and Q-factor (Arya et al. 2022; Kumari et al. 2021a, 2021c). Parameters incorporated in the proposed system are tabulated in Table 1.

3 Results and Discussion

The performance of the proposed PON is measured in OptiSystem software.

Figure 2 presents the performance of the proposed PON with different split ratio in terms BER versus distance at 10 Gbps data rate for both upstream as well as downstream direction. It is seen that the maximum achieved transmission distance at 10^{-3} BER limit is >40 km for upstream for different split ratio. While for downstream transmission distance for 32, 64, and 128 split ratio is 40, 35, and 30 km, respectively, at BER limit. Thus, the upstream performs better than downstream transmission over varied distance for all split ratio. Also, it is depicted that proposed PON performs for 1:32 split ratio followed by 1:64 and 1:128.

Figure 3 represents received power vs. data rate of the proposed PON with different split ratio over 10 km distance for both upstream as well as downstream direction. It is noted that the highest achieved data rate in upstream >80 Gbps for all split ratio at −3 dBm received power limit. Also, for downstream transmission the maximum obtained data rate is 80, 70, and 60 Gbps for 32, 64, and 128 split ratio, respectively. Therefore, PON offers better data rate performance in upstream than downstream for all split ratio and out of all split ratio, 1:32 shows best performance with high received power.

Figure 4 depicts Q-factor vs. SNR of the proposed bidirectional PON with different split ratio over 10 km fiber at 10 Gbps data rate. It is noted that the maximum SNR obtained for 32, 64, and 128 split ratio is 15, 10, and 6 dB respectively in upstream, at Q-factor limit of 6. Also, the maximum obtained SNR in downstream for 32, 64, and 128 split ratio is 18, 17, and 16 Db, respectively, at Q-factor limit. Besides this it is noted that with increase in SNR, the system performance also increases with best performance by 1:32 split ratio.

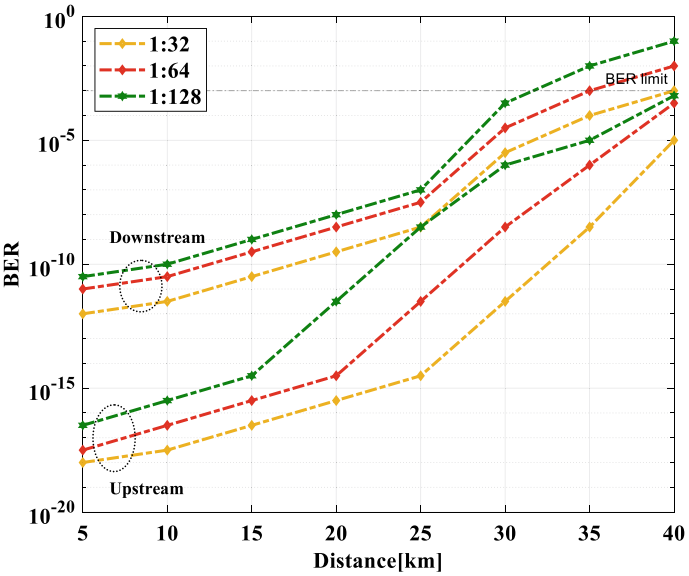


Fig. 2 BER versus distance of proposed bidirectional PON with different split ratio

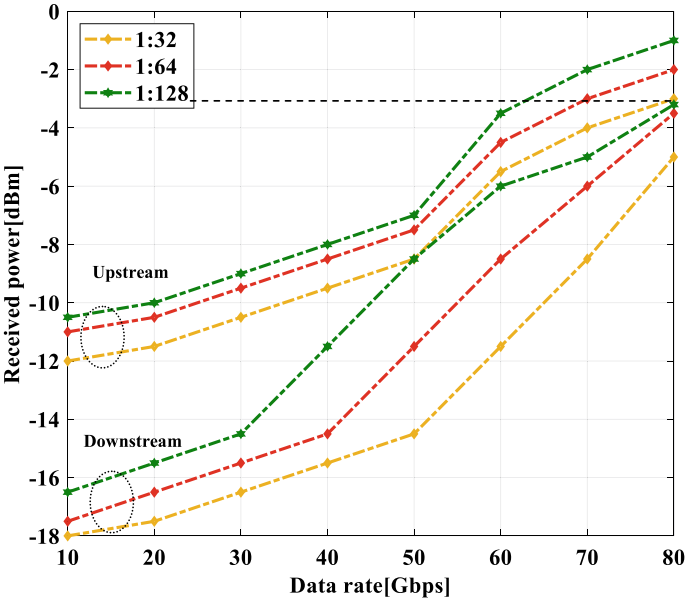


Fig. 3 Received power versus data rate of proposed bidirectional PON with different split ratio

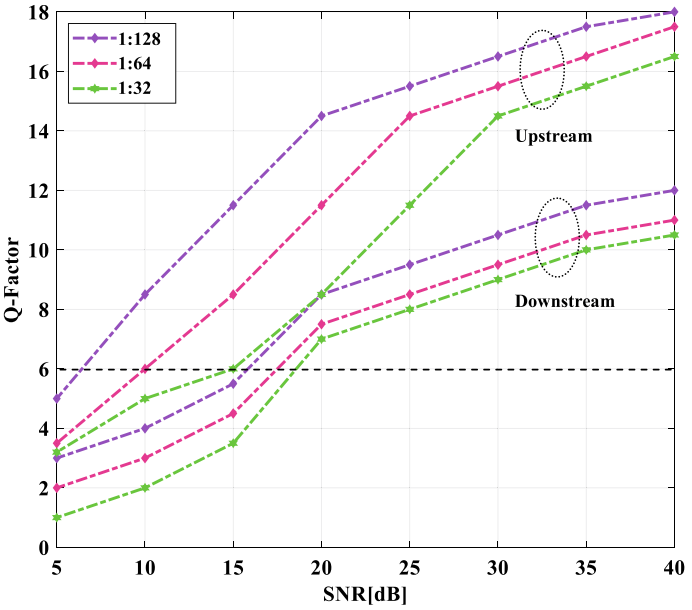


Fig. 4 Q-factor versus SNR of proposed bidirectional PON with different split ratio

4 Conclusion

In this work, an 80/80 Gbps PON with 1:32/64/128 split ratio based on NG-PON3 has been proposed. It is concluded that the proposed PON offer faithful transmission range of 40 km in both uplink and downlink transmission at 10 Gbps for 32 split ratio. Also, high SNR of 18 and 15 dB in uplink and downlink transmission respectively can be obtained. NG-PON3-based PON can be used for various rural to urban areas applications.

References

Arya V, Choubey H, Sharma S, Chen T (2022) Image enhancement and features extraction of electron microscopic images using sigmoid function and 2D-DCT. IEEE Access 10:76742–76751

Chaudhary S, Tang X, Wei X (2018) Comparison of Laguerre-gaussian and donut modes for MDM-WDM in OFDM-Ro-FSO transmission system. AEU—Int J Electron Commun 93:208–214

Kumari M, Arya V (2022) Investigation of high-speed hybrid WDM-OCDMA-PON system incorporating integrated fiber-FSO link under distinct climate conditions. Opt Quantum Electron 54(12):1–29

Kumari M, Sheetal A, Sharma R (2021b) Performance analysis of a full-duplex TWDM-PON using OFDM modulation with red LED visible light communication system. Wirel Pers Commun 119(3):2539–2559

Kumari M, Sheetal A, Sharma R (2021c) Performance analysis of symmetrical and bidirectional 40 Gbps TWDM-PON employing m-QAM-OFDM modulation with multi-color LDs based VLC system. Opt Quantum Electron 53(8):1–29

Kumari M, Sharma R, Sheetal A (2021) A hybrid next-generation passive optical network and visible light communication for future hospital applications. Optik (Stuttg) 242:166978

Mas Machuca C, Wosinska L, Chen J (2015) Assessment methodology of protection schemes for next generation optical access networks. Opt Fiber Technol 26:82–93

Singh M, Grover A, Kumari M, Sheetal A, Sharma R, Malhotra J (2021) A hybrid wavelength-mode division multiplexing-based inter-satellite optical wireless communication link. Optoelectron Adv Mater Rapid Commun 15(9–10):448–458

Van Veen D, Houtsma V, Porto S, Basavanhally N, Bolle C, Schmuck H (2019) Wavelength-stable burst-mode laser for next-generation PONs. J Opt Commun Netw 11(2):A155–A165

All-*p*-Type Digital Circuits Using Single Gate and Double Gate Organic Field Effect Transistors

Vidhi Goswami, Brijesh Kumar, and Richa Yadav

Abstract This paper presents comparative analysis of electrical behavior of double gate and single gate organic field effect transistors (OFETs). Charge carrier modulation makes double gate (DG) organic transistor a better choice for electronic equipments due to the presence of two conducting channels. A significant enhancement of 46%, 67%, and 40% is obtained in transconductance, ON current, and mobility, respectively. Further, this paper compares two configurations of all-*p*-type organic inverter circuits, i.e., zero V_{gs} load logic (ZVLL) and diode load logic (DLL). Their electrical responses are compared to analyze the effect of both the devices. Zero V_{gs} load logic configuration produces a maximum voltage gain (A_v) of value 16 using DG-OFET while SG-OFET configuration yields a maximum A_v of 6.27. Substantial improvement in propagation delay of 66% in DLL logic and 53% in ZVLL logic using DG as compared to SG-OFET are achieved. Furthermore, 2-input NOR gates are also implemented using same inverter circuits and their characteristics are compared. In NOR gate design, DG ZVLL configuration outperforms other configurations.

Keywords Organic · Double gate · DLL · ZVLL · Inverter · NOR

1 Introduction

Uncomplicated fabrication process and distinguished applications in the arena of lower cost, large size, wide area foldable screens, and modern electronic equipments make organic field effect transistors (OFETs) a valuable component in the field

V. Goswami (✉) · B. Kumar · R. Yadav
Indira Gandhi Delhi Technical University for Women, Delhi 110006, India
e-mail: igdtuwvidhi@gmail.com

B. Kumar
e-mail: brijeshkr@igdtuw.ac.in

R. Yadav
e-mail: richayadav@igdtuw.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_7

of organic electronics (Kumar et al. 2014; Tripathi et al. 2020; Yang et al. 2023). Recent studies (Baruah and Baishya 2023; Maurya et al. 2022; Negi et al. 2021, 2019a, 2018, 2019b; Wang et al. 2023) have come up with remarkable applications of OFETs in light-sensors, OLEDs, organic photovoltaic, smart cards, RFID tags, memory devices, and flexible/rollable displays. Exceptional advantages of using organic materials are mechanical flexibility and easier processing at room temperature. Enormous research work is being performed in the area of OLEDs and its variations (Negi et al. 2019a, 2018, 2019b). Main applications of a transistor are as amplifier and switch. Value of V_{th} (threshold voltage) determines the performance of a transistor as a switch. V_{th} is the voltage bias applied at gate terminal required to switch the transistor from a low conducting depletion region to a high conducting accumulation region. V_{th} is the property of a device dependent upon the doping which cannot be changed after its fabrication. A technology to externally control V_{th} can lead to a highly regulated operation of an electronic device. In single gate (SG) OFETs, V_{th} can be modified by changing the dielectric material or using a gate material of a distinct work function. An alternate solution to this is the use of double gate (DG) OFETs (Sekitani et al. 2005).

2 DG-OFET and Its Operating Mechanism

2.1 Double Gate Organic Field Effect Transistor (DG-OFET)

Double gate (DG) OFETs possess a top gate and its insulator layer; source (S) and drain (D) contacts; and bottom gate and its dielectric (Fig. 1). Charge carrier distribution accumulated by first gate in the channel can be electrostatically modified by biasing the second gate. Therefore, V_{th} can be regulated using second gate but at the expense of another gate layer and additional fabrication process (Sekitani et al. 2005). A lot of research and studies (Baruah and Baishya 2023; Cui and Liang 2005; Goswami et al. 2012; Horowitz and Hajlaoui 2001; Sekitani et al. 2005) have already been reported for many advantages of DG-OFETs such as higher ON current, threshold voltage control, and higher transconductance. A comparative analysis of electrical performances of SG- and DG-OFETs and their applications in digital circuit design has already been published in Goswami et al. (2012).

2.2 Operating Mechanism of DG-OFET

A DG-OFET operates in three modes viz a viz top gate mode, bottom gate mode, and double gate mode based on the biasing applied at the two gate electrodes. In the first two modes, a DG-OFET operates as a SG device with one gate biased and another kept to ground (Cui and Liang 2005). Threshold voltage, V_{th} of device can

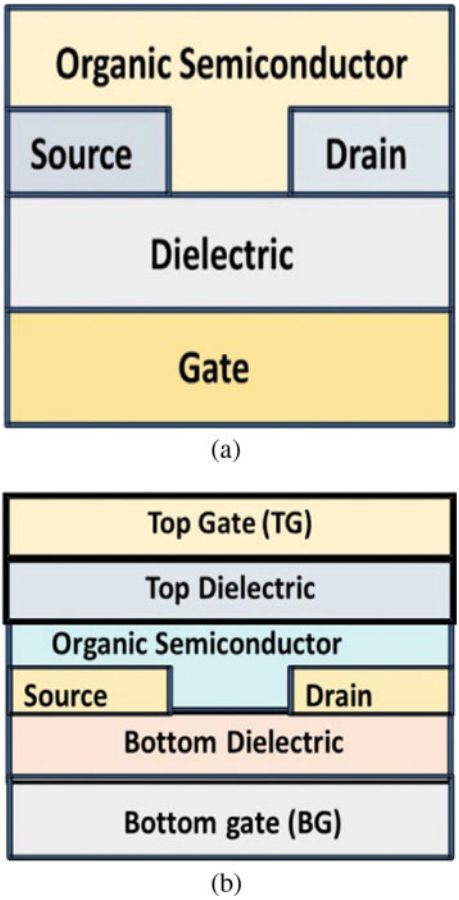


Fig. 1 Schematic front view of the device **a** SG-OFET **b** DG-OFET

be modified by changing the voltage at second gate (Sekitani et al. 2005). Numerous advantages of DG-OFETs such as increased gate modulation, steeper subthreshold slope with a higher on–off current have been reported (Cui and Liang 2005). Double gate devices outperform single gate devices due to reduced leakage current and enhanced device stability (Goswami et al. 2012). The structure of a DG transistor is shown in Fig. 2. The biases on the top and the bottom gate electrodes determine the current in the DG-OFET at any particular drain voltage.

Software Used and Simulation Conditions

The SG and DG device structures are designed using Silvaco Atlas 2-D simulator. Organic semiconductor layer (OSC) is implemented using pentacene, a p-type organic semiconductor (Cui and Liang 2005). Circuits are simulated using standard software package, i.e., Atlas/Silvaco, taking into consideration the effects of the

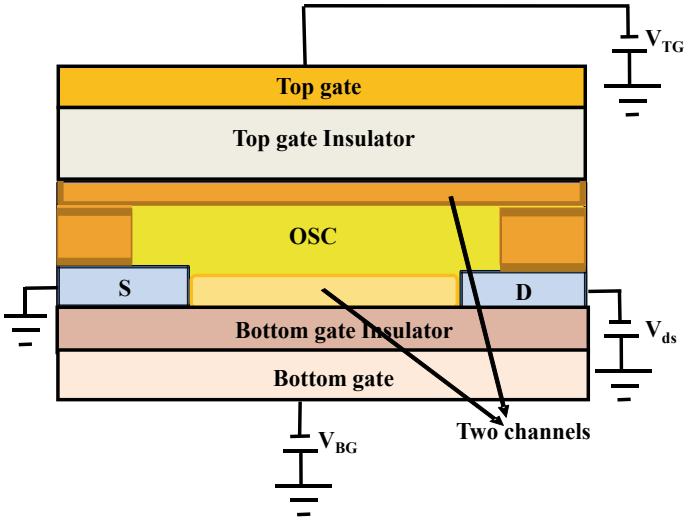


Fig. 2 Pictorial representation of two channels in a DG-OFET

Poole-Frenkel model and Gaussian distribution. Although there is no direct consideration for the effects of morphology of the materials used and the traps, but the zero-field mobility (μ_0) term in the expressions in these models accounts for the dependence of current mechanism in transistors on morphology of the organic materials (Horowitz and Hajlaoui 2001).

3 All-*p* Digital Circuits Using DG-OFET

3.1 DLL and ZVLL Inverter Circuits

In this segment, electrical characteristics of two implementations of all-p-type inverters, diode load logic, and zero V_{gs} load logic (DLL and ZVLL) are compared (Goswami et al. 2013). The pull-down *n*-type transistor in a standard complementary MOS inverter is substituted by a *p*-type organic transistor as represented in Figs. 3a, b and 4a, b.

3.2 NOR Gate Circuit

2-input NOR gates are also implemented using these inverter circuits. V_1 and V_2 are the two inputs and V_{OUT} is the output voltage. Their performances are compared on the basis of the transfer characteristics obtained when one of the input (V_1 or V_2) is

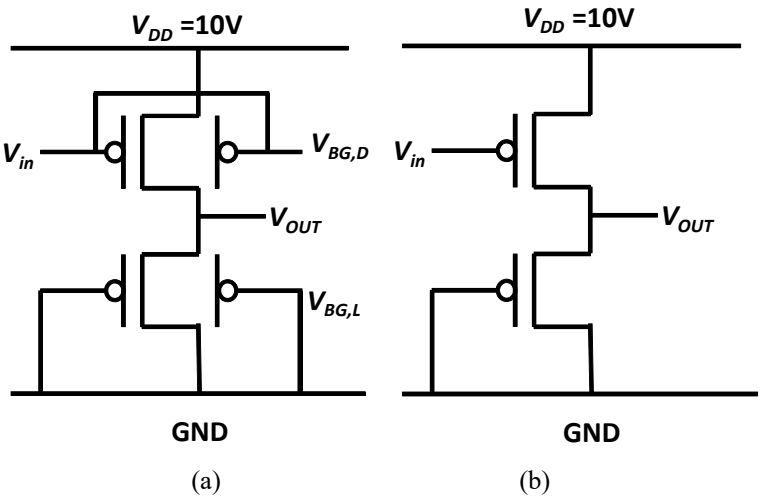


Fig. 3 Circuit diagrams of inverters in DLL configurations **a** SG-OFET and **b** DG-OFET

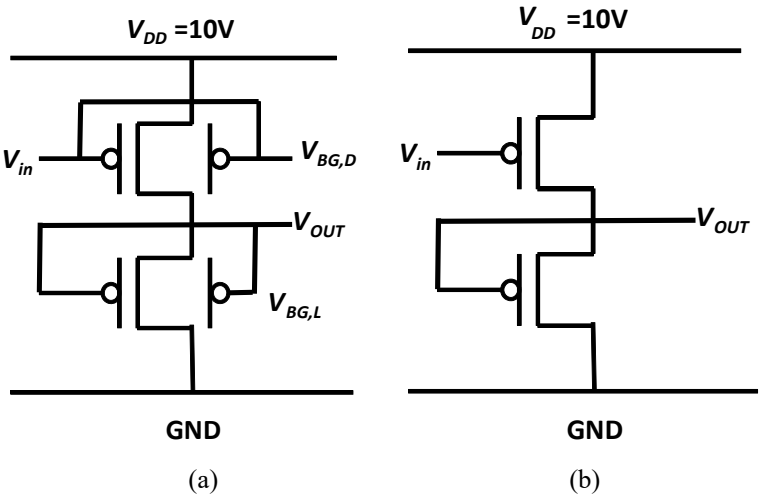


Fig. 4 Circuit diagrams of inverters in ZVLL configurations **a** SG-OFET and **b** DG-OFET

kept at logic "0" and the other one is swept from 0 to 10 V. A NOR gate should behave as an inverter in such a case. Fig. 5a and b show the circuit diagrams of NOR gate in DLL configurations using SG-OFET and DG-OFET, respectively. Figure 6a and b show the circuit diagrams of NOR gate in ZVLL configurations using SG-OFET and DG-OFET, respectively.

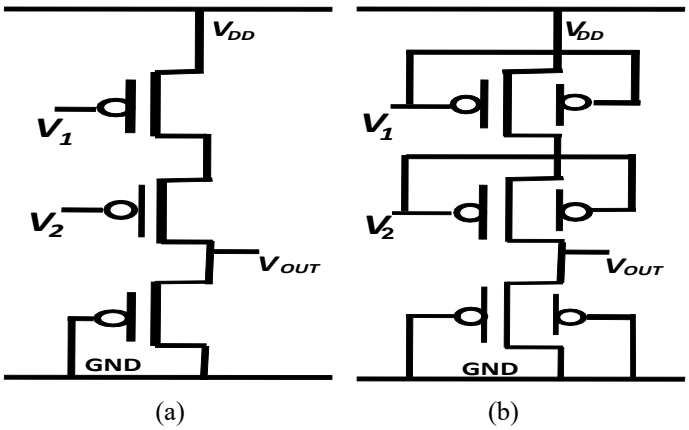


Fig. 5 Circuit diagrams of NOR gate in DLL configurations **a** SG-OFET and **b** DG-OFET

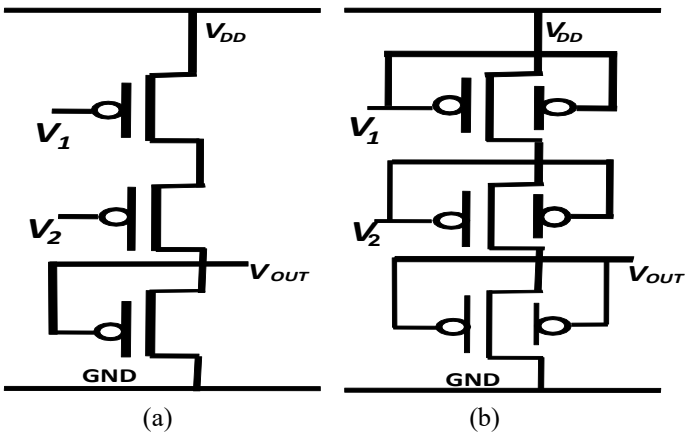


Fig. 6 Circuit diagrams of NOR gate in ZVLL configurations using **a** SG-OFET and **b** DG-OFET

4 Digital Circuit Analysis Using SG-OFET and DG-OFET

4.1 SG-OFET and DG-OFET Devices Characteristics

It can be seen from the input characteristics shown in Fig. 7 that a DG mode results in a higher output current and lower threshold voltage than a SG-OFET.

Figure 7 shows the input characteristics of DG and SG transistors in continuous scale at drain-source voltage, $V_{ds} = -2.5$ V. Drain characteristics for single gate and double gate OFETs are demonstrated in Fig. 8a and b. Electrical performance parameters obtained by simulating the devices in ATLAS are represented

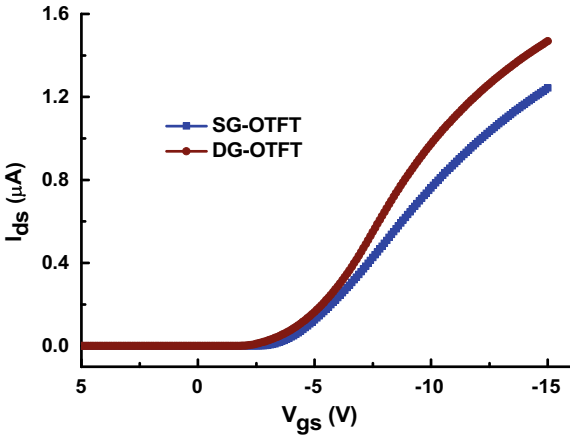


Fig. 7 Input characteristics of SG and DG devices

in Table 1. Electrical characteristics and essential performance parameters obtained from numerical simulator, i.e., Silvaco Atlas are found to be in agreement with the already published experimental results in Cui and Liang (2005). It is evident from the data in Table 1 that DG organic transistor shows boosted performance in comparison to the SG transistor. This boosted performance is by the virtue of existence of charge carrier modulation in dual gate OFET.

4.2 Comparison of Performance of Inverter Circuits

In this section, a relative study for diode load and zero V_{gs} load inverters employing DG and SG devices is performed. It can be observed from the voltage transfer characteristics (VTC) of the inverters that the output voltage swing is more prominent in DG-OFETs as compared to SG-OFETs in case of both the configurations of inverter circuits. Critical voltages and essential electrical parameters are at par in case of DG as compared to SG-OFETs (Goswami et al. 2013). At $V_{DD} = 10$ V, ZVLL configuration with DG-OFET as structural element produces a larger output swing of 8.08 while DLL configuration produces an output swing of 6.4. Noise Margin (high as well as low) are larger in case of inverters designed with SG-OFETs rather than DG-OFETs. A higher value of voltage gain is obtained in ZVLL configuration inverters in contrast to DLL configuration for both SG and DG-OFETs. ZVLL inverters with DG-OFETs perform better than other three configurations in terms of propagation delays (τ_R (μs), τ_F (μs), τ_{PHL} (μs), τ_{PLH} (μs), τ_{p} (μs)).

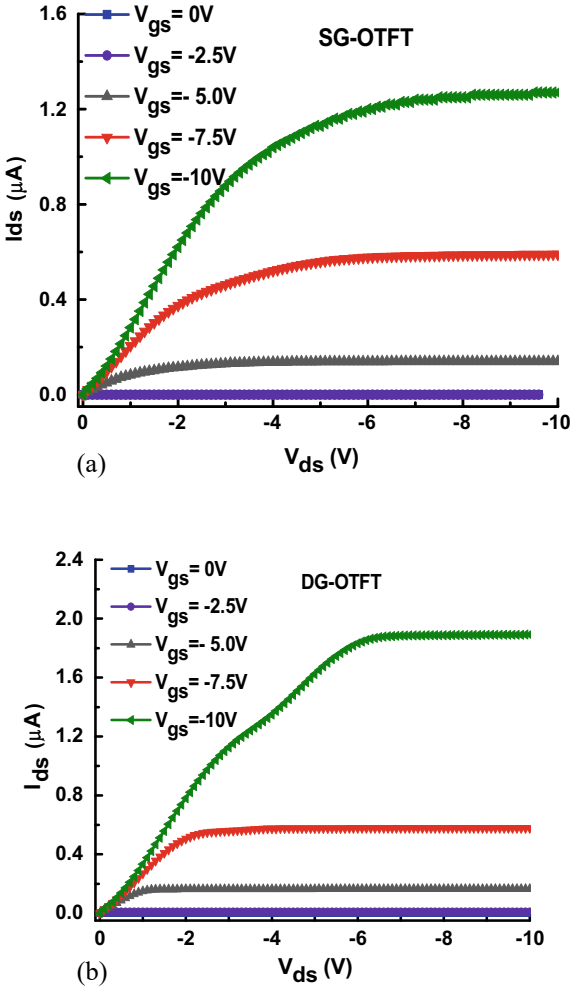


Fig. 8 Drain characteristics of **a** SG-OFET and **b** DG-OFET

Table 1 Extracted electrical performance parameters for DG-OFET and SG-OFET

Parameters		μ (cm^2/Vs)	V_{th} (V)	I_{max} (μA)	I_{on}/I_{off}	g_m (μS)
DG-OFET	Exp.	0.100	−2.20	−1.1	3.80×10^3	0.14
	Sim.	0.088	−2.13	−1.4	3.97×10^3	0.19
	Analytical	0.096	−2.00	−1.17	4.50×10^3	0.16
SG-OFET	Exp.	0.020	−2.00	−0.40	3.20×10^3	0.05
	Sim.	0.063	−2.61	−1.2	1.28×10^3	0.13
	Analytical	0.028+	−1.70	−1.08	3.10×10^3	0.03

Table 2 Characteristics table for 2-input NOR gate

Inputs		Output
V_1	V_2	
0	0	1
0	1	0
1	0	0
1	1	0

4.3 Comparison of Performance of NOR Circuits

Table 2 shows the truth table for a NOR gate. Figures 9 and 10 show the transfer characteristics of NOR gates implemented in DLL and ZVLL configuration using DG and SG devices. As can be observed from the characteristics, DG ZVLL configuration outperforms other configurations same as the case with inverter designs.

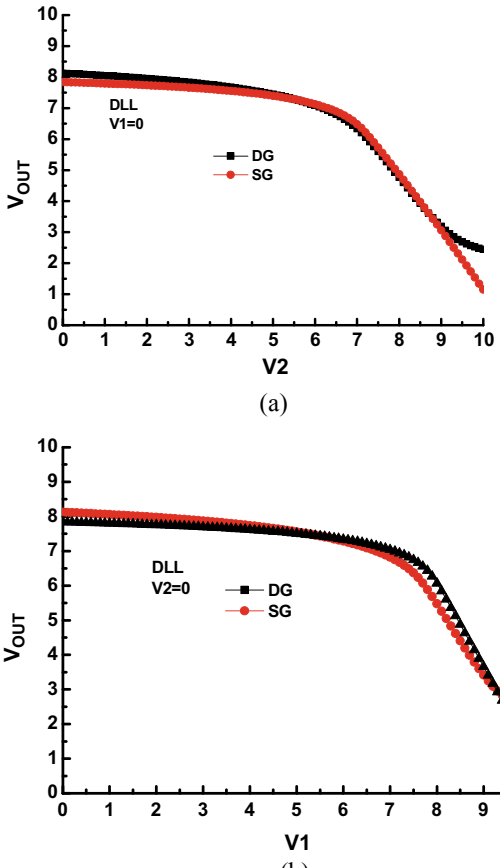


Fig. 9 VTC for NOR gate in DLL configuration with DG and SG-OFETs at **a** $V1 = 0$ and **b** $V2 = 0$ at $V_{DD} = 10$ V

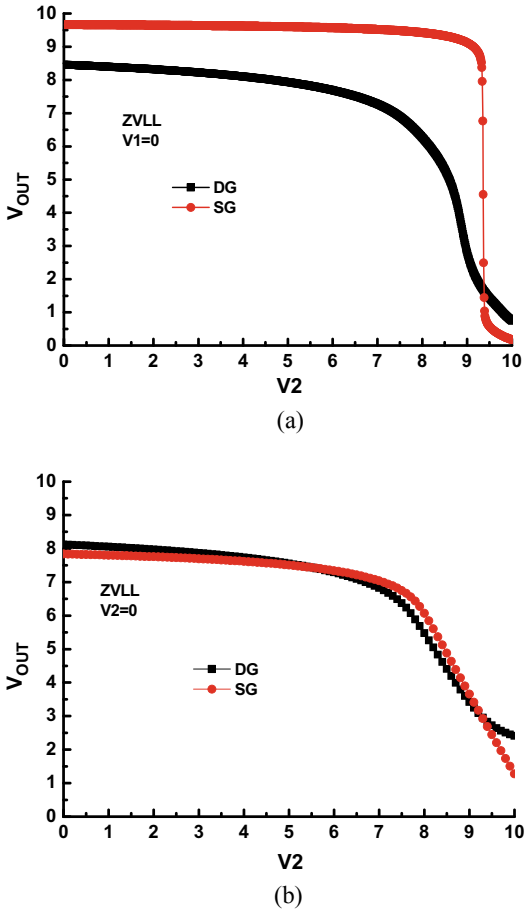


Fig. 10 VTC of NOR gate in ZVLL configuration with DG and SG-OFETs at **a** $V1 = 0$ and **b** $V2 = 0$ at $V_{DD} = 10$ V

5 Conclusion

It is evident from the observations that the DG-OFETs present finer performance in contrast to SG-OFET sin terms of ON–OFF current ratio and sub threshold slope. Also, the DG-OFET shows improved current ON–OFF ratio, especially by controlling OFF current with poor top interfaces. Hence, they are more suitable and reliable for robust applications. Comparing among inverter circuits, it is revealed that the gross performance of ZVLL inverters is better than DLL configuration. DG-OFETs outperform SG-OFETs in terms of all parameters namely propagation delays, voltage gain and voltage swing by virtue of lesser threshold voltage and larger ON current. However, they extract more current from input voltage sources and hence consume more power. Substantial improvement in propagation delay of 53% for ZVLL and 66% for DLL are obtained using DG as compared to SG-OFET. Furthermore, 2-input

NOR gates are also implemented using these inverter circuits and their characteristics are compared. DG ZVLL configuration proves to be outperforming in NOR gate design. In this paper, it is demonstrated that a DG-OFET technology proves to be a practical alternative to achieve more reliability and robustness in designing of flexible digital electronic circuits but at the expense of higher power consumption.

References

Baruah K, Baishya S (2023) Numerical assessment of dielectrically-modulated short- double-gate PNPN TFET-based label-free biosensor. Microelectron J 133:105717

Cui T, Liang G (2005) DG pentacene organic field-effect transistors based on a nano assembled SiO_2 nanoparticle thin film as the gate dielectric layer. Appl Phys Lett 86(6):064102-1–064102-3

Goswami V, Kumar B, Kaushik BK, Yadav KL, Negi YS, Majumder MK (2012) Effect of dielectric thickness on performance of DG organic field effect transistors. In: Processing international conference on communications, devices and intelligent systems (CODIS 2012). IEEE, Kolkata

Goswami V, Kumar B, Kaushik BK, Negi YS, Yadav KL (2013) Analysis of static and dynamic performance of organic inverter circuits based on dual and single gate OTFTs. IET Circ, Devices Syst 7(6):345–351

Horowitz G, Hajlaoui ME (2001) Grain size dependent mobility in polycrystalline organic field-effect transistors. Synth Met 122:185–189

Kumar B, Kaushik BK, Negi YS (2014) Organic thin film transistors: structures, models, materials, fabrication and applications—a review. Polym Rev 54(1):33–111

Maurya P, Mittal P, Kumar B (2022) Performance improvement for organic light emitting diodes by changing the position of mixed-interlayer. Main Group Chemistry

Negi S, Mittal P, Kumar B (2018) Impact of different layers on performance of OLED. Microsyst Technol 24(12):4981–4989

Negi S, Mittal P, Kumar B (2019a) Analytical modelling and parameters extraction of multi-layered OLED. IET Circ Devices Syst 13(8):1255–1261

Negi S, Mittal P, Kumar B (2019b) Organic LED based light sensor for detection of ovarian cancer. Microelectron Eng 218:111154–111164

Negi S, Mittal P, Kumar B (2021) Modeling and analysis of high-performance triple hole block layer organic LED based light sensor for detection of ovarian cancer. IEEE Trans Circ Syst—I 68(8):3254–3264

Sekitani S, Kato TY, Someya T (2005) Control of threshold voltage of organic field-effect transistors with double gate structures. Appl Phys Lett 87(2):023509–023511

Tripathi S, Kumar B, Dwivedi BK (2020) Study on formation and characterization of kesterite CZTSSe thin films deposited by thermal evaporation technique for solar cell applications. J Mater Sci: Mater Electron 31(26):8308–8315

Wang LM, Cao L, Ji WH, Du R, Tang M, Chen SL, Yan LL, Xiao Y, Zhang JR (2023) Application of neutron scattering in organic photovoltaic materials. Chin J Struct Chem 100023

Yang CM, Yang YC, Jiang BH, Yen JH, Su XM, Chen CP (2023) An organic semiconductor obtained with a low-temperature process for light-addressable potentiometric sensors. Sens Actuators, B Chem 381:133449

Rotor Unbalance Severity Detection Using Maximum Overlap Discrete Wavelet Transform

Sonalika Bhandari, Sachin Taran, and Varun Sangwan

Abstract The rotor unbalances a critical fault that increases stress at rotational parts like bearings and gears, resulting in higher power consumption and early machinery wear. The general behavior of the vibration spectrum under this fault changes with strength and rotational speed. To address this problem, the presented work proposes frequency domain data fusion of vibration signals obtained from sensors placed at three different locations. The fused signal retains maximum spectral information, which decomposes into a multi-scale matrix using energy-preserving maximum overlap discrete wavelet transform. To analyze the severity of unbalance, the decomposed scale matrix is encoded into a contour plot using the mean absolute deviation of individual scales as iso-reference lines. Finally, a two-stage classification is performed using a convolutional neural network. The proposed method is tested using a publicly available dataset from Fraunhofer Institute for Integrated Circuits. The results show an overall classification accuracy of 97.05% for unbalance severity which is significantly better than other studies using single-sensor data.

Keywords Unbalance fault · Data fusion · MODWT · Contour plot

1 Introduction

The advancement in networks, processors, and sensors has together revolutionized multiple fields. It has opened a plethora of opportunities for the development of real-time condition monitoring systems (RT-CMS). These systems have strong use cases

S. Bhandari (✉) · S. Taran · V. Sangwan
Delhi Technological University, New Delhi, India
e-mail: sonalikabhandari_2k21spd13@dtu.ac.in

S. Taran
e-mail: taransachin2@gmail.com

V. Sangwan
e-mail: sangwanvarun@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_8

in areas where 24-h supervision of all components is not possible, like automated industries, renewable energy plants, intelligent vehicles, etc. RT-CMS aims to create an early warning in case of any probable failure. This is done based on sensor data analysis and intelligent AI-based classification. A critical area of innovation lies in developing algorithms with low time complexity and moderate processing cost.

A typical drive train or rotating machine comprises shafts, gears, and bearings. These parts often get worn out due to operating conditions causing misalignment, bearing, or unbalanced faults. A moving system works at a characteristic frequency under healthy conditions, but once a fault occurs, the behavior of sensor data changes. Hence, implementing RT-CMS in industrial rotating machines helps to solve a fundamental problem of early fault detection, which scrapes downtime requirements. This work focuses on detecting unbalance faults of multiple strengths at varying rotational speeds. An unbalance is a highly critical fault that arises due to uneven distribution of mass that misaligns the center of mass of the shaft and axis of rotation. This increases stress at other rotating parts, leading to more power consumption and early machinery wear. Thus, developing an algorithm suitable for real-time processing becomes vital to avoid unwanted repair costs.

Most datasets available for testing algorithms focus only on bearing health monitoring, but from the mechanical standpoint, the stress caused by unbalanced faults extravagate bearing faults. During the fledging stage, research focused on traditional time–frequency transformation (Al-Badour et al. 2011)-based methods followed by manual spectral analysis. An early work (Atoui et al. 2013) used a three-level discrete wavelet transform (DWT) followed by the fast Fourier transform (FFT) of the approximate coefficient to establish unbalanced detection based on spectrum visualization. Authors in Lu et al. (2015) focus on developing a method for detecting unbalance of different strengths using empirical mode decomposition (EMD) followed by novel dimensionality reduction, which achieves an accuracy of 98.13% with support vector machine (SVM) under constant rotating speed. A detailed review of unbalanced detection using current and vibration signals for different methods like time domain, frequency domain, multi-resolution analysis, and envelop analysis is covered in Mizanur and Uddin (2017). A multi-modal data acquired at 35 Hz from a vibration sensor and infrared thermal imaging is used in Mian et al. (2022) to detect misalignment and unbalanced faults. It is observed that works based on sensor fusion suffer a primary drawback of improving accuracy at additional hardware and processing cost. Research in Ma and Chu (2019) proposes a decision-level fusion using an ensemble model of convolutional residue network, auto-encoder, and deep belief network (DBN) to classify multiple faults with an accuracy of 98.08% for data under various speeds. Transfer learning has also been explored in Lee et al. (2005); the paper used a pre-trained VGG19 model to input Mel frequency spectrogram images of vibration signal obtained under fixed unbalance strength of 3.2 gm. It is noted from the survey that sensor fusion and deep learning (DL) for unbalance detection has undergone a paradigm shift over the past decade with advanced developments in the DL community.

In this work, we are trying to address the inconsistency of existing methods by developing an automated algorithm capable of unbalance detection under varying

strength factors and rotational speed using data fusion, contour encoding, and DL. The major contribution of this work is listed as follows:

- It proposes a complementary data fusion strategy for fusing data from three vibration sensors to construct a single signal with maximum relevance.
- It proposes a new encoded statistical contour plot for signal decomposed using maximum overlap discrete wavelet transform.
- A two-stage warning system for unbalanced detection and severity analysis.

The paper is organized into four sections: Sect. 2 explains the proposed methodology, Sect. 3 covers the dataset description and experimental results, and Sect. 4 mentions the conclusion.

2 Methodology

The work proposes a three stage method for identifying unbalance and classifying its severity based on ISO standards for small machines as mild, moderate, and severe as shown in Fig. 1. The following sub-section discusses the details of each stage.

2.1 Data Fusion

Fusion refers to the process of amalgamating information from different sources. This can be done at three levels. First is the sensor level, where multi-domain sensors record data that gets processed differently, as proposed in Mian et al. (2022). The second type is data level, where sensor data is combined based on mathematical or

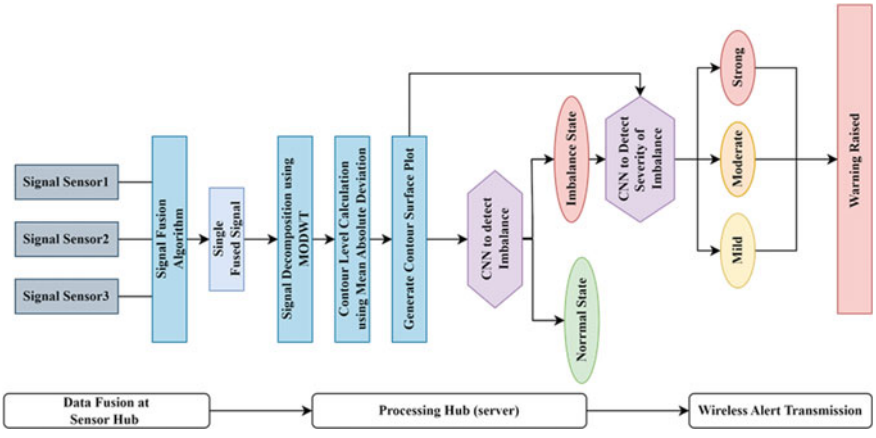


Fig. 1 Flowchart for the proposed methodology

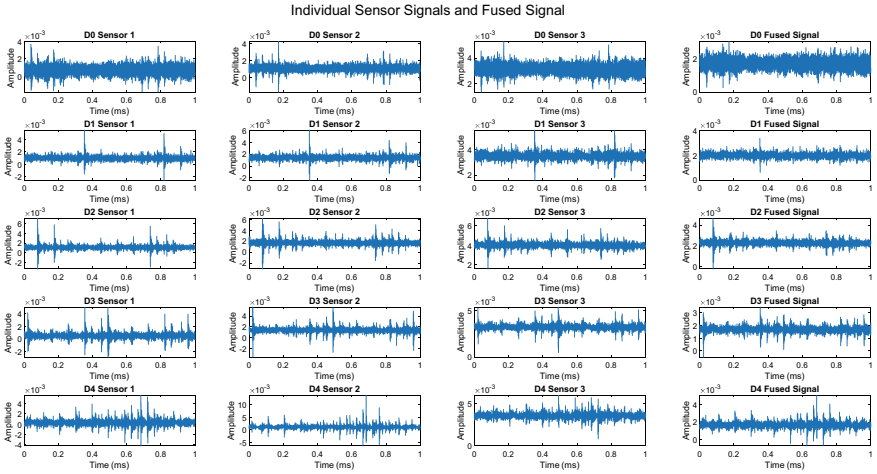


Fig. 2 Representation of fused signals obtained for different strengths of unbalance

domain knowledge to form a single signal with relevant and consistent information. Third is decision-level fusion involving an ensemble of multiple algorithms (Ma and Chu 2019) to complement each other's drawback in the decision-making phase.

This work uses the frequency domain for performing data level fusion of three vibration sensors. Fault signals are associated with fault characteristic frequency hence frequency domain has been chosen for data fusion. Two of the sensors are placed in an orthogonal manner at the bearing block and one at the motor mount. Firstly, FFT is calculated for signals S1, S2, and S3 to obtain frequency spectrum F1, F2, and F3, respectively. Then point wise mean is calculated and inverse FFT (IFFT) is performed to obtain the fused signal as given by (1). The frequency domain is ideal for fusion because the region of interest for unbalanced machine fault lies at the amplitude of peak frequency. With variation in speed, the peak frequency varies and often becomes difficult to distinguish in the spectrum obtained under the real-world environment with ambient noise. The recording of individual sensors and the obtained fused signal of a signal segment of 1s is shown in Fig. 2.

$$X_{\text{Fused}} = \text{IFFT}\left[\sum_{n=1}^{N} \frac{F1(n) + F2(n) + F3(n)}{3}\right] \tag{1}$$

2.2 *Maximum Overlap Discrete Wavelet Transform*

Wavelet-based signal decomposition (Al-Badour et al. 2011) is among the most suitable method for the feature generation of non-stationary signals. It uses a time-shifted

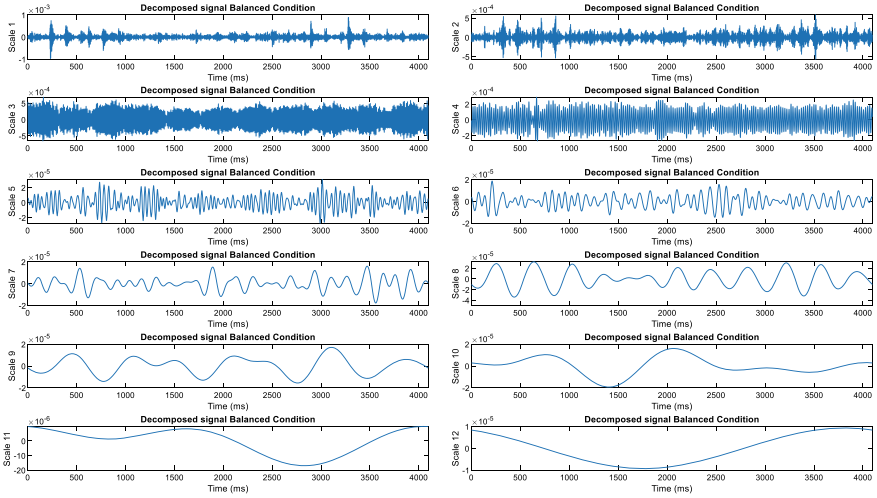


Fig. 3 MODWT of the fused signal under no unbalance condition

and scaled collection of basis functions of the selected mother wavelet. Traditionally DWT has been used for machine fault detection (Atoui et al. 2013) but it suffers from an inherent disadvantage of loss of data due to down-sampling and stringent requirement of 2n samples in a signal. Owning to this MODWT was proposed as a revised version of DWT which does not decimate the decomposed signal at each scale. This makes MODWT redundant in nature as well as energy-preserving decomposition. Hence X_{Fused} can be obtained from the summation of coefficients from all scales. This work uses higher-order Daubechies asymmetrical wavelet 'db40' to ensure maximum localization (Mizanur and Uddin 2017) with non-linear phase response. The decomposition of fused signal for no unbalance and maximum unbalance is shown in Figs. 3 and 4, respectively.

2.3 Contour Plot

The work in the literature has significantly explored time–frequency images like spectrogram, scalogram, Hilbert transform, and synchrosqueezing transform as potential options for the CNN network. In general, the computational cost of calculating wavelet-based scalograms is very high, which introduces latency in the system for real-time monitoring. To overcome this, a contour plot has been proposed using mean absolute deviation-based iso-response z slices. Historically contour plots have been widely used for elevation representation in meteorology, geology, and physics (Seadawy et al. 2020).

We have proposed a novel application of statistically encoded contour plot to visualize the decomposed signal by relating it with the variance at each scale. A

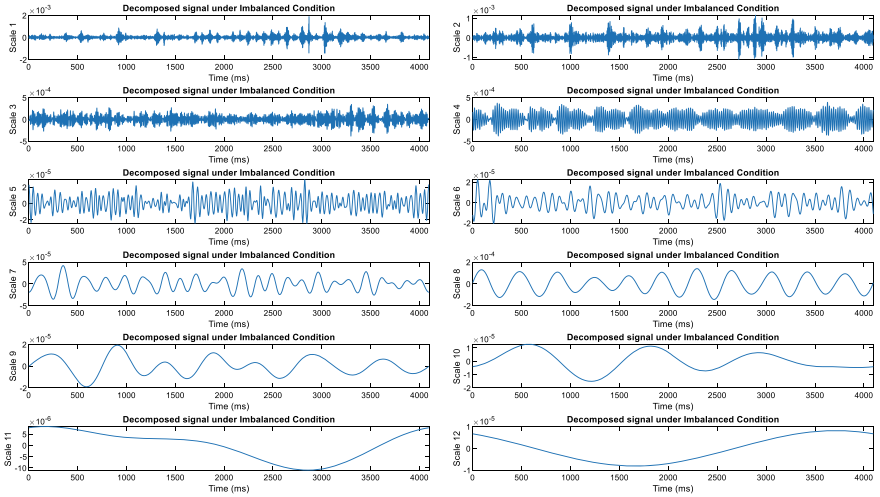


Fig. 4 MODWT of the fused signal under maximum unbalance condition

level matrix with 13 values is created for each decomposed segment using MAD (2) where m is the coefficients of a given scale, μ is the mean of the scale, and N is the total number of coefficients (Elamir 2022). Figure 5 shows the zoom version of obtained contour plot for samples of individual classes for development and evaluation signals. It can be seen that with the increase in unbalance strength, the plot captures the detailed variation of the signal.

$$L = \frac{1}{N}\sum |x_m - \mu| \tag{2}$$

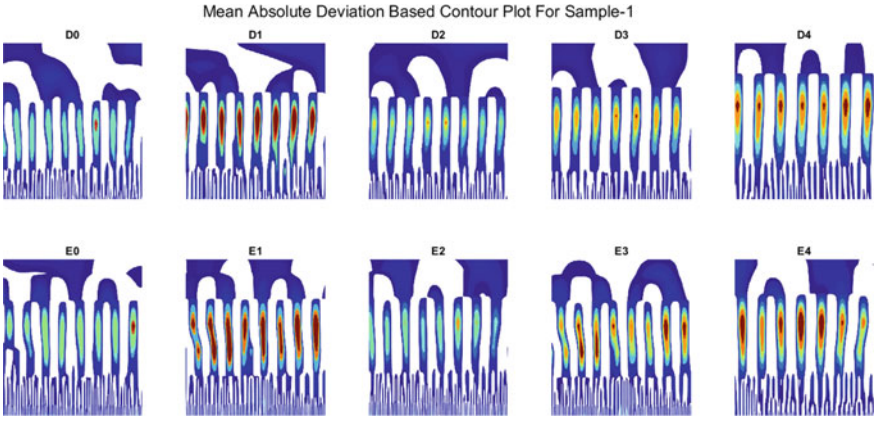


Fig. 5 Encoded contour plot of the decomposed signal matrix for all classes

2.4 Convolutional Neural Network

CNN has proved its capability for classification problems related to machine failure (Lee et al. 2005; Ma and Chu 2019) over time. It is a popular choice among researchers because of its ability to learn contour lines from complex images using filter weights adjustment (Shen et al. 2015). CNN is realized using a set of recurring units consisting of a convolutional, activation, and pooling layer. In this work, a dual-stage CNN is used to improve the accuracy of the severity-based warning system. Stage 1 is a coarse CNN structure with two convolution blocks and a dropout layer. The shallow net learns more generalized weights for classifying the rotor as balanced or unbalanced in case of unbalancing the stage 2 network triggers and raises a warning based on detected severity. The input image size for best performance is taken as 256 × 256. 'Leaky Relu' with alpha 0.2 is used as an activation function, while 'softmax' is used for classification, and 'sparse cross entropy' is the loss function minimized during the training process. The architecture of CNN for both stages, along with the first filter response of the convolutional layer on the input image, is shown in Fig. 6.

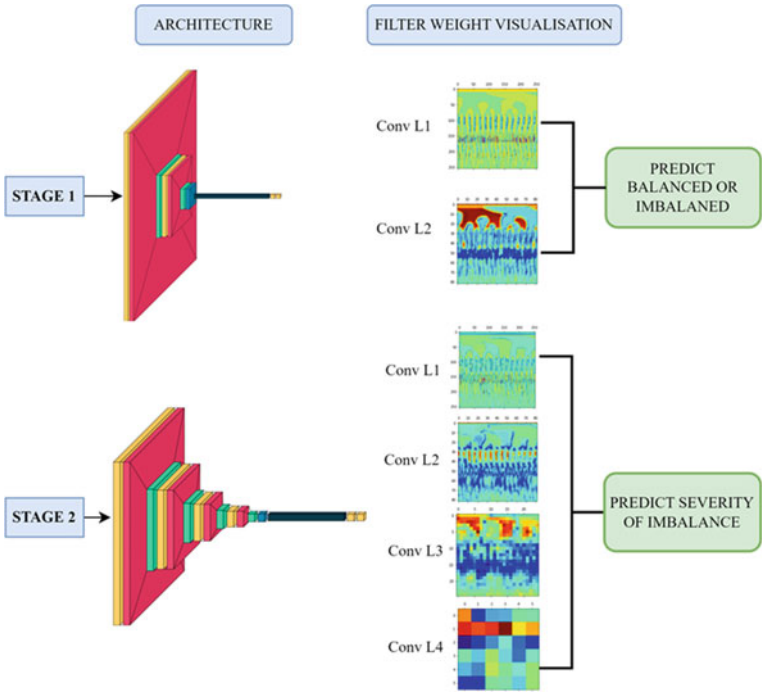


Fig. 6 Architecture and first filter weight visualization of convolutional layer of CNN (Shen et al. 2015)

3 Experimental Results

3.1 Dataset

The method for automated detection of unbalance and severity classification is verified using a publicly available dataset from Fraunhofer Institute for Integrated Circuits (Mey et al. 2020). The speed of the motor is varied from 550 to 2500 RPM for recording different datasets for development (Dx) and evaluation (Ex). The measurement setup and speed variation are shown in Fig. 7. The setup employs a 3D printed disc to introduce unbalance of different strengths viz., 0–no unbalance and 4–maximum unbalance. The data is sampled at 4096 samples per second and a window of 1s is considered for segmentation. Table 1 summarizes the details of different conditions considered in the dataset.

Unbalance force is equivalent to the centrifugal force also it can be noted that the strength of unbalance (UF) for a point mass is proportional to the product of mass times the radii (3) and its effects get amplified at higher rotational speeds (Mizanur and Uddin 2017).

$$F = mr\omega^2 \tag{3}$$

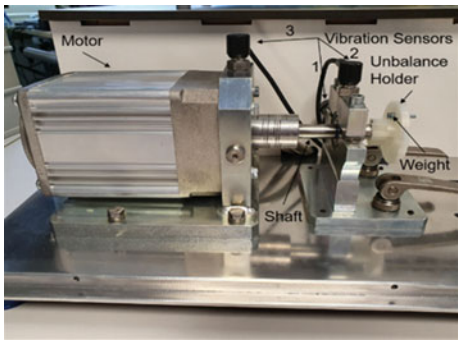


Fig. 7 Measurement setup for recorded dataset (Mey et al. 2020)

Table 1 Summary of dataset from Fraunhofer institute

Parameter	Severity of fault based on ISO standard 10,816				
	Normal	Mild	Mild	Moderate	Strong
Dataset	D0 and E0	D1 and E1	D2 and E2	D3 and E3	D4 and E4
Radi (mm)	–	14	18.5	23	23
Mass (g)	0	3.281	3.281	3.281	6.614
UF (mm g)	0	45.9	60.7	75.5	152.1

3.2 Results and Discussion

This section discusses the results obtained from the dual-stage classification process and severity detection of unbalance. The system configuration used for the experimental study is i5 9th, 2.4 GHz, and 16 GB RAM. The speed of classification using mentioned system configuration and the proposed method is approximately 2800 samples per second. The recorded wall time for generating the encoded contour image is 0.52 ms, while for generating a scalogram with the same data is 2.9 ms. Hence the proposed method speeds up the classification process by five times.

As mentioned in Sect. 2 a segment of 1s length with 4096 data points from three sensors is fused in the frequency domain to form a single, decomposed signal, and a contour MAD-based input image for CNN is obtained. The data is divided into 6000 images from each development file (Dx) to form a training dataset and 1500 images from the evaluation file (Ex) to create a testing dataset. Further, the training set is split into two parts, with 4800 and 1200 from all individual files. This is used for training and validation, respectively.

Evaluation for two stages is conducted where stage 1 is a general qualitative analysis using coarse CNN to detect an unbalanced or balanced rotor operation state. Accuracy in validation and test data has been considered for describing the model performance, as summarized in Table 2. Initially, a pairwise analysis is done to understand the distinguishability between the two classes. It is observed that the unbalance strength of 45.9mm g is the most difficult to distinguish from the balanced case, and it shows the lowest test accuracy of 98.3%. The final test accuracy is calculated by considering samples of 1E, 2E, 3E, and 4E as unbalance class and 0E as balanced class. This gives an accuracy of 99%, comparable to the other methods summarized in Table 3.

The second stage involves quantitative analysis of severity as mild (1E & 2E), moderate (3E), and severe (4E) based on ISO standards to raise the alarm. It is seen that the proposed method is capable of raising a correct warning with a test

Table 2 Summary of accuracy obtained for different classification tasks

Train dataset	Validation accuracy	Test dataset	Test accuracy
STAGE 1-CNN (2 class-balance/unbalance detection)			
0D + 1D	98.67	0E + 1E	98.3
0D + 2D	99.24	0E + 2E	98.62
0D + 3D	99.84	0E + 3E	99.39
0D + 4D	100	0E + 4E	99.45
0D + (1D + 2D + 3D + 4D)	99.46	0E + (1E + 2E + 3E + 4E)	99
STAGE 2-CNN (3 class-severity detection)			
(1D + 2D) + 3D + 4D	99.7	(1E + 2E) + 3E + 4E	98.42
Overall warning accuracy (4 class-normal/mild/moderate/severe)			97.05

Table 3 Comparison of work with existing methods in the literature

Classification method	Features	Classes	Speed	Average accuracy (%)
Proposed CNN	MODWT-MAD-contour images	2	Time-varying	98.72
		4		97.05
SVM (Ma and Chu 2019)	Frequency spectrum	2	Constant	94
FCN (Shen et al. 2015)	FFT (single sensor)	2	Time-varying	98.6
1D-CNN (Shen et al. 2015)	Raw vibration signal	2	Time-varying	93.6
CNN (Kumar et al. 2021)	TF image angular domain	2	Time-varying	98.1
SVM (Gangsar et al. 2021)	Raw time/frequency domain	2	Multiple speeds	91.66
D-CART (Deng et al. 2020)	Dynamics	4	Multiple speeds	90
SVM (de Sá Só Martins et al. 2021)	Hybrid features	4	Constant	93.2–98.2
Multi-DBN (Yan et al. 2019)	Raw vibration signal	4	Constant	75
	Fused orbit plots + vibration			86.46–100

accuracy of 98.42% which is significantly higher compared to the literature for datasets with varying rotational speeds. The confusion matrix for the overall warning system obtained by evaluating the complete two-stage method end-to-end using test data of all classes (0E, 1E, 2E, 3E, and 4E) is shown in Fig. 8. The observed accuracy for the 4-layer CNN model with batch normalization and dropout layer neither over fits nor under fits the data. Also, the results establish that the MAD iso-reference-based contour plot of the MODWT decomposed signal is a good representation of localized information. The study conducted in this paper is limited in the selection of iso-reference where options like energy, median frequency, etc., can also be explored.

Work addressing the two-class problem on the same dataset is presented in Mey et al. (2020). It compared the performance of multiple algorithms where maximum accuracy of 98.6% is obtained using a single-sensor data for FFT-based fully connected neural network (FCN) with two hidden layers for unbalance detection. Other summarized works use personal datasets like research in Gangsar et al. (2021) that combines three statistical features from both the time and frequency domain to feed into an SVM classifier. The results showed the accuracy was a vital function of operating speed, and the method is unsuitable for classification at lower speeds. Work in Deng et al. (2020) uses data recorded at multiple speeds to develop and test classification and regression trees using dynamic-based features (D-CART). The method achieved 90% accuracy for qualitative analysis of severity detection. Research in

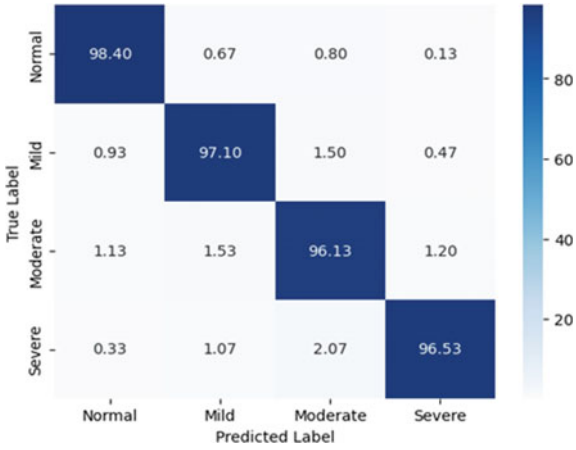


Fig. 8 Confusion matrix for overall warning system using test data

Yan et al. (2019) used heterogeneous information from vibration signals and shaft orbital plots to extract features using a deep belief network (DBN) automatically. The method showed an accuracy of up to 100% using both sensor data and only 75% using the vibration sensor under constant motor speed.

4 Conclusion

This paper presented an automated unbalanced detection system for RT-CMS capable of performing severity detection at varying rotational speeds. In general, to deal with the variation of speed, multi-modal data is used, which increase resource requirement for the processing and transmission of data. The proposed work addresses this problem using frequency domain data fusion of vibration signals obtained from transducers mounted at three locations. The point averaging of the spectrum is performed to incorporate complete spectral information without any frequency loss. The resultant signal uses the energy-preserving nature of MODWT and novel encoding of the decomposed matrix into a contour plot based on the MAD statistic of each scale. Finally, for classifying, CNN is used in a two-stage strategy to classify the severity of fault with 97.05% accuracy, significantly better than existing methods for unbalanced severity detection. This distributed architecture and use of single modal data for the proposed work makes it suitable for real-time application. Future scope for improving the accuracy further can be explored using adaptive level encoding for contours and increasing the labeled data for more advanced deep neural networks.

References

Al-Badour F, Sunar M, Cheded L (2011) Vibration analysis of rotating machinery using time–frequency analysis and wavelet techniques. Mech Syst Sign Process 25(6):2083–2101

Atoui I et al (2013) Fault detection and diagnosis in rotating machinery by vibration monitoring using FFT and wavelet techniques. In: 2013 8th international workshop on systems, signal processing and their applications (WoSSPA). IEEE

de Sá Só Martins DHC et al (2021) Diagnostic and severity analysis of combined failures composed by unbalance and misalignment in rotating machines. Int J Adv Manuf Technol 114:9–10

Deng H et al (2020) A high-speed D-CART online fault diagnosis algorithm for rotor systems. Appl Intell 29–41

Elamir EAH (2022) On uses of mean absolute deviation: shape exploring and distribution function estimation. arXiv preprint arXiv:2206.09196

Gangsar P, Pandey RK, Chouksey M (2021) Unbalance detection in rotating machinery based on support vector machine using time and frequency domain vibration features. Noise Vib Worldwide 52(4–5):75–85

Kumar A et al (2021) Tacho-less sparse CNN to detect defects in rotor-bearing systems at varying speed. Engineering applications of artificial intelligence, p 104

Lee SW et al (2020) Apply VGGNet-based deep learning model of vibration data for prediction model of gravity acceleration equipment. arXiv preprint arXiv:2005.10985

Lu L, Yan J, de Silva CW (2015) Dominant feature selection for the fault diagnosis of rotary machines using modified genetic algorithm and empirical mode decomposition. J Sound Vibr 344:464–483

Ma S, Chu F (2019) Ensemble deep learning-based fault diagnosis of rotor bearing systems. Comput Ind 105:143–152

Mey O et al (2020) Machine learning-based unbalance detection of a rotating shaft using vibration data. In: 2020 25th IEEE international conference on emerging technologies and factory automation (ETFA), vol 1. IEEE

Mian T, Choudhary A, Fatima S (2022) Multi-sensor fault diagnosis for misalignment and unbalance detection using machine learning. In: 2022 IEEE international conference on power electronics, smart grid, and renewable energy (PESGRE). IEEE

Rahman MM, Uddin MN (2017) Online unbalanced rotor fault detection of an IM drive based on both time and frequency domain analyses. IEEE Trans Ind Appl 53(4):4087–4096

Seadawy AR, Arshad M, Dianchen L (2020) Dispersive optical solitary wave solutions of strain wave equation in micro-structured solids and its applications. Physica A 540:123122

Shen W et al (2015) Deepcontour: a deep convolutional feature learned by positive-sharing loss for contour detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition

Yan J, Yuanyuan H, Guo C (2019) Rotor unbalance fault diagnosis using DBN based on multi-source heterogeneous information fusion. Procedia Manuf 35:1184–1189

Self-attention-Based Deep Learning Approach for Machine Translation of Low Resource Languages: A Case of Sanskrit-Hindi

Nandini Sethi, Amita Dev, and Poonam Bansal

Abstract The automated translation of natural language text using software is known as machine translation (MT). Natural language's intrinsic complexity and incompatibility have made machine translation (MT) a difficult process that is full of difficulties, especially in comparison to a physical conversion. The development of neural-based methods has brought machine translation outcomes closer to what people would expect. The seq-to-seq strategy, which uses encoder and decoder pairs, recurrent NN, complicated convolutions, and transformers, is one of the most modern deep learning techniques. This work proposes a self-attention deep learning architecture for MT that is built on a transformer with 8 layers of encoders and 8 layers of decoders and has all layers completely dedicated to multi-head attention. The important source and outcome of earlier layers that flow into the subsequent layer are weighted summarised to highlight the complete contribution of the proposed model. Comparing this process inputs that are not hybrid, the transformation produced is more exact. A parallel corpus for the Sanskrit-Hindi language pair is created in this study, and the model is assessed using the created corpus. The tests are carried out on Google Collab instances with GPUs, and the outcomes reveal an accuracy of 53.4 BLEU, which is an improvement over the earlier work.

Keywords Corpus · Machine translation system · Sanskrit · Hindi · Neural machine translation · Self-attention · Transformer

N. Sethi (✉) · A. Dev · P. Bansal
Indira Gandhi Delhi Technical University for Women, New Delhi 110006, India
e-mail: nandinisethi2104@gmail.com

A. Dev
e-mail: vc@igdtuw.ac.in

P. Bansal
e-mail: poonambansal@igdtuw.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_9

1 Introduction

A branch of computational linguistics called "machine translation", is the practise of automating, assisting, or facilitating language translation using software. Applications for machine translation include education, international communication, the development and exchange of cultures, technical support services, and trade. One of the most difficult challenges in artificial intelligence is language modelling, which involves many different features such language interpretation, reasoning, and the use of common or public information. Machine translation systems have a long way to go before they can be said to have a sufficient degree of accuracy, despite various research efforts. Semantic web MT reconstructs a substantial volume of material in a fashion that is akin to the human mind and that is available on the Internet. The information from connected multilingual phrases needed for MT has been automatically acquired using SMT analysis of statistical data. Semantic disambiguation is one translation problem that uses statistical translation, which is the application of statistics-based approaches (Albat 2015). The fundamental benefit of NMT is that it requires less feature engineering while maintaining or even increasing the effectiveness and precision. This aids researchers in finding solutions to issues where they lack specialised knowledge or find it challenging to manually develop the rules (Vinyals and Le 2015).

In this article, we introduce a deep MT network that creates the input for the following layer both in encoders and decoders by fusing the key input of layers with the result of earlier sub-layers. The suggested model includes three key components: pre-processing, encoder-decoder, and SoftMax layer. It is referred to as the enhanced hybrid input attention-based transformer. The main benefits of the suggested model are increased translation accuracy and primary input and output combining from earlier sub-layers. Our own custom-designed parallel corpus for the Sanskrit-Hindi language pair is used to evaluate our model. The model has been trained utilising 80% of the intended dataset on Google Colab servers with GPUs. The remaining 20% of the dataset is then used to test the model on the same platform. The Bilingual Evaluation (Papineni et al. 2002), a technique for assessing the translation, is the most often used evaluation criterion in MT. This metric is what we use to gauge how well our work stacks up against the canon. The accuracy of our experimental results is 53.4 BLEU, which is a considerable improvement over the prior study.

2 Related Work

Rule-based and SMT approaches have been the main topics of previous MT research (Misu et al. 2012). It is quite challenging to construct a system with an expanding domain using such techniques, which frequently rely on human rule and grammar construction. It has been demonstrated that a variety of data-driven machine learning techniques significantly help with the interpretation and synthesis of natural lan-

guages (Indurkhya and Damerau 2010; Ba et al. 2016). The NMT approach is used by most contemporary MT algorithms. Building a single network model that can be jointly changed to improve translation performance is the fundamental goal of NMT (Bahdanau et al. 2014). When string of words and neural machine translation were contrasted, Junczys-Dowmunt et al. (2016) discovered that neural machine translation used the n-gram technique and had greater precision than phrase-based translation.

Additionally, several advances have significantly increased the effectiveness of a subset of NMT called translation utilising deep learning employs machine learning from beginning to end, but in a different way. The majority of NMT models in use today are not deep networks, and the best single NMT models perform far worse than their deep network equivalents (Zhou et al. 2016). Gated recurrent units (GRU), long short-term memory (LSTM), and recurrent neural networks (RNN) have been acknowledged as the state of the art in sequence modelling. Machine translation systems are assessed for quality using automated techniques. A variety of NMT implementations employ a dual-step recurrent neural network utilising encoding and decoding techniques to acquire the generalised model to improve the accuracy of translation as opposed to training the different translation machine components independently (Cho et al. 2014; Wołk and Marasek 2015). The best outcomes in NMT can be obtained, in accordance with Zhang and Zheng (2016), by utilising an encoder-decoder design (Junczys-Dowmunt et al. 2016; Kalchbrenner and Blunsom 2013).

3 Dataset Creation

A corpus of parallel source-target sentence pairings was created using manual back translation, using Sanskrit and Hindi as the native and target languages, respectively. The parallel sentences were manually constructed using data from a variety of Internet sources, including Wikipedia, the Bhagavat-Geeta, Multilingual Question Papers, Parallel Dictionary Words, and other websites. Since analogous sentences were not available in the Wikipedia source, we manually translated the sentences into Sanskrit. Sanskrit phrases were also translated into Hindi after being collected from various blogs on the Internet. The parallel source and target sentence train and test sets were split in an 80:20 ratio. This work will act as the baseline paper for the Sanskrit and Hindi low-resource language pair.

4 System Description

Our suggested architecture, shown in Fig. 1, is divided into three main phases, which are described.

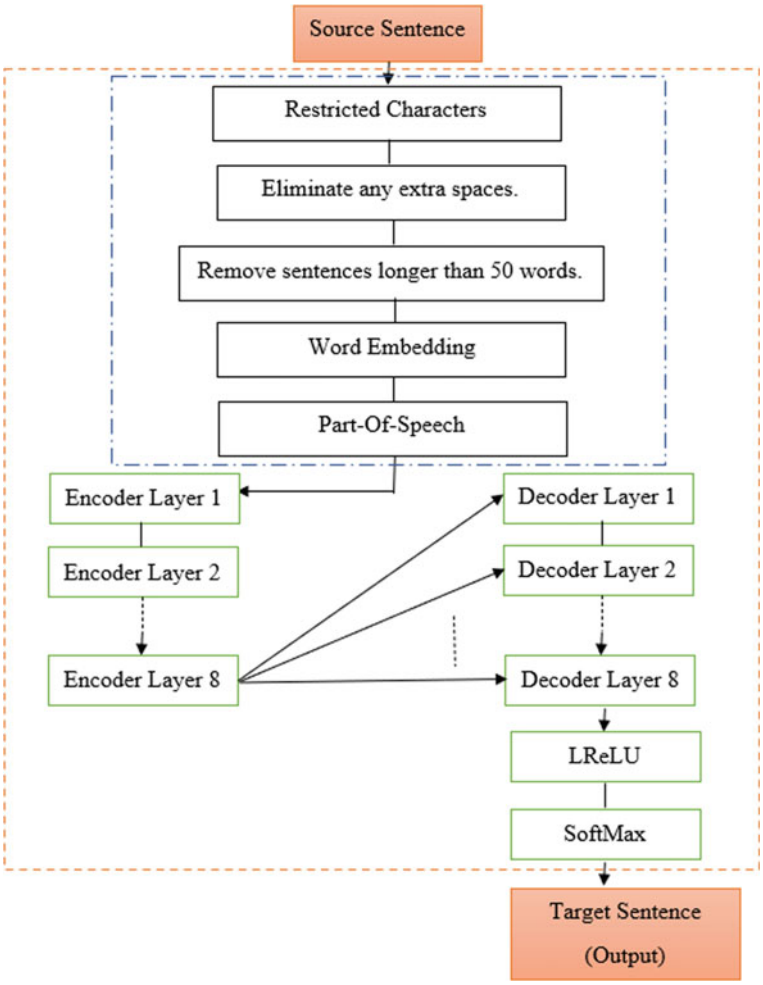


Fig. 1 Proposed translation system architecture

4.1 Pre-processing of the Data

Now that the dataset has been sorted, the sorted data's outputs have been rendered consistent and share the similar structure. Three steps make up the pre-processing phase:

Eliminate unusual characters and lengthy sentences The sentences that are being entered are first written in the input language, and then we eliminate any unlawful typescripts like '! "# $ percent & () * +, −. /:; =>? @ [] _. Additionally, because phrases larger than 50 words are exceptionally long and unnecessarily lengthen processing time, we eliminate them from our analysis.

Embedding of Words Embedded words are numeric vector that encode words in the dictionary and have several applications in NLP. Every word must be converted into a set of values so that the net can interpret them. This is the basic notion underlying word embedding. Every word must be represented by a separate integer in the data input for the embedding layer. In this instance, the TF-IDF technique will be used to tokenize the input.

Positional Encoding The network does not comprehend the grammatical structure while presenting the input sequence, so we must insert a positioning encoder step to recognise the syllables in the input phrase. This is accomplished by adding a vector to the input vector that is the same size. These vectors adhere to a particular pattern that the model is taught to follow by the learning process. This makes it easier to determine where each word should go in a sentence or how far apart the words should be.

4.2 Encoding Layer

The result of the pre-processing step serves as the input for the first step of the encoder phase. The suggested design uses two substrates for each encoder layer: a "self-note" as the first substrate and a fully-connected neural network as the second. The "self-note" describes how a phrase's words are related to one another.

Multi-head Attention The encoder design uses the self-attention mechanism to count the number of words in the input phrase that are connected to one another. This process involves two steps:

- Each of the encoder input vectors is used to create the query, key, and value vectors in the first stage of processing. Weight matrices are used to build these vectors.
- The activator function of the leaky ReLU is used by the self-attention mechanism to permit a minimal, non-zero gradient while the unit is not in use. The ultimate attention score is determined in step three using Eq. (1).

$$\text{attention}(Q.K.V) = \text{Softmax}((QK^T)/\sqrt{(d_k)}) * V \tag{1}$$

We employ multi-head attention to precisely improve the relationship among words in a phrase.

4.3 Decoding Layer

In order to feed the decoding layers, the result of the last encoding step is turned into a set of V and K vectors of attention. Every layer on the decoding side consists of three substrates, two of which are comparable to the encoding layer and one of which is

referred to as "multi-head masked attention" and helps with certain decoding stages like the focus input. Contrary to the one on the encoder side, the self-attention sub-layer on the decoding side only allows looking at past places in the output sequence. Before generating the multi-head attention score, a mask multi-head attention layer is used to achieve this. Other than the Q matrix, which replaces the tier before it and gets the K and V composite from the encoding result, the encoder and decoder layer operates like an octave note. As we proceed deeper into the network, the impact of the information decreases, thus we aggregate the outputs from this tier with the data input on both sides to adjust for this.

4.4 SoftMax Layer

The output of the decoding stack is a vector of decimal numbers. With the aid of a linear layer backed by the SoftMax function, this vector can be converted into words. Using a fully-connected network known as a linear network, the vectors produced by decoder stacks are converted into the significantly larger vectors defined as log likelihood. The amount of the phrase collection is indicated by the size of this vector. For every cell has a unique word. The SoftMax function converts the output points into a range of values (0...1).

5 Experiments

Due to their high computing requirements, deep neural networks can be taught using computers; however, training on general-purpose CPUs is sometimes extremely time-consuming and ineffective. However, because to their success in handling deep learning workloads, accelerators with high inherent SIMD parallelism, such as GPUs, are now used as the industry standard for offload execution in deep learning systems. The Keras framework, based on the TensorFlow framework from Google, was one of the deep learning frameworks that supported accelerators in this case. The suggested network's entire training and testing procedure was carried out on Google Colab, a cloud-based service for deep learning tasks, with unrestricted support of GPU. For the testing phase, 20% of the sentences from the target dataset were removed. The networks were then tested for 100 epochs. In order to construct the loss function, we used sparse categorical cross entropy, a statistic often used in multi-class NLP applications, such as machine translation. 128 rounds of training phase are offered to the network throughout the training phase. On the network, each training session lasts roughly one minute. Using eight encoding and decoding steps and training on 20K parallel corpus created from Sanskrit and Hindi, we train our heuristics architecture. Based on several experiments, we concluded that each epoch required at least 20,000 sessions to get the best accuracy. Today's most well-known optimizer is Adam, a technique that shortens training time by altering the learning rate while training.

Table 1 Results comparison based on BLEU score

System	BLEU score	Modelling technique	Citation
Sanskrit-Hindi	Better by 24% than rule-based approach	CBMT	Singh et al. (2020)
Sanskrit-Hindi	51.6%	Direct + RBMT	Sethi et al. (2022)
Sanskrit-Hindi	53.4%	Transformers with self-attention	Proposed system

Table 1 demonstrates the proposed method's superiority in terms of BLEU correctness by contrasting the outcomes of our technique testing on the 20% of the phrases of the designed dataset with other studies that have undertaken equivalent research. Due to the hybrid model we utilise in contrast to the appropriate amount of layers for input data, the findings in Table 1 demonstrate that our experiment's BLEU measure is superior to that of comparable approaches.

6 Conclusion

The machine translation method put forward in this study is based on the self-attention-based transformer technique. It uses a system for multiple heads of attention, a technique for self-attention. Every network tier encodes and decodes the self-attention mechanism. By modifying the model framework and merging layer productions, the problem of superfluous computing and computation error that plagued prior transformer models is eliminated, ensuring that the outcome was least affected by the computational mistake. The suggested technique yields 53.4 BLEU, which is the highest value among comparable past experiments.

References

Albat TF (2015) Systems and methods for automatically estimating a translation time including preparation time in addition to the translation itself. Google Patents. US Patent 9,128,929

Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473

Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259

Indurkhya N, Damerau FJ (2010) Handbook of natural language processing. Chapman and Hall/CRC

Junczys-Dowmunt M, Dwojak T, Hoang H (2016) Is neural machine translation ready for deployment? A case study on 30 translation directions. arXiv preprint arXiv:1610.01108

Kalchbrenner N, Blunsom P (2013) Recurrent continuous translation models. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1700–1709

Misu T, Georgila K, Leuski A, Traum D (2012) Reinforcement learning of question-answering dialogue policies for virtual museum guides. In: Proceedings of the 13th annual meeting of the special interest group on discourse and dialogue, pp 84–93

Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp 311–318

Sethi N, Dev A, Bansal P, Sharma DK, Gupta D (2022) Hybridization based machine translations for low-resource language with language divergence. ACM Trans Asian Low-Resour Lang Inf Process

Singh M, Kumar R, Chana I (2020) Corpus based machine translation system with deep neural network for Sanskrit to Hindi translation. Procedia Comput Sci 167:2534–2544

Vinyals O, Le Q (2015) A neural conversational model. arXiv preprint arXiv:1506.05869

Wołk K, Marasek K (2015) Neural-based machine translation for medical text domain. Based on European medicines agency leaflet texts. Procedia Comput Sci 64:2–9

Zhang J, Zong C (2016) Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 1535–1545

Zhou J, Cao Y, Wang X, Li P, Xu W (2016) Deep recurrent models with fast-forward connections for neural machine translation. Trans Assoc Comput Linguist 4:371–383

Reinforcement Learning Method for Identifying Health Issues for People with Chronic Diseases

Sunita Dhote, Michaelraj Kingston Roberts, and K. P. Sridhar

Abstract Healthcare systems will inevitably be tasked with determining the best method for identifying and modelling the risks associated with chronic diseases. For these reasons, it is thought that clinically examining medical records using traditional and machine learning technologies could be a beneficial, real, and most crucially cost-effective alternative to human medical professionals. Understanding the significance of early detection in preventing the worst effects of such diseases is crucial. Manually diagnosing diseases is often a time-consuming and inaccurate process for doctors; from de-identified healthcare data and narrative texts, information is extracted. This system has been used to overcome the problem of missing information using both structured data (SD) and unstructured data (USD) by the reinforcement learning-based multimodal risk identification system (RL-MRIS). The prediction accuracy and the convergence speed have been analysed for various data sets. It is based on the AutoLearn algorithm (ALA), which can identify a tool for determining elements in a data set and the broad variations of chronic diseases. The prediction accuracy and the convergence speed have been obtained using this identification system.

Keywords Reinforcement learning · Structured data · Unstructured data · Chronic diseases · Multimodal identification · Prediction accuracy

S. Dhote (✉)
Department of Management Technology, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
e-mail: sunitandhote@gmail.com

M. K. Roberts
Faculty of Engineering, Department of ECE, Sri Eshwar College of Engineering, Coimbatore, India
e-mail: king.pane@gmail.com

K. P. Sridhar
Faculty of Engineering, Department of ECE, Karpagam Academy of Higher Education, Coimbatore, India
e-mail: Capsridhar@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_10

1 Overview of Reinforcement Learning Methods with Chronic Diseases

The healthcare system may improve patients' quality of life and significantly lower the soaring cost burden by more effectively managing chronic illnesses and preventing hospitalisations and serious consequences (Hoque et al. 2022). The prevalence of chronic diseases is a major challenge in the healthcare industry (Battineni et al. 2020). The medical report claims that the rising mortality rate in humans is attributable to chronic disorders (Remy-Jardin et al. 2020). Over 70% of a patient's income is used up by the cost of therapy for this illness. It is crucial to lessen the patient's mortality risk as much as possible (Yajurved and Prasad 2022). Despite the widespread adoption of reinforcement learning techniques in the healthcare sector, there is still a lack of empirical evidence on the efficiency of existing methods for systematically classifying patient records (Ikemura et al. 2021a). Heart disease, stroke, cancer, diabetes, respiratory illnesses, and arthritis are all examples of chronic diseases (Abubeker and Baskar 2022; Abubeker and Baskar 2022; Feng et al. 2021). For quite some time, chronic diseases have topped the lists of leading killers and disablers in countries with high per capita incomes (Karthik and Sudha 2018). The system aims to rapidly determine which data points will be the most useful in developing prediction models for cardiovascular disease (Ikemura et al. 2021b). Preventing the worst effects of these diseases requires that they can be detected and predicted as early as possible (Bhola et al. 2021). Physicians often struggle with making a correct diagnosis when relying solely on manual methods (Hassoun et al. 2023). Patient demographics, medical analysis reports, and disease histories are all examples of what can be found in healthcare data (NagabhushanaRao et al.). Therefore, it is essential that the patient's living conditions and environmental factors be recorded alongside the disease information (Kor et al. 2022). The RL-MRIS identification approach places greater emphasis on identifying missing information utilising both structured data and unstructured data (Kingston Roberts et al. 2021).

When compared to solely using structured data, outcomes that incorporate both types of information are much more reliable combination of the two (Moore et al. 2021). When combined with the structured data that includes patient demographics, disease details, living habitats, and laboratory test results, the unstructured data that provides doctors' notes on patients' illnesses and their descriptions of their symptoms and hardships offers invaluable insight (Babu). In the reinforcement learning-based multimodal risk identification system (RL-MRI), rare diseases may be easily detected by combining reinforcement learning techniques with pre-processing the data collected for constructing a data set for missing values, and selecting the key characteristics needed for successful illness prediction can improve prediction accuracy while decreasing the time required (Srivani et al. 2020).

2 Related Work

The prevalence of chronic diseases has been the subject of numerous studies authored by researchers and medical professionals. Many different methods have been explored and developed to use reinforcement learning to categorise chronic diseases.

Alanazi (2022) reported convolutional neural network (CNN) that helps to validate automatic feature extraction and disease prediction more effectively. Further. K-nearest neighbour (KNN) calculation helps to find the exact match in the data set based on the patient's disease symptoms. Getu et al. (2022) suggested that the Cumulative Index to Nursing and Allied Health Literature (CINAHL) was chosen because of the clarity with which they describe the various modelling approaches used to develop CD diagnostic and usage models for specific illnesses, including their advantages and disadvantages. A lack of trust in machine learning due to a bad diagnosis may discourage policymakers from using prediction models.

Shruthigedam et al. (2020) deal with the methods of stress detection that have been implemented in the light of the aforementioned sensory devices including wearable sensors, electrocardiogram, electroencephalogram, and photoplethysmography (EEP); a deep learning-based wearable-sensor-based multimodal stress detection system is proposed. Padmanaban et al. (2016) Models are proportional to the impact of the behaviour of learning algorithms as measured by a set of data mining indicators. Naive Bayes and decision tree (NBDT) methods have been used to examine the data set's ability to classify data. To overcome these issues, the present RL-MRI and ALA regression models increase the accuracy and risk predictions. The detailed process of operation of our model is discussed in following section.

3 Reinforcement Learning Multimodal Risk Identification System

This paper proposes the extraction of patterns from data, knowledge acquisition, and the application of that knowledge to tasks like risk prediction which are all within the preview of reinforcement learning (RL). Among the leading causes of death in the world, cardiovascular disease comprises heart disease and stroke. If the patient modifies their behaviour, they may be able to lessen the influence on detecting and avoiding cardiovascular disease. Diagnostic uncertainty, variation in treatment, safety response due to suboptimal generalisability of clinical trial results, complexity in risk, and limited integration of information at the point of care all pose difficulties in reinforcement learning when it comes to providing the right care to the right patient. In Fig. 1, the ability of multimodal risk identification system with respect to random classifier is presented.

In this MRI model, among the many reinforcement learning algorithm classifier is the one most commonly employed to categorise data. Classification is the process of

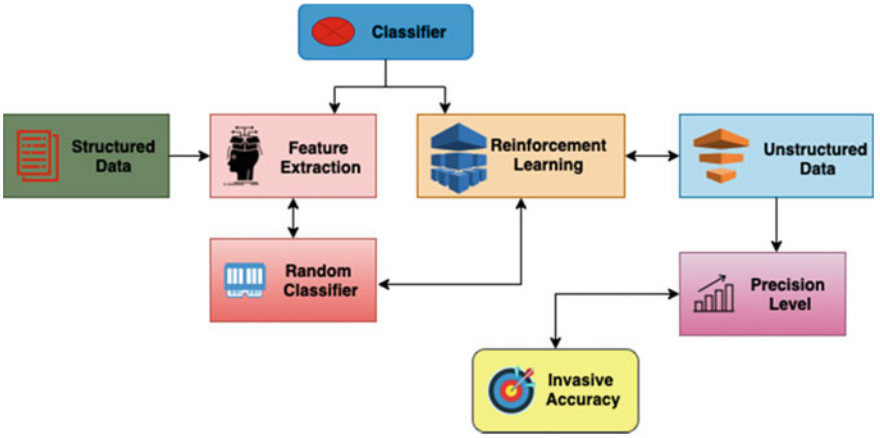


Fig. 1 Ability of multimodal risk identification system

organising all known data into hierarchies based on their shared characteristics and differences. To categorise data, a computer employs an algorithm, which can be a classifier. The classifier needs to evaluate both our structured and unstructured data. Unstructured data is crucial with more difficult piece of the data in the field of reinforcement learning. Data that has been structured can be easily stored and accessed through tools like databases and spreadsheets. On the other hand, unstructured data information will lack a standard format. While there is no shortage of data sources, the four most important for reinforcement learning models are as follows. The four types of data include numbers, categories, time series, and free-form text. The main objective of feature extraction is to remove irrelevant data from a data set, and data reduction facilitates model creation with reduced machine effort and accelerates the learning and generalisation phases of machine learning.

Feature extraction turns text and images into number features that machine learning systems can read. Feature selection applies machine learning theory to number data. It helps trained algorithms extract the most important traits, but it destroys data. Reinforcement learning teaches robots by rewarding and punishing behaviour. Reinforcement learning robots can sense their environments, plan and execute suitable reactions, and be better through trial and error. Reinforcement learning can train a programme to make wise choices. This could transform decision-making and optimisation. It is a popular machine learning method that forecasts accuracy by combining the results of many random classifiers on data set groups. A data set is formed from a mode risk-trained group of random classifiers.

Figure 2 shows the purposes of developing and training machine learning models; "data pre-processing" is the process of transforming raw data into a form that is usable for these purposes by removing redundant information and arranging the data in a more logical fashion. The accuracy of a classification prediction is measured by the model's precision.

$$L\theta(R) = \frac{1}{1 + e^{-(\beta+\beta x)}} \tag{1}$$

where R represents the regression, β represents the data deviation, and χ is standard deviations of the mean if an instrument or method is accurate. During invasive testing, medical instruments are used to make incisions. The regression $L\theta(R)$ is represented by modified data

$$s(f, r) = \sum_{n=0}^{N-1} S(n)\left[w(n - r)e^{-\frac{j2\pi fn}{N}}\right] \tag{2}$$

In Eq. (2), the modified data is $s(f, r)$, energy is denoted by $s(n)$, the frequency is represented by f, the window function is $w(n)$, and r as the time domain representation of frequency is represented by n. The number of samples is denoted by the symbol N. If k latent factors are present, then the original R matrix can be approximation as

$$R_{MXN} = PN_{MXK} \tag{3}$$

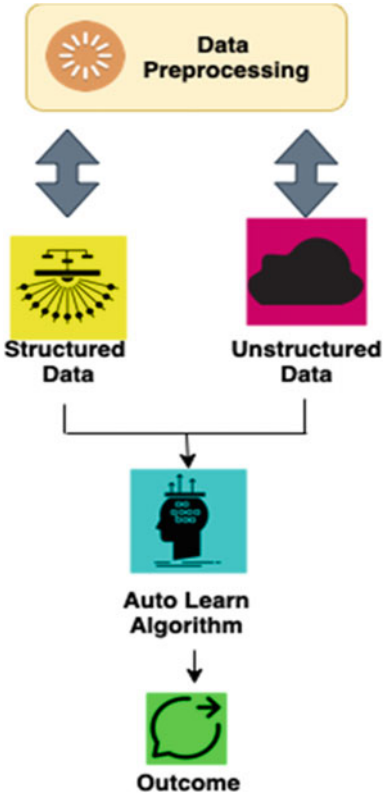


Fig. 2 Analysis of data pre-processing

In Eq. (3), each element value can be written as $r_{uv} = P_u T_U Q_V$, where P_u is the vector of the user factor, which indicates the patient's preference to these potential factors, and Q_V is the vector of the feature attribute factor. The P_U and Q_V values in the above formula are unknown, and we can transform this problem into an optimisation.

$$H(e) = -\sum_{x \in X} q(e) \log_2(q(e)) \tag{4}$$

As shown in Eq. (4), $q(e)$ is used for data classification, which means the optimum set of features is recognised by e factor. $H(e)$ refers to virtual environment, and the data layers of information have been modified in the equation below.

$$H(e) = \frac{1}{1 + e^{-s}} + f(E_{nm}) \tag{5}$$

As shown in Eq. (5), E_{nm} is added by connecting the values of training algorithm. The values are adjusted by the training data. The processed data $H(e)$ can be expressed in the form of signals s. The parameter $f(E_{nm})$ defines the transfer function of learning numbers from nth output respect to mth input. The number of residuals is widely used as functions in Eq. (6),

$$H(x) = \frac{1}{1 + e^{-s}} + f(A_{nm}) \tag{6}$$

This $H(x)$ function determines the output A, calculations of the proportion of correct predictions to the sum of correct positive and wrong negative predictions, as well as the proportion of correct predicted values to the sum of correct positive and incorrect negative predicted values.

$$K = \frac{1}{2} \sum_{N=1}^{i} \sum_{M=1}^{j} \left(B_{NM} - B_{NM}^{*} \right)^2 \tag{7}$$

Equation (7) determines the K values, where $B_{NM} - B_{NM}^{*}$ is the actual transfer function, NM is the Nth output and the Mth input, and K weight values are constant.

Figure 3 shows logistic regression's predictive and category uses. In the following cases, a logistic regression model can help teams identify data anomalies that indicate theft predicting what will happen. Prediction is not limited to the future. Raw data is analysed to produce knowledge. Quantifiers are terms or morphemes used with numbers or names to indicate countable or measurable items. In Eq. (8), it is preferable to identify significant attributes from a large number of data sets. Suppose D_x is the set of samples where the data derived from regression learning RL is determined for a random variable I_a using following equation:

$$RL(D_X, I_a) = H(D_x) - \sum_x \left(\frac{(D_{x=1})}{k} \right) I_a \tag{8}$$

Figure 4 shows that the reinforcement learning can help determine the cause of an algorithm's success when it consistently produces the correct answer or the reward indicating it made a good decision. The foundation of RL is the feedback loop between a system and its surroundings.

The regression contains one or more independent variables that represent outcomes considering logistic function. The data is randomly divided between a training set and a testing set, or k-fold cross validation is used to randomly split the data k times. That number is a rough benchmark for the model's efficacy. Equation (9) determines the large data variables. Let $x = x_1, x_2, \ldots x_n$ be independent variables, and $\theta = \theta_1, \theta_2, \ldots \theta_n$ parameters are considered.

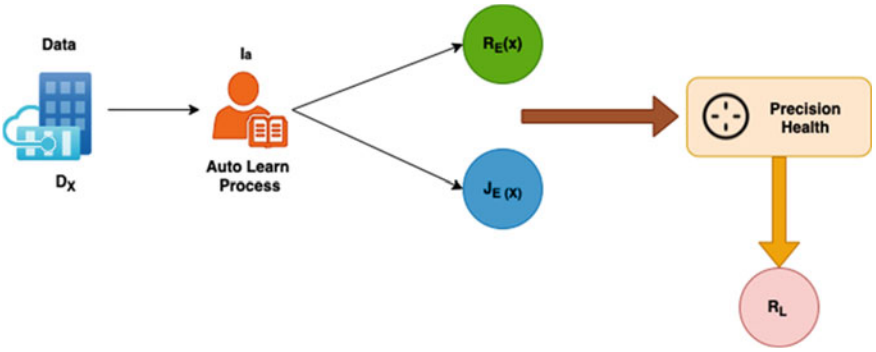


Fig. 3 Processing of data

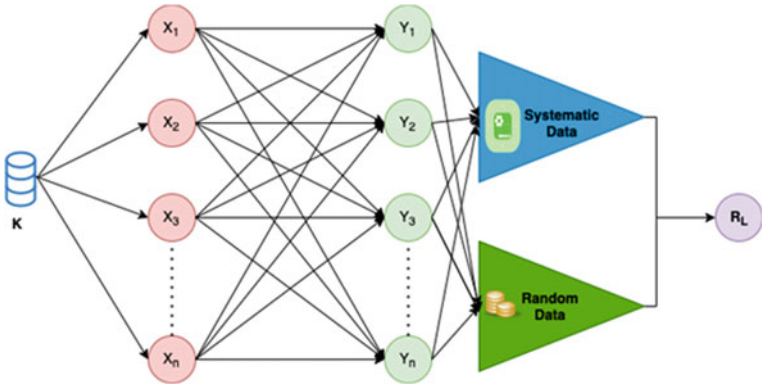


Fig. 4 Distribution of data

$$R_\theta(x) = \frac{1}{1 + e_{y\theta^x}} \tag{9}$$

where the combination of regressions functions is systematic and random as shown in Eq. (10).

$$J(\theta) = MS_\theta + \sum_{n=1}^{\infty}\left(a_n \mathrm{K}\frac{n\pi x}{L} + b_n \mathrm{K}\frac{n\pi x}{L}\right) \tag{10}$$

The health of a population is measured based on precision medicine which can help them for effective disease diagnosis, prophylaxis, and management.

$$R_L = \sqrt{(X_1 - Y_1)} + \sqrt{(X_2 - Y_2)} + \sqrt{(X_3 - Y_3)} \ldots \sqrt{(X_n - Y_n)} \tag{11}$$

Equation (11) shows that the X and Y axes include the input data used for training, whereas the plots contain the test data. The test data plots that are the closest to the ideal are then selected. Whenever possible, use an odd number for the nearest R_L value. By employing the bagging method to train many decision trees simultaneously, it generates a forest of these models.

4 Experimental Results and Discussion

The proposed RL-MRIS model experimental results have been executed, and the overall prediction ratio, convergence speed, prediction accuracy, performance ratio, and error rate are predicted. With further study, reinforcement learning can be applied with greater certainty. Equation (8) performs the prediction ratio of the risk identification of the proposed data. To validate the state of data based on the fixed function $R_L(Dx, Ia)$ in correlation with reinforcement learning. When applied to reinforcement learning yields favourable outcomes. Q-learning enables a wide variety of experiments to be conducted.

When the learning curve stops increasing and becomes flat, the R_L algorithm has converged. The variable data has been obtained through Eq. (10) which provides the following data sets. When set to 0, no new information is added to the values and autolearning is disabled. Using a large value like 0.9 expedites the learning process. The performance ratio of the reinforcement learning algorithm is provided with the sum of the structured data with the sum of the unstructured data with the function of missing values. The cross-validation of the data is generated through Eq. (7), and the ratio has been increased accordingly with 96% of identification.

The error rate is determined from Eq. (9) as the unit of data to the total number of data missing in the model where the error rate is predicted below 5% as the measure of the prediction error of the model is classified with respect to the proposed model. The systematic errors are measured with specific cause, whereas the random errors

caused due to random circumstances during the measurements and the negligent errors have been processed (Roberts and Anguraj 2020). The proposed RL-MRIS model improves the accuracy, convergence speed, ratio, and reduces the error rate when compared to other existing methods such as EEP, CNN-KNN, NBDT, AUPCR, and CINAH accordingly.

5 Conclusion

This paper proposed a method to identify the health issues for people with chronic diseases by reinforcement learning algorithm such as RL-MRIS and ALA. In RL-MRIS method, the performance of the algorithm is compared with the existing methods and provides the analysed results. Thus the missing information in the proposed system RL-MRIS can be detected with the high performance, and the identification and prediction of the diseases are accurate. The results are provided with the prediction accuracy of 97.56%, convergence speed of 98.45%, prediction ratio of 97.8%, accuracy of 98.9%, and the error rate of 19.45% when compared to other existing approaches.

References

Abubeker KM, Baskar S (2022) Internet of things based real-time hyperglycemia and hypoglycemia monitoring using wearable biosensors. In: 2022 IEEE international conference on electronics and renewable systems (ICEARS), pp 487–493

Abubeker KM, Baskar S (2022) Wireless sensor and wireless body area network assisted biosensor network for effective monitoring and prevention of non-ventilator hospital-acquired pneumonia. Frontiers Sustain Cities 16:4

Alanazi R (2022) Identification and prediction of chronic diseases using machine learning approach. J Healthc Eng 2022

Babu V. Survey paper on a machine learning methodology for diagnosing chronic kidney disease. Journal homepage: www.ijrpr.com ISSN, 2582, 7421.

Battineni G, Sagaro GG, Chinatalapudi N, Amenta F (2020) Applications of machine learning predictive models in the chronic disease diagnosis. J Personalized Med 10(2):21

Bhola G, Garg A, Kumari M (2021) Comparative study of machine learning techniques for chronic disease prognosis. In: Computer networks and inventive communication technologies. Springer, Singapore pp 131–144

Feng Y, Wang Y, Zeng C, Mao H (2021) Artificial intelligence and machine learning in chronic airway diseases: focus on asthma and chronic obstructive pulmonary disease. Int J Med Sci 18(13):2871

Getu ER, Samuel M, Bonventre JV, Celi LA, Mattie H (2022) Machine learning for acute kidney injury prediction in the intensive care unit. Adv Chronic Kidney Dis 29(5):431–438

Hassoun S, Bruckmann C, Ciardullo S, Perseghin G, Di Gaudio F, Broccolo F (2023) Setting up of a machine learning algorithm for the identification of severe liver fibrosis profile in the general US population cohort. Int J Med Inform 170:104932

Hoque MR, Mumu JR, Wanke P, Azad MAK (2022) Application of machine learning techniques in chronic disease literature: from citation mapping to research front. Int J Ind Syst Eng 42(2):193–210

Ikemura K, Bellin E, Yagi Y, Billett H, Saada M, Simone K, Stahl L, Szymanski J, Goldstein DY, Gil MR (2021a) Using automated machine learning to predict the mortality of patients: prediction model development study. J Med Internet Res 23(2):e23458

Ikemura K, Bellin E, Yagi Y, Billett H, Saada M, Simone K, Stahl L, Szymanski J, Goldstein DY, Gil MR (2021b) Using automated machine learning to predict the mortality of patients with COVID-19: prediction model development study. J Med Internet Res 23(2):e23458

Karthik S, Sudha M (2018) A survey on machine learning approaches in gene expression classification in modelling computational diagnostic system for complex diseases. Int J Eng Adv Technol 8(2):182–191

Kingston Roberts M, Kumari S, Anguraj P (18 March 2021) Certain investigations on recent advances in the design of decoding algorithms using low-density parity-check codes and its applications. Int J Commun Syst 34(8):e4765

Kor CT, Li YR, Lin PR, Lin SH, Wang BY, Lin CH (2022) Explainable machine learning model for predicting first-time acute exacerbation in patients with chronic obstructive pulmonary disease. J Personalized Med 12(2):228

Moore R, Ashby K, Liao TJ, Chen M (2021) Machine learning to identify interaction of single-nucleotide polymorphisms as a risk factor for chronic drug-induced liver injury. Int J Environ Res Public Health 18(20):10603

NagabhushanaRao M, Mohana RM, Talasila V, Sureshkumar M. Prediction of chronic diseases at an early phase using machine learning approach. Turk J Physiotherapy Rehabil 32:3

Padmanaban KA, Parthiban G (2016) Applying machine learning techniques for predicting the risk of chronic kidney disease. Indian J Sci Technol 9(29):1–6

Remy-Jardin M, Faivre JB, Kaergel R, Hutt A, Felloni P, Khung S, Lejeune AL, Giordano J, Remy J (2020) Machine learning and deep neural network applications in the thorax: pulmonary embolism, chronic thromboembolic pulmonary hypertension, aorta, and chronic obstructive pulmonary disease. J Thorac Imaging 35:S40–S48

Roberts MK, Anguraj P (2020) A comparative review of recent advances in decoding algorithms for low-density parity-check (LDPC) codes and their applications. Arch Comput Methods Eng 28(4):2225–2251

Shruthigedam, Shafiee NSM, Mutalib S (2020) Prediction of mental health problems among higher education student using machine learning. Int J Educ Manag Eng (IJEME) 10(6):1–9

Srivani M, Mala T, Murugappan A (2020) A survey on precision treatment for humans using cognitive machine learning techniques. In: Handbook of research on emerging trends and applications of machine learning. IGI Global, pp 79–106

Yajurved J, Prasad PS (2022) Analysis of chronic disease (Liver) prediction using machine learning. J Positive Sch Psychol 5489–5496

Metrics Evaluation of Bell Pepper Disease Classification Using Deep Convolutional Neural Network (DCNN)

K. Sowmiya and M. Thenmozhi

Abstract Crop disease is one of the major issues in agricultural fields. It reduces the food production process and produces a huge economic loss for farmers in agricultural lands. Deep learning models are built for detecting and identifying diseases at an early stage to help the farmers and provide food security. In this paper, transfer learning approach is used to identify and classify the diseased portion of leaves in farming lands. The benchmark dataset was taken from the Internet (Kaggle) for bell pepper leaf for identifying the diseases such as pepper leaf spot, mosaic virus, blight, damping off disease, verticillium wilt, blossom end rot, yellowing leaves, curling leaves, and healthy leaves. Ten pre-trained models are built for recognizing diseases and compared to obtain better accuracy. Adam optimizer and binary cross-entropy is used in the model for calculating loss function. In this paper, experimental results are presented on bell pepper diseases for MobileNetV2 with better accuracy of 99.42%. The various performance metrics such as accuracy, precision, recall and F1-score, ROC curve are used to determine the accuracy of the model.

Keywords Crops · Deep learning · Bell pepper leaf · Transfer learning

1 Introduction

Animal farming and plants growth in the farming field are known as agriculture. The Indian economy is based on agriculture and primary source of income for farmers. Plant disease is important problem faced by farmers in day-to-day life in the agriculture which cause loss to the plant as well as plant produce and in charge of keeping

K. Sowmiya
Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

M. Thenmozhi (✉)
Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
e-mail: thenmozm@srmist.edu.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_11

a close eye on material and financial losses, and it is due to deficiency in nutrients, damages in pesticides, mutations of genetics, or illness in biotic. Computer vision plays vital role for detecting and predicting the plant diseases in the agriculture. Convolutional neural network (CNN) is subpart of artificial neural network and mainly used in image classification or object recognition. It plays most important role for solving computer vision problems. Without the help of human's instructions, computer can recognize characteristics of the objects. This study mainly focuses on the plant disease classification and identification with locating and segmenting the plant and is achieved by the deep learning models such as AlexNet, VGGNet, MobileNet, GoogleNet, DenseNet, and ResNet which are pre-trained models used to predict and classify the bud diseases in agriculture. The main focus of the paper is to deal with the bell pepper disease classification using different models and comparison of those models for accuracy. Section 2 explains the literature review. Section 3 explores various pre-trained models and their methodology. Section 4 deals with the experimental results, and Sect. 5 deals with the conclusion.

2 Literature Review

2.1 Leaf Disease Types

The following are the types of bell pepper leaf diseases which are due to bacteria or viruses or fungus such as pepper leaf spot, mosaic virus, blight, damping off disease, verticillium wilt, blossom end rot, yellowing leaves, curling leaves, and healthy leaves. These are the most common leaf diseases in farming land. The images are types of bell pepper diseased leaves in the dataset that can identify and recognize at early stage. It can be due to temperature, humidity, moisture of soil, pH level in soil, type of soil, and fertility of soil. The diseases in the pathogens (biotic) are consisted of fungus, bacteria, or virus infectious portion in the diseased leaf and (abiotic) are temperature, humidity, moisture of soil, pH level in soil, type of soil, and fertility of soil. Figure 1 explains about the types bell pepper leaves.

2.2 Related Work

Aditya Rajbongshi et al. explain that only mango leaf disease is taken with five specific classes, but in future more leaf diseases need to be predicted. Compare with other models, DenseNet201 model attains better accuracy of 98.00% and also seven more metrics are performed at the early stage (Hanh and Manh 2022).

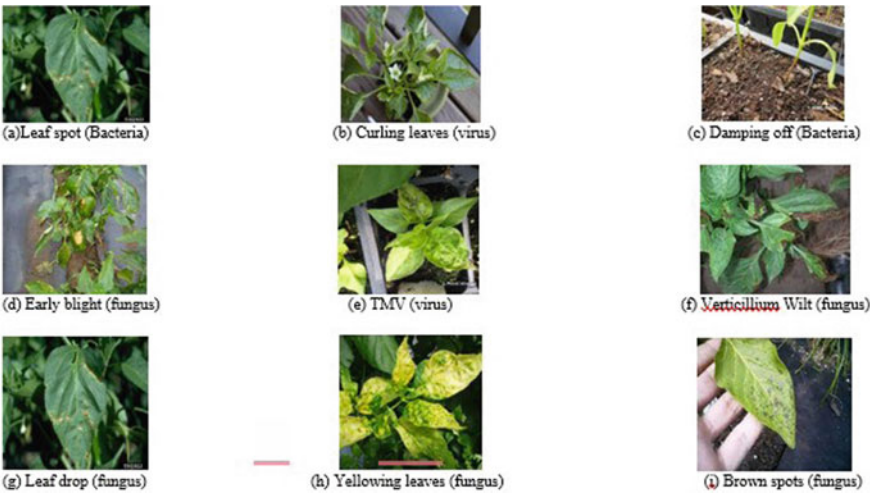


Fig. 1 Types of leaf diseases in bell pepper

Igor V. Arinichev et al. explain that four CNN models are compared such as DenseNet121, ResNet18, SqueezeNet1.0, and GoogleNet, but DenseNet121 will be performed better with accuracy of 95.57% with 20 epochs for rice leaf diseases. To improve the quality of multi-crop diseases images with real time agricultural field (Arinichev et al. 2021).

Ümit Atila et al. explain about EfficientNet model for proving that manually tough job for detecting leaf diseases in the agricultural field. It is also compared with other models such as AlexNet, VGGNet, ResNet, and Inception-Net, but EfficientNet performs well and attains better accuracy with 99.91% for B4 model and 99.97% for B5 model in both original and augmented datasets. It has multicrops in both original and augmented datasets (Atila et al. 2021).

Bui Thi Hanh et al. explain about leaf disease prediction at an early stage. EfficientNet B3 and B5 are used to classify the diseased leaves with better accuracy (99.97%) with 45 epochs for both augmented and non-augmented datasets. It has been trained, tested, and validated in an efficient manner (Hanh et al. 2022).

Borja Espejo-Garcia et al. explains about deep nutrient deficiency problem which will be overcome with help of images in pre-trained models. Two benchmark datasets are taken to recognize the sugar beet and orange leaf diseases. About 98.52 and 98.65% accuracies are obtained for both different datasets. In the future, the object detection approach is used as a classifier for nutrient deficiency problems (Espejo-Garcia et al. 2022).

Sedrick Scott Keh et al. explain about the problem in this paper which is used to take single leaf images that were classified and recognized. Three models are compared such as VGG16, ResNet101, and DenseNet16. Out of these models, DenseNet16 performs very well with 96.2% accuracy and improved a lot for avoiding fluctuations rate. Noisy student model (labeled and unlabeled datasets) is the semi-supervised self-training model which is used to do classification and results test set as 98.2% (Keh 2012).

Yeong Hyeon Gu et al. explains multi-classification leaf disease diagnostic proposed method used to do classify more diseases. The proposed method is used to provide solution as the input leaf diseased images are more or less than equal to classified output images. It is used to recognize both apple and pear leaf diseased images with six variety of diseases. Dimensionality reduction method used in proposed method to speed up execution of model also reduces time consumption of recognition rate (Gu et al. 2022).

Sophia L. Sanga et al. explain about identification of banana leaf diseases at early stage using transfer learning techniques. VGG16, ResNet18, ResNet50, ResNet152, and InceptionV3 were five models used to identify banana disease with better accuracy 99.2% achieved by ResNet152 model (Sanga et al. 2020).

Madallah Alruwaili et al. explain about detection of olive leaf disease in efficient way with help of deep learning model that is different from other models. The proposed method enhances its performance and achieved better performance metrics such as accuracy (99.11%), precision (99.49%), recall (99.11%), F1-score (99.29%). The aim of this proposed method is to reduce the yield loss and help the farmers by high production rate (Alruwaili et al. 2019).

Rajinder Kumar M. Math et al. explain about how the crop diseases are identified and protected by factors affecting disease in environmental changes and it influence the reducing economic loss of farmers. In this paper, grapes leaf disease diagnosis is done by help of deep learning models and they are compared with each other to which model get better accuracy (99.34%). The models used in this report are InceptionV3, ResNet50, VGG16, VGG19, and Xception. It also concentrates on other metrics such as precision, recall, and F1-score (Math and Dharwadkar 2022).

Muhammad Hammad Saleem et al. explain about leaf disease detection and classification that are identified at initial stage to save farmer's crop production and minimize the economic loss in agricultural fields. It gives solution to problem by visualizing more diseases recognized in RGB images. When compared to EfficientNetV2, the accuracy obtained by the EfficientNetB4 model (noisy student approach) is 98.65%. EfficientNetV2 is important for removing undesirable portion of leaf (Saleem et al. 2019).

3 Methodology

3.1 Dataset

A publicly available benchmark dataset is taken from Plant Village, Kaggle which has bell pepper leaf with 2475 images (https://www.kaggle.com/datasets/emmarex/plantdisease). All the images used in this dataset will be 224 * 224 * 3, but for Xception, Inception, and InceptionResNet, image size will be 299 * 299 * 3. It consists of bell pepper leaf diseases which are due to bacteria or virus or fungus. Healthy leaf is represented by class 0 and diseased leaf is represented by class 1. Binary classification is used to classify the crops by two outcomes healthy (0) or diseased. The Hyper-parameters used in this research work are (i) **Activation Function**—Sigmoid, (ii) **Loss Function**—Binary Cross-Entropy, (iii) **Optimizer**—Adam. The evaluation metrics used in bell pepper leaf disease binary classification are accuracy, precision, recall, F1-score, confusion matrix, specificity, sensitivity, ROC curve, etc. Table 1 explains about the train (80%), test (10%), and validation (10%) split ratio of dataset.

Figure 2 explains about CNN methodology architecture diagram.

Table 1 Dataset split

S. No.	Class	Train (80%)	Validation (10%)	Test (10%)	Total count
1	Class 0—Healthy	1035	221	222	1478
2	Class 1—Diseased	698	149	150	997

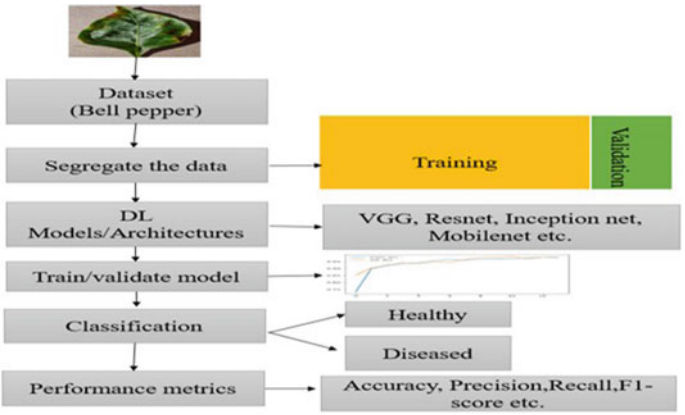


Fig. 2 Architecture diagram of methodology

Table 2 Fine-tuning of models

S. No.	Hyperparameter	Value
1	Batch size	32
2	Epochs	30
3	Optimizer	Adam
4	Loss entropy	Binary cross
5	Learning rate	0.001

3.2 CNN Architecture

Convolutional neural networks (CNN) are a type of artificial neural network used to analyze visual imagery in deep learning. Convolution is the process of multiplication of matrix input and filter to form feature maps for extracting important features in convolutional neural network (CNN). Convolutional neural networks are composed of many layers of artificial neurons. It takes mathematical form of summation of weights and bias with activation function of inputs in the neurons. Every layer takes activation function to the input in order to produce desirable output. The features (even main complex features also) are extracted in convolutional layer which are taken as edges or corners for better recognition.

Data dimensionality reduction is accomplished in the pooling layer by reducing the amount of the data. Additionally, data computation time should be kept to a minimum. It has two types such as max pooling and average pooling. Max pooling is used to take maximum value of input data with feature map and average pooling is used to take average value of input data with feature map. Compared with average pooling, max pooling performs better. Fully connected layer is used to do classification and it takes input as nonlinear function, i.e., Rectified Linear Unit (ReLU) function. Backpropagation is conducted to every iteration of training after the flattened output is given to a feedforward neural network. Softmax activation function is used for classification that is appropriate for both binary and multi-class classifications.

Table 2 explains models' fine-tuning using hyper-parameters.

3.3 Transfer Learning

Transfer learning is the approach used for reusing the existing model's knowledge to latest model. It will be used for generalizing the training model and also able to solve the complex problems. It can also be beneficial for training the models by fewer datasets and able to provide better accuracy when compared to training from scratch the models.

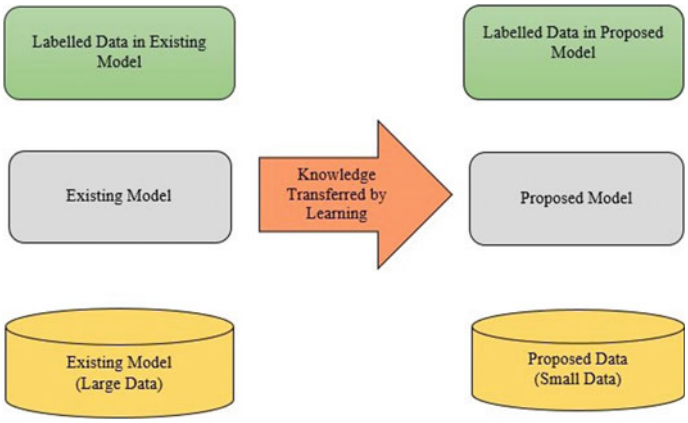


Fig. 3 Diagram for transfer learning

General Procedure for Transfer Learning:

Step 1: In large dataset, pre-trained CNN models are trained.
Step 2: Then, frozen some of the layers (convolutional layers) for adjusting weights.
Step 3: While training the model parameters, join with classifiers.
Step 4: Train the model by trained classifiers for data availability.
Step 5: After the model's training gets completed, fine-tune parameters and unfreeze those convolutional layers (Fig. 3).

4 Experimental Results

In this research, the comparative analysis of pre-trained models is used to do classification (Class 0 and Class 1) with 30 epochs (automatically stopped by early stopping). It has both accuracy and loss executed graph results for evaluating the performance of model to be fitted in correct way. The performance metrics such as accuracy, loss, precision, recall, and F1-score are used to evaluate the model's performance which are compared to be predicted correctly and executed with results. Receiver Operator Characteristic (ROC) curve is used for evaluating classification of both binary and multi-classification problems (Tables 3 and 4).

4.1 Performance Evaluation Metrics

The Table 4 (above table) briefly explains about precision, recall, and F1-score calculation for classification results in all models. All models give results after running 30

Table 3 Evaluating metrics of models

S. No.	Pre-trained models	No. of epochs	Accuracy	Loss	Precision	Recall	F1-score
1	VGG (16/19)	30	0.9593	0.1248	0.99	0.98	0.97
2	ResNet (150/152)	30	0.7817/0.7309	0.4405	0.92	0.79	0.89
3	Inc-ResNet	30	0.9746	0.1273	0.98	0.99	0.94
4	InceptionV3	30	0.9492	0.2023	0.99	0.98	0.98
5	MobileNetV2	30	0.9949	0.0152	0.98	0.99	0.98
6	DenseNet (12/169)	30	99.49	0.003	0.99	0.99	0.99
7	Xceptionet	30	0.9695	0.104	0.97	0.98	0.99

Table 4 Confusion matrix of models

S. No.	Pre-trained models	TP	TN	FP	FN
1	VGG (16/19)	142	47	7	1
2	ResNet (150/152)	149	48	0	0
3	Inc-ResNet	142	47	1	0
4	InceptionV3	142	47	7	1
5	MobileNetV2	140	47	1	9
6	DenseNet (121/169)	149	47	1	0
7	Xceptionet	142	48	7	1

epochs (runs up to certain epochs due to early stopping) in order to make model in better way. The following equations (such as accuracy, sensitivity, specificity, precision and recall) are used to calculate the performance metrics of the training datasets, such as (Figs. 4, 5, and 6),

$$\begin{aligned}
\text{Accuracy} &= (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + \text{FN}),\\
\text{Sensitivity} &= \text{TP}/\text{TP} + \text{FN},\\
\text{Specificity} &= \text{TN}/\text{TN} + \text{FP},\\
\text{Precision} &= \text{TP}/\text{TP} + \text{FP},\\
\text{Recall} &= \text{TP}/\text{TP} + \text{FN},\\
\text{F1-score} &= 2 * (\text{precision} * \text{recall}/\text{precision} + \text{recall}).
\end{aligned}$$

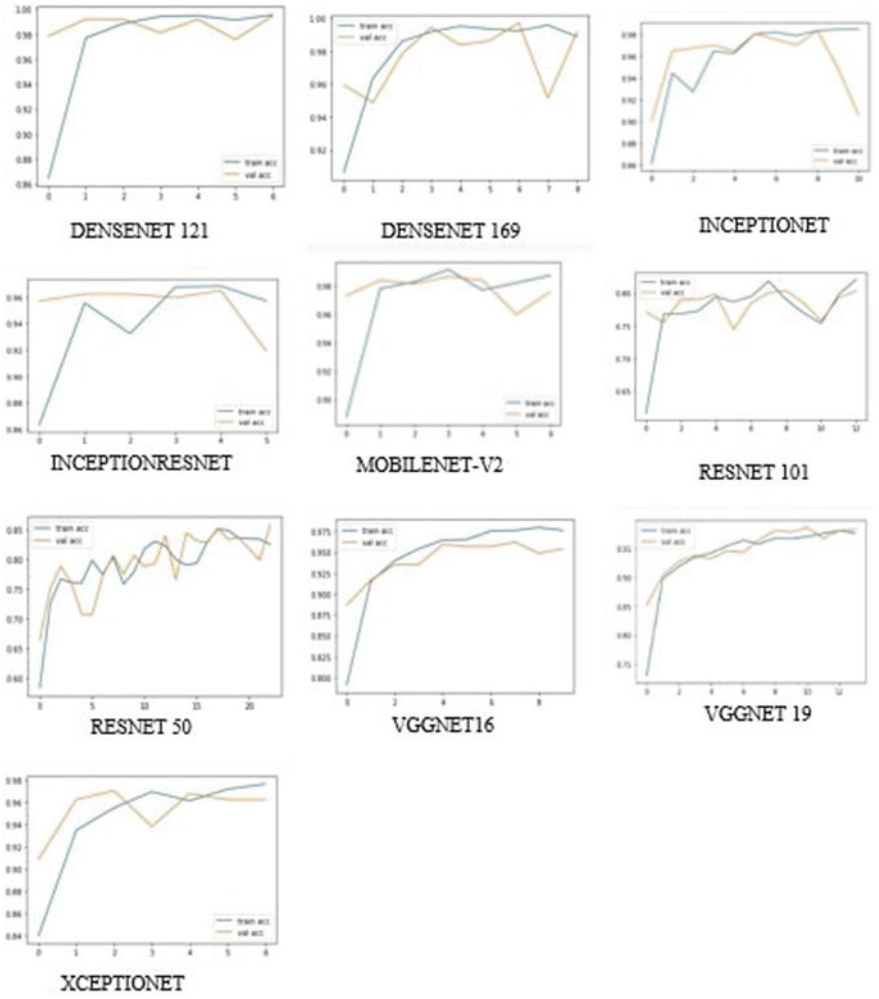


Fig. 4 Models—accuracy

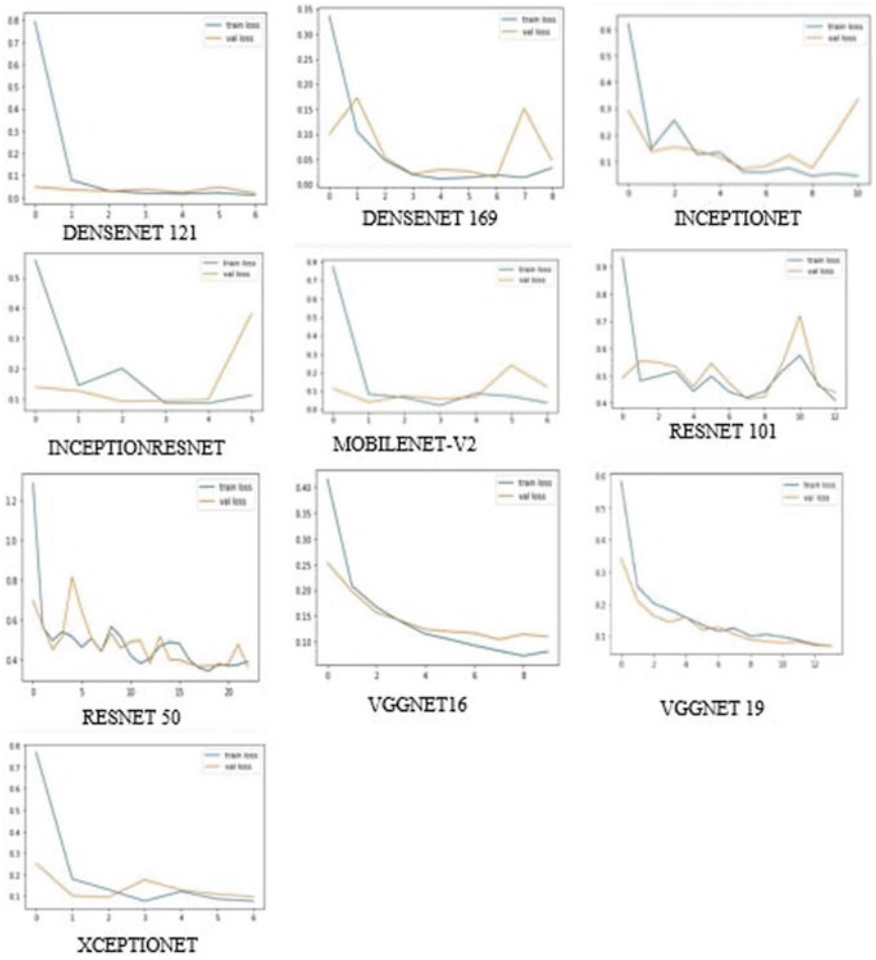


Fig. 5 Models—loss

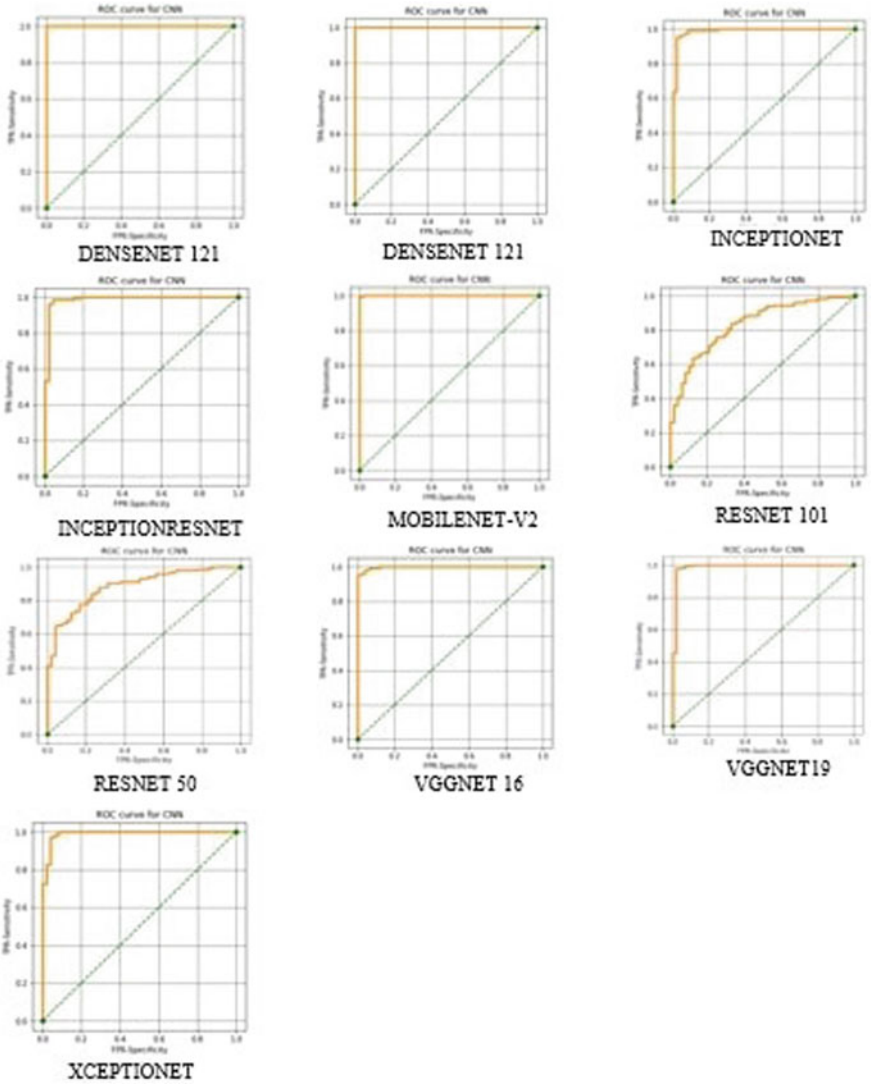


Fig. 6 Models—ROC curve

5 Conclusion

In this paper, ten pre-trained models are compared to get the best among other models and classify them appropriately. The bell pepper leaf disease benchmark dataset is used for classifying both healthy and diseased leaves. Since much research is going on leaf disease classification and prediction and considered a hot topic of the last decade. The experimental results show about performance metrics of the

pre-trained model's better performance. Adam optimizer is used as the optimization algorithm and binary cross-entropy is used as the loss function. The compared pre-trained models using CNN can detect and classify the leaf disease detection. It will be very useful for the farmers to identify what kind of disease is in the leaves. Out of all models, MobileNetV2 has better accuracy and minimum loss rate with 30 epochs (with early stopping). Finally, we come to the conclusion that the experimental results of MobileNetV2 have obtained 99.49% accuracy and performed well. The future work will be improving accuracy for the collection of multi-crops with varieties of diseases and how they will be used for multi-classification of leaf disease classification.

References

Alruwaili M, Alanazi S, Abd El-Ghany S, Shehab A (2019) An efficient deep learning model for olive diseases detection. Int J Adv Comput Sci Appl Sci Inf (SAI) Organization Limited

Arinichev IV et al (2021) Rice fungal diseases recognition using modern computer vision techniques. Int J Fuzzy Logic Intell Syst 21(1):1–11

Atila Ü et al (2021) Plant leaf disease classification using EfficientNet deep learning model. Eco Inform 61:101182

Espejo-Garcia B et al (2022) Using EfficientNet and transfer learning for image-based diagnosis of nutrient deficiencies. Comput Electron Agric 196:106868

Gu YH, Yin H, Jin D, Zheng R, Yoo SJ (2022) Improved multi-plant disease recognition method using deep convolutional neural networks in six diseases of apples and pears. Agriculture 12(2):300

Hanh BT,Van Manh H, Nguyen N-V (2022) Enhancing the performance of transferred efficientnet models in leaf image-based plant disease classification. J Plant Dis Prot 129(3):623–634

Keh SS (2020) Semi-supervised noisy student pre-training on efficient net architectures for plant pathology classification. arXiv preprint arXiv:2012.00332

Math RKM, Dharwadkar NV (2022) Early detection and identification of grape diseases using convolutional neural networks. J Plant Dis Prot

Saleem MH, Potgieter J, Arif KM (2019) Plant disease detection and classification by deep learning. Plants

Sanga S, Mero V, Machuve D, Mwanganda D (2020) Mobile-based deep learning models for banana diseases detection. arXiv preprint arXiv:2004.03718

Analysis of Heart Disease Prediction Using Various Machine Learning Algorithms

G. Sakthipriya, Y. Suresh, C. Varnisha, R. Sindhu, and R. Shivraj

Abstract Every year due to heart diseases around 12 million deaths have been confirmed by the World Health Organization. Cardiovascular disease is responsible for about one in every 4 deaths in the World, and the risk only rises with age. Even though this disease is very common, it can be reduced using smart predictions and diagnoses based on relevant medical data. This paper focuses on the relevant risk factors that are causing the heart diseases. There are two categories created from these risk factors. The patient's age, sex, and family history make up the first group. The patient's lifestyle-related risk factors are included in the second category. These factors include maximum blood pressure, smoking, a high level of cholesterol, and inactivity. We have used algorithms like Logistic Regression, Support Vector Classifier, and Random Forest to predict the risk level of heart diseases and we made a comparative study on these three algorithms.

Keywords Heart disease prediction · Machine learning · Logistic regression · Support vector machine (SVM) · Random forest

1 Introduction

In the total population of the deaths caused by cardio vascular diseases half of the population is in developed and developing countries. Shortness of breath, swelling feet, and generalized weakness are common signs of heart disease (Nikhar and Karandikar 2019). To prevent the heart diseases and stroke certain work has to be

G. Sakthipriya (✉) · Y. Suresh · C. Varnisha · R. Sindhu · R. Shivraj
Department of Information Technology, Sona College of Technology, Salem, Tamil Nadu 636005, India
e-mail: sakthipriyag.19it@sonatech.ac.in

Y. Suresh
e-mail: sureshy@sonatech.ac.in

R. Shivraj
e-mail: sharkshiv7@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_12

done. This helps in medical support and care if the disease is predicted in advance doctors can treat the people as per their symptoms (Savita et al. 2020).

A model that uses machine learning to forecast diseases uses data from users to make its predictions. Based on the information, symptoms, or necessary parameters given by the user, it makes a diagnosis for the patient or user and displays the results accordingly. The information to predict the heart disease might include age, sex, cholesterol, old peak, fasting blood sugar (fbs), etc. (Ali et al. 2019). It's a system that predicts whether or not the patient is affected by heart disease. To understand about the diseases, the user just has to give the information in the form of input as features and then the model will tell the person is affected by the heart disease or not. Using this kind of models, the healthcare providers can also help the people as soon as possible just by asking health information and symptoms (Beyene and Pooja 2018).

There are number of ML systems which have worked on this basis. But what is unique about our work is to do comparative analysis among three machine learning algorithms and to find the best fitting algorithms among them. And use that algorithm for larger dataset. The reason for developing this model is because nowadays doctors are using many methodologies and technologies to diagnosing the diseases (Ramesh et al. 2022). The exact analysis of the diseases helps for the successful treatment for the patient. The ultimate reason behind developing this model is to beat the heart diseases at its early stage and help the doctors to treat them (Bashir et al. 2019).

2 Literature Survey

Machine learning works similar to the human. A human learns from the experiences they undergo, similarly a machine learning algorithm also learn from the experiences that is the data we give the algorithm to train it. Various machine learning algorithms are useful in prediction of heart disease in medical field such as Regression, Linear Regression, Naive Bayes Classifier, Logistic Regression, Bayes theorem, Decision Tress, K-Nearest Neighbor Classifier (KNN), Entropy, XG-Boost, Ada-Boost, Support Vector Machines (SVM), Random Forest, and K-means Algorithm (Pe et al. 2021). In addition to these algorithms, the deep learning algorithms like Long Short-Term Memory Networks (LSTMs), Convolutional Neural Networks (CNNs) are more useful for predictions (Kumar et al. 2020). The data mining technique can also be used, this technique can include using a single algorithm or combination of two or more algorithms for prediction (Gandhi and Singh 2015).

According to the researches that are done, heart diseases have become so common nowadays. Researchers say that women are more prone to get heart disease than men (Gomathi et al. 2022). It is always recommended to get to know about the heart disease earlier and visit doctor for the diagnosis and to know about the intensity of that disease. For predictions, it is found that Random Forest, Decision Tress, and ANN contribute a lot and gives better results and can identify the factors that

influence much (Akilandeswari et al. 2021). The input features, in addition to the algorithms used, are crucial in the disease prediction process (Akhil et al. 2013).

This paper mainly focuses on improving the accuracy in the prediction in order for the final results to be precise. Various researches have been undertaken by more number of researchers for the heart disease prediction.

Mohan et al. (2019) proposed a hybrid machine learning algorithm which is called as Hybrid Random Forest Linear Method (HRFLM). This algorithm is a combination of two algorithms that include Random Forest (RM) and Linear Regression Method (LM). They divided their work into 4 stages where in the first stage they partition the dataset, in the second stage with few rules they classify the input dataset. In the third stage they extract features using Less Error Classifier. The Less Error Classifier finds the difference between the maximum and the minimum error rate. In the fourth stage, they implement their Hybrid Algorithm (HRFLM).

Singh et al. (2019) proposed a system where they have tried their work with 5 different algorithms that include Random Forest, Support Vector Classifier, Logistic Regression, Linear Regression, Decision Tree. The Random Forest Algorithm they applied gave them the best accuracy of 85.81% with 75 number if trees that are splitted.

Rajdhan et al. (2020) have analyzed the accuracy score, F-score, precision, and recall of the algorithms like the Logistic Regression, Decision Tree, Random Forest, and Naïve Bayes. They got better accuracy for Decision Tree. The accuracy is around 90.16%.

Rindhe et al. (2021) have analyzed seven algorithms which includes XG-Boost, SVM, LR, RF, Naïve Bayes, Decision Tree, Ada-Boost. The found that XG-Boost have given them the highest accuracy of 81.3%.

Shalev-Shwartz and Ben-David (2014) have used three different algorithms that include SVM, Random Forest, and Artificial Neural Network (ANN). From that SVM has given them the accuracy of 84%.

From the references we have inferred, it is found that the Support Vector Machine (SVM) and the Random Forest (RF) and Logistic Regression algorithm gives better results and accuracy.

3 Proposed Methodology

In this work, we have collected dataset with attributes that play an important role in identifying the disease of the heart. This dataset is trained and a model is built in such a way that it will predict whether the person is affected by any kind of heart disease or not. We have used three algorithms like Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF) algorithms to train the model.

We have used the Cleveland dataset with 303 tuples and 14 attributes including one target column for comparing the accuracies. Age, sex, chest pain (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood pressure (fbs), resting electrocardiographic results (restecg), exercise induced angia (exang), old depression

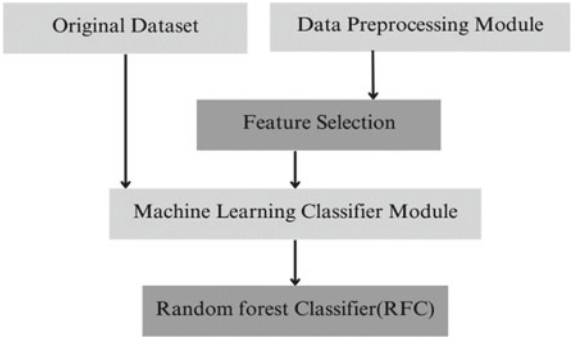


Fig. 1 Proposed classifier

induced by exercise (peak), peak exercise (slope), number of major vessels colored by fluoroscopy (ca), thalassemia-type blood disorder (thal), and dependent variable (target) are among the 14 attributes. All the three algorithms are applied on this dataset and the accuracies are compared and the algorithm with better accuracy is chosen for the final prediction.

The proposed system is developed with the help of various diagrams and designs as the base like the Architecture diagram, the sequence, activity, and the object diagram. Keeping base as all these diagrams a predictive approach was developed.

3.1 Classifiers for Machine Learning Proposed

The suggested method was applied to the dataset after the dataset had been thoroughly examined, and a highly well-liked ensemble method called Random Forest Classifier was then utilized.

Figure 1 shows the proposed system where from the original dataset, processes like preprocessing and feature selection will be done followed by applying the classifier module which is the Random Forest Algorithm (RFC). In the preprocessing stage the dataset is checked for null values and those null values are removed, i.e., data cleaning happens. In the feature selection stage only the required and the most important attributes that contribute a lot in the heart disease prediction are taken and the other unnecessary attributes are removed. Now, the RFC algorithm is applied to the dataset.

3.2 Details of the Dataset

The dataset we have taken for final prediction is the Framingham dataset which has 16 features. This includes Sex: utilizes the following format to display the person's gender, 0 = female, 1 = male, Age: shows an individual's age, education, current smoker: a grownup who has smoked 100 cigarettes or more throughout their lives

and is still smoking, CigsPerDay: number of cigarettes smoked per day, BPMeds: Blood Pressure Medicines—whether the patient takes any blood pressure medicines, Prevalent Stroke: tells about the patient's history of strokes, Prevalent Hyp: whether or not the person has hypertension, and Diabetes: whether or not the person has diabetes. Compares a person's fasting blood sugar level to 120 mg/dl. 0 (false) if fasting blood sugar is less than 120 mg/dl, 1 (true) otherwise, Tot Chol: shows the serum cholesterol in mg/dl (unit), SysBP: tells the systolic blood pressure, DiaBP: tells the diastolic blood pressure, BMI: tells the body mass index, Heart Rate: shows the highest heart rate a person has ever reached, Glucose: tells the glucose level in the blood, and TenYearCHD: it is the target value. Using all the 14 features, we predicted the target value whether or not the patient has coronary heart disease.

3.3 Feature Selection

Feature selection simply means selecting the features that contribute a lot in the dataset and those which helps in giving the optimized result. Feature selection is commonly used for large dimensional dataset with 300 and more columns. The feature selection techniques are broadly classified into two types like the supervised feature selection techniques and the unsupervised techniques where the labeled and unlabeled data are used for the feature selection, respectively. Feature selection play an important role in getting the accurate results. Having unnecessary features in the dataset will increase the computational time as well as will not give good results. The common techniques of feature selection include Variance threshold, Information Gain, Mean Absolute Difference, Correlation Coefficient, etc. In our project the Framingham dataset taken has got only 16 features and among those features all the columns were considered to be contributing in predicting the heart disease expect the education column which got constant values. We dropped that column with the variance threshold technique. And with the remaining features the prediction was done.

3.4 Algorithm Used

Random Forest Classifier (RFC). A non-parametric approach called Random Forest can be applied to classification and regression problems. They are one of the most popular ensemble methods that belong to a certain category called bagging method. Ensemble method helps a lot in increasing the performance. A Random Forest Classifier and Regressor is a bagging technique. In bagging technique, we have many base learner models which can be considered as M1, M2, and so on. These models can be basically called as decision trees.

For example, consider the dataset D where we have m number of rows and n number of columns. We take a sample of records $D'(D' < D)$ with the help of row

sampling and feature sampling and give it to the base learners M1, M2, M3, and so no. The base learner models are nothing but the decision trees because in other simpler words Random Forest can be considered as a group of decision trees. The row sampling happen with replacement. Now, the decision trees will be trained on the given data. Through Boot strapping we have splitted as decision trees which can also be called as the base learners. And with the help of bagging we now take the results given by each decision tree according to the majority number of votes. In other words if the results given by Decision Tree 1 (DT1) is 1 and the results of DT2, DT3, and DT4 are 1, 0, and 1, respectively, then according to majority the final result will be taken as 1.

4 Result Analysis

In our work, the Framingham dataset that is used includes all the necessary features that helps in determining the heart disease of the patient. All the 16 features except the dependent variable column and the education column contribute a lot in the prediction of the disease. But all the existing works were found to be using a smaller dataset with the features that are not sufficient to predict very accurately. Out of 4238 individual rows or entries in the Framingham dataset 3594 patients do not have the heart disease which is predicted as 0 and the remaining 644 are identified having a heart disease which is predicted as 1.

Also, a comparative analysis with the three algorithms is done with the Cleveland dataset. The accuracies are obtained as tabulated as shown in Table 1.

Comparative Analysis

Figure 2 shows the comparative study that is done on the three algorithms. The Cleveland dataset is used for this study. The accuracies of all these algorithms were compared. From the study we inferred that SVM gave us the least accuracy of 81.97%. Logistic Regression gave an accuracy of 88.25% and Random Forest gave a maximum accuracy of 90.16%. So, the RF algorithm is chosen as the best algorithm for the final prediction.

Figure 3 shows the final result of those patients who have the heart disease and those who do not. For the final prediction the Framingham dataset is used. Then, the Random Forest Algorithm is used which gave us the better accuracy in the comparative study.

Table 1 The accuracy of the respective algorithm that are obtained

Algorithms	Accuracy (%)
Logistic regression	88.25
SVM	81.97
Random forest	90.16

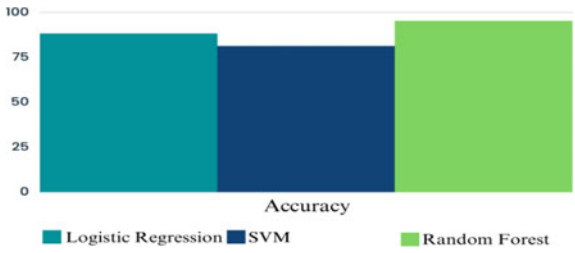


Fig. 2 Comparative study on the ML algorithms

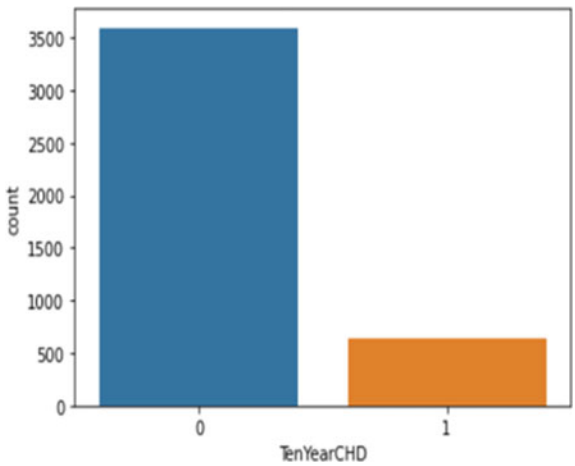


Fig. 3 Final prediction

5 Conclusion

We conducted a comparison study in this paper, and encouraging results were obtained. We came to the conclusion that this analysis was better served by machine learning methods. For comparative analysis we have used 3 algorithms including Random Forest Classifier, Support Vector Classifier, and Logistic Regression. We have used the Cleveland dataset for comparative analysis and Random Forest Classifier gave an accuracy of 90.16%. Then, using the Framingham dataset we have built a model and predicted whether the person has heart disease or not using RFC and we achieved a hundred percent accuracy for the dataset we used. The methods which are used for evaluation are classification report and accuracy score.

References

Akhil J, Deekshatulu B, Chandra P (2013) Classification of heart disease using K-nearest neighbor and genetic algorithm. Procedia Technol 10:85–94. https://doi.org/10.1016/j.protcy.2013.12.340

Akilandeswari J, Jothi G, Naveenkumar A, Iyyanar P, Paramasivam ME (2021) Detecting pulmonary embolism using deep neural networks. Int J Perform Eng 17(3):322–332

Ali L, Niamat A, Khan J, Amiri Golilarz N, Xingzhong X, Noor A, Nour R, Bukhari SAC (2019) An optimized stacked support vector machines based expert system for the effective prediction of heart failure. IEEE Access 7:54007–54014. https://doi.org/10.1109/ACCESS.2019.2909969

Bashir S, Khan Z, Khan F, Anjum A, Bashir K (2019) Improving heart disease prediction using feature selection approaches, pp 619–623. https://doi.org/10.1109/IBCAST.2019.8667106

Beyene C, Kamat P (2018) Survey on prediction and analysis the occurrence of heart disease using data mining techniques. Int J Pure Appl Math 118:165–173

Gandhi M, Singh SN (2015) Predictions in heart disease using techniques of data mining. In: 2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE), Greater Noida, India, pp 520–525. https://doi.org/10.1109/ABLAZE.2015.7154917

Gomathi JJ, Emilyn AS, Thamburaj AS, Vinod Kumar D (2022) Real time air pollution pre diction in urban cities using deep learning algorithms and IoT. In: 2022 7th international conference on communication and electronics systems (ICCES), Coimbatore, India, pp 340–343. https://doi.org/10.1109/ICCES54183.2022.9835991

Kumar A, Kumar P, Srivastava A, Ambeth Kumar VD, Vengatesan K, Singhal A (2020) Comparative analysis of data mining techniques to predict heart disease for diabetic patients. https://doi.org/10.1007/978-981-15-6634-9_46

Mohan S, Thirumalai C, Srivastava G (2019) Effective heart disease prediction using hybrid machine learning techniques. IEEE Access 7:81542–81554. https://doi.org/10.1109/ACCESS.2019.2923707

Nikhar S, Karandikar AM (2019) Prediction of heart disease using machine learning algorithms. In: 2019 1st international conference on innovations in information and communication technology (ICIICT), Chennai, India, pp 1–5. https://doi.org/10.1109/ICIICT1.2019.8741465

Pe R, Subasini C, Katharine A, Kumaresan V, Gowdhamkumar S, Nithya T (2021) A cardiovascular disease prediction using machine learning algorithms. Ann Romanian Soc Cell Biol 25:904–912

Rajdhan A, Agarwal A, Sai M, Ghuli P (2020) Heart disease prediction using machine learning. Int J Eng Research 9. https://doi.org/10.17577/IJERTV9IS040614

Ramesh P, Jeyaswamidoss J, Vijayakumar V (2022) Hybrid artificial neural networks using customer churn prediction. Wirel Personal Commun 124. https://doi.org/10.1007/s11277-021-09427-7

Rindhe B, Ahire N, Patil R, Gagare S, Darade M (2021) Heart disease prediction using machine learning. Int J Adv Res Sci Commun Technol, 267–276. https://doi.org/10.48175/IJARSCT-1131

Savita, Sharma G, Rani G, Dhaka V (2020) A review on machine learning techniques for prediction of cardiovascular diseases, pp 237–242. https://doi.org/10.1109/PDGC50313.2020.9315747

Shalev-Shwartz S, Ben-David S (2014) Understanding machine learning. In: From theory to algorithms. Cambridge University Press, Cambridge, UK

Singh Y, Sinha N, Singh S (2019) Heart disease prediction system using random forest, pp 613–623. https://doi.org/10.1007/978-981-10-5427-3_63

Analysis of Agricultural Commodities Prices Using BART: A Machine Learning Technique

Eva Mishra and R. Murugesan

Abstract The price analysis of agricultural commodities has both practical and theoretical aspect. In stark contrast to financial markets, this study purposes applying the Bayesian additive regression tree (BART)—a machine learning technique, for the evaluation and comparison of monthly prices of twenty-five agricultural commodities spanning 2011–2019. The maximum posterior inference function of the BART helps the model evaluate each variable's marginal effects. The model has the advantage of selecting the intrinsic variable in order to find solution for the problems. The model identifies key predictors and essential interactions among twenty-five independent variables, which are relevant for decision making on the market for agricultural commodities. Overall, the analysis showed that the Bayesian additive regression tree performs significantly better.

Keywords Bayesian additive regression trees · Agricultural commodities prices · Machine learning technique · Built-in variable analysis

1 Introduction

A crucial issue with both academia and industries is the analysis of the prices of agricultural goods for determining policy and fostering sustainable growth (Drachal 2019). The increase in food prices causes political and economic instability (Davenport and Funk 2015) effecting directly farmers (Thiyagarajan et al. 2015). Although studies on what factors influence agricultural prices are very common, literature does not look at work related to the ties between large numbers of spot commodity prices.

E. Mishra (✉) · R. Murugesan
Department of Humanities and Social Sciences, National Institute of Technology, Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India
e-mail: evamishra.nitt@gmail.com

R. Murugesan
e-mail: rmurugan@nitt.edu

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_13

Previous studies that include Autoregressive Integrated Moving average, Generalized Auto Regressive Conditional Heteroskedasticity, and Random Forest are unable to understand the influence of time-variation of several possible factors on the agricultural prices (Liu et al. 2020). From Bayesian econometrics, a natural solution is achieved. Bayesian econometrics although having usefulness in modeling agricultural prices, it has been less explored as concluded from the literature review (Andersson and Karlsson 2008; Mazur 2018; Osathanunkul et al. 2018). Bayesian additive regression tree's (BART) built-in variable collection is used to fix the problem of small n period compared to large no of p explanatory variables. Among predicting variables, the nonlinearity and interaction effects which are generally allowed in BART provides the ability to tackle the question of large tree models continuing to overfit by applying prior distributions to streamlining the fit of each node, such that every tree is only a trifle of the response variable variance. In comparison, by having a linear model for the response parameter, the Bayesian method helps one to compensate for an uncertainty function, tree, and forecast (Prüser 2019).

2 Methodology

2.1 BART Model

Bayesian additive regression tree (BART)—a machine learning process and nonparametric regression, is used for model-free prediction by keeping track of the frequencies of predictor inclusion (Ajidarma and Irianto 2019). It is represented as:

$$Y = f(Q) + \varepsilon \approx K_1^L(Q) + K_2^L(Q) + \cdots + K_l^L(Q) + \varepsilon, \quad \varepsilon \sim N_n(0, \sigma I_n) \quad (1)$$

Here,

Y	$n \times 1$ response vector
Q	$n \times$ p design matrix
ε	$n \times 1$ vector of noise
L	Distinct regression trees, marked by K
L, K	Denotes tree structure
L	Denotes parameter at the terminal nodes.

2.2 BART Model Fitting

To perk up one from the four moves: (1) Grow, (2) Prune, (3) Change, and (4) Swap and to get a posterior distribution, Metropolis–Hastings analysis and Markov chain Monte Carlo (MCMC) sampler have been utilized. At last, the draws from the full

conditionals $p(K_j, L_j \mid W_j, \sigma)$, are acquired utilizing the following two independent methods.

(1) Computing $p(K_j \mid W_j, \sigma) \propto p(K_j)\, p(W_j \mid K_j, \sigma)$
(2) Drawing from $p(L_j \mid K_j, W_j, \sigma)$.

The conjugate prior on the terminal node parameters $p(\mu_{ij} \mid K_j)$ allows the μ_{ij} marginalized parameters out of the full conditional for $W_j \mid K_j, \sigma^{-2}$:

$$p\left(W_j|K_j, \sigma^{-2}\right) = \prod_{i=1}^{b_j} \int_{-\infty}^{+\infty} p\left(W_{ij}|\mu_{ij}, \sigma^{-2}\right) \times p\left(\mu_{ij}\right) p(\sigma - 2)\mathrm{d}\mu_{ij} \tag{2}$$

Integrating out the fitting complexity is highly reduced by partial residual's conditional distribution calculated from each tree (K_j) represented as μ_{ij}. Dimensionality change during which trees are grown and pruned, is not considered. Posterior variable importance score is furnished on the basis of importance of the variables' rank. In uniform to the proportion of times a variable is selected in all posterior MCMC samples, the variable inclusion proportion is done.

2.3 Data and Sources

This machine learning technique has been used on twenty-five wholesale agricultural commodities prices index (WPI) spanning 2011–2019. The twenty-five agricultural commodities are paddy, wheat, jower, bajra, maize, barley, ragi, gram, arhar, moong, masur, urad, peas, rajma, vanaspati, mustard oil, soybean oil, sunflower oil, groundnut oil, castor oil, rice barn oil, palm oil, rapeseed oil, copra oil, and cottonseed oil. This WPI obtained from the website of Reserve Bank of India (RBI). The entire data analysis has been executed in R version 3.6.2.

3 Results and Discussion

In the design of a BART Machine model, the default hyperparameters settings are m equal to 50.4, q equal to 0.9, k equal to 2, α equal to 0.95, ν equal to 3, β equal to 2, and probabilities of the Prune/Grow/Change phases are 28%/28%/44%. The burn-in Gibbs samples was set to 250 and post-burn-in samples were fixed at 1000. The variation in number of tree m effect on performance can be explained as there is no enhancement of performance by increasing the m value greater than 50. A slight better in-sample performance had been observed in the cross-validated BART Machine model ($L1$ equal to 8.18, $L2$ equal to 0.68, and Pseudo-R^2 equal to 0.978) consecutively a better out-of-sample performance ($L1$ equal to 21.05, $L2$ equal to 4.40, and Pseudo-R^2 equal to 0.858) which were evaluated via predict function and

fits for the first seven rows are: 98.09665, 98.01807, 97.35929, 98.94781, 99.69777, 98.96554, and 98.15722.

3.1 Model Diagnostics

Diagnostics of the error variances in Fig. 1 displaying a QQ-plot (to determine normality) in addition to a residual-by-predicted plot (to determine homoscedasticity) illustrate the natural and homoscedastic distribution of residuals.

The study, by exhibiting convergence statistics, started the test of the consistency of the outcomes for the model. The importance given to the stability of four main segments: the error variance (apex left), the approval rate disposing prior information (apex right), the number of leaves (ground left), and the tree depth (ground right) is displayed in Fig. 2a where the upper row' subplots has results for the burn-in period. Further the four-in-one figure shows a stable evolution of the three parameters and also observation of the approval rate across iterations.

Figure 2a deciphering convergence diagnostics for the BART model with a cross-validation. Upper-left subplot: by MCMC iteration, σ^2. Values till the initial top to bottom gray line encountered in the graph are burn-in from the MCMC chain of the computing core. All the plots segregated by gray lines are the post-burn-in iterations from every 4 computing essence used while the model development. Upper-right subplot: percent acquiring of proposals from Metropolis–Hastings along with the m trees where each value point is an iteration. Burn-in iterations have been portrayed by the point after and before the gray vertical line. Each computing essence is shown in different color. Ground-left subplot: mean total count of leaves through iteration over the m trees (after-burn-in only when calculating essence segregated by vertical gray lines). Ground-right subplot: average tree depth by iteration around the m trees (post-burn-in only where computing cores are divided by top to bottom gray lines).

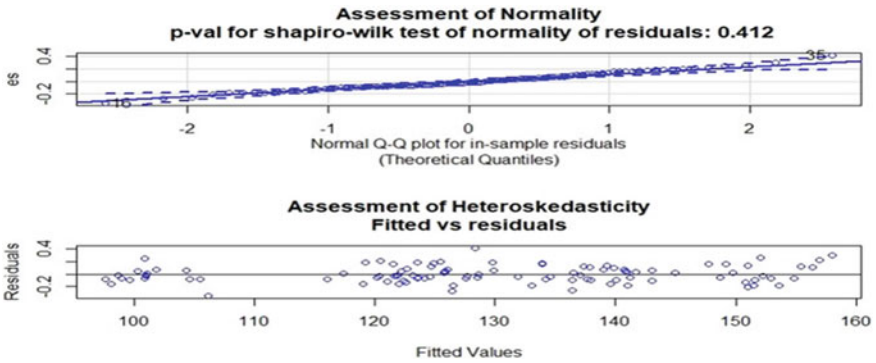


Fig. 1 Diagnostics for error distribution of BART—by using the Shapiro–Wilk test (above) and residual plot to evaluate heteroscedasticity (below)

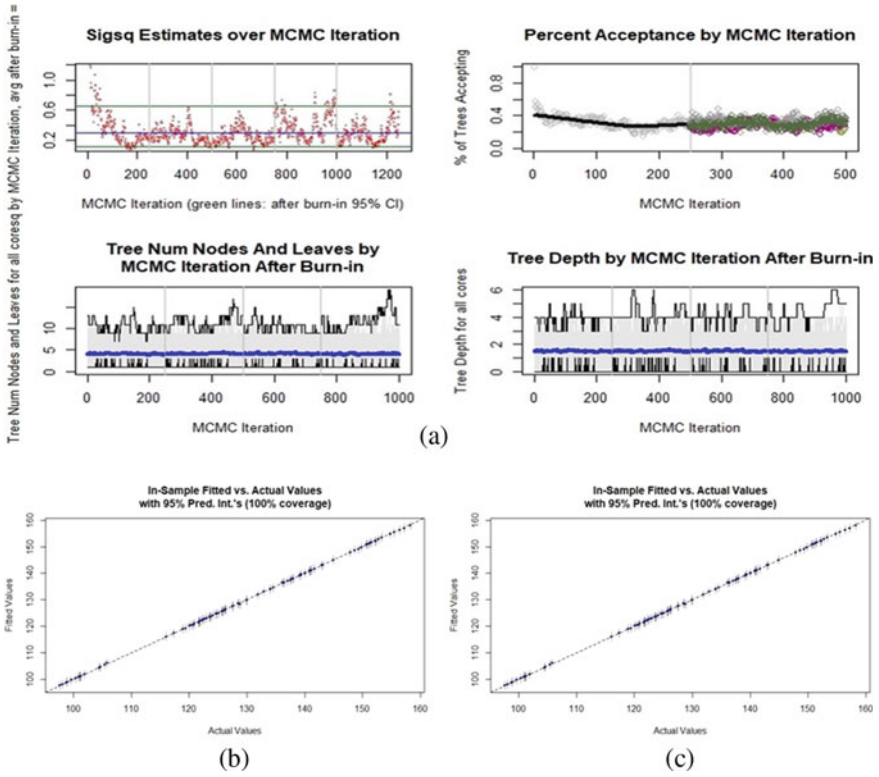


Fig. 2 **a** Convergence diagnostics for the cross-validated BART machine model. **b** and **c** Left-segments illustrate credible intervals; Right-segments illustrate prediction intervals

Figure 2b demonstrates how prediction responded to the initial output in terms of sample with credible intervals of 95%. Figure 2b (left and right) now reveals a close forecast to the initial original plot with predictive intervals of 95%. Figure 2c is showing fitted and actual dataset response values. The divisions are 95% credible intervals (left) or 95% forecasting intervals (right) for estimation. True response during the specified gap is represented by green dots and other indicated by red dots.

3.2 Partial Dependence Plots (PDP)

Selected partial dependence plots are shown in Fig. 3, where we give stress for few main forecasters (as calculated in the context of the comparative significance) defined in Fig. 4a. Figure 3a is the PDP plots along with the variable paddy's 2.5 percentile and the 97.5 percentile. By turning over much of the paddy inventory, the price will be expected to rise by about Rs. 1000. Figure 3b with the 2.5 percentile and the 97.5

percentile and PDP for the variable ragi. The black PDPs plot and credible intervals of 95% drawn in color blue for variables in the dataset of agricultural commodity prices. The plotted points are at the 5 percentile, 10 percentile, 20 percentile …, 90 percentile, and 95 percentile of predictor values.

Lines drawn by linear interpolation between the points estimates PDP. To explain the most suitable threshold out of 1 (Global Max, and 2) Global SE, the "best" procedure is selected by cross-validation on out-of-sample RMSE. Figure 4a indicates the finding by α equal to 0.05 of the three variable selection procedures. The upper plot is signified "Local" procedure. The threshold level evaluated by the permutation distributions must exceed for a specified variable are shown by the green line. The ratio of the measured data with variable inclusion are points plotted. When the variable is included and the selected value is above than the green bar, the solid dots are shown in the diagram; open dots are shown if the variable is not included. "Global SE" and "Global Max" thresholds both are displayed by the lower plot. Cut off for "Global Max" is shown by the red line and the variables which pass this threshold illustrated as solid dots.

The "Global SE" procedure threshold is represented by the blue line. Asterisks are the factors which exceed this cut off but not up to the mark "Global Max". Both have not open dots surpassed limits.

3.3 Relative Importance

The ratio of time is relative importance. The BART machine model selects a predictor to divide all splitting rules from the posterior drawing (Kapelner and Bleich 2013). It shows the amount of contribution of the different predictors to the construction of the BART models. "Inclusion proportions" are understood to be post-burn-in MCMC iterations, corrected by splitting rules in the m trees following the development of a BART model and exposure to variable importance value found as barley, ragi, maize, paddy, rice bran oil, moong, and mustard oil (the top eight predictors). The sum-of-trees model drawn upon the ratio of instances that a variable is chosen out of all splitting rules from the posterior reflects for a specific inclusion proportion. Data on agricultural commodity prices were averaged over 100 model constructions in the cross-validated BART Machine model to obtain reliable estimates for several posterior modes in the case of distribution of sum-of-trees (Bleich et al. 2014). The eight zero inclusion proportion predictors show one attribute identically.

3.4 Interaction Effect Detection

This analysis also examined the model's interactions among different predictors, as seen in Fig. 4b. It indicates the highest importance for the relationship between ragi and barley, followed by barley and paddy. The interactions between the forecasters

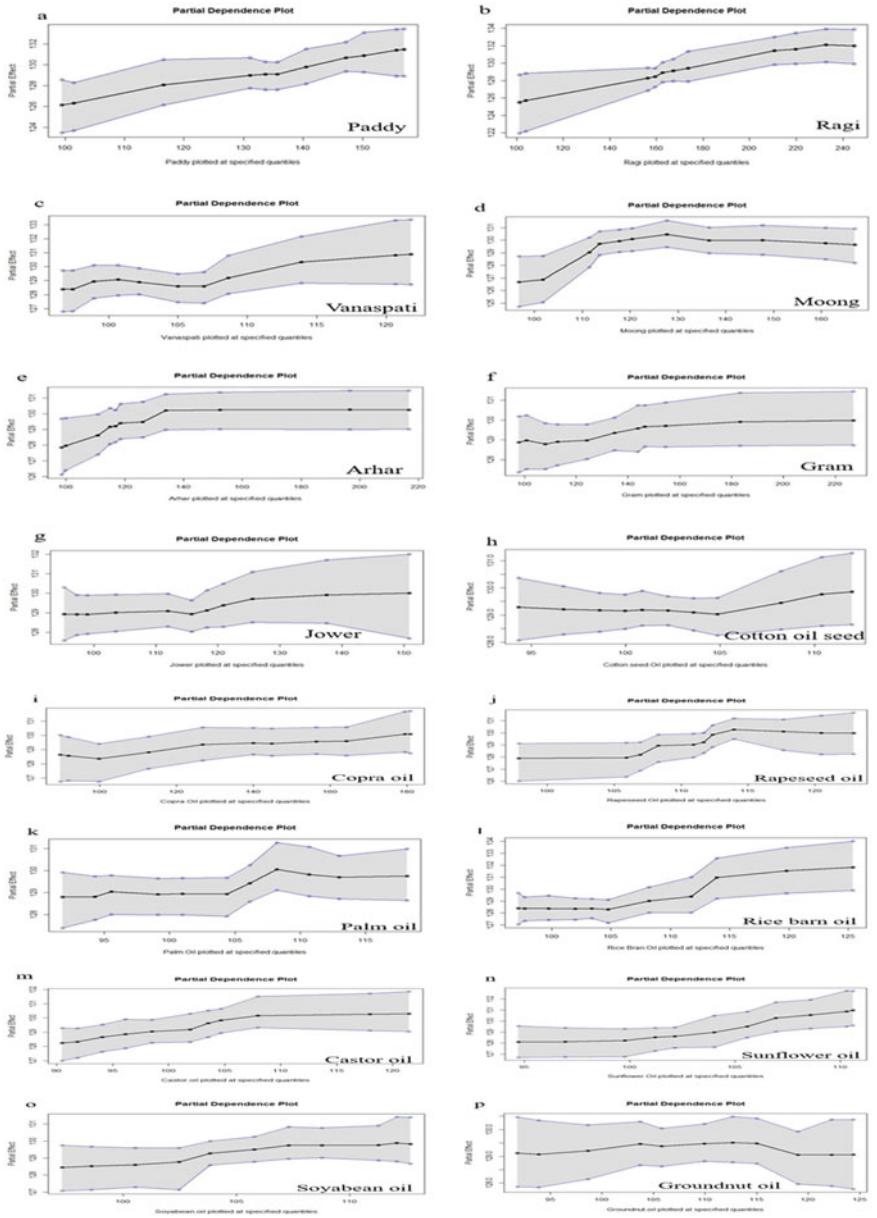


Fig. 3 Partial dependence plots—**a** Paddy, **b** ragi, **c** vanaspati, **d** moong, **e** arhar, **f** gram, **g** jower, **h** cotton seed oil, **i** copra oil, **j** rapeseed oil, **k** palm Oil, **l** rice bean oil, **m** castor oil, **n** sunflower oil, **o** soybean Oil, **p** groundnut oil, **q** mustard oil, **r** cotton seed oil

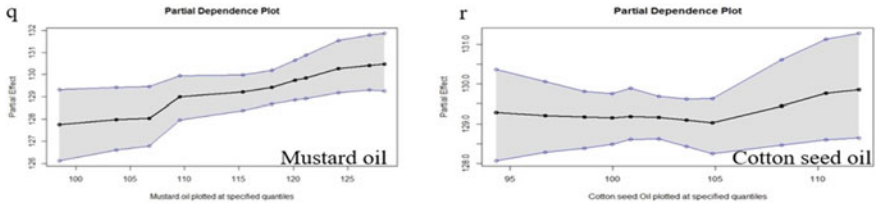


Fig. 3 (continued)

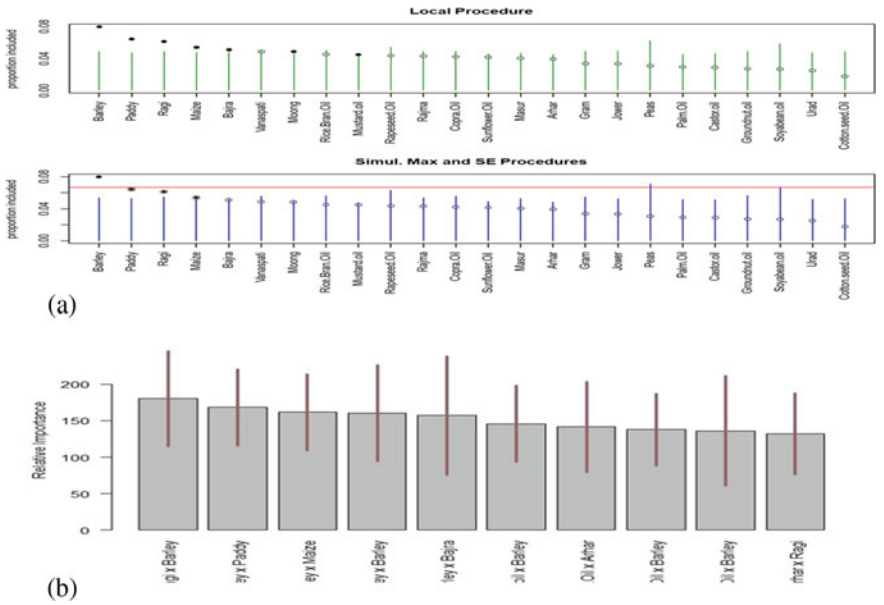


Fig. 4 **a** The visualization of the variable selection procedures. **b** Interaction between variables (the top ten) in the chosen model

are easily apprehensive of the innate interdependencies of each other in the real world. The segments atop the bars are confidence intervals of 95%.

4 Conclusion

This study executes forecasting one of the agricultural commodity prices with many of its agricultural commodities called as predictors using BART drawn from machine learning literary studies. The BART model has been considered as essential owing to the reason that it shows the intrinsic variable selection in order to find solution for the problems. Nonlinear and interactional effects are found to be existing naturally

in the BART model. Considering the importance of other key variables in the prices of agricultural commodities it is a matter of serious concern for budget, policies, and businesses. Our study evidently proves that the BART model is strong support to existing methods for such datasets. In case of problems related to figuring out nonlinearities in the dataset and catching reciprocal effects between the predictors, the model is found to be giving satisfactory results, although nonlinear and interactional effects are natural in the said model. While determining the rejection of forecast efficiency by studying linear efficiency, the model' algorithmic approach to the data is also considered to be good.

References

Ajidarma P, Irianto D (2019) Application of Bayesian additive regression trees to analyze the growth of United States electric automobile industry. IOP Conf Ser Mater Sci Eng 598(1):012035

Andersson MK, Karlsson S (2008) Bayesian forecast combination for VAR models. In: Chib S, Griffiths W, Koop G, Terrell D (eds) Bayesian econometrics. Emerald Group Publishing Limited: Bingley, UK, pp 501–524

Bleich J, Kapelner A, George EI, Jensen ST (2014) Variable selection for BART: an application to gene regulation. Ann Appl Stat, 1750–1781

Davenport F, Funk C (2015) Using time series structural characteristics to analyze grain prices in food insecure countries. Food Secur 7(5):1055–1070

Drachal K (2019) Analysis of agricultural commodities prices with new Bayesian model combination schemes. Sustainability 11(19):5305

Kapelner A, Bleich J (2013) BART machine: machine learning with Bayesian additive regression trees

Liu K, Koike A, Mu Y (2020) Price risks and the lead-lag relationship between the futures and spot prices of soybean, wheat and corn. Asian J Econ Model 8(1):76–88

Mazur B (2018) Cyclical fluctuations of global food prices: a predictive analysis. In: Foundation of the Cracow University of Economics, Proceedings of the 12th Professor Aleksander Zelias international conference on modelling and forecasting of socio-economic phenomena, Cracow, Polska, pp 8–11

Osathanunkul R, Khiewngamdee C, Yamaka W, Sriboonchitta S (2018) The role of oil price in the forecasts of agricultural commodity prices. In: International conference of the Thailand econometrics society. Springer, Cham, pp 422–429

Prüser J (2019) Forecasting with many predictors using Bayesian additive regression trees. J Forecast 38(7):621–631

Thiyagarajan S, Naresh G, Mahalakshmi S (2015) Forecasting volatility in Indian agri commodities market. Global Bus Finance Rev 20:95–104

Deep Learning Model-Based Approach for Agricultural Crop Price Prediction in Indian Market

Eva Mishra, R. Murugesan, and Deba Prasad Dash

Abstract The Indian economy depends largely on the agricultural sector. The price fluctuation in the agricultural market is a major issue for farmers and consumers. In this work, deep learning and machine learning model were used in designing an algorithm for agricultural crop price prediction. The SVR model is largely used for time series data prediction. The SVR model with RBF and polynomial kernel was applied to predict the crop price. The SVR with polynomial kernel achieved better prediction accuracy compared to the SVR RBF model. The RNN model achieved the lowest MSE in predicting crop price. The prediction accuracy for SVR and RNN models was compared by evaluating the models' MAE, MSE, and RMSE values. The RNN model predicted the rice crop price with the lowest MSE, i.e., 0.003. Finally, the deep learning model proved to work better compared to the machine learning model in predicting Indian crop prices.

Keywords Deep learning · Agricultural crop price · Time series prediction · RNN · SVR · RBF kernel · Polynomial kernel

E. Mishra (✉) · R. Murugesan
Department of Humanities and Social Sciences, National Institute of Technology, Tiruchirappalli, Tiruchirappalli, Tamilnadu, India
e-mail: evamishra.nitt@gmail.com

R. Murugesan
e-mail: rmurugan@nitt.edu

D. P. Dash
Department of Electrical & Instrumentation Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India
e-mail: deba.dash@thapar.edu

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_14

1 Introduction

Pre-independence and post-independence, agriculture is the primary source of income in the Indian economy. Two third of the population relies on agriculture directly or indirectly in this economy (Das et al. 2009; Ahmad et al. 2011; Mahto et al. 2021). India has got the second rank for agricultural productivity in the world. The agricultural industry makes up around 18% of the GDP in the Indian economy (Mahto et al. 2021; Sabu and Kumar 2020). Agricultural products are necessary for people to live their daily lives. So the agricultural commodity market is a large proportion of the product market. But there is a major problem in this market, i.e., the quick large price fluctuation in the agricultural commodity prices. It is affecting the people's daily food expenditure, policy makers, traders, brokers, farmers, and investors' life. So to resolve this problem, many models are implemented for forecasting the price of the agricultural commodity. The forecasted agricultural price will help the farmers know the crop price. According to that, they can plan for the next crop. Also, policymakers and investors can manage their decision with supply and demand analysis. The consumer can improve their purchasing power by knowing the forecasting price.

From the previous literature analysis, many statistical models were implemented, i.e., ARIMA, SARIMA, Holt winter, GARCH, etc., to predict agricultural crop prices. But recently, economists have been using machine learning and deep learning techniques like LSTM (Ly et al. 2000), artificial neural network, random forest, decision tree, etc., to predict agricultural crop prices. There is very limited work in this area. To fill this gap, the forecasting accuracy of two different models, i.e., Recurrent Neural Network (RNN) and Support Vector Regression (SVR) models, are compared in this work. In this work two different types of kernels, i.e., polynomial and radial basis function (RBF) were used for SVR model, and prediction accuracy were compared. For the accuracy of the result, root mean square (RMSE), mean square error (MSE), and mean absolute error (MAE) were evaluated for the model. Seven agricultural commodity prices, such as rice, wheat, onion, potato, gram, banana, and groundnuts were selected based on the Indian consumption point of view. The data was monthly basis which was collected from January 2000 till September 2022.

2 Literature Review

Paul et al. (2022) investigated the seventeen brinjal market data in Odisha to forecast the price of the brinjal, which will be helpful for the farmers, and traders. They used four machine learning models: GRNN, SVR, RF, and GBM. They concluded that the GRNN model is better than other models in forecasting the wholesale price of brinjal. SVR and Holt winter exponential smoothing method was used by Kanchymalay et al. (2017) to investigate the correlation between the crude palm oil price and selected vegetable oil prices. Based on their work, the author concluded that SVR performed

better compared to the Holt winter smoothing method. The methods like ARIMA, SVM, and ARIMA-SVM hybrid models were applied to predict garlic prices by Wang et al. (2018). The data was collected from 2010 to 2017 on a monthly wholesale price basis. The ARIMA-SVM model achieved the best result RMSE value of 1.11. The other method, ARIMA, resulted in 1.46 RMSE value, and for SVM, it is 1.69.

Different machine learning algorithms ANN, SVR, ELM, and RF were applied by Zhang et al. (2020), to forecast the prices of piglet, pork, white bran, layer chick, corn, pig grain, mutton, broiler feed, broiler, and wool. A total of 522 samples were used to evaluate the efficiency of a different machine learning algorithm in predicting the agricultural product prices mentioned above. They concluded that the other model were optimal for different products. Weng et al. (2019) used the ARIMA, BP, and RNN models to predict the prices of cucumber, tomato, and eggplant. The models were applied to predict the prices for both short-term and long-term periods. The ARIMA model performed well for periodical data. Overall, the RNN method was better than all other methods. Shengwei et al. (2017), conducted a study to forecast the wholesale fruit price on a monthly basis using the SVR model. The model resulted in MAE equal to 0.1062, NMSE equal to 0.1624, and RMSE equal to 0.1318. As per the observation based on the result, the SVR model was found to be efficient in predicting fruit prices. Xiong et al. (2015) proposed multi output Support Vector Regression and Vector Error Correction model in order to grasp the nonlinear and linear pattern of agricultural crop future prices. Agricultural crop (cotton, corn) prices were collected from the Chinese market and used in this research to validate the model. The cotton data was collected from January 4, 2010 to December 31, 2013 with total sample size of 959. The corn price was collected from January 4, 2006 to December 31, 2013 with total sample size of 1658. The authors claim that VECM-MSVR method performed better compared to the other models like ARIMA-ANN, ARIMA-MSVR, MSVR, VECM, and SSVR. The RNN and SVR model were applied for price prediction apart from agricultural commodity prices. Che and Wang (2010) proposed SVR and ARIMA model in order to predict short-term electricity prices. The hybrid model SVR-ARIMA was proposed by the author for price prediction. The resulted obtained in this research outperforms the existing neural network approaches. The SVR-ARIMA model achieved the RMSE value 0.7515 for the first week and 1.0747 for the second week data. The model accuracy was compared with other models like NN, NN-ARIMA, SVR, ARIMA, and ARIMA-SVR. The SVR-ARIMA was found to be the best model compared to all other approaches. Duan et al. (2017) predicted the aquatic product price using genetic algorithm SVR model. The SVR model along with RBF kernel was used in this research for price prediction. The experimental data was collected from January 2011 to December 2015 from the Chinese market. The GASVR and BP neural network model were compared and GASVR model was found to be superior.

3 Methodology

3.1 Data Description

The Reserve Bank of India (RBI) website, operated by the Government of India, provided the dataset for this study. From January 1, 2000, to September 1, 2022, it has seven independent time series data for wholesale agricultural commodity prices. Rice, wheat, gram, banana, peanut, onion, and potato are the seven agricultural products.

3.2 Price Prediction Approach

In neural networks, it is essential to give normalized inputs to the activation functions in order to get proper training output. In our data the different crop prices have different range of price variations. In order to make all the crop prices in same range the input was normalized before giving it to the RNN and SVR model for training. The data was normalized using Minmax scaler function and given as an input to the RNN model. Mathematically the Minmax scaler function can be written as (Han et al. 2012) (Table 1):

$$D = (y - y_{\min}) \big/ (y_{\max} - y_{\min}) \tag{1}$$

where

D Normalized data
y Raw data
$y_{\min}$ Minimum value of raw data
$y_{\max}$ Maximum value of raw data.

The data was normalized using Standard scaling function before given as an input to the SVR model. Mathematically the Standard scaling function can be written as (Han et al. 2012):

Table 1 Steps followed in price prediction

Steps of price prediction
• Data collection from the website for seven agricultural products
• Normalization of data
• Dividing the normalized data into two sections, i.e., testing and training dataset
• Designing the machine learning (SVR)/deep learning model (RNN)
• Training the model using the training dataset
• Testing the efficiency of the model using the testing dataset
• The extraction of MAPE, RMSE, MAE for model accuracy comparison

$$D = \frac{y - \mu}{\sigma} \tag{2}$$

where

D Normalized data
y Raw data
σ Standard deviation of the y
μ Average of the y.

In our model the entire dataset was divided into training and testing dataset. Total 66% of data was used for the training purpose and 34% of the data was used further testing purpose. The normalized data was directly given as an input to the RNN and SVR models. There was no feature engineering involved before giving as input to deep learning or machine learning model.

3.3 Support Vector Regression (SVR)

This method is based on the principle of the Support Vector Machine. SVR (Paul et al. 2022) is a regressor, i.e., utilized for predicting continuous ordered variables. Regression analysis means the method try to find a relationship between two variables. In case of SVR tries give the best value within the given range/boundary. Equation 3 gives the details about the training samples (Paul et al. 2022):

$$Z = \{(a_k, b_k)\}_{k=1}^{V} \tag{3}$$

where

$a_k \in S^v$ Input vector
$b_k \in S$ Output vector
S Size of sample set.

Then the nonlinear SVR calculating formula is given below (Paul et al. 2022);

$$f(a) = w^t \vartheta(a) + c \tag{4}$$

where

ϑ Nonlinear mapping function which is from actual input space into a higher dimensional space
$w \in S^{o_i}$ Weighted vector
c Bias term
t Transpose.

The coefficients w and c are calculated from sample by minimizing the given regularized risk function (Paul et al. 2022):

$$S(\varphi) = \frac{1}{2}||w||^2 + D\left[\frac{1}{V}\sum_{k=1}^{V} M_\varepsilon(b_k, f(a_k))\right] \tag{5}$$

where

$\frac{1}{2}w^2$	Regularized term
$\frac{1}{V}\sum_{k=1}^{V} M_\varepsilon(b_k, f(a_k))$	Empirical error (It is calculated by Vapnik ε intensive loss function)
D, ε	User-determined hyper-parameter.

Vapnik loss function is shown by Paul et al. (2022):

$$M_\varepsilon(b_k, f(a_k)) = \begin{Bmatrix} |b_k - f(a_k)| - \varepsilon & |b_k - f(a_k)| \geq \varepsilon \\ 0 & |b_k - f(a_k)| < \varepsilon \end{Bmatrix} \tag{6}$$

where

b_k Actual value.

Two different kernel functions can be presented mathematically as follows:
The polynomial kernel is given below (Patle and Chouhan 2013):

$$H(Q_1, Q_2) = \left(j + Q_1^T Q_2\right)^k \tag{7}$$

where

k Degree of kernel
j Constant.

The radial basis function (RBF) is presented as (Patle and Chouhan 2013):

$$Z(Q_1, Q_2) = \exp\left(-\frac{||Q_1 - Q_2||^2}{2\sigma^2}\right) \tag{8}$$

where

σ	Hyper-parameter
$Q_1 - Q_2$	Euclidean.

3.4 Recurrent Neural Network (RNN)

A form of neural network known as an RNN is used to analyze sequential input and has found extensive use in speech recognition, natural language processing, and

other areas (Weng et al. 2019; Williams and Zipser 1989; Cho et al. 2014; Miao et al. 2015). Since the nodes among the hidden layers are linked, the input of the hidden layer includes the hidden layer's most recent output and the result of the input layer. This is the fundamental idea behind RNN: the network will memorize the prior information and use it to calculate the current output (Weng et al. 2019).

The adaptive moment estimation is an algorithm for optimization techniques for gradient descent. It is more effective when there is large amount of data. It is given below (Mustapha et al. 1743; Kim and Choi 2021):

$$r_{t+1} = r_t - \alpha w_t \tag{9}$$

where

$$w_t = \vartheta w_{t-1} + (1 - \vartheta)\left[\frac{\delta L}{\delta r_t}\right] \tag{10}$$

w_t Aggregate of gradient at present time period
w_{t-1} Aggregate of gradient at previous time period
r_t Weighted time t period
r_{t+1} Weighted at time $t + 1$
δL Derivative loss function
δr_t Derivative of weight at present time period
ϑ Moving average parameter.

Rectified linear unit function is known as the *ReLU*. It is a linear function that the output input directly if it is positive, otherwise, output will be zero.

4 Result and Discussion

In this paper RNN deep learning model and SVR model were applied to predict seven different agricultural crop prices. In order to implement the RNN model, the data was first rearranged in a 2D matrix form. The first layer of RNN model was the sequential layer. Two dense layers were added after the sequential layer. The input matrix was of a dimension of 130. The first sequential layer has 32 nodes with an activation function with *relu* activation function. The first dense layer had eight nodes with *relu* activation function. To compile the model, mean square error loss was considered and to optimize the model, the *adam* optimizer in this work. Total of 93 samples were taken for the testing of the model. In the SVR approach, the same test matrix size was considered for evaluating the model. The RBF and polynomial kernel prediction accuracy were compared, and the results are tabulated in Table 2. The data was segregated into the testing and training dataset. The training set was given as an input to the deep learning and machine learning model. The testing data was used to evaluate the prediction accuracy of different model.

Table 2 Results of RNN and SVR

Commodity	RNN			SVR					
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
				RBF			Polynomial		
Rice	0.003	0.05	0.059	0.014	0.096	0.118	0.004	0.054	0.064
Wheat	0.004	0.05	0.063	0.011	0.078	0.107	0.004	0.045	0.054
Gram	0.0049	0.04	0.070	0.091	0.239	0.302	0.0004	0.021	0.021
Onion	0.021	0.09	0.1438	0.151	0.328	0.389	0.082	0.248	0.287
Potato	0.006	0.05	0.077	0.097	0.188	0.311	0.014	0.1035	0.121
Groundnuts	0.006	0.03	0.077	0.020	0.128	0.141	0.010	0.086	0.100
Banana	0.0073	0.04	0.085	0.062	0.171	0.24	0.004	0.054	0.064

Seven different agricultural crop prices were used in this research to calculate the efficiency of the RNN and SVR models in predicting the prices. As observed from the research, the RNN model predicted the price more accurately than the SVR model for polynomial kernel of degree 1 and radial basis function (RBF) kernel. The RNN model predicted the rice price accurately with MSE equal to 0.003. The SVR model with RBF kernel predicted the price with low accuracy compared to the SVR with the polynomial kernel. The SVR with the polynomial kernel predicted the rice price accurately with MSE equal to 0.004. Figure 1a represents the actual and predicted rice price for a time period between January 2015 and January 2022 using the RNN model. From the graph it can be observed that the predicted rice price followed the trend of actual price. But there is no overlap between the actual and predicted price which is reflected in RMSE, i.e., 0.06. The RNN model was implemented to predict the wheat price. The model predicted the wheat price with MSE equal to 0.004. The SVR with polynomial kernel achieved the similar MSE, i.e., 0.004. The SVR with RBF kernel did not perform well as indicated by MSE equal to 0.011. Figure 1b represent the actual and predicted price of wheat using the RNN model. It is seen in the graph that both the actual and predicted price have same trend of price fluctuation with many over lapping. The SVR with polynomial kernel predicted the gram price with lowest MSE equal to 0.0004. The RNN model predicted the gram price with MSE equal to 0.0049. SVR with RBF kernel failed to predict the price with MSE equal to 0.091. Overall for gram price SVR with polynomial kernels achieved the lowest MSE. Figure 1c shows the price fluctuation of gram. The actual and the predicted price followed the same trend in price fluctuation with many overlapping. The overlapping indicates the actual and predicted price were equal and this is also reflected with MSE equal to 0.0049. The RNN model forecasted the onion price with highest accuracy compared to SVR model using both the kernel. The RNN model predicted the onion price with MSE equal to 0.021. The SVR model with polynomial and RBF kernel failed to predict the onion price. Figure 1d draws the graph of actual and predicted price of onion. At many instances the actual and the predicted price

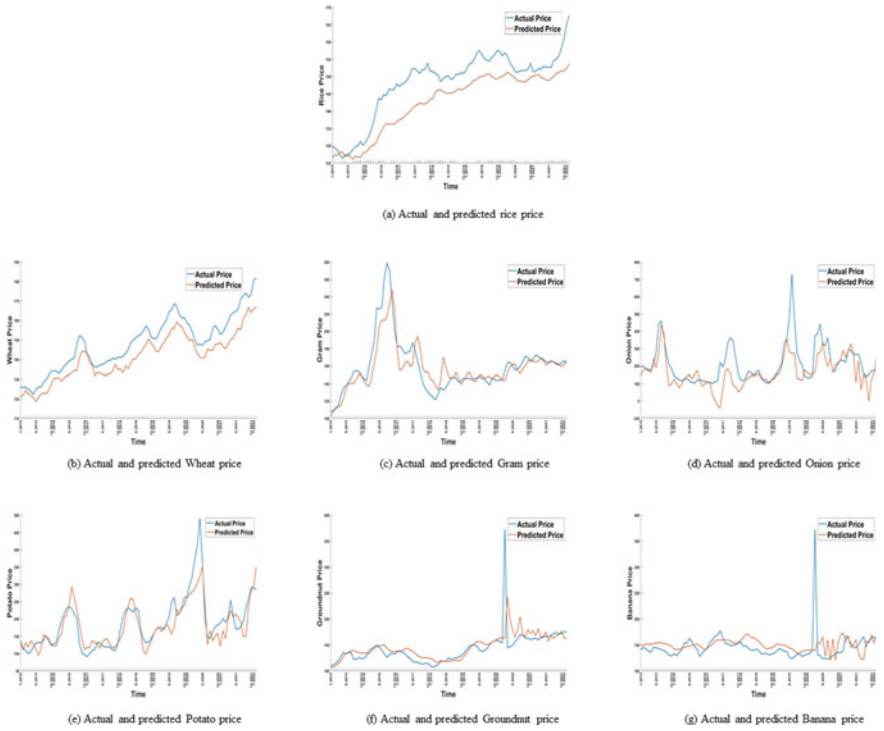


Fig. 1 **a** Actual and predicted rice price, **b** actual and predicted wheat price, **c** actual and predicted gram price, **d** actual and predicted onion price, **e** actual and predicted potato price, **f** actual and predicted groundnut price, **g** actual and predicted banana price

were equal indicated by the overlapping. The RNN model predicted the potato and groundnut price accurately with MSE equal to 0.006.

The SVR model with RBF kernel forecasted the potato and groundnut price with MSE equal to 0.097 and 0.020, respectively. The SVR polynomial model forecasted the potato and groundnut price less accurately compared to the RNN model. Figure 1e, f represents the actual and predicted price plot for potato and groundnut, respectively. It can be observed from the graph that the predicted price followed the price variation trend of actual price. Also a lot of overlapping was seen in the plot indicating the exact prediction of the actual price. The banana price was predicted efficiently by both RNN model and SVR polynomial model. For the RNN model MSE is equal to 0.007 and for the SVR polynomial, it is 0.004. Figure 1g shows overlapping between actual and predicted banana price. It is also observed that the RNN model failed to predict sudden change in the price which is reflected as a spike in the graph.

Figure 2 represents the 3D dot plot between RMSE, MSE and MAE for RNN, SVR, RBF, and SVR polynomial model. As it is observed from the graph the RNN model has the lowest RMSE, MAE, and MSE for the five agricultural product rice,

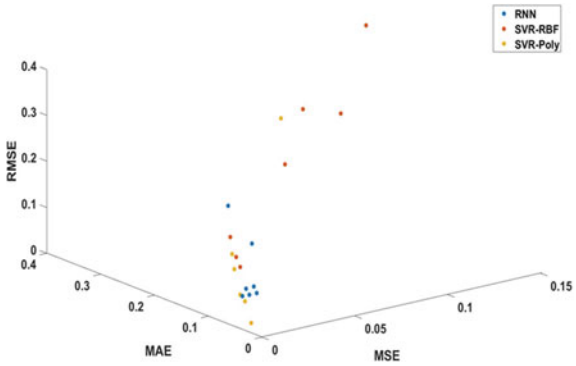


Fig. 2 3D plot of RMSE, MSE, MAE for RNN, SVR, and RBF

gram, onion, potato, and groundnut. For other two agricultural product that is rice and banana the MSE is less than 0.05 indicating the significant prediction. So the RNN model is recommended for prediction of these agricultural commodity prices.

The training and testing accuracy of the RNN and SVR models is given in Table 3. The RNN model works efficiently both for training and testing dataset as observed in Table 3. The MSE value for rice is equal to 0.0001 for training dataset and 0.003 for the testing dataset. The wheat price data after being applied to the RNN model was predicted efficiently with MSE value 0.001 for the training dataset and 0.004 for the testing dataset. The lowest MSE value was observed for onion price for the training dataset. The results indicate that the model was efficient in price prediction both for training and testing dataset. The SVR model achieved MSE value equal to 0.003 for the training dataset and 0.004 for the testing dataset using the rice crop price. The lowest MSE value was achieved for the gram crop price testing dataset, i.e., 0.0004. From the table it can be concluded that the SVR polynomial model achieved good MSE value for rice, wheat, and gram crop prices. But The SVR model fails to predict other crop prices efficiently.

The major contribution of this paper are as follows:

- Design the agricultural price prediction model
- Efficiency of machine learning model (SVR) and deep learning model (RNN) are compared
- The design models is helpful for the farmers, policy makers, investors, agro-based industries, etc.
- The model was applied on the Indian dataset. A very few literature have been found in agricultural price prediction involving Indian crop price. This work will motivate the future researchers to design more efficient models for Indian agricultural crop prices.

Table 3 Comparison of training and testing result

Commodity	RNN training			RNN testing			SVR training (polynomial)			SVR testing (polynomial)		
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
Rice	0.0001	0.009	0.01	0.003	0.05	0.059	0.003	0.043	0.05	0.004	0.054	0.064
Wheat	0.001	0.007	0.01	0.004	0.05	0.063	0.004	0.003	0.06	0.004	0.045	0.054
Gram	0.00025	0.01	0.015	0.0049	0.04	0.070	0.007	0.005	0.08	0.0004	0.021	0.021
Onion	0.000031	0.004	0.0055	0.021	0.09	0.1438	0.7	0.8	0.8	0.082	0.248	0.287
Potato	0.0001	0.009	0.01	0.006	0.05	0.077	0.14	0.5	0.37	0.014	0.1035	0.121
Groundnuts	0.0001	0.007	0.01	0.006	0.03	0.077	0.30	0.05	0.55	0.010	0.086	0.100
Banana	0.000093	0.007	0.0096	0.0073	0.04	0.085	0.012	0.06	0.11	0.004	0.054	0.064

5 Conclusion

The price prediction plays a vital role for farmers in planning the next agricultural crop. This also plays an important role in designing policies by policy makers and government officials. In this research seven agricultural commodity prices were taken from the Indian agricultural market for the period January 2000 till September 2022. Two different models were evaluated for accuracy in price prediction. The RNN model was found to be the best compared to the SVR model for both RBF and polynomial kernel. The RNN model achieved the lowest MSE, i.e., 0.003 for the agricultural product rice. The RNN model achieved MSE value 0.004 for wheat, 0.0049 for gram, 0.021 for onion, 0.006 for potato and groundnut, and finally 0.0073 for banana price. The lowest MAE was achieved for the groundnut, i.e., 0.03 and the lowest RMSE was achieved for rice, i.e., 0.059. The SVR model also predicted the agricultural crop price with some accuracy but less than RNN. The SVR model with RBF kernel achieved the lowest MSE, MAE, and RMSE for wheat price, i.e., 0.011, 0.078, and 0.107, respectively. The SVR with polynomial kernel achieved the lowest MSE, MAE, and RMSE for gram price, i.e., 0.0004, 0.021, 0.021, respectively. So in this work the RNN model is proposed for agricultural crop price prediction in Indian context. In future different deep learning models will be evaluated for forecasting the different product price in worldwide context. The transform learning model can also be explore in agricultural price prediction. The model can be implemented for practice in real life and the efficiency of the proposed model can be discuss in terms of benefit achieved by the farmers, investor, and policy makers.

Abbreviations

SVR	Support Vector Regressor
RBF	Radial Basis Function
RNN	Recurrent Neural Network
MAE	Mean Absolute Error
MSE	Mean Square Error
NMSE	Normalized Mean Square Error
RMSE	Root Mean Square Error
GDP	Gross Domestic Product
ARIMA	Autoregressive Integrated Moving Average
SARIMA	Seasonal Autoregressive Integrated Moving Average
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
LSTM	Long Short-Term Memory Network
GRNN	Generalized Regression Neural Network
RF	Random Forest
GBM	Gradient Boosting Machine
SVM	Support Vector Machine

ANN	Artificial Neural Network
ELM	Extreme Learning Machine
BP	Back Propagation
VECM	Vector Error Correction Model
MSVR	Multi-output Support Vector Regressor
SSVR	Single-output Support Vector Regression
GASVR	Genetic Algorithm Support Vector Regression
RBI	Reserve Bank of India

References

Ahmad J, Alam D, Haseen MS (2011) Impact of climate change on agriculture and food security in India. Int J Agric Environ Biotechnol 4(2):129–137

Che J, Wang J (2010) Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling. Energy Convers Manage 51(10):1911–1917

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation

Das A, Senapati M, John J (2009) Impact of agricultural credit on agriculture production: an empirical analysis in India. Reserve Bank of India Occasional Papers 30(2):75–107

Duan Q, Zhang L, Wei F, Xiao X, Wang L (2017) Forecasting model and validation for aquatic product price based on time series GA-SVR. Trans Chin Soc Agric Eng 33(1):308–314

Han J, Kamber M, Pei J (2012) Data preprocessing. Morgan Kaufmann Series in data management system. In: Data mining, 3rd edn, pp 83–124

Kanchymalay K, Salim N, Sukprasert A, Krishnan R, Hashim URA (2017) Multivariate time series forecasting of crude palm oil price using machine learning techniques. IOP Conf Ser Mater Sci Eng 226(1):012117

Kim KS, Choi YS (2021) HyAdamC: A new adam-based hybrid optimization algorithm for convolution neural networks. Sensors 21(12):4054

Ly R, Traore F, Dia K (2021) Forecasting commodity prices using long-short-term memory neural networks, vol. 2000. International Food Policy Research Institute

Mahto AK, Alam MA, Biswas R, Ahmed J, Alam SI (2021) Short-term forecasting of agriculture commodities in context of Indian market for sustainable agriculture by using the artificial neural network. J Food Qual, 1–13

Miao Y, Gowayyed M, Metze F (2015) EESEN: end-to-end speech recognition using deep RNN models and WFST-based decoding. In: 2015 IEEE workshop on automatic speech recognition and understanding (ASRU), pp 167–174

Mustapha A, Mohamed L, Ali K (2021) Comparative study of optimization techniques in deep learning: application in the ophthalmology field. J Phys Conf Ser 1743(1):012002

Patle A, Chouhan DS (2013) SVM kernel functions for classification. In: 2013 international conference on advances in technology and engineering (ICATE). IEEE, pp 1–9

Paul RK, Yeasin M, Kumar P, Kumar P, Balasubramanian M, Roy HS, Gupta A (2022) Machine learning techniques for forecasting agricultural prices: a case of brinjal in Odisha, India. Plos one 17(7)

Sabu KM, Kumar TM (2020) Predictive analytics in agriculture: forecasting prices of arecanuts in Kerala. Procedia Comput Sci 171:699–708

Shengwei W, Yanni L, Jiayu Z, Jiajia L (2017) Agricultural price fluctuation model based on SVR. In: 2017 9th international conference on modelling, identification and control (ICMIC). IEEE, pp 545–550

Wang B, Liu P, Chao Z, Junmei W, Chen W, Cao N, Wen F (2018) Research on hybrid model of garlic short-term price forecasting based on big data. CMC: Comput Mater Continua 57(2):283–296

Weng Y, Wang X, Hua J, Wang H, Kang M, Wang FY (2019) Forecasting horticultural products price using ARIMA model and neural network based on a large-scale data set collected by web crawler. IEEE Trans Comput Soc Syst 6(3):547–553

Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1(2):270–280

Xiong T, Li C, Bao Y, Hu Z, Zhang L (2015) A combination method for interval forecasting of agricultural commodity futures prices. Knowl-Based Syst 77:92–102

Zhang D, Chen S, Liwen L, Xia Q (2020) Forecasting agricultural commodity prices using model selection framework with time series features and forecast horizons. IEEE Access 8:28197–28209

Enhanced Intracranial Tumor Strain Prediction and Detection Using Transfer and Multilevel Ensemble Learning

Premanand Ghadekar, Ajinkya Mahajan, Aditya Bodhankar, Diksha Prasad, Shivani Mahajan, and Riya Dhakalkar

Abstract Ample amounts of research has been already done when it comes to detection of brain tumors in a computerized way. Considering that the manual approach is highly time consuming and erroneous, digitalizing the whole process can be very helpful and can lead to early detection of brain tumors. And this early detection in turn can minimize the risks since the required treatment can be provided in time. The research previously done is basically focused on machine learning models which are in some cases very weak and can lead to less accurate results. Thus, this approach uses analysis of symptoms well before in advance, transfer learning, ensemble learning, and deep learning methods to tackle the problem. The optimized Densenet model achieved a great accuracy of 93% on training set with 15% loss. It achieved validation accuracy of 90% with loss of 16%. The Resunet model was clubbed with this in the next level of ensemble learning, where it achieved the accuracy of 96% with loss of 10% on training set and 91% accuracy with 14% loss on validation set. The training was done on 60 epochs. Analysis of factors affecting intracranial tumors includes use of Random Forest algorithms that gave 93% accuracy which is higher than other models.

P. Ghadekar · A. Mahajan (✉) · A. Bodhankar · D. Prasad · S. Mahajan · R. Dhakalkar
Department of Information Technology, VIT, Pune, India
e-mail: narendra.ajinkya@vit.edu

P. Ghadekar
e-mail: premanand.ghadekar@vit.edu

A. Bodhankar
e-mail: aditya.bodhankar@vit.edu

D. Prasad
e-mail: diksha.prasad@vit.edu

S. Mahajan
e-mail: vijay.shivani@vit.edu

R. Dhakalkar
e-mail: riya.dhakalkar@vit.edu

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_15

Keywords Machine learning · Ensemble learning · Transfer learning · Deep learning · Multilevel ensemble learning

1 Introduction

Brain tumor or brain cancer is extra cell growth inside the brain which is not normal. A tumor in the brain can be caused due to several reasons like when normal cell changes (mutates) or even when cancer from some other body part reaches the brain. Because of the complex pattern of the lesion's locations, despite extensive research into the subject, there are still some limitations in this field. Due to the small region's appearance as a healthy region, detecting the small amount of lesions in that region is very difficult. The task of drawing out and selecting significant attributes is definitely a difficult task because doing so directly reduces accuracy of classification. Although these models are computationally demanding, CNN assist in extracting informative features. For the analysis of intracranial tumors, a simple model is still required. Brain tumors can be treated depending upon the stage. There are four major stages of brain tumor:

Grade 1 brain tumor is when some cells in the brain are damaged and the growth of tumor is very slow. This one can be targeted through basic surgery. Grade 2 brain tumors have abnormal cells which can be dangerous but they spread slowly. These can be treated through surgery but there's a chance that they might recur. So multiple sessions are usually considered. Grade 3 is where the cells are severely abnormal and they spread quickly in different parts of the brain.

The Ensemble Learning approach uses Densenet and Perceptron in its first level. These models give an accuracy of 93% and 88%, respectively. Thereafter, Resunet model is used in level 2 and it gives an accuracy of 81%. These models are clubbed together and outcomes are decided by majority voting approach. The Random Forest model predicts whether a person has tumor or not by considering several parameters like grade, age, recurrent and metastatic spread, family history, head injury and symptoms, this model worked with 93% accuracy.

The proposed model has followed below given flow:

Level 1: Using multilevel ensemble learning with Densenet, Perceptron, and Resunet models. Majority voting is taken for each model and models are trained on multiple imagery datasets.
Level 2: Thereafter, Random Forest binary classifier algorithm is used to train model on the dataset and predict the tumor based on 14 categories and 181 symptoms.
Level 3: Transfer learning approach is used so that there is no need to train models exhaustively and computation cost is saved. Random Forest classifier is applied on the CSV dataset which is not imagery. This predicts tumor or no tumor using the symptoms from the datasets. Densenet and Resunet are applied on the MRI image of the tumor for attributes like size, grade, severity, recovery time and location.

2 Literature Review

The authors (Amin et al. 2022a) proposed a work in which deep attributes were extracted through the IV3 model, in that the vector of the score was extracted through the softmax and then passed to the QVR for differentiation of glial tumor, meningiomas tumor, tumor does not exist, craniopharyngioma. Post categorization, cancer pictures are given as an input for the segmentation network, in which an area which has the actual infection is segmented in order to know the critical stage of cancer. The outcomes are validated over a couple of reference datasets: Kaggle, 2020 BRATS, as well as locally gathered pictures. The system gives accuracy score more than 89%.

The authors of this paper (Brunese et al. 2020) put forth a system that predicts the various levels of brain cancer through analyzing the MRI pictures of brain. A process for predicting the blocks of an ensemble learning model has been given. The goal of that ensemble learning model is to segregate between various brain cancer levels utilizing radiomic-attributes that are non-invasive. Validation of the attributes' performance is done through hypothesis testing.

In this paper, the authors (Sahoo et al. 2022) proposed a model in which, picture in painting on MRI pictures of the brain is done so as to focus the tumors that are existing in the image. Thereafter, the highlighted pictures are then passed to an ensemble model for train process. Three Convolutional Neural Networks are utilized as initial level classifiers and then the results achieved from them are again provided to one Multilayer Perceptron for further train process and final categorization.

The authors in the paper (Amin et al. 2022b) stated that the issue in brain cancer prediction usually comes through differences in the actual tumor's area, form, and proportion. The focus of this study was coming up with a detailed review on prediction of brain cancer using the MRI pictures in order to aid the medical professionals. The study went through brain tumor anatomy, different datasets, enhancing methods, segmentation, attribute extraction, quantum machine learning, deep learning, classification, transfer learning for analysis.

The research (Xenya and Wang 2021) proposed provided a framework to predict the actual area and form of tumors in the brain using a segmentation model. It even utilizes one multilevel ensemble learner model for classifying tumors in the brain from the MRI pictures dataset. To find out the form attribute of a particular tumor, a statistical measuring method utilizing area attributes has been used. The system for classification includes 3 initial learning models, 2 base level ensemble learners, and one eventual ensemble learner.

3 Proposed Methodology

The Enhanced Intracranial tumor strain prediction and detection is possible using Transfer and multilevel ensemble learning. The model requires similar types of multiple imagery datasets that are collected from Kaggle repository. One more dataset

which is not imagery, a CSV file dataset is used for Random Forest classifier. A stable and more accurate model is proposed that makes use of multilevel ensemble learning, where the stage 1 includes two powerful ML classifiers: (1) Densenet, (2) Perceptron. The model will be used at ensemble learning level 2 where Resunet model shall be included for predictive analysis of factors like size and location of tumor.

3.1 Models Used

DenseNet—It is a type of Convolutional Neural Networks that have fully connected layers which use linear operation where every input neural node is connected to every output neural node using an activation function g, meaning they have n^2 (inputs outputs) which is huge for a huge network. Resunet—Residual Network acts as an encoder-decoder architecture specially developed for semantic segmentation.

The U-net uses 3 * 3 matrix convolution followed by a ReLU activation function (Fig. 1).

3.2 Algorithm

1. Three datasets with 2 imagery ones and one CSV file have been acquired from Kaggle repository.
2. The imagery datasets are visualized using functions like size, tight-layout, aspect ratio, surface-based render, and pixelated render. SK image library is used.
3. After visualization, images are modified and refined in order to simplify accurate prediction. Removing noise, cropping images, conversion to gray scale, resizing images to 200 × 200 size, random patch extraction, and pixel labeling of images has been done. This completes the preprocessing and makes the datasets ready for the models.
4. For ensemble learning level 1, Densenet from keras and Perceptron from sklearn are applied on the datasets. Their weighted average has been taken into consideration.

   Perceptron function:

$$f(x) = 1; \quad \text{if } w.x + b > 0 \tag{1}$$

5. In the ensemble learning level 2, Resunet from keras has been applied on the datasets and again the majority voting is taken with level 1 output. This ensemble model detects the size, grade, location, and severity of the tumor and also predicts possible recovery time.
6. Numerous parameters like, Grade belonging to class I, II, III, or IV, Age, Symptoms, Tumor residual type, Tumor location, Function neurologic status ranging from 10 to 99, Metastatic and Recurrent tumor observed, N-nitrous compounds

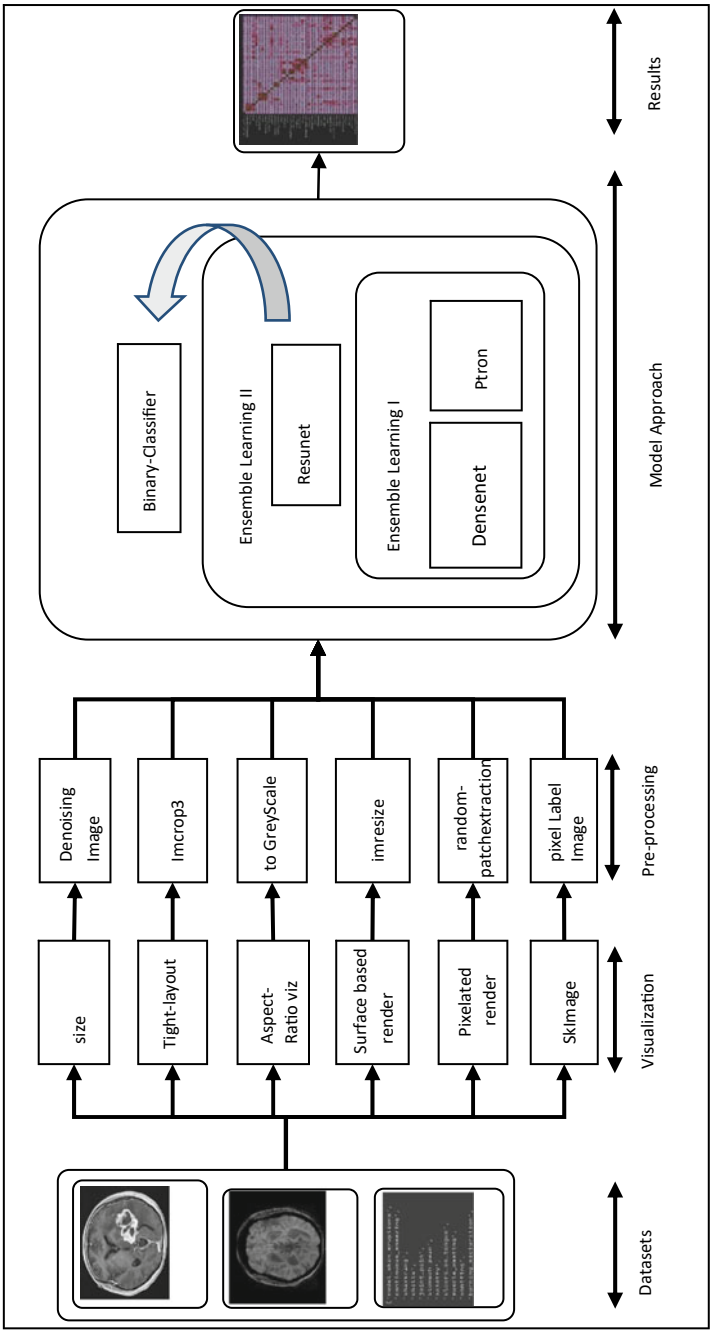


Fig. 1 Proposed flow diagram of intracranial strain tumor prediction and analysis

present, Head injury, Exposure to infections and viruses, Family history, and Exposure to radiation. The symptoms include basic symptoms but covers most of them making more than 100 rows including nodal skin, head ache, skin rash, etc. are considered and a sample output of small dataset is considered as recovery time may vary from person to person depending on the immune system of one.

7. However, the parameters are very prominent when considering brain tumor and the reason for its cause. For example people with greater exposure to radiation like X-rays are more likely to diagnose with brain tumor. The parameters are analyzed and RF classifier is implemented and outputs are generated.
8. Finally, cv2.imread() function is given an image, the ensemble learning models give out size and location of the tumor in the image. The model also gives severity level, tumor level, and recovery time of the tumor through the segmented image. As the recovery time is highly variable, we have considered a dataset consisting of parameters of few patients. The binary classifier gives out the tumor stage. Epochs are generated which contains all the essential information.

3.3 Data Augmentation

Data augmentation is carried out to expand the dataset to avoid overfitting. Although there are three different imagery datasets that are being used here, the data is still not enough and the model might still overfit. To address this issue, the dataset is expanded by applying operations like horizontal and vertical flipping of the images, rotating the images randomly, shearing the images randomly, random width as well as height shifting and zooming the images to random degrees. If the images in dataset are not enough, this might lead to overfitting of the model and result in performance degradation during the testing phase. Therefore, the expansion of dataset was done using several different techniques. Horizontal and vertical flipping of the pictures, random rotation of the pictures, random shear operations applied on the pictures, random shifts in width and height, and random degree of zooming on the pictures was done. All these different operations aided in enough data available for training as well as testing (Fig. 2).

(a) Insertion of Augmented Images into Dataset

(b) Data Shuffling

Fig. 2 Data expansion

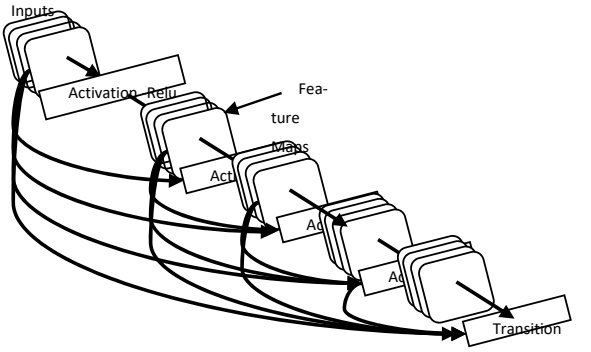


Fig. 3 DenseNet optimization approach

3.4 Densenet Optimization

The Densenet201 model is available in the keras applications framework. In this model, average pooling is used. Freezing of the layers has been done to avoid unwanted modifications. Hidden layer is defined where the activation parameter is set to 'relu'. Classification layer is defined where the activation parameter is set to 'softmax'. Finally the model is compiled with parameters: loss: categorical_ crossentropy, optimizer: adam, metric: accuracy. The model is trained and validated on 30 epochs. The results are visualized graphically and in tabular formats. The results are discussed further in the study. Brain tumor or brain cancer is extra cell growth inside the brain which is not normal. The research does a predictive analysis on brain tumors and predicts the grade at which the brain tumor is. Ensemble learning and transfer learning approaches are followed in order to detect brain tumors (Fig. 3).

3.5 Image Processing

Image processing is carried out on the dataset. Image processing is used to identify the region that is highlighted that is recognized as tumorous. We extract the part by giving lower and upper bounds of color ranges. We fetch the results and a cv2 window (Fig. 4).

3.6 Attribute Extraction

The technique for minimization of dimensions that works with segregating initial batch of the unfiltered data and thereafter, bringing it down to more comprehensible collections, consists drawing out attribute. This makes it much easier to work on whenever required feature extraction is a crucial step in the construction of any

Fig. 4 Sample tumor image extracted from

classification job since it seeks to obtain the essential data that distinguishes each class. Feature vectors are created by separating out the most important aspects from photos. Feature vectors are produced by removing pertinent characteristics from photos. Classifiers match the input unit to the intended output unit using these feature vectors. Prognosis values are counted of the target column that is the tumor column. This column is then visualized using bar plot. Correlation matrix is visualized using the heatmap to see the correlation of all the different attributes with the target attribute. Random Forest classifier is now applied on the dataset with criterion for feature selection as entropy. The accuracy achieved by this model is 93%. It accurately predicts if the person has tumor or not looking at their symptoms.

3.7 Ensemble Learning

In the first level, two models: Densenet and Perceptron respectively are used and later on in the second level, Resunet is used. We have implemented both bagging and boosting techniques where the bagging technique is implemented using majority voting as it proved faster than boosting techniques and more stable (Fig. 5).

This gives a greater accuracy and proper prediction of tumor and helps in predictive analysis. Majority voting and boosting concepts are used to implement the approach. Though ensemble learning gives better outcomes due to multiple models, there are some setbacks here. It increases the computational complexity to a certain level. In the proposed approach the usage of ensemble learning increased the computational complexity by 5/3, i.e., 1.66 times as compared to single models. The current systems make use of linear computation and single model making the system slower and generating similar type and amount of outputs over data, but very unstable over huge datasets (Fig. 6).

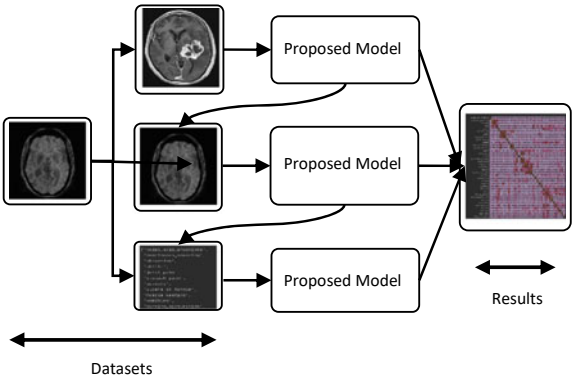


Fig. 5 Boosting ensemble learning approach

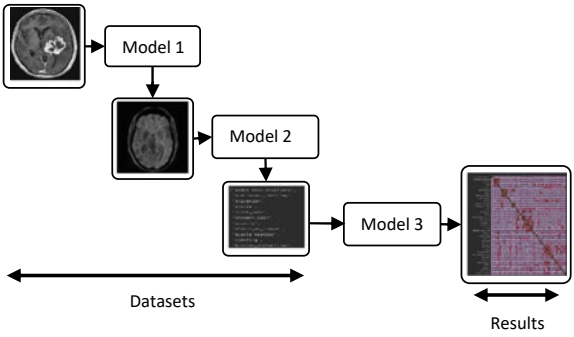


Fig. 6 Linear executing with no ensemble learning

4 Experimentation

The imagery datasets contain images which are distinct. These images are initially visualized to get an understanding about their size, aspect ratio, etc. The images are found to be very different in terms of size and other parameters. The tumor is categorized into one of the four classes: meningioma_tumor, no_tumor, pituitary_tumor, glioma_tumor. Label encoding is done here after, assigning codes 0 to 3 for every tumor category. All the labels are then converted to categorical values. The level one models are applied on data with 60 epochs. The accuracy achieved with Desnenet is 85% on validation dataset. Resunet achieved an accuracy of 81%. Ensemble learning increased the computational complexity by 5/3 times that is 1.66 times. Binary classifier Random Forest is utilized to categorize brain tumor or no tumor. Accuracy and loss of both validation data and train data are visualized graphically and also in a tabular format. The Random Forest classifier applied on the CSV dataset gives 93% accuracy.

5 Results and Discussion

The validation loss and training loss with validation accuracy and training accuracy is visualized below. Table 1 also explains in detail the metrics of performance. The Densenet model's outcomes were: loss: 21%, accuracy: 91%, val_loss: 46%, val_ accuracy: 85%, which properly categorizes the tumor into respective class. This aids in predicting the severity level and recovery time. The segmented image gives size and location of the tumor as well. Thereafter the Resunet and Opencv model highlight the tumor portion in the image.

Resunet loss visualization here shows how training and validation loss has been constantly decreasing which is great for the model. This, in turn increases the accuracy since the loss and accuracy are inversely proportional to each other. Resunet achieved loss of 10% and accuracy of 96% on training dataset and loss of 65% and accuracy of 81% on validation dataset (Fig. 7).

The predictive analysis is implemented using many important tumor parameters that are, Grade belonging to class I, II, III or IV, Age, Symptoms, Tumor residual type, Tumor location, Function neurologic status ranging from 10 to 99, Metastatic and Recurrent tumor observed, N-nitrous compounds present, Head injury, Exposure to infections and viruses, Family history and Exposure to radiation (Fig. 8).

The RF classifies whether the person has tumor or not using 181 symptoms has an accuracy of 93%. The criterion for feature selection is selected as entropy. The 93% accuracy precisely predicts if the person has brain tumor or not. The dataset

Table 1 Resunet evaluation matrices

	Precision	Recall	F1-score	Support
0	0.81	0.65	0.72	156
1	0.88	0.86	0.87	82
2	0.75	0.97	0.83	165
3	0.90	0.82	0.86	165
Accuracy			0.82	574

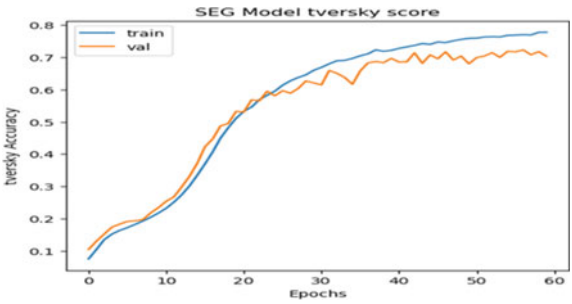


Fig. 7 Resunet training-validation accuracy trade-off

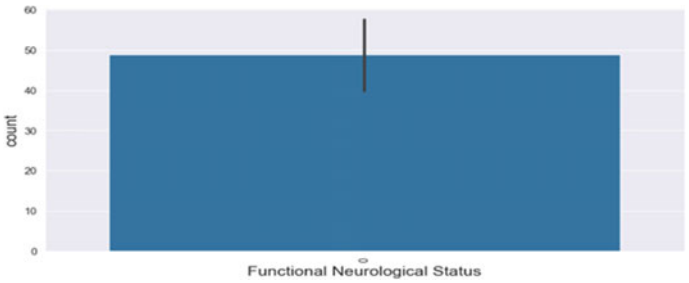


Fig. 8 Function neurological status and count relation with TP values

Table 2 Comparison with existing models

Paper	Accuracy achieved	
A new system on Brain Cancer prediction using Transfer Learning and Ensemble Learning along with QVC (Amin et al. 2022a)	a. Tumor/no tumor classification: 93.33% b. Tumor type classification: 99%	
Brain Cancer prediction using Radiomic features and Ensemble Learning (Brunese et al. 2020)	Ensemble size	Accuracy (%)
	5	97.37
	25	97.57
	125	97.56
Brain Cancer prediction system using Machine Learning methods and analysis by experimentation (Ramkumar et al. 2021)	97.79%	
A professional approach for Brain Cancer prediction using Fuzzy C-means along with super resolution and CNN as well as extreme ML (Özyurt et al. 2020)	98.33%	

is trained on around 4800 rows so it is properly trained and can predict accurately (Table 2).

6 Conclusion

The Enhanced Intracranial Brain Tumor detection and prediction using transfer and multilevel ensemble learning has proved to have great results. This might help the medical professionals around the world to diagnose patients with brain tumor in advance and thus help them provide required treatment. This might even aid in minimizing the risk of fatality due to late diagnosis or incorrect diagnosis.

The system was created using multiple weak machine learning models like Densenet and Perceptron. The key, however, was to combine the results of these

to make a stronger model. This not only reduced the system complexity and bulkiness, but also improved the overall performance. The further combination of these models with Resunet, which is a pre-trained model, but here, it was applied on an acquired dataset, enhanced the accuracy furthermore, leading to even better outcomes in predictive analysis. The Random Forest classifier application on the dataset gave good accuracy and was able to correctly diagnose the image having tumor or not using the symptoms.

References

Amin J, Anjum MA, Sharif M, Jabeen S, Kadry S, Moreno Ger P (2022a) A new model for brain tumor detection using ensemble transfer learning and quantum variational classifier. Comput Intell Neurosci, 3236305. https://doi.org/10.1155/2022/3236305. PMID 35463245; PMCID PMC9023211

Amin J, Sharif M, Haldorai A et al (2022b) Brain tumor detection and classification using machine learning: a comprehensive survey. Complex Intell Syst 8:3161–3183. https://doi.org/10.1007/s40747-021-00563-y

Brunese L, Mercaldo F, Reginelli F, Santone A (2020) An ensemble learning approach for brain cancer detection exploiting radiomic features. Comput Methods Progr Biomed 185:105134. ISSN 0169-2607.https://doi.org/10.1016/j.cmpb.2019.105134

Özyurt F, Sert E, Avcı D (2020) An expert system for brain tumor detection: Fuzzy C-means with super resolution and convolutional neural network with extreme learning machine. Med Hypotheses 134:109433, ISSN 0306–987. https://doi.org/10.1016/j.mehy.2019.109433

Ramkumar G, Prabu RT, Singh NP, Maheswaran U (2021) Experimental analysis of brain tumor detection system using Machine learning approach. Materials Today: Proceedings, ISSN 2214–7853. https://doi.org/10.1016/j.matpr.2021.01.246

Sahoo DK, Das A, Mohanty MN, Mishra S (2022) Brain tumor detection using inpainting and deep ensemble model. J Inf Optim Sci. https://doi.org/10.1080/02522667.2022.2091094

Xenya MC, Wang Z (2021) Brain tumour detection and classification using multi-level ensemble transfer learning in MRI dataset. In: 2021 international conference on artificial intelligence, big data, computing and data communication systems (icABCD), pp 1–7. https://doi.org/10.1109/icABCD51485.2021.9519361

Detection and Classification of Blood Cancer Using Deep Learning Framework

Manish Prajapati, Santos Kumar Baliarsingh, Jhalak Hota, Prabhu Prasad Dev, and Shuvam Das

Abstract Leukocytes, which are generated in the bone marrow, account for around 1% of all blood cells. Blood cancer develops as a result of the uncontrolled multiplication of these white blood cells. The proposed study offers a reliable method to analyze the blood cancer. The two classes of blood cancer are acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Here a deep learning framework is employed to detect blood cancer automatically from input images and completely eliminate the possibility of human error. We have utilized the deep convolutional neural networks in our framework to distinguish leukemic cells from healthy blood cells. Then we have extracted the deep features from blood cancer images using CNN and trained the model followed by classification. We have conducted the experiments on publicly available dataset, namely SN-AM. Comprehensive experiments' results demonstrate that our developed method has obtained the accuracy of 98.71%, recall of 97.48%, precision of 98.37%, and F1-score of 97.87%, achieving better performance compared to the previous state-of-the-art (SOTA) approaches.

Keywords Acute lymphoblastic leukemia · Acute myeloid leukemia · Classification · Digital image processing · Deep learning

M. Prajapati · S. K. Baliarsingh (✉) · J. Hota · P. P. Dev · S. Das
KIIT Deemed to Be University, Bhubaneswar, India
e-mail: santos.baliarsinghfcs@kiit.ac.in

J. Hota
e-mail: jhalak.hotafcs@kiit.ac.in

P. P. Dev
e-mail: prabhu.devfcs@kiit.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_16

1 Introduction

A lymphoid blood cell cancer called acute lymphoblastic leukemia (ALL) promotes the growth of immature lymphocytes (Cho et al. 2022). In addition to harming bone marrow, damaged white blood cells can also compromise the immune system. It also affects the ability of bone marrow to process platelets and red blood cells. Further, these malignant cells have the capacity to travel via the circulation and seriously injure other organs and systems of the body, incorporating the liver, kidney, heart, and brain as well as induce the emergence of additional lethal tumors. The International Agency for Research on Cancer (IARC), which compiles statistics for the whole world, recorded 437,033 leukemia cases and 303,006 deaths as of 2023. The three primary cell types that makeup blood are platelets, red blood cells (RBC), and white blood cells (WBC), each of which performs a specific function. A component of blood is plasma. Oxygen is transported by RBC from the lungs to the body's tissues, and the opposite is also true. The body uses white WBC to ward off illnesses and infections. Platelets help in blood clotting and bleeding control. Blood cancer leukemia is identified by an excess of immature WBC, which often impacts RBC and platelets. The ratio of WBC to all other cells in our body is 1000:1. One white blood cell and one thousand RBCs are indicated by this expression. The two categories of WBC most likely to develop cancer are lymphoid and myeloid cells. Lymphocytic or lymphoblastic leukemia is the name given to leukemia produced by lymphoid cells, whereas myelogenous or myeloid leukemia is the name given to leukemia caused by myeloid cells (Shaheen et al. 2021; Baig et al. 2022). Leukemia is split into two groups: acute and chronic, depending on how fast the cells are multiplying. In acute leukemia, young cells, also known as immature blasts, are typically aberrant blood cells that do not function normally. These cells are growing quickly. If acute leukemia is not treated right immediately, it quickly grows worse. Chronic leukemia produces mature functional cells in addition to young blood cells. The growth of blasts in chronic leukemia is slow. The illness takes longer to worsen.

Our contributions to the current work are summarized as follows:

- The aim of our study is to create an automated cancer detection approach using blood cell images. This approach makes use of a convolution network to determine whether a blood cell is contaminated by cancer. Images of cancerous blood cells sometimes contain ambiguity, are superimposed with other diagnoses, and have similarities to different benign abnormalities. The changes affect how frequently cancer is detected by physicians.
- We initially train the CNN employing the data from the training set to determine the weights for the proper filters in each of the three convolutional sublayers and the weights that result in the fewest mistakes in each of the two fully connected layers. The cross-entropy loss and validation error are then calculated using the data from the validation set to assess the convolution network. We carry out the procedure again until we have trained the convolution network across ten epochs. Following that, the efficiency of the CNN is measured using test dataset.

The remainder of the paper is structured as follows: Literature review is presented in Sect. 2. The proposed work is presented in Sect. 3. Section 4.2 presents the results and discussion. Lastly, we conclude the work and show the future work in the section.

2 Literature Review

To identify leukemia, researchers have developed a number of techniques. CNN, a deep learning-based methodology, has become one of the most widely used techniques for medical image classification in recent years (Thanh et al. 2018). The standard approach for this strategy includes many predefined steps, such as classification, segmentation, clustering, morphological filtering, feature selection, or extraction and evaluation.

Mandal et al. (2019) developed an approach for the detection of cancer cells by extracting the important features from the blood cell images followed by gradient boost and support vector machine (SVM) classifier. They found some vital features, namely the presence of adjacent nuclei and the measure of irregularity in the shape of a nucleus.

Rehman et al. (2018) studied a system for categorizing normal and ALL into several subtypes in stained myeloid pictures. For reliable classification results, the model was trained on images of bone marrow using deep learning methods. Results of various classifiers, such as Naive Bayesian, K-nearest neighbors, and SVM, were then compared with the research findings.

Tran et al. (2018) employed SegNet, a deep learning model using semantic segmentation to segment blood cells in a peripheral blood smear image. It is a cutting-edge technique to separate the red and white blood cells within blood smear images. In semantic segmentation, every pixel of an image was assigned to a class label. The SegNet architecture used a set of encoder and decoder to create feature maps.

Kumar et al. (2020) proposed an optimized dense CNN to eradicate the probability of errors. After being trained on cell images, the model recovers the best attributes from the preprocessed images. The authors claimed that the overall performance of the model outperformed other existing ML models like SVM, decision tree, random forest, and naive Bayes classifiers.

3 Methodology

The aim of our work is to create an automated cancer detection system that detects the input blood cell images to identify whether a blood cell is malignant or not using a convolutional neural network (CNN). Blood cancer cell image identification is often imprecise which makes it difficult for physician to diagnose it consistently. This variation in diagnoses can be reduced if cancer detection can be automated in expert physician level. Such an advancement is not only beneficial in clinical settings but

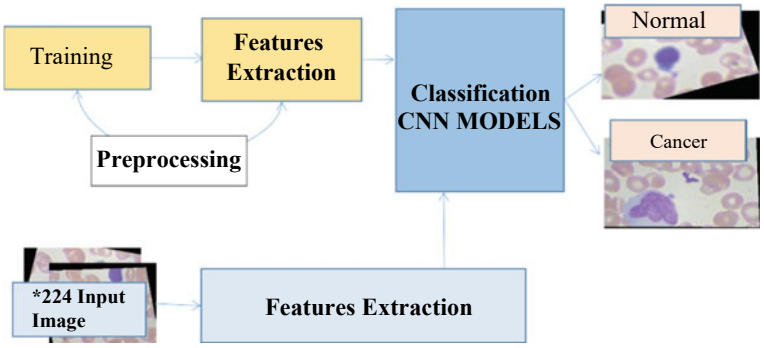


Fig. 1 Deep learning framework for detection of blood cancer

also provides crucial support to populations without access to diagnostic imaging specialists.

Our proposed deep learning framework for the detection of blood cancer is depicted in Fig. 1. We utilized a convolutional neural network (CNN) for the extraction of features and classification of blood cells. Our CNN has 11 layers, including convolutional, pooling, fully connected, and dropout layers. The input layer processes the color input image where each color channel is processed individually. The first six convolutional layers apply a varying number of 3×3 filters to the image, and the max pooling sublayer reduces the image size to half. The output of the flatten layer is a 4800 * 1D array. The fully connected layer with rectified linear unit (ReLU) activation maps 4800 input values to 64 output values, and the dropout layer drops 50% of input values to reduce overfitting. Finally, the fully connected layer with sigmoid activation maps 64 input values to binary class labels. We trained the CNN using the training set and evaluated its performance on the validation and test sets. The training was repeated for ten epochs until we achieved the best cross-entropy loss and validation error.

4 Experiments

4.1 Dataset

The performance of the suggested deep learning framework is measured through experiments on a publicly available dataset, namely SN-AM (Ghaderzadeh et al. 2022). We have split the dataset into two categories, comprising a total of 4961 images for training. The first category consisted of 2483 images depicting healthy patients, while the second category contained 2478 images depicting patients with blood cancer. The performance of the mode was evaluated on a test dataset of 1240 images, with 620 images in each category. Each image has a resolution of 224 * 224.

4.2 Results and Discussion

We have conducted experiments on SN-AM dataset. All experiments are performed on Google Colaboratory using Python programming language. For the evaluation of our proposed model, we have considered four metrics, namely recall, accuracy, f1-score, and precision. Then, we have compared the results obtained from the developed framework with previous state-of-the-art methods as presented in Table 1. The performance of the developed model is significantly improved the accuracy by 10.85%, 4.19%, 3.05%, 2.29%, 1.82% as compared to Ding et al. (2019), Bukhari et al. (2022), Ghaderzadeh et al. (2021), Aftab et al. (2021), and Khamparia et al. (2020), respectively.

A class of people without cancer is represented by 0, while a class of people with cancer is represented by 1. The confusion matrix of the developed method shown in Fig. 2 calculates various parameters depending on true positives, true negatives, false negatives, and false positives.

Table 1 Comparison of experimental results with state-of-the-art approaches

Methods	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Ding et al. (2019)	94.44	91.88	93.15	87.86
Bukhari et al. (2022)	96.16	96.19	96.87	94.52
Ghaderzadeh et al. (2021)	97.12	95.23	96.87	95.66
Aftab et al. (2021)	95.45	97.45	96.10	96.42
Khamparia et al. (2020)	96.34	95.56	97.35	97.89
Proposed method	**98.37**	**97.48**	**97.87**	**98.71**

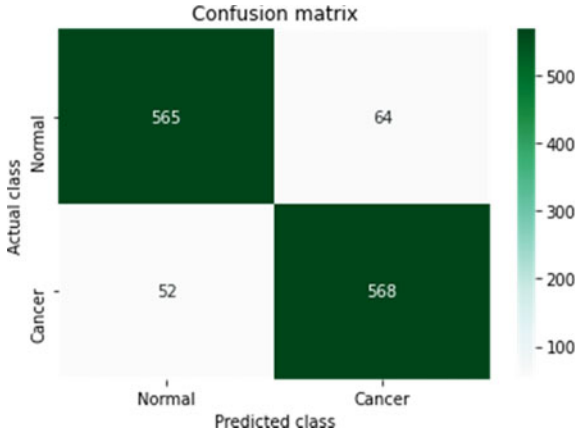


Fig. 2 Confusion matrix obtained by the proposed model

5 Conclusion

In this study, we have proposed a deep learning framework to automatically detect cancer from the blood cell images. We have conducted experiments on publicly available dataset, namely SN-AM. From the experimental results, we have found that our developed model achieved the accuracy of 98.71% that outperforms the best available state-of-the-art methods by a significant increase of 1.82%. In future, we will expand the dataset with noisy images that do not require much preprocessing, in order to address the problem of prediction based on actual blood cell images.

References

Aftab MO, Awan MJ, Khalid S, Javed R, Shabir H (2021) Executing spark BIGDL for leukemia detection from microscopic images using transfer learning. In: 2021 1st international conference on artificial intelligence and data analytics (CAIDA). IEEE, pp 216–220

Baig R, Rehman A, Almuhaimeed A, Alzahrani A, Rauf HT (2022) Detecting malignant leukemia cells using microscopic blood smear images: a deep learning approach. Appl Sci 12(13):6317

Bukhari M, Yasmin S, Sammad S, El-Latif A, Ahmed A (2022) A deep learning framework for leukemia cancer detection in microscopic blood samples using squeeze and excitation learning. Math Probl Eng 2022

Cho S, Tromburg C, Forbes C, Tran A, Allapitan E, Fay-McClymont T, Reynolds K, Schulte F (2022) Social adjustment across the lifespan in survivors of pediatric acute lymphoblastic leukemia (all): a systematic review. J Cancer Survivorship, 1–17

Ding Y, Yang Y, Cui Y (2019) Deep learning for classifying of white blood cancer. In: ISBI 2019 C-NMC challenge: classification in cancer cell imaging: select proceedings. Springer, pp 33–41

Ghaderzadeh M, Asadi F, Hosseini A, Bashash D, Abolghasemi H, Roshanpour A (2021) Machine learning in detection and classification of leukemia using smear blood images: a systematic review. Sci Program 2021:1–14

Ghaderzadeh M, Aria M, Hosseini A, Asadi F, Bashash D, Abolghasemi H (2022) A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images. Int J Intell Syst 37(8):5113–5133

Khamparia A, Gupta D, de Albuquerque VHC, Sangaiah AK, Jhaveri R (2020) Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning. J Supercomput 76:8590–8608

Kumar D, Jain N, Khurana A, Mittal S, Satapathy SC, Senkerik R, Hemanth JD (2020) Automatic detection of white blood cancer from bone marrow microscopic images using convolutional neural networks. IEEE Access 8:142521–142531

Mandal S, Daivajna V, Rajagopalan V (2019) Machine learning based system for automatic detection of leukemia cancer cell. In: 2019 IEEE 16th India council international conference (INDICON). IEEE, pp 1–4

Rehman A, Abbas N, Saba T, ur Rahman SI, . Mehmood Z, Kolivand H (2018) Classification of acute lymphoblastic leukemia using deep learning. Microscopy Res Tech 81(11):1310–1317

Shaheen M, Khan R, Biswal RR, Ullah M, Khan A, Uddin MI, Zareei M, Waheed A (2021) Acute myeloid leukemia (AML) detection using alexnet model. Complexity 2021:1–8

Thanh T, Vununu C, Atoev S, Lee S-H, Kwon K-R (2018) Leukemia blood cell image classification using convolutional neural network. Int J Comput Theory Eng 10(2):54–58

Tran T, Kwon O-H, Kwon K-R, Lee S-H, Kang K-W (2018) Blood cell images segmentation using deep learning semantic segmentation. In: 2018 IEEE international conference on electronics and communication engineering (ICECE). IEEE, pp 13–16

Deep Learning-Based Multi-label Image Classification for Chest X-Rays

Maya Thapa and Ravreet Kaur

Abstract Radiology employs radiation and imaging technologies, such as chest X-ray images, to diagnose and treat disorders. Chest X-rays (CXRs) are useful in emergency diagnosis and treatment; however, patient care is hampered by staff workload and subjective test interpretation. The use of automated systems is being widely explored for the removal of existing issues. Further, deep learning is emerging and giving promising results for quick and efficient image analysis. This paper gives a systematic review of automated multi-label classification of CXR images and a meta-analysis of the CheXpert CXR image dataset. This paper shows the comparison of the results of evaluation metrics like accuracy, F1-score, precision, and recall for multi-label classification by adopting different ways of handling the uncertainties for all 14 observations of the dataset. The performance of the model is assessed using test data that has not yet been observed. With no patients from the training set appearing in the test set, 200 trials from 200 patients were randomly selected from the whole dataset. Our experimental setup gives results that are an improvement upon earlier work; thus, this study will provide guidance as well as a higher accuracy level for the next chest radiography studies.

Keywords Multi-label classification · CheXpert · Deep learning · Accuracy · Precision · Recall · F1-score

1 Introduction

Numerous deep learning (DL) models are being investigated to perform chest X-ray (CXR) analysis, particularly for image classification (Çallı et al. 2021). However, the majority of these methods focus on identifying certain diseases such as pneumonia

M. Thapa (✉) · R. Kaur
University Institute of Engineering and Technology, Panjab University, Chandigarh, India
e-mail: 21310.maya.cse@gmail.com

R. Kaur
e-mail: ravreetkaur@pu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_17

(Szepesi and Szilágyi 2022), tuberculosis (Imam et al. 2020), or lung cancer (Tekade and Rajeswari 2018). While this is going on, using CXRs to properly detect the presence of various chronic thoracic illnesses is still a difficult undertaking that needs a large amount of investigation. For disease diagnosis and medical treatment, reliable image analysis is essential as a lot of chest radiography images are now produced annually (Alam et al. 2022). But, images from the chest X-ray dataset are difficult to categorize for the presence of a particular condition. As a result, radiologists frequently assign an unclear label (value $= -1$) to these images. Thus, necessary steps need to be taken to handle these values. Therefore, in this study, we have adopted one of the architectures of DL to perform multi-label classification on the publicly available datasets and followed different approaches to handle the uncertainties.

2 Related Work

Most research has used, well-designed convolutional neural network (CNN) architectures like VGG (Simonyan and Zisserman 2015), DenseNet (Huang et al. 2017), and ResNet (He et al. 2016) to implement various deep learning models. To predict the target dataset, authors in Huang et al. (1457) thoroughly examined how different transfer learning techniques affected deep CNN models for the multi-label categorization of CXR pictures by combining them with model fine-tuning. Based on the attention mechanism, Sriker et al. (2022) produced a mean AUC of 0.811 where each class in the CheXpert (Irvin et al. 2019) dataset is given a unique attention mask. Chen et al. (2022) discussed the framework that actively explores the semantic similarities across images, which amplifies the performance of X-ray images for multi-label classification and was able to obtain a mean AUC of 0.82 (u-ones)–0.83 (u-zeros). Pillai (2022) created a simple classifier to identify various chest diseases from a chest X-ray image. From an input image, it generates a label vector that specifies which of the 14 illness classes the image belongs to. The models obtain an overall accuracy of 87% and a ROC of about 0.78. A supervised multi-label classification framework was presented by authors in Pham et al. (2021) for anticipating the existence of five common thoracic diseases. Pham et al. (2021) achieved a mean AUC of 0.94. These studies, as previously said, imply that dealing with multi-labeled datasets is worthwhile because doing so may allow for a more nuanced view of the results, and handling uncertainty for a particular dataset may provide us greater insight into all the anticipated outcomes.

3 Experimental Setup

We used a DenseNet-121 architecture based on transfer learning and trained it using the initial weights from a previously trained network using ImageNet data. The CheXpert dataset, which consisted of frontal and lateral images of 65,240 patients, was used in this study. CheXpert has 14 labeled observations that are classified as negative, positive, or uncertain for each image, making it one of the largest publicly available datasets. Out of these, 13 labels indicate typical thoracic diseases, and the label "No finding" indicates the absence of all diseases. This dataset comes in two variations: a high-resolution (about 439 gigabytes) dataset and a down-sampled resolution dataset (about 11 gigabytes). For our study, we used 223,648 low-resolution images. We adopted different ways for handling the uncertainties: first, all of the ambiguous labels were viewed negatively; second, we followed an approach similar to Pillai (2022), where labels were divided into u-zero (labels treated negatively) and u-one (labels treated positively); and lastly, all the uncertain labels were treated positively. Common evaluation metrics like AUROC, accuracy, F1-score, precision, and recall are used for analyzing the performance of a model. The input size of images was scaled to 224 * 224 pixels, and we used the Adam optimizer for backpropagation (Fig. 1).

4 Results

We experimented on the CheXpert dataset described in Sect. 3. In an 80:20 proportion, the train set was split into training and validating sets. The training set contained 178,731 images and the validation set contained 44,683 images. To estimate the model performance, we forecast the multi-classification labels using unseen test data containing 234 images. This paper compares the value of the metric with the work of Pillai (2022) and has achieved better performance as shown in bold terms (Table 1).

4.1 Test Metrics for Labels—AUROC and Accuracy

To evaluate the classification performance, AUROC is used which simultaneously represents how well a model performs for several thresholds (Classification: ROC Curve and AUC| Machine learning| Google Developers 2023). To measure the proportion of correctly classified instances, we used accuracy metrics (Tables 2 and 3).

In terms of overall AUROC and accuracy, the uncertain labels considered as positive depicted better performance as compared to others.

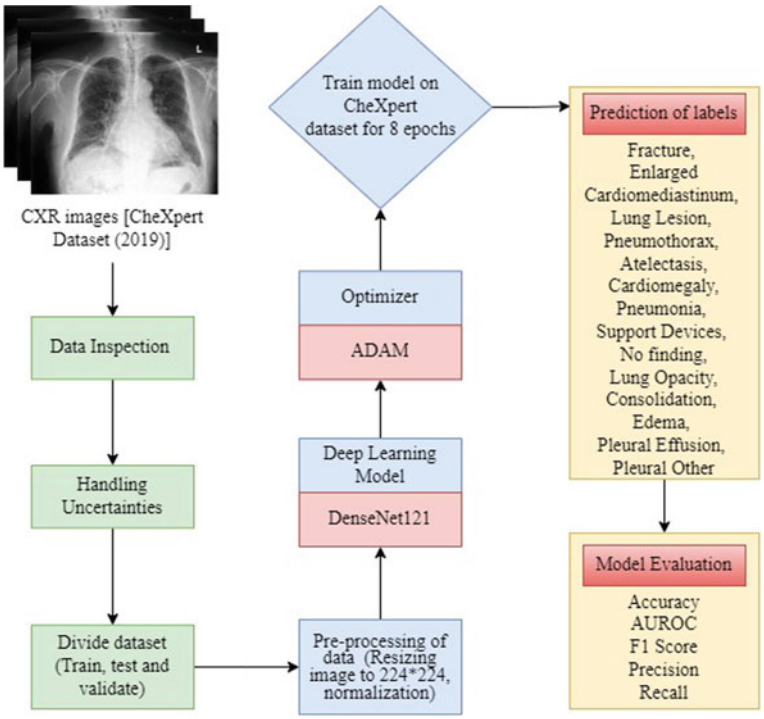


Fig. 1 Architecture of the methodology

Table 1 Testing metrics

Uncertain labels	AUROC	Accuracy	F1-score	Precision	Recall
Negative (u-zeros)	0.804058	0.868567	0.246796	0.342498	0.214448
Negative (except Atelectasis, Edema) (Pillai 2022)	0.817182	0.850503	0.228082	0.352581	0.228562
Positive (u-ones)	**0.842069**	**0.878557**	**0.319126**	**0.442229**	**0.274749**

4.2 Confusion Matrix

We employed a confusion matrix to depict the summary of correct and incorrect predictions because, in rare circumstances, inaccurate accuracy results may be obtained due to an unbalanced data collection (Figs. 2, 3, and 4).

When the anomalous labels were set as positive, the model was able to predict at least one Enlarged cardiomediastinum and consolidation which was unable to predict when uncertain labels were treated differently. And for other labels, such as Cardiomegaly, an exceptional performance was observed.

Table 2 CheXpert labels AUROC results on the test set

Labels	Uncertain labels		
	Negative	Negative except Atelectasis, Edema	Positive
Fracture	NaN	NaN	NaN
Enlarged Cardiomediastinum	0.506862	0.529761	0.546789
Lung lesion	0.163090	0.716738	0.562232
Pneumothorax	0.716814	0.845686	0.811394
Atelectasis	0.831981	0.813149	0.814773
Cardiomegaly	0.826896	0.802179	0.859408
Pneumonia	0.668142	0.688053	0.878319
Support devices	0.925749	0.930164	0.883950
No finding	0.867481	0.850967	0.884264
Lung opacity	0.906452	0.895356	0.896531
Consolidation	0.910599	0.845771	0.896728
Edema	0.889359	0.911581	0.906996
Pleural effusion	0.921530	0.907230	0.918313
Pleural other	0.901288	0.845494	0.944206
Mean	0.772018	0.814009	0.831069

Table 3 CheXpert labels' accuracy results on the test set

Labels	Uncertain labels		
	Negative	Negative except Atelectasis, Edema	Positive
Enlarged Cardiomediastinum	0.529915	0.534188	0.538462
Atelectasis	0.658120	0.666667	0.747863
Cardiomegaly	0.747863	0.713675	0.794872
Lung opacity	0.769231	0.606838	0.799145
Support devices	0.876068	0.824786	0.807692
No finding	0.824786	0.833333	0.846154
Consolidation	0.858974	0.858974	0.858974
Edema	0.880342	0.867521	0.867521
Pleural effusion	0.884615	0.841880	0.867521
Pneumonia	0.965812	0.965812	0.965812
Pneumothorax	0.965812	0.893162	0.965812
Fracture	1.000000	1.000000	0.995726
Lung lesion	0.995726	0.995726	0.995726
Pleural other	0.995726	0.995726	0.995726

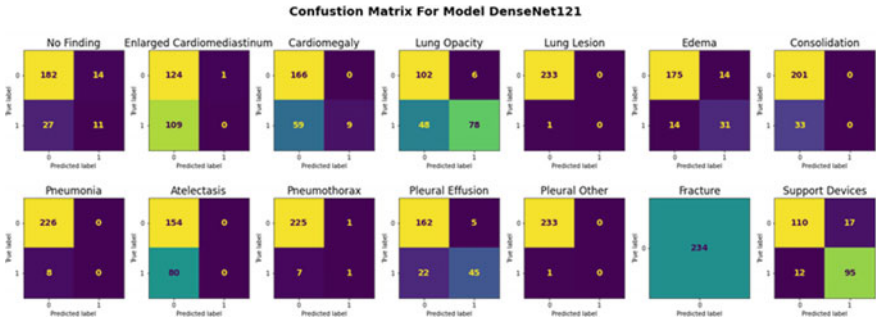


Fig. 2 Confusion matrix: uncertain labels treated as negative

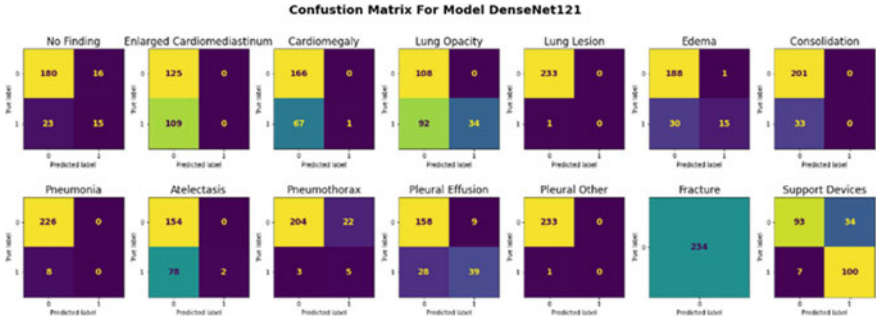


Fig. 3 Confusion matrix: uncertain labels treated as negative except Atelectasis, Edema

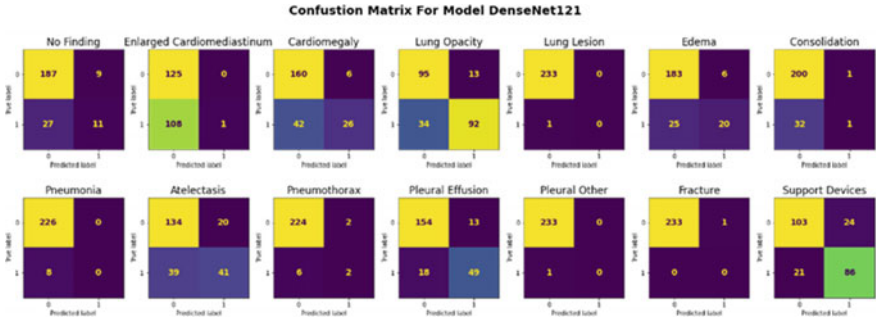


Fig. 4 Confusion matrix: uncertain labels treated as positive

5 Discussion and Conclusion

We explored the result of a pre-trained model on a multi-label classification problem by following different approaches to handle the uncertainties and concluded that the model relies on annotated data, thus having high-quality annotated data for training

these models which is very important. We got 88% accuracy on test data and were able to achieve a ROC of about 0.84. In the future, we will extend our work to perform more exhaustive comparisons as well as explore interpretable deep neural networks using explainable artificial intelligence (XAI) techniques to strengthen our analysis even further. These techniques can help us to provide insights into how the model is making its predictions and can help to validate the results that we have obtained so far. We may discover some potential biases or restrictions that could be present and modify the model to increase its accuracy and dependability by developing a deeper knowledge of how it functions.

References

Alam MU, Baldvinsson JR, Wang Y (2022) Exploring LRP and Grad-CAM visualization to interpret multi-label-multi-class pathology prediction using chest radiography. In: 2022 IEEE 35th international symposium on computer-based medical systems (CBMS). IEEE, Shenzen, China, pp 258–263

Çallı E, Sogancioglu E, van Ginneken B, van Leeuwen KG, Murphy K (2021) Deep learning for chest X-ray analysis: a survey. Med Image Anal 72:102125

Chen B, Zhang Z, Li Y, Lu G, Zhang D (2022) Multi-label chest X-ray image classification via semantic similarity graph embedding. IEEE Trans Circ Syst Video Technol 32:2455–2468

Classification: ROC Curve and AUC| Machine learning| Google Developers. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc. Last accessed 20 Feb 2023

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Las Vegas, NV, USA, pp 770–778

Huang G-H, Fu Q-J, Gu M-Z, Lu N-H, Liu K-Y, Chen T-B (2022) Deep transfer learning for the multilabel classification of chest X-ray images. Diagnostics, 1457

Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Honolulu, pp 2261–2269

Imam OT, Haque M, Shahnaz C, Imran SA, Tariqul Islam M, Islam MT (2020) Detection of tuberculosis from chest X-ray images based on modified inception deep neural network model. In: 2020 IEEE international women in engineering (WIE) conference on electrical and computer engineering, WIECON-ECE. IEEE, Bhubaneswar, India, pp 360–363

Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, Marklund H, Haghgoo B, Ball R, Shpanskaya K et al (2019) Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 590–597, Atlanta, Georgia

Pham HH, Le TT, Tran DQ, Ngo DT, Nguyen HQ (2021) Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels. Neurocomputing 437:186–194

Pillai AS (2022) Multi-label chest X-ray classification via deep learning. J Intell Learn Syst Appl 14:43–56

Simonyan K, Zisserman (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations, ICLR, San Diego, CA, USA

Sriker D, Greenspan H, Goldberger J (2022) Class-based attention mechanism for chest radiograph multi-label categorization. In: 2022 IEEE 19th international symposium on biomedical imaging (ISBI). IEEE, Kolkata, India, pp 1–5

Szepesi P, Szilágyi L (2022) Detection of pneumonia using convolutional neural networks and deep learning. Biocybern Biomed Eng 42:1012–1022

Tekade R, Rajeswari K (2018) Lung cancer detection and classification using deep learning. In: 2018 fourth international conference on computing communication control and automation (ICCUBEA). IEEE, Pune, India, pp 1–5

Fine-Tuning the Deep Learning Models Using Transfer Learning for the Classification of Lung Diseases from Chest Radiographs

H. Mary Shyni and E. Chitra

Abstract Lung diseases are one of the main sources of death across the globe which might prompt lung cancer when left unattended for an extensive stretch of time. X-ray imaging is the fundamental stage in clinical imaging for patients associated with lung oddities. However, because of the intricate morphology of the chest, radiologists have a difficult time visually interpreting the chest radiographs. The purpose of this study is to develop a medical image interpretation model for diagnosing multiple lung diseases by identifying abnormalities in chest X-ray images using transfer learning. The suggested approach has experimented with the four classes of the COVID-19 radiography dataset. The MobileNet V2 architecture performed effectively with the preprocessed dataset.

Keywords Deep learning · Lung diseases · Multi-classification · Transfer learning

1 Introduction

Lung diseases are an aggregate term for various types of lung contaminations and are one of the main sources of death across the globe that are set off by viral, bacterial, and contagious microbes (Wu et al. 2020). At the point when lung infections are not dealt with for quite a while, they might prompt lung cancer. The process of gas exchange is aided by the lungs, which are an essential part of the respiratory system. The lung gets inflamed and inflexible when a lung infection develops gradually, preventing the alveoli from fully expanding. The alveolar walls thicken as the disease worsens, limiting gas exchange (Sundaresan et al. 2011).

H. Mary Shyni · E. Chitra (✉)
SRM Institute of Science and Technology, Kattankulathur, Chennai, India
e-mail: chitrae@srmist.edu.in

H. Mary Shyni
e-mail: mh3006@srmist.edu.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_18

X-ray screening is still the first and most cost-effective procedure for lung diseases (Anis et al. 2020). The chest's intricate anatomical structure still makes it difficult to interpret, despite its benefits (Seah et al. 2021). By learning the intricate characteristics of the radiographs, deep learning methods aid radiologists in making precise decisions (Sriporn et al. 2020).

X-ray images are used in this study to identify lung diseases using the transfer learning method. The COVID-19 radiography database in the Kaggle repository created by M.E.H. Chowdhury et al. was used to experiment with the four pre-trained models: DenseNet 201, Inception V3, MobileNet V2, and VGG 16. The performance of the models tested using various metrics reveals that MobileNet V2 outperformed other models in terms of output.

The structure of the paper is as follows: The most recent studies on deep learning for multi-classification are summarized in Sect. 2. Section 3 describes the dataset modeling and network modeling of the proposed approach. Section 4 provides the experimental parameters and obtained results and Sect. 5 concludes the research with future direction.

2 Related Works

By optimizing the ResNet 50 architecture and training the model with various input sizes at each stage, Muhammad Farooq and Abdul Hafeez proposed a framework. The COVIDx dataset was used to perform multi-classification in order to distinguish Covid-19 from other pneumonia infections. With 41 epochs, the model was accurate to 96.23% (Farooq and Hafeez 2003). In (Apostolopoulos and Mpesiana 2020), two publicly accessible datasets with three classes each were used to evaluate five pre-trained models. A new network was used to classify the features derived from the previously trained models. With an accuracy of 93.48%, VGG 19 outperformed other models.

Min Hong et al. proposed a four-class classification model by center cropping the chest X-ray images with an aspect ratio of one to one, which resulted in an accuracy of 96.1%. The lookahead Nadam optimizer is applied to the base model of EfficientNet B7. The features that are extracted and concatenated with multi-GAP are categorized with the help of the Softmax function (Hong et al. 2021). MobileLungNet V2 was created by F.M. Javed Mehedi Shamrat to classify 14 categories of lung diseases from chest X-ray images. Preprocessing the X-ray images with a Gaussian filter and optimizing the model's hyperparameters over 300 epochs led to a classification accuracy of 96.97% (Shamrat and Mehedi 2023).

Using soft computing techniques, the method suggested in Goyal and Singh (2021) categorizes lung diseases from chest radiographs. Since the model is based on feature normalization, its performance has increased to 95%, while its computational cost has decreased. In (Kim et al. 2022a), the authors fine-tuned EfficientNet V2 before using it to distinguish between normal cases and other lung-infected cases. To obtain

the model's ideal hyperparameters, it was adjusted and regularized, and this resulted in a detection accuracy of 82.20%.

The introduction of the DNet architecture features a direct link between two successive layers, making it possible to form capsules of higher quality that aid in the extraction of complex features. By correctly classifying 14 types of lung diseases from X-rays, the model achieved an AUC of 0.867 (Karim et al. 2022). The model was trained for disease classification with multiple patches of the lung region to overcome data limitations. With the majority voting system, the model achieved an accuracy of 91.9% by utilizing the transfer learning approach of ResNet-18 (Oh et al. 2020).

3 Materials and Methodology

The suggested framework intends to categorize the provided chest X-ray pictures into one of four classes: (i) COVID-19, (ii) lung opacity, (iii) normal, and (iv) pneumonia. Figure 1 presents the workflow of the proposed framework. Dataset collection, data preprocessing, transfer learning, and model training and classification are the principal stages involved in this work.

3.1 Dataset Modeling

For the purpose of identifying lung diseases, four classes of X-ray images were used to evaluate the four well-known pre-trained models trained from the ImageNet dataset. The COVID-19 radiography database compiled by M.E.H. Chowdhury et al. served as the source for the lung disease dataset which can be found in the repository for Kaggle (Chowdhury et al. 2020). There are 3616 images with COVID-19 infection, 6012 images with lung opacity, 10,192 normal X-rays, and 1345 images with pneumonia infection in the database. All the images are saved as PNG files with a dimension of 299 × 299. The model was tested with 300 images from each class

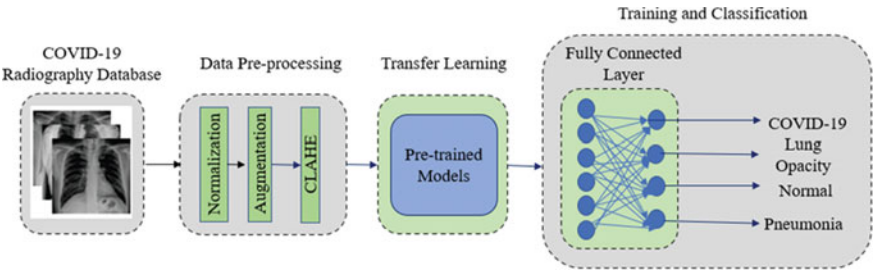


Fig. 1 Workflow of the proposed framework

Table 1 Dataset description after augmentation

Types	Class	Total images	Training set	Validation set	Test set
COVID-19	Class 0	10,848	9948	600	300
Lung Opacity	Class 1	10,220	9320	600	300
Normal	Class 2	10,192	9292	600	300
Pneumonia	Class 3	10,760	9860	600	300

or roughly 3% of the total. The model was validated using 600 images, roughly 6% from each class, which aids in the selection of the optimized model.

All the images in the database were subjected to normalization, augmentation, and CLAHE preprocessing techniques. In order to avoid data variation that could diverge the training process (Huang et al. 2009), the lung images were normalized between 0 and 1. The class disparity nature of the X-ray database would affect the performance of the model (Hasib et al. 2012). To balance the classes, augmentation methods like rotation, flipping, scaling, shearing, and shifting have been applied to the samples. The description of the dataset for training the model following augmentation is presented in Table 1. CLAHE was used to improve the visual quality of the images while also preventing noise amplification (Sanagavarapu et al. 2021).

3.2 Network Modeling

The COVID-19 radiography database was used to train four commonly used deep learning models, namely DenseNet 201, Inception V3, MobileNet V2, and VGG 16. Only the final layers were retrained to learn the complex features from the lung diseases dataset, while the model was learning the basic features from the ImageNet dataset (Kim et al. 2022b). The Softmax function was used to fine-tune the final dense layer, resulting in four outputs that correspond to the four database classes. With an epoch value of 30, the model was trained and validated, with early stopping to prevent poor generalization (Ferro et al. 2023). The four-class database classification was done using the optimal model that was saved from the early stopping phase.

4 Experiments and Results

For the precise detection of four distinct lung diseases, an experimental evaluation of the four deep learning models DenseNet 201, Inception V3, MobileNet V2, and VGG 16 was carried out.

4.1 Experimental Parameters

The actual size of the images in the database 299 × 299 was used to train the Inception V3 network. The remaining three networks were trained with the resized image size of 256 × 256 pixels. With a batch size of 32 and a 0.001 learning rate, the network was to improve the performance of the model.

4.2 Results

Five metrics, which are listed in Table 2, were used to assess the trained models. With an average accuracy score of 90.92%, MobileNet V2 has demonstrated superior performance. It has correctly categorized COVID-19 cases with 97% accuracy, lung opacity class with 91.33% accuracy, normal cases with 81.33% accuracy, and cases of viral pneumonia with 94% accuracy. But in the case of the best-performed model MobileNet V2, more normal cases were misclassified as having lung infections.

The least-performed model Inception V3 misclassified more of the lung opacity cases which led to 77.67% of classification accuracy with that class. VGG 16 classified most of the pneumonia cases correctly with an accuracy of 96.33%. The classification and misclassification of the test data using the four pre-trained models are shown in the confusion matrix in Fig. 2. The ROC curve depicted in Fig. 3 shows

Table 2 Evaluation metrics of the four pre-trained models

Model	Class	Accuracy	Precision	Recall	F1-score	AUC
DenseNet 201	Class 0	0.90	0.91	0.90	0.91	0.98
	Class 1	0.85	0.90	0.85	0.87	0.98
	Class 2	0.89	0.79	0.89	0.83	0.96
	Class 3	0.92	0.99	0.92	0.96	0.99
Inception V3	Class 0	0.86	0.76	0.86	0.81	0.96
	Class 1	0.78	0.93	0.78	0.85	0.97
	Class 2	0.87	0.79	0.87	0.83	0.96
	Class 3	0.92	0.99	0.92	0.95	1.00
MobileNet V2	Class 0	0.97	0.87	0.97	0.92	0.99
	Class 1	0.91	0.88	0.91	0.90	0.98
	Class 2	0.81	0.90	0.81	0.85	0.98
	Class 3	0.94	1.00	0.94	0.97	1.00
VGG 16	Class 0	0.88	0.91	0.88	0.89	0.98
	Class 1	0.89	0.85	0.89	0.87	0.98
	Class 2	0.84	0.83	0.84	0.83	0.96
	Class 3	0.96	0.98	0.96	0.97	1.00

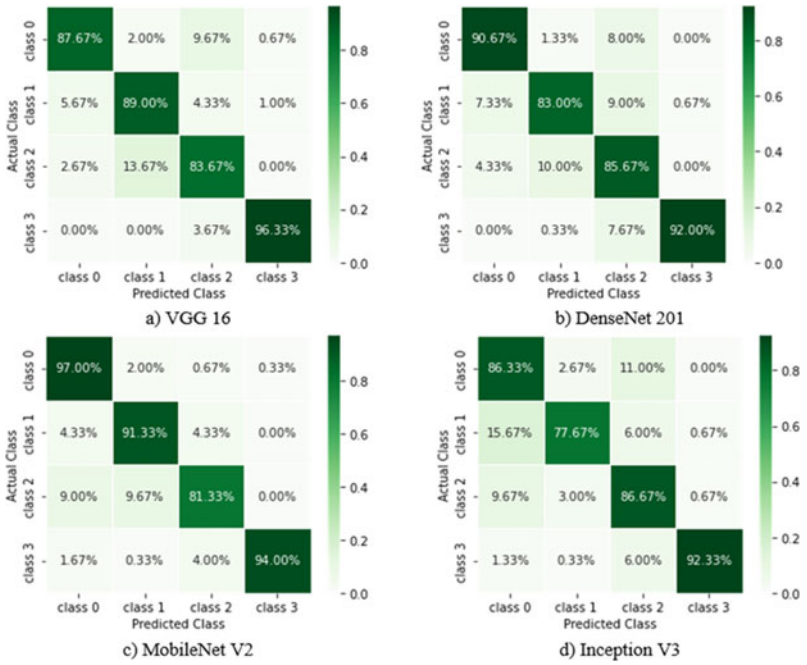


Fig. 2 Confusion matrix for the four pre-trained models

the performance of each model. Each ROC curve is highlighted with four different colors to indicate the four different lung diseases.

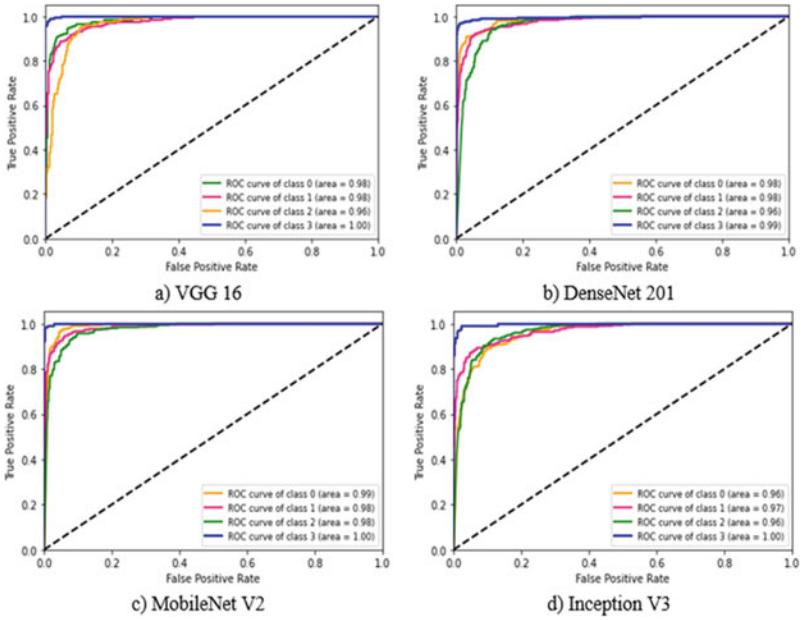


Fig. 3 ROC curve for the four pre-trained models

5 Conclusion

Early diagnosis of lung anomalies aids in stopping the further progression of the infection in the lung area, potentially saving many lives. It takes a lot of time to train a deep neural network from scratch to classify lung diseases. In this study, the deep learning models DenseNet 201, Inception V3, MobileNet V2, and VGG 16 only had their final layers trained, which might save a lot of time. Better outcomes were obtained using the fine-tuned MobileNet V2 model, which scored 90.92% on accuracy. In the future, we intend to develop a customized deep learning model to classify more lung related diseases.

References

Anis S et al (2020) An overview of deep learning approaches in chest radiograph. IEEE Access 8:182347–182354

Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43:635–640

Chowdhury MEH et al (2020) Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 8:132665–132676

Farooq M, Hafeez A (2020) Covid-resnet: a deep learning framework for screening of covid19 from radiographs. arXiv preprint arXiv:2003.14395

Ferro MV et al (2023) Early stopping by correlating online indicators in neural networks. Neural Netw 159:109–124
Goyal S, Singh R (2021) Detection and classification of lung diseases for pneumonia and Covid-19 using machine and deep learning techniques. J Ambient Intell Hum Comput, 1–21
Hasib KM et al (2020) A survey of methods for managing the classification and solution of data imbalance problem. arXiv preprint arXiv:2012.11870
Hong M et al (2021) Multi-class classification of lung diseases using CNN models. Appl Sci 11(19):9289
Huang L et al (2020) Normalization techniques in training DNNs: methodology, analysis and application. arXiv preprint arXiv:2009.12836
Karim F et al (2022) Towards an effective model for lung disease classification: using Dense Capsule Nets for early classification of lung diseases. Appl Soft Comput 124:109077
Kim S et al (2022a) Deep learning in multi-class lung diseases' classification on chest X-ray images. Diagnostics 12(4):915
Kim HE, Cosa-Linan A, Santhanam N et al (2022b) Transfer learning for medical image classification: a literature review. BMC Med Imaging 22:69
Oh Y, Park S, Ye JC (2020) Deep learning COVID-19 features on CXR using limited training data sets. IEEE Trans Med Imaging 39(8):2688–2700
Sanagavarapu S, Sridhar S, Gopal TV (2021) COVID-19 identification in CLAHE enhanced CT scans with class imbalance using ensembled resnets. In: 2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS). IEEE
Seah JCY et al (2021) Effect of a comprehensive deep-learning model on the accuracy of chest x-ray interpretation by radiologists: a retrospective, multireader multicase study. The Lancet Digital Health 3(8):e496–e506
Shamrat FM, Mehedi J et al (2023) High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. Comput Biol Med, 106646
Sriporn K et al (2020) Analyzing lung disease using highly effective deep learning techniques. Healthcare 8(2)
Sundaresan A et al (2011) Model-based optimal PEEP in mechanically ventilated ARDS patients in the Intensive Care Unit. Biomed Eng Online 10:1–18
Wu J-X et al (2020) Multilayer fractional-order machine vision classifier for rapid typical lung diseases screening on digital chest X-ray images. IEEE Access 8:105886–105902

A Systematic Approach for Effective Apgar Score Assessment in 1 and 5 min Using Manifold Machine Learning Algorithms

S. Shaambhavi and G. Bhavani

Abstract The Apgar score was first introduced by Dr. Virginia Apgar in 1952, and its mainly used to find the well-being of newborn infants by using the five important aspects such as heartbeat, reflex irritation, color, muscle tone, and respiration. Apgar was an assessment tool that is performed at 1 and 5 min of a child's birth. There is a potential impact of the score on medical decision making and providing care to newborn infants. Since a low Apgar score results in harmful interventions. So here in this research paper, the analysis and comparison of R^2 Score of various machine learning algorithm models is done. Here, this comparison mainly focuses on improving the R^2 score of the Apgar value. Additionally, the use of the Apgar score is a very important subject in terms of research and discussion among healthcare professionals. The Apgar score has also been evaluated in various populations including preterm infants and newborns with congenital anomalies. This research paper has analyzed the comparison of 13 different machine learning algorithms and the R^2 Score was obtained as 96.5% for the hybrid approach. So, utilizing machine learning to predict the Apgar score will increase R^2 Score, boost neonatal research and help us better understand the physiology and health of newborns.

Keywords Apgar score · Machine learning algorithms · Data analysis · Decision making

S. Shaambhavi · G. Bhavani (✉)
Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai 625015, India
e-mail: gbicse@tce.edu

S. Shaambhavi
e-mail: shaambhavisivappugaz5@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_19

1 Introduction

The Apgar score was first introduced by Dr. Virginia Apgar in 1952 and its mainly used to find the well-being of newborn infants by using the five important aspects such as heartbeat, reflex irritation, color, muscle tone, and respiration. The Apgar score is an assessment tool performed using 1 and 5 min of a child's after birth. It is used to identify the risk of newborn and help in clinical decision making. For the five criteria, a value will be set from 0 to 2 and the overall score is set to 10. So the overall score of 7–10 states that the baby is in good condition and 4–6 states that the baby is moderately being abnormal and the score of 0–3 states that low in terms of infant and later preterm infant. These scores were also implemented to assess neonatal asphyxia without the need for blood gas analyses. The mortality rate was 244 per 100 for infants whose Apgar score at 5 min was 1–3. Also in preterm infants, the mortality rate was 315 per 1000 with an Apgar score of 0–3. Deaths were also more common among the infants who have the Apgar score of 0–3 than with 7–10. The low Apgar score can be caused due to variety of reasons such as preterm birth, maternal medical conditions, fetal monitoring errors, c-section errors, delays, etc. The babies with low Apgar scores should be immediately moved to the neonatal intensive care unit (NICU) so that the baby receives appropriate care. In recent years there is a tremendous growth in using machine learning techniques and all the predictions made using machine learning has been accurate. So using machine learning algorithms in predicting the Apgar score can lead to improve R^2 Score and give very better results.

Recent studies its shown that,

- The child with a low Apgar score at 5 min confers high risk than a child with a low Apgar score at one minute.
- Changes in the Apgar score value at one and five minutes will increase the risk of cerebral palsy, and epilepsy and also neurological morbidity can be influenced by this change.
- Infants failing to achieve the Apgar score of 7 within 5 min from birth should be repeated the process of calculating the Apgar score every 5 min until the score is reached 7.

The machine learning algorithms can be trained on a large number of clinical datasets and using demographic information which will help to identify patterns and relationships as they are not easily recognizable by human experts. So it will lead to accurate predictions of Apgar score and also help in assisting the neonatal well-being. This paper mainly focuses on implementing machine learning models for predicting the Apgar score and exploring the potential benefits. Since each technique has its strengths and weakness an analysis of various machine learning algorithms is being used for prediction. The algorithm which gives us the best R^2 Score and that does not overfit will be chosen for further proceeding. The algorithm will be tested on other evaluation metrics such as MSE, MAE, RMSE, R-Squared, and EVS is evaluated. Apart from the five main criteria some other features are also explored

such as gestational age, birth weight of the child, mode of delivery, and mother's age have been included for prediction. It is also important to note that the use of these Apgar score should be guided by a commitment to promote the well-being of the infant and not used to discriminate against certain groups of newborn based on factors like race, ethnicity, or socioeconomic status. By providing clear and accurate information, healthcare providers can help parents and families to understand the significance of the score. The Apgar score continues to be a very valuable tool for over 70 years in evaluating the physical condition of the newborn infant.

2 Literature Review

In (Park et al. 2022) monitoring the fetal heart rate is the main important aspect of this paper as its a very complex purpose and they have proposed it using machine learning algorithms such as linear and logistic regression. In (Abdallah et al. 2021) due to an unhealthy lifestyle the babies are having low birth weight and low Apgar scores. This paper has used binary logistic regression under multivariate analysis using SPSS and this was significant. In (Ibrahim et al. 2006) in this paper they have analyzed the risk factors for low Apgar scores using artificial neural networks. In (Cnattingius et al. 2020) since a lower Apgar score resulted in neonatal death they performed the statistical analysis and also provide prognostic information about the survival of the infants. In (Chao et al. 2022) they discussed detecting the COVID-19 in infants and then when they used the Apgar score it was a poor indicator. In (Jensen et al. 2012) they have used the 5-min Apgar score as a marker of perinatal asphyxia and it was investigated and it was a statistical analysis. In (Sahak et al. 2009) in this paper the cry of the premature and mature infants with low Apgar score was analyzed using principle component analysis and it provides the foundation for the design of an automatic algorithm to replace the manual Apgar scoring system. In (Rotariu et al. 2014) they identified the dynamic relationship between intrapartum fetal heart rate variability and uterine pressure for normal and hypoxic features and the results are compared to documented fetal outcome through low pH and Apgar values. In (Tarimo et al. 2022) they have proposed a system to correctly classify the neonates who have a low Apgar score from imbalanced learning. In (Khatibi et al. 2022) it involves the early prediction of the neonate 5 min Apgar score before or during birth.

3 Dataset

3.1 About Dataset

The features like Apgar score, gender, birth weight of the child, mother's age, heartrate, respiratory rate, muscle tone, reflex irritability and color, and mode of delivery are being chosen to predict the Apgar score. The dataset which had some features was obtained from KU and the five vital features were included from the nearby clinic.

3.2 Pre-processing

The dataset had a lot of missing values. The data set was filled with mean, mode, and median according to the respective feature. The pre-processing was done using the standard scaler library. The features that were not required were removed from the dataset.

4 System Design

In Fig. 1 the design of the system has been displayed. The system design involves collecting the dataset. Pre-processing the dataset using standard scaler techniques and then training using various machine learning algorithms and then testing their performance and finally validating the results of the algorithm.

5 Machine Learning Models for Apgar Score Prediction

5.1 Random Forest

Random forest is also considered to be a more robust and flexible algorithm that will handle a large number of input variables which can be categorical and continuous and also handles the non-linear relationship between features and target variables. This has been a very popular algorithm that is being used to solve classification and regression problems. This has been an ensemble method that combines multiple decision trees to produce a more accurate and stable prediction. In this algorithm, each tree is trained on a different subset of data and finally, all the predictions of the trees are combined to obtain the final prediction. This process is also called as

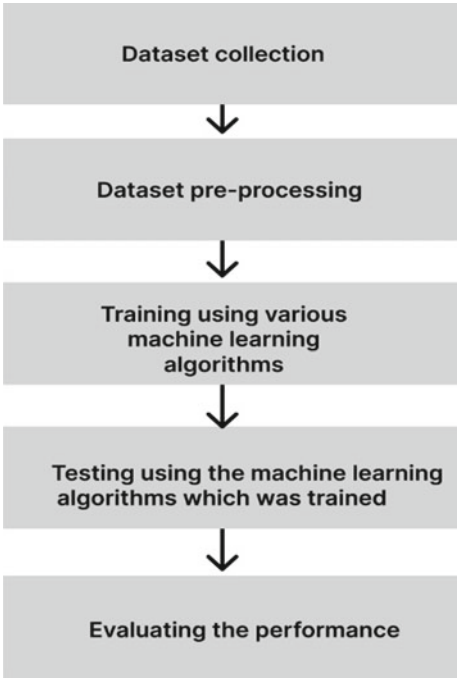


Fig. 1 System design

random subspace method. Random forest is a versatile and powerful algorithm. The random forest model has given a R^2 Score of 81%.

5.2 Gradient Boosting Regressor

This algorithm is an ensemble model which is used for both regression and classification problems. This algorithm mainly focuses on combining the multiple weak models and producing a more powerful and accurate prediction. This algorithm mainly works by training the weak models and also the weights are updated for the respective data point based on the errors made by previous models. The weights that were updated gives the difficult to predict the samples so that subsequent models are being focused and the R^2 Score of the sample is being predicted. And at last, the final prediction is made by combining the prediction of all the trees using a weighted average, where the weight is directly proportional to the accuracy of each tree. The gradient boosting regressor model has given a R^2 Score of 83%.

5.3 Adaptive Boosting Regressor

This is one of the ensemble models that is being widely used. This algorithm aims in building a model and it will also give equal weights to all the data points that are present and then it will start assigning higher weights to points that have been classified wrong. So now all the points which have higher weight will be given more importance in the next training model and this will get repeated until the error is becoming low. This is also simple to implement and its also being computationally efficient as it takes training only on a small number of weak models. The Adaptive Boosting Regressor model has given a R^2 Score of 80%.

5.4 BootStrap Aggregating

This is one of the ensemble learning models which is used for regression and classification tasks. This algorithm will combine multiple models and provides a more accurate prediction. The algorithm works in a way that it trains the multiple models on different subsets of data which are randomly sampled data with replacement and it creates a different version of the training data for each model which allows the model to be trained on slightly different information and the final model prediction is done by combining the predictions of all models using a majority vote or also by taking the weighted average depending on the task. The BootStrap Aggregating model has given a R^2 Score of 72%.

5.5 K-Nearest Neighbors

It's a simple and powerful algorithm that is being used for regression and classification. This algorithm works on the idea that similar examples are close to each other in feature space which is assigning a label to a new data point based on the label of its nearest neighbors in the feature spaces. This algorithm never makes assumptions about the underlying data distribution and this algorithm is very well suited for a wide range of problems. In this algorithm, the value of K is a user-defined parameter and the value of K determines the size of the neighborhood which makes the algorithm be used for predictions. If K is being a small value the algorithm will be mostly influenced by the outliers and may also result in over fitting of the model. But if the value is large this algorithm will be less likely influenced by outliers but it may result in under fitting of the model. This algorithm is also very easy to implement and also requires less amount of training data. The K-Nearest Neighbors model has given a R^2 Score of 66%.

5.6 Extreme Gradient Boosting

The Extreme Gradient Boosting algorithm is used for both classification and regression type of problems and also this algorithm is an ensemble learning model which is combining the prediction of multiple weak models in order to produce a stronger prediction model. This algorithm also efficiently handles the missing values without requiring significant pre-processing. It's also possible to train a large number of datasets in a reasonable amount of time. It also includes several forms of regularization which will help prevent the over fitting problem. The Extreme Gradient Boosting model has given a R^2 Score of 80%.

5.7 Categorical Boosting

It is an open-source library method. This method is an efficient, fast, and scalable implementation of gradient boosting for decision trees. This model handles categorical features and is often used for multiclass classification and binary classification models. It includes the regularization techniques such as feature selection and model parameter tuning which helps in preventing the over fitting model. It uses a highly optimized implementation of gradient boosting so that it can train models several times faster than other gradient boosting libraries. It can work well for large datasets. The Categorical Boosting model has given a R^2 Score of 82%.

5.8 Light Gradient Boosting

It's a gradient boosting library that provides the implementation of gradient boosting for decision trees. This algorithm is well suited for large datasets and it's faster than other gradient boost libraries. It uses a highly optimized implementation of gradient boosting and is very fast. It's designed to handle a large number of large datasets and is also a very efficient implementation of gradient boosting for decision trees. It also additionally provides GPU acceleration for training. This algorithm also supports parallel and distributed training which is making it easy to scale to large datasets. The Light Gradient Boosting model has given a R^2 Score of 82%.

5.9 Decision Tree

This is a tree-based model which is used for supervised learning. This algorithm is a graphical representation of all possible solutions to a decision that is based on conditions. The tree will be having internal nodes, which will be representing the

condition or tests for that node and the leaf nodes will be representing the outcomes or decisions and branches which is representing the possible path in the tree which is completely based on the value of the input features. These are easy to understand and the logic behind this algorithm is that it is represented in a tree format structure. This will handle the missing values in the data as they will be choosing the best spilt based on the information available. This can handle both numeric and categorical features. The decision tree model has given a R^2 Score 66%.

5.10 Linear Regression

Linear regression is a supervised machine learning model which is mainly used for the purpose of regression. This method is actually a statistical model used to analyze the relationship between target variable and one or more independent variable. It is main task is to find the best linear relationship between the variables. The linear regression model has given a R^2 Score of 57%.

5.11 Support Vector Machine

This is a supervised learning algorithm that is mainly used for classification and regression analysis. They are based on the concept of finding the hyperplane in a high-dimensional space which can separate it into different classes in the best possible way. The hyperplane is chosen such that the margin or the distance between the hyperplane will be maximized. These are called support vectors. This algorithm can handle non-linear relationships between the features and the target variable by using the support vectors. They can be very much effective to handle a large number of datasets. This algorithm also requires the selection of the kernel function. The support vector machine model has given a R^2 Score of 77%.

5.12 Multilayer Perceptron

Multilayer perceptron which is known as artificial neural networks is a type of machine learning model that mimics the human brain. It consists of multiple layers of interconnected nodes in which each layer processes the information from the previous layer information that was received. It can handle non-linear relationships between the input and target variables. It can be designed with any number of layers and nodes so that it's possible to handle large and complex datasets. These are trained using stochastic gradient descent which is an optimization algorithm. The Multilayer perceptron model has given a R^2 Score of 77%.

5.12.1 Hybrid Model

This model is based on an ensemble learning model where multiple models are combined and make predictions on the new data and then the predictions are combined to give a better result. In this hybrid model, we have used the Extreme Gradient Boosting and Adaptive boosting algorithm.

Extreme Gradient Boosting algorithm is used for both classification and regression type of problems and also this algorithm is an ensemble learning model which is combining the prediction of multiple weak models in order to produce a stronger prediction model. This algorithm also efficiently handles the missing values without requiring significant pre-processing. It's also possible to train a large number of datasets in a reasonable amount of time. It also includes several forms of regularization which will help prevent the overfitting problem.

Adaptive boosting is one of the ensemble models that is being widely used. This algorithm aims in building a model and it will also give equal weights to all the data points that are present and then it will start assigning higher weights to points that have been classified wrong. So now all the points which have higher weight will be given more importance in the next training model and this will get repeated until the error is becoming low. This is also simple to implement and its also being computationally efficient as it takes training only on a small number of weak models.

The combined predictions of the model will be used and trained using a random forest regressor on the predictions that were combined. The stacking model has given a R^2 Score of 96.5%.

6 R^2 Score Comparison

In Table 1 the R^2 Score of all algorithms that have been done is compared.

In Fig. 2 the graph of the hybrid algorithm's MSE, MAE, RMSE, R-Squared, and EVS is shown. In Fig. 3 the R^2 Score of all models has been presented.

7 Conclusion

This paper has analyzed the different algorithms of machine learning such as Bootstrap aggregating, Gradient boosting, Adaptive boosting, Random forest, K-Nearest Neighbors, Extreme Gradient Boosting, Categorical boosting, Hybrid algorithm, Light Gradient Boosting, Decision Tree, Linear regression, Support vector machine, Multilayer perceptron. And from this, the hybrid model performed well giving a good R^2 Score of around 96.5%. This method will be helpful in the medical field by providing automated results and resulting in good R^2 Score of the model. In the future focusing on further refinement in the score and adding more features and exploring the relationship to neonatal outcomes will be done.

Table 1 A comparison of algorithm R^2 score

Algorithm	R^2 score (%)
Random forest	81
Gradient boosting	83
Adaptive boosting regressor	80
Bootstrap aggregating	67
K-nearest neighbors	66
Extreme gradient boosting	80
Categorical boosting	82
Light gradient boosting	82
Decision tree	64
Linear regression	57
Support vector machine	71
Multilayer perceptron	77
Hybrid model	96.5

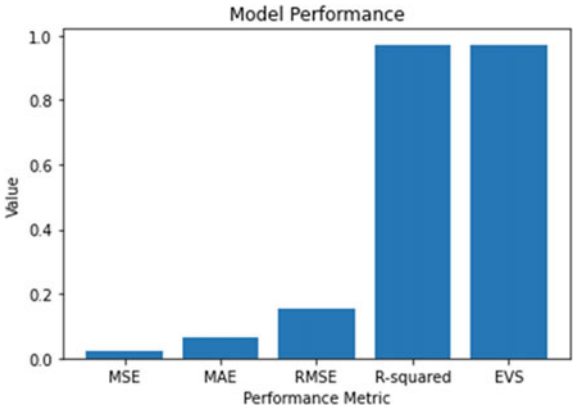


Fig. 2 The graph showing additional assessment metrics for a hybrid prediction model

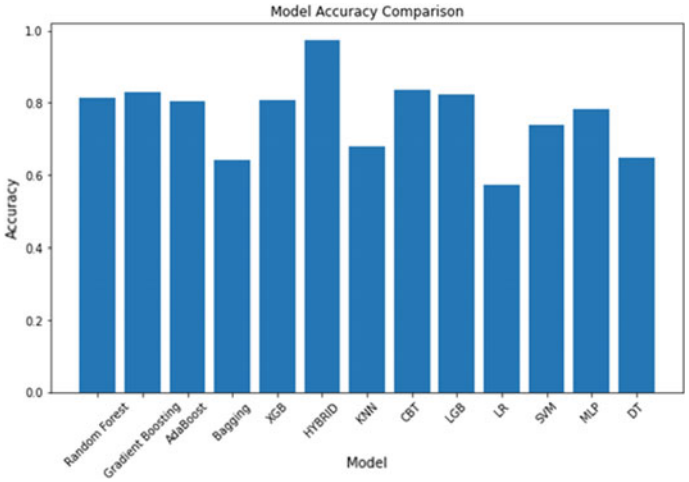


Fig. 3 The graph compares the R^2 score of all the models

References

Abdallah AY, Joho AA, Yahaya JJ (2021) Influence of maternal lifestyle behaviors on birth weight and Apgar score. Int J Afr Nurs Sci 15:100334. ISSN 2214-1391. https://doi.org/10.1016/j.ijans.2021.100334

Chao M, Menon C, Elgendi M (2022) Validity of Apgar score as an indicator of neonatal SARS-CoV-2 infection: a scoping review. Front Med (Lausanne) 8:782376. https://doi.org/10.3389/fmed.2021.782376. PMID 35087845; PMCID PMC8787091

Cnattingius S, Johansson S, Razaz N (2020) Apgar score and risk of neonatal death among preterm infants. N Engl J Med 383(1):49–57. https://doi.org/10.1056/NEJMoa1915075. PMID: 32609981

Ibrahim D, Frize M, Walker RC (2006) Risk factors for Apgar score using artificial neural networks. Conf Proc IEEE Eng Med Biol Soc 2006:6109–6112. https://doi.org/10.1109/IEMBS.2006.259591. PMID: 17946357

Jensen LV, Mathiasen R, Mølholm B, Greisen G (2011) Low 5-min Apgar score in moderately preterm infants; association with subsequent death and cerebral palsy: a register-based Danish national study. Acta Paediatr 101(2):e80–e82. https://doi.org/10.1111/j.1651-2227.2011.02516.x. PMID 22054115

Khatibi T, Farahani A, Sepehri MM, Heidarzadeh M (2022) Distributed big data analytics method for the early prediction of the neonatal 5-minute Apgar score before or during birth and ranking the risk factors from a national dataset. AI 3:371–389. https://doi.org/10.3390/ai3020023

Park TJ, Chang HJ, Choi BJ, Jung JA, Kang S, Yoon S, Kim M, Yoon D (2022) Machine learning model for classifying the results of fetal cardiotocography conducted in high-risk pregnancies. Yonsei Med J 63(7):692–700. https://doi.org/10.3349/ymj.2022.63.7.692. PMID 35748081; PMCID PMC9226828

Rotariu C, Pasarica A, Costin H, Nemescu D (2014) Spectral analysis of fetal heart rate variability associated with fetal acidosis and base deficit values. In: 2014 international conference on development and application systems (DAS), Suceava, Romania, pp 210–213. https://doi.org/10.1109/DAAS.2014.6842457

Sahak R, Mansor W, Khuan LY, Zabidi A, Yasmin F (2009) An investigation into infant cry and Apgar score using principle component analysis. In: 2009 5th international colloquium on signal

processing & its applications, Kuala Lumpur, Malaysia, pp 209-214.https://doi.org/10.1109/CSPA.2009.5069218

Tarimo CS, Bhuyan SS, Zhao Y et al (2022) Prediction of low Apgar score at five minutes following labor induction intervention in vaginal deliveries: machine learning approach for imbalanced data at a tertiary hospital in North Tanzania. BMC Pregnancy Childbirth 22:275. https://doi.org/10.1186/s12884-022-04534-0

Data Pre-processing Techniques for Brain Tumor Classification

Neha Bhardwaj, Meenakshi Sood, and S. S. Gill

Abstract Brain tumor detection and classification is a main concern owing to the global fatalities caused by it. Computer-aided design (CAD) techniques for classification of brain tumor are benefitting radiologists and doctors for error-free detection and correct prognosis. Convolutional neural network (CNN) is the most sought-after framework in deep learning for brain cancer detection due to its robust nature and efficient handling of large datasets. Pre-processing has a pivotal role in brain tumor classification framework architecture. Benchmark datasets acquired for training and testing of the CNN framework have to be pre-processed before being fed to the framework. Pre-processing techniques like data shuffling, resizing, normalization, and augmentation are done to enhance the image quality for its effective analysis. It also increases the reliability of the model by decreasing the learning time. The problem of underfitting and overfitting is also overcome by adhering to pre-processing techniques prior to feeding the dataset to the designed framework in deep neural nets. In this paper, benchmark dataset for brain tumor classification has been pre-processed in two different manners before being fed to the deep CNN model for the classification of brain tumor and results are compared.

Keywords Augmentation · Resizing · Normalization · CNN · Classification

N. Bhardwaj (✉)
ECE Department, NITTTR Chandigarh, Chandigarh, India
e-mail: neha.ece21@nitttrchd.ac.in

M. Sood
CDC Department, NITTTR Chandigarh, Chandigarh, India
e-mail: meenakshi@nitttrchd.ac.in

S. S. Gill
IMEE Department, NITTTR Chandigarh, Chandigarh, India
e-mail: ssg@nitttrchd.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_20

1 Introduction

Brain tumor is one of the major illnesses worldwide causing significant mortalities. A tumor is an abnormal mass or the development of cells in any part of the body. Since the brain is the primary organ of the human body, any malfunctioning and growth in it needs timely addressing for a proper diagnostic evaluation. Tumors are categorized mainly as benign and malignant. Benign tumors advance inside the skull but outside the brain tissue. Benign tumors generally do not metastasize. These tumors are mostly non-cancerous. Malignant tumors whereas are cancerous and may invade surrounding body parts and hamper their functioning (Amin et al. 2018). Therefore, proper detection is crucial for timely diagnosis and its removal. Depending on the disease's severity, gliomas are categorized from Grade 1 to Grade 4, according to the World Health Organization, in which Grade 1 tumors are less severe and comparatively easily curable than Grade 4 tumors who metastasize and require radiation therapy for its complete eradication (Johnson et al. 2017; https://www.cancer.net/cancer-types/brain-tumor/grades-and-prognostic-factors).

Owing to the complex structure of brain, professional medical staff is required to detect and categorize brain tumor (DeAngelis 2001; Tazin et al. 2021). The current methods of detecting brain tumors are manual, invasive, and rely on histopathological analysis. Existing manual detection methods generally require human intervention for tumor detection in the brain and thus are biased or subjective in nature whereas automated frameworks aid in locating tumors with minimal human intervention (Irmak 2021; Shafi et al. 2021).

Computer-aided diagnosis systems can benefit radiologists in detecting brain tumors earlier in the clinic. Researchers have developed a number of automated systems for detecting brain tumors, including machine learning techniques, supervised or non-supervised, transfer learning, and deep neural models (Muhammad et al. 2021). Machine learning accomplishes a specific task without using unambiguous instructions and relying on patterns in a supervised or unsupervised manner (Deng and Yu 2014). Deep learning frameworks generally take an edge over the existing methods due to its ability to handle large datasets effectively with negligible error. Computer-aided fully automated detection methods can help clinicians and radiologists in enhancing their diagnostic abilities and reducing the time for a correct prognosis (Shafi et al. 2021; Çınarer et al. 2020). CNN is the most popular deep learning model (Çinar and Yildirim 2020). It can handle multiple datasets in an array due to its robust nature. Figure 1 shows the workflow for brain tumor classification. The first step for training of any framework includes the acquisition of benchmark dataset and its pre-processing (Rahman and Islam 2022). To reduce computational intricacy while maintaining image information, the acquired image database has to be pre-processed before being fed to the framework for training and testing. Pre-processing helps in the reduction of unwanted biases in the image dataset, if any, and for enhancement of specific qualities required for the application under study (Masoudi et al. 2021). Various steps in pre-processing exist like data augmentation,

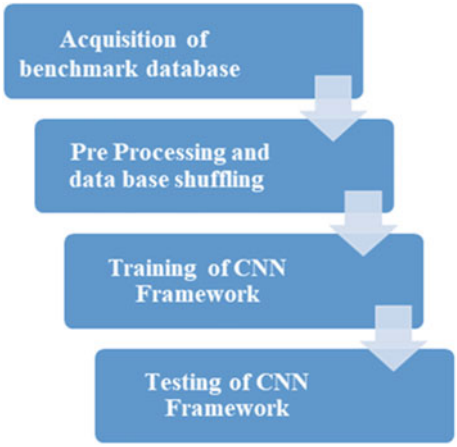


Fig. 1 Workflow of brain tumor classification

resizing, and normalization are done to enhance image quality (Vishvanathan et al. 2019).

The paper presents various pre-processing techniques like data augmentation, resizing, and normalization on Kaggle benchmark dataset, and its effect on the performance of brain tumor classification framework was studied. There was an increase in classification accuracy when the dataset was subjected to pre-processing techniques. The paper is organized as follows: Sect. 2 shows the related work in this field. Section 3 shows the various pre-processing techniques. Results and discussions followed by conclusion conclude the paper.

2 Related Work

The recent advancement in artificial intelligence (AI) and deep learning (DL) has resulted in significant attainment in correct diagnosis of brain glioma (Swati et al. 2019). Deep learning techniques are generally preferred owing to its ability to handle large datasets efficiently with less human intervention. Also, convolutional neural networks are employed due to its robust nature and compatibility with image classification and recognition tasks (Abbasi et al. 2019). CNNs generally require less pre-processing and can handle multiple datasets in array. It is possible to improve clinicians and radiologists diagnostic abilities and curtail the time frame for a correct prognosis by employing computer-assisted fully automated detection methods (Shafi et al. 2021). S. Vishvanathan et al. analyzed the effect of data pre-processing on disease classification and the disease undertaken was brain tumor and the accuracy achieved was enhanced when the dataset was subjected to pre-processing techniques (Vishvanathan et al. 2019). M. Hashemi et al. did image resizing with deformation by using zero padding technique and compared it with conventional method of increasing the size by zooming and concluded that resizing the images gave better

results before being fed to the CNN (Hashemi 2019). M. M. Badža et al. normalized and resized the brain MRI images to 256 * 256 pixels and augmented the dataset by flipping and rotation and obtained an accuracy of 96.56% (Badža and Barjaktarović 2020). A. ul Haq et al. resized image size of brain tumor MRI images into 512 * 512 on Kaggle dataset and obtained an accuracy of 97%. The optimizer used was SGD (ul Haq et al. 2022). C. Fan et al. reviewed various data pre-processing techniques like missing value calculation, data division, data scaling and concluded it to be an indispensable part in machine and deep learning algorithms (Fan et al. 2021). J. Amin et al. discussed various pre-processing techniques in machine learning like linear, nonlinear, fixed, multiscale, and pixel-based techniques. Also the publically available datasets, feature extraction methods, and enhancement techniques for brain tumor detection and its further classification (Amin et al. 2021). R. Vankdothu et al. used adaptive filter in pre-processing phase to remove noise in brain MRI images of Kaggle dataset containing 3000 MRI images and achieved an accuracy of 95% (Vankdothu and Hameed 2022). C. Srinivas et al. resized 256 brain tumor images from Kaggle dataset to 256*256, and all the images were normalized; brain tumor classification was done on pretrained models like VGG16, ResNet, and obtained accuracy was observed as a performance parameter (Srinivas 2022). H. A. Shah et al. used data augmentation as a pre-processing step in brain MRI dataset and fine-tuned Efficient Net B0 base model and achieved classification accuracy of 98.86% (Shah et al. 2022). F. J. Díaz-Pernas et al. did data augmentation on 3064 MRI images of three types of brain tumor using elastic transform and obtained an accuracy of 97.3% (Díaz-Pernas et al. 2021). K. Dang employed data augmentation image pre-processing technique using gamma correction on 335 MRI images obtained from BraTs 2019 database and obtained an accuracy of 93% on testing dataset of brain tumor multi-classification (Dang et al. 2022). Y. Xie et al. carried out image pre-processing techniques like resizing, augmentation, and image registration on different brain MRI datasets collected from different hospitals and achieved an accuracy of 91.63% in distinguishing gliomas of three different types (Xie et al. 2015). S. Alsubai et al. converted MRI images collected to grayscale and applied thresholding techniques to convert images into binary images, resized the size to 224 * 224, and obtained brain tumor prediction accuracy of 99% (Alsubai et al. 2022). N. Bhardwaj et al. developed CNN framework for brain tumor classification, and accuracy of 99.4% was achieved (Bhardwaj et al. 2022).

3 Data Acquisition

Benchmark dataset, which is a combination of the following three datasets: Figshare, SARTAJ, and Br35H have been used for classification from Kaggle (https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection; https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset). The dataset contains 4117 MRI images of three different types of tumor—glioma, meningioma, and pituitary. The dataset is balanced dataset. The dataset has 1321 Glioma tumor images,

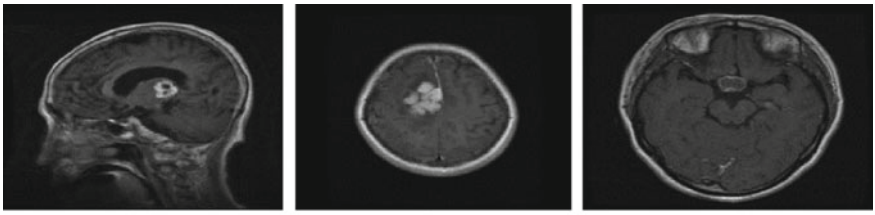


Fig. 2 Brain tumor MRI: **a** glioma, **b** meningioma, and **c** pituitary

1339 Meningioma images, and 1457 Pituitary tumor images. Figure 2 shows the pictorial representation of three tumors acquired from the same dataset.

3.1 Data Augmentation

Data augmentation increases the size of limited training samples to aid the better training of the deep learning model (Fan et al. 2021). Many transformations like flipping, rotation, cropping, and padding come under this category. Data augmentation helps in overcoming the problem of overfitting and underfitting in deep learning models.

3.2 Data Resizing

Resizing allows images to be a uniform size without any information loss. It is an important pre-processing step in machine and deep learning. Also, the image dataset being used for training may contain images of different sizes. To maintain uniformity, all image dataset is resized to one value before being fed to the neural net. Figure 3 shows brain tumor image being resized to 200 and 150, respectively. Pixel value of 200 is selected for resizing all the images before feeding them to the classification network (Hashemi 2019).

3.3 Data Shuffling

Database is shuffled before feeding it to the deep neural net so that the same image set is not always fed to the framework for training and testing. This operation in deep learning helps overcoming the problem of overfitting and converges the performance. The dataset of all the three types of tumor was shuffled to avoid repetition of the same dataset being fed to the framework every time.

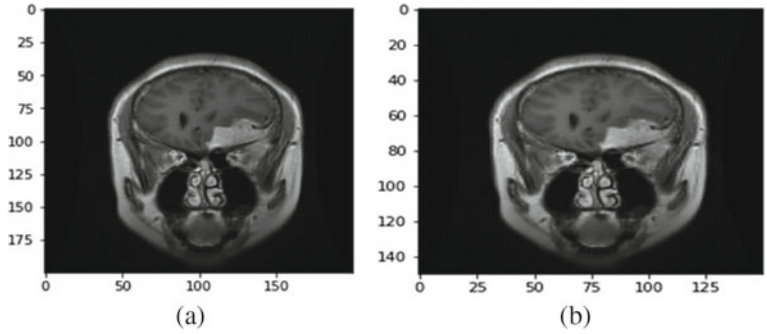


Fig. 3 Brain tumor image resizing: **a** 200, **b** 150

3.4 Data Normalization

Data normalization puts the data under consideration on a similar scale. It is done to change the pixel values in images. Brain MRI datasets are collected from multiple sources using different scanners with varying resolutions. Hence, the collected data has to be normalized to lessen the impact of differences between scanners and their respective scanning parameters. Image normalization improves image contrast for better comprehension. For most images, the pixel values lie between 0 and 255. Normalizing the dataset helps in faster convergence of the deep net while training (Nikhil 2023). The pixel value of the entire dataset was normalized to maintain the similarity scale. Max normalization technique has been used to normalize the dataset.

3.5 Training and Testing Data Split

The database acquired is split into training and testing datasets. Generally, validation split of 0.3 or 0.2 is used for the split. It is a convenient way to ascertain the performance of predictive deep or machine learning models. Training dataset is used to fit the deep net or machine learning model, and testing data is used to evaluate the model (Sarker 2021). In this brain tumor classification framework, data split of 0.3 or 0.2 is generally employed.

4 Results and Discussions

1. The benchmark dataset acquired is pre-processed and divided with validation split of 0.3 (2881 for training and 1236 for testing), resized to 200-pixel value with a validation split of 0.3 was fed to deep learning framework designed for

brain tumor classification and training accuracy of 99.4% and testing accuracy of 94.09% was achieved after 15 epochs as shown in Fig. 4.

2. The dataset acquired was pre-processed and divided with validation split of 0.2 (3293 for training and 824 for testing), and resized to 150-pixel value was fed to deep learning framework designed for brain tumor classification and training accuracy of 98.75% and testing accuracy of 93% was achieved after 15 epochs as shown in Fig. 5.

Table 1 shows the representation of the results obtained.

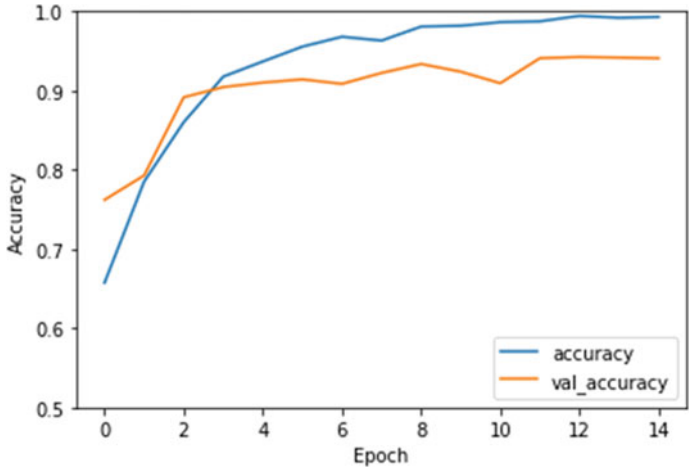


Fig. 4 Accuracy versus epochs on the above pre-processing techniques

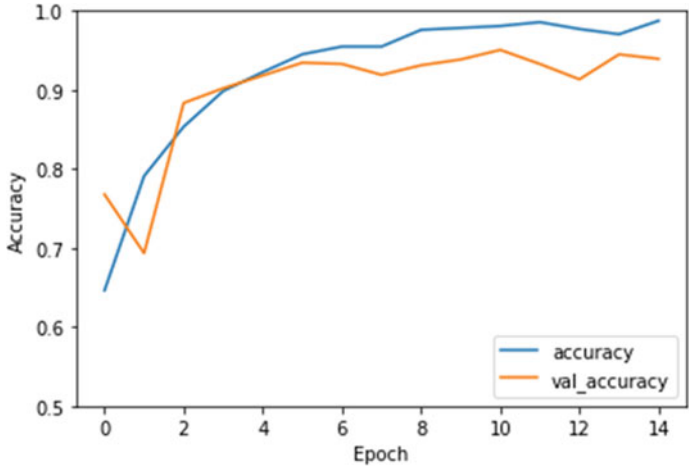


Fig. 5 Accuracy versus epochs on the above pre-processing techniques

Table 1 Pre-processing techniques on dataset acquired and the accuracy achieved

S. No.	Dataset	Validation split	Training dataset	Testing dataset	Pixel value	Training accuracy (%)	Testing accuracy (%)
1	4117	0.3	2881	1236	200	99.4	94.09
2	4117	0.2	3293	824	150	98.75	93

5 Conclusion

Pre-processing of the dataset is an important step and improves the accuracy and reliability of the developed framework. Pre-processing steps also shorten the training time of the model and fastens the model inference. The paper analyzes the effect of different sets of pre-processing techniques on the performance of brain tumor classification model. In the future, parameters of the CNN framework will be tuned and optimized for the detection and multiclassification of brain glioma.

References

Abbasi M, Eslami B, Rezaei Z (2019) Brain tumor classification using deep learning methods, vol 16, no Special Issue. Springer Singapore. https://doi.org/10.5812/iranjradiol.99160

Alsubai S, Khan HU, Alqahtani A, Sha M, Abbas S, Mohammad UG (2022) Ensemble deep learning for brain tumor detection. Front Comput Neurosci 16. https://doi.org/10.3389/fncom.2022.1005617

Amin J, Sharif M, Yasmin M, Fernandes SL (2018) Big data analysis for brain tumor detection: deep convolutional neural networks. Futur Gener Comput Syst 87:290–297. https://doi.org/10.1016/j.future.2018.04.065

Amin J, Sharif M, Haldorai A, Yasmin M, Nayak RS (2021) Brain tumor detection and classification using machine learning: a comprehensive survey. Complex Intell Syst

Badža MM, Barjaktarović MC (2020) Classification of brain tumors from MRI images using a convolutional neural network. Appl Sci 10(6)

Bhardwaj N, Sood M, Gill S (2022) Deep learning framework using CNN for brain tumor classification. In: 2022 5th international conference on multimedia, signal processing and communication technologies (IMPACT), Aligarh, India, pp 1–5. https://doi.org/10.1109/IMPACT55510.2022.10029043

Br35H: Brain tumor detection 2020 | Kaggle. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection. Accessed 3 Aug 2022

Brain Tumor MRI Dataset | Kaggle. https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset. Accessed 18 Jan 2023

Brain tumor: grades and prognostic factors | Cancer.Net. https://www.cancer.net/cancer-types/brain-tumor/grades-and-prognostic-factors. Accessed 20 July 2022

Çinar A, Yildirim M (2020) Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. Med Hypotheses 139(March):109684

Çınarer G, Emiroğlu BG, Arslan RS, Yurttakal AH (2020) Brain tumor classification using deep neural network. Adv Sci Technol Eng Syst 5(5):765–769. https://doi.org/10.25046/AJ050593

Dang K, Vo T, Ngo L, Ha H (2022) A Deep learning framework integrating MRI image preprocessing methods for brain tumor segmentation and classification. IBRO Neurosci Reports 13(June):523–532. https://doi.org/10.1016/j.ibneur.2022.10.014

DeAngelis LM (2001) Brain tumors. Med Prog N Engl J Med 114(2):114–123

Deng L, Yu D (2014) Deep learning: methods and applications. Found. Trends® Signal Process 7(3–4):197–387. https://doi.org/10.1561/2000000039

Díaz-Pernas FJ, Martínez-Zarzuela M, González-Ortega S, Antón-Rodríguez M (2021) A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network. Healthcare 9(2). https://doi.org/10.3390/healthcare9020153

Fan C, Chen M, Wang X, Wang J, Huang B (2021) A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front Energy Res 9(March):1–17. https://doi.org/10.3389/fenrg.2021.652801

Hashemi M (2019) Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. J Big Data 6(1). https://doi.org/10.1186/s40537-019-0263-7

Irmak E (2021) Multi-classification of brain tumor MRI images using deep convolutional neural network with fully optimized framework. Iran J Sci Technol - Trans Electr Eng 45(3):1015–1036

Johnson DR, Guerin JB, Giannini C, Morris JM, Eckel LJ, Kaufmann TJ (2017) 2016 updates to the WHO brain tumor classification system: what the radiologist needs to know. Radiographics 37(7):2164–2180. https://doi.org/10.1148/rg.2017170037

Masoudi S et al (2021) Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. J Med Imaging 8(01):1–14. https://doi.org/10.1117/1.jmi.8.1.010901

Muhammad K, Khan S, Del Ser J, Albuquerque VHCD (2021) Deep learning for multigrade brain tumor classification in smart healthcare systems: a prospective survey. IEEE Trans Neural Networks Learn Syst 32(2):507–522. https://doi.org/10.1109/TNNLS.2020.2995800

Nikhil B, Image data pre-processing for neural networks. Becoming Human: Artificial Intelligence Magazine. https://becominghuman.ai/image-data-pre-processing-for-neural-networks-498289068258. Accessed 28 Jan 2023

Rahman T, Islam MS (2022) MRI brain tumor classification using deep convolutional neural network. Springer Singapore. https://doi.org/10.1109/ICISET54810.2022.9775817

Sarker IH (2021) Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci 2(6):1–20. https://doi.org/10.1007/s42979-021-00815-1

Shafi ASM, Rahman MB, Anwar T, Halder RS, Kays HME (2021) Classification of brain tumors and auto-immune disease using ensemble learning. Inf Med Unlocked 24:100608. https://doi.org/10.1016/j.imu.2021.100608

Shah HA, Saeed F, Yun S, Park JH, Paul A, Kang JM (2022) A robust approach for brain tumor detection in magnetic resonance images using Finetuned EfficientNet. IEEE Access 10:65426–65438

Srinivas C et al (2022) Deep transfer learning approaches in performance analysis of brain tumor classification using MRI images. J Healthc Eng 2022. https://doi.org/10.1155/2022/3264367

Swati ZNK et al (2019) Brain tumor classification for MR images using transfer learning and fine-tuning. Comput Med Imaging Graph 75:34–46. https://doi.org/10.1016/j.compmedimag.2019.05.001

Tazin T et al (2021) A robust and novel approach for brain tumor classification using convolutional neural network. Comput Intell Neurosci 2021. https://doi.org/10.1155/2021/2392395

ul Haq A, Li JP, Khan S, Alshara MA, Alotaibi RM, Mawuli CB (2022) DACBT: deep learning approach for classification of brain tumors using MRI data in IoT healthcare environment. Sci Rep 12(1):1–14. https://doi.org/10.1038/s41598-022-19465-1

Vankdothu R, Hameed MA (2022) Brain tumor MRI images identification and classification based on the recurrent convolutional neural network. Meas Sensors 24(August):100412. https://doi.org/10.1016/j.measen.2022.100412

Vishvanathan S, Vidyapeetham AV, Kp S, Vidyapeetham AV (2020) ICICCT 2019—system reliability, quality control, safety, maintenance and management, no August 2019. https://doi.org/10.1007/978-981-13-8461-5

Xie Y et al (2022) Convolutional neural network techniques for brain tumor classification (from 2015 to 2022): review, challenges, and future perspectives. Diagnostics 12(8). https://doi.org/10.3390/diagnostics12081850

A Novel Approach for Detection of Lumpy Virus

Veena Kumari, Sahil Sandip Thorat, Sree Pravallika Pulikallu, Nayansi, Anmol Kaur, Tanvi Kumari, and Amritpal Singh

Abstract Recent years have seen a rise in breakouts and extensive skin infections caused by the lumpy skin virus. A convolutional neural network has been used to identify risk variables, predict the possibility of future outbreaks, and detect the virus in its early stages in order to solve this issue. The potential of this technology to address issues in public health is illustrated by the employment of CNNs in the fight against the lumpy skin virus. To process images, TensorFlow technology is employed. With the use of object, pattern, and feature recognition, this technology analyses and classifies images. CNN, a type of deep learning technique, is used in image processing by TensorFlow and Keras to evaluate and comprehend the information included in the images.

Keywords LSD · LSDV · CNNs · TensorFlow · Lumpy skin disease · Convolutional neural networks

1 Introduction

This scenario, cattle and other bovine animals are vulnerable to lumpy skin disease, a very contagious viral illness (LSD). It is brought on by the LSDV, which belongs to the Poxviridae family viruses. The skin and mucous membranes of affected animals acquire big, rigid, elevated, and painful nodules (lumps), which are indicative of the condition. These lumps, known as nodules, can be as little as peas or as big as

V. Kumari · S. S. Thorat (✉) · S. P. Pulikallu · Nayansi · A. Kaur · T. Kumari · A. Singh
Department of Computer Science and Technology, Lovely Professional University, Phagwara, Punjab, India
e-mail: devendrathorat408@gmail.com

V. Kumari
e-mail: chsveena2001@gmail.com

A. Singh
e-mail: amritpal.17673@lpu.co.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_21

grapefruits. The head, neck, and legs of the animal tend to have the most lumps, though they can arise anywhere on the body.

Symptoms of LSD include fever, loss of appetite, weight loss, swelling of hind and fore legs, decreased milk production, and decreased fertility in affected animals. In severe cases, the disease can lead to death. The disease is most common in young animals but can affect all age groups of cattle.

Direct contact with sick animals or encountering contaminated materials like feed, water, or equipment are the two main ways that LSD is transmitted. Arthropod vectors, like ticks and mosquitoes, can also transmit the virus. Given that it can result in large production losses and trade restrictions, the disease is seen to be of great economic significance.

Laboratory testing and clinical signs and symptoms are frequently used to make the diagnosis of LSD. There are few available treatments, which mostly involve managing the sick animals and providing supportive care.

The most effective strategy for limiting and preventing LSD is vaccination. There are vaccines on the market that have been demonstrated to lessen the prevalence and severity of the illness in afflicted herds.

In addition to vaccination, good biosecurity practices, such as preventing contact between infected and uninfected animals and controlling vectors with insecticides, are important for preventing and controlling LSD outbreaks.

As an outcome of the LSD, it is a highly contagious viral disease that affects cattle and other species of cattle (LSDV). Its distinctive feature is the development of large, rigid, raised, and painful nodules on the skin and mucous membranes of affected animals. It might result in considerable production losses and trade limitations. Laboratory testing and clinical signs and symptoms are typically used to get a diagnosis. The disease can be controlled and prevented most successfully by vaccination and proper biosecurity procedures.

2 Literature Survey

The highly contagious viral disease known as lumpy skin disease is communicable among cattle and other bovine animals (LSD). The illness is brought on by the Poxviridae family member lumpy skin disease virus (LSDV). The disease's most obvious sign is the development of firm, raised, and painful nodules (lumps) on the skin and mucous membranes of affected animals. These lumps, which can range in size from pea-sized to grapefruit-sized, can develop anywhere on an animal's body, but they are most usually found on the head, collar, and limbs.

A study by Magori-Cohen et al. (2012) described the transmission pathways of lumpy virus disease through mathematical modeling and evaluation. The study was based on a dataset consisting of a 2006 LSD outbreak in ten separate groups of cattle in a large diary format. A transmission model was used to evaluate the outbreak dynamics and risk factors of LSD. From this study, it appears that LSD can be transmitted in three different ways, indirect exposure within groups of the herd,

shared drinking water or direct contact, and through the milking process. In their study, they concluded that the probability of transmission of LSD through indirect contact, i.e. bugs, insects, etc., was higher.

In accordance with research done in Ethiopia by Mesfin (2019), the introduction of preventative measures like immunisation, movement restrictions, and quarantine significantly reduced the prevalence of LSD. In Tanzania, another study discovered that the use of bed nets coated with insecticide in addition to immunisation was successful in limiting LSD epidemics in cattle.

According to research by Kumar et al. (2021), Ranchi, India, is the epicentre of LSD outbreaks spread through cattle. Primary goat kidney cell scab skin lesions were used to isolate the virus. According to phylogenetic studies based on nucleotide sequencing of LSD virus (LSDV) ORF011, ORF012, and ORF036, the isolated virus (LSDV/Bos taurus-tc/India/2019/Ranchi) is closely related to Kenyan LSDV strains. In Vero cells, they also altered the isolated virus. The cytopathic effect (CPE) occurred only in the fourth blind passage after the isolated LSDV-infected Vero cells with high viral titres after adoption. In addition, this work represents the first effective LSDV isolation in India and provides details of the life cycle of LSDV as it occurs in Vero cells.

Based on a study by Roche et al. (2020), LSDV was originally discovered after an outbreak in Zambia in 1929, and its first full genome sequence was discovered in 2001. The LSDV virus, a member of the Capri poxvirus genus, is responsible for the sickness. Animals with the virus shed it through their oral and nasal secretions, contaminating common feed and water troughs. Hence, the disease can spread through tainted feed and water or by direct contact with the disease's vectors. It is a contagious vector-borne disease that mostly affects host animals like cows and water buffalo. Vectors that spread it include mosquitoes, certain biting flies, and ticks. In the immunological response, LSDV mimics or shares antigenic similarities with the sheep pox virus (SPPV) and goat pox virus (GTPV). Mortality is minimal, and the overall infection rate is believed to be 7%. The disease's mortality rate is less than 10%, while its morbidity ranges from 2 to 45%.

Clinical indications in British cattle exposed to the South African (Neethling) strain of the lumpy skin disease virus (LSDV) were noted during a three-week period, according to a study by Carn and Kitching (1995). A clinical scoring system was employed to assess the impact of the various infection pathways. Six out of seven of the animals infected via the conjunctival sac in the experiment were fully unresponsive to infection, and neither clinical symptoms nor serum-neutralising antibodies were found in the two animals. This study also shows that LSDV is only seldom transferred by animal-to-animal contagion and that parenteral virus inoculation is necessary to initiate infection.

According to a study by Al-Salihi et.al. (2015), the prevalence of LSDV in Iraq was 7396 infected cows and 227 infected calves, with morbidity and mortality rates of 9.11% and 0.51%, respectively. Also, he mentions clinical signs like persistent high pyrexia (40–41.5 °C), severe sadness, anorexia, and distinctive lumps all over the skin that are 3–8 cm in diameter. Within a 10 km radius of the outbreak, ring vaccination and live sheep pox vaccines were administered. He also mentions that the LSDV

infections in Iraq were placed between February and October 2014 and between August and October 2013. Weather conditions throughout these times ranged from moderate to hot, which favoured bug growth and spread.

Shalala Zeynalova et al. (2016), a study of the Azerbaijan Veterinary Service, deployed to border rayons in response to an outbreak in the neighbouring Iranian country to investigate whether any livestock in southern Azerbaijan was infected was done. LSD infections were first identified in the rayons of Bilasuvar, Jalilabad, Ujar, and Aghdash. Examining samples of blood, lesions, and important organs that had perished required the use of real-time polymerase chain reaction. In Azerbaijan, 2,762 cattle were discovered to have clinical symptoms consistent with LSD between June and November 2014. Using real-time PCR, 269 out of 269 samples tested positive for the LSD virus. A total of 33 cattle died, or 1.2% of those showing clinical indications of the disease. To lessen the spread of disease, preventative measures such travel restrictions, vector control, and vaccination were implemented.

Gezahegni Alemayehu et al. (2013) conducted a study in which the risks of LSD introduced by traded Borena bulls to the market chain were evaluated along with their consequences. The risk analysis approach was used in the evaluation. The odds of emission and exposure varied from insignificant to extremely high. The possibility of exposure was extremely high. During the LSD epidemic examination at six feedlot operations in and around Adama, 681 (6.1%) and 204 (1.8%) of the 11,189 bulls were detected. LSD was reckoned to have resulted in a massive economic loss of $667,785.6 USD. Because the relative risks for LSD are greater than negligible, the study demonstrates that Ethiopian veterinary services should thoroughly research disease prevention and control strategies along the chain.

Yang Wang et al. (2017) research presented the UIE-Net, a CNN-based network, and provided an end-to-end framework for enhancing underwater image quality. Both the colour correction and haze removal tasks are used to train the UIE-net. This combined training method allows for the simultaneous learning of a solid feature representation, for both tasks. The proposed learning architecture uses a pixel disruptive technique to more efficiently extract the inherent properties of local patches, which significantly boosts convergent speed and accuracy. The authors added 200,000 training photographs for UIE-net training using the physical underwater imaging model. UIE-net outperforms earlier methods, according to experiments on benchmark.

The Neethling strain outbreaks are referred to as a prototype of LSD, a Capri poxvirus (CaPVs), one of the eight genera in the Chordopoxvirinae subfamily of the Poxviridae, in a paper by Mulatu and Feyisa (2018). African and Asian domestic ruminants were affected by this virus. Also, he lists a variety of symptoms that can affect cattle, including body-wide lumps of skin and mild-to-severe fever. He further claims that infected animals exhibit lymphadenitis, emaciation, and lameness. Regardless of the mechanism of infection, he claims that general clinical symptoms persist for 7–14 days in experimentally infected calves and for 2–5 weeks in cases that arise naturally. Also, he claims that one or two nodules develop within two days of the start of a fever. Conditions Clinical symptoms in cattle with a moderate infection include salivation, ocular and nasal discharge, agalactia, and emaciation.

On the body, nodular lesions that are painful and hyperemic may also be seen. A mixture of antimicrobials, anti-inflammatory drugs, supportive care, and antiseptic solutions were beneficial in the treatment trials Salib and Osman carried out to avert LSD problems and preserve lives.

Abutarbush et al. (2013) stated that there were states in the Bani Kenanah region of the Irbid governorate, which is located near Jordan straddles Israel and Syria, and there were two cases, where all districts in Irbid Governorate were quickly affected by the disease. During the month following the outbreak, data were collected on the epidemiology of the disease and the number of infected animals on farms. A survey was conducted on 41 dairy farms where milk yield decreased between 0 and 100%. Anti-inflammatory drugs and broad-spectrum antibiotics were the main treatments used in affected animals. Treatment costs per animal ranged from £0 to £84.3.

The focus of this research, according to Annandale et al. (2010), was to identify the site of lumpy skin disease virus (LSDV) persistence in bulls excreting the virus in semen over a period of more than 28 days, to ascertain whether the virus is present in all fractions of semen, and to look at lesions that develop in the genital tract. Six post-pubertal bulls that had negative serology results were experimentally infected with a virulent LSDV field isolate. After the bulls were put down, tissue samples were taken from their genitalia and sent for histological examination, immunoperoxidase staining, viral isolation, and PCR. Bulls that had entire semen samples collected on day 28 post-inoculation that had proven positive by PCR underwent these procedures. According to the study, LSDV lives and excretes in the testis and epididymis of bulls for extended periods of time, and viral DNA was discovered in all fractions of seminal fluid.

An unsupervised chromatic segmentation technique is described in a study by Albiol et al. (2001), that can be used to separate skin-detected pixels into a collection of homogeneous regions for use in face identification applications. The chrominance and luminance data are employed successively during the algorithm's two stages of execution. Each stage uses a brand-new algorithm that combines techniques for region- and pixel-based colour segmentation. It is claimed that the algorithm has proven successful on a variety of test photos.

Dhillon and Verma (2020) published a study that contains a thorough analysis of numerous deep architectures and models that highlight particular model properties. The study went on to describe how CNN architectures work and their individual components before going into great detail about several different CNN models, including the traditional LeNet model as well as AlexNet, ZFNet, GoogleNet, VGGNet, ResNet, ResNeXt, SENet, DenseNet, Xception, and PNAS/ENAS. It is evident that the authors' main research interests revolve around three important applications of deep learning architectures: the detection of small armaments, humans, and wild animals. In-depth review summaries are also provided for each model, the systems, the database, the application, and the claimed accuracy.

3 Methodology

As there has not been any detection work on this disease to date, the novelty of the study work is that we can determine whether specific cattle have lumpy virus without encountering it. Additionally, the accuracy obtained is quite good.

The best outcomes in the fields of image identification, classification, and regression have been obtained using deep learning models. We can analyse enormous volumes of organised and unstructured data using deep learning since we just need to input the objects rather than the features, which subsequently detect the higher order features. We can improve and hone the precision with the aid of additional hidden layers. Several machine learning methods have been outperformed by deep neural networks.

Google created the open-source package TensorFlow to help in the creation and training of machine learning models. Used it for image processing called TensorFlow Image Processing (TF-IP). This module provides a set of functions and operations for image manipulation and analysis, such as image resizing, cropping, and flipping, as well as filtering and transform operations such as blurring, sharpening, and rotating. With TensorFlow's image processing capabilities, building, and training powerful image-based models for a variety of applications, including image classification, object recognition, and segmentation was done.

During our work, use of the Keras neural network library, which is based on TensorFlow, was done. It provides a user-friendly API for building and training machine learning models for image processing. Used several preprocessing functions to work with image data, such as image resizing, normalisation, and data augmentation. These functions can be easily integrated into a machine learning pipeline to prepare data for training models. For image classification applications, Keras offers a few pre-trained (CNN) models, such as VGG16 and ResNet50.

These models, which are simple to adapt to new datasets or utilise as feature extractors for transfer learning, have been employed by us during this time. For assessing image classification models, Keras has built-in measures like accuracy, precision, and recognition.

Convolutional neural networks (CNNs) are a particular kind of neural network that are frequently employed for image processing applications and contain various layers as,

Input layer: This layer takes the input image and forwards it to the next layer.
Convolutional layer: This layer applies a set of adaptive filters to the input image and extracts the features that are important for the task at hand. The output of a convolutional layer is a set of feature maps.
Pooling layer: In this layer, the output of the previous layer is down sampled, reducing the spatial size of the feature maps, and increasing their robustness to minor variations in the input image.
Dropout layer: In this layer, some neurons of the network are randomly turned off during training, preventing overfitting and improving the generalisation performance of the network.

Table 1 Structure of the neural network

Layer type	Dimensions	Output
Convolutional	5 × 5 × 4	16
Max pooling	2 × 2—Stride: 2	–
Convolutional	5 × 5 × 16	32
Max pooling	2 × 2—Stride: 2	–
Convolutional	5 × 5 × 32	64
Max pooling	2 × 2—Stride: 2	–
Convolutional	5 × 5 × 64	128
Max pooling	2 × 2—Stride: 2	
Fully connected	5 × 5 × 128	1024
Fully connected	1024	256

Batch normalization layer: In order to lessen the effects of internal covariate shift and increase network stability and speed, this layer normalises the output of the preceding layer.
Activation layer: The output of the preceding layer is given a nonlinear activation function in this layer, enabling the network to learn intricate representations of the input image.
Flattening layer: This layer converts the previous layer's output into a one-dimensional vector and prepares it for input to a fully linked layer.
Fully linked layer: This layer establishes connections between every neuron in the previous layer's output and every neuron in the layer after it, enabling the network to learn intricate feature combinations.
Output layer: This layer generates the network's final output, which could be a class label for a classification task or a pixel-by-pixel prediction for a segmentation task.

The CNN is the structure of the neural network employed in this study. Convolutional layers, pooling layers, fully linked layers, and dropout layers are all used in this kind of network, as previously mentioned. 100 × 100-pixel standard RGB photos make up the input that we utilised. The structure of the neural network we employed for this research is shown in Table 1. The overall backend work is illustrated in Fig. 2.

3.1 Dataset

Given dataset with images, provide a practical knowledge of the system, to determine whether cattle will have lumps on their skin or not. For this technique, cattle with lumps and cattle without lumps combinedly gathered over 2,000 random images, which are collected manually and through public platforms which were then sorted and displayed in Fig. 1.

Fig. 1 Dataset images of healthy cattles and diseased cattles

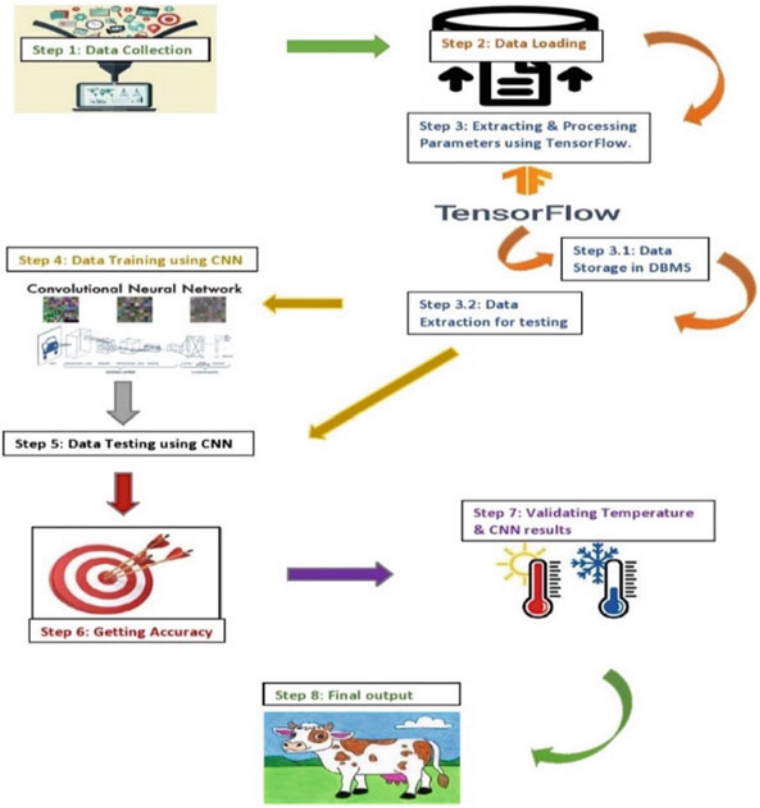


Fig. 2 Overall backend working of the machine learning code

3.2 Pseudo Code

1. Import required libraries.
2. Defining constants.
3. Learning rate, min_lr, lr_reduction_factor, patience, verbose.
4. Set image size < -(100,100).
5. Set input shape < -(100,100,3).
6. Sending processed images to DBMS.
7. Give path for train dataset.
8. Path for test dataset.
9. Define function to create charts for accuracy and loss in training process.
10. Define function for plotting confusion matrix.
11. Define function for randomly change in hue and saturation of images.
12. Define generator function.
13. Define function for converting image from RGB to HSV and grayscale.
14. Define function for model training.

$$\text{Covn2D} \rightarrow y(i, j) = (I, f)(x, y) = \Sigma_{-\infty}^{\infty} \Sigma_{-\infty}^{\infty} I(x - u, y - v) f(u, v), \tag{1}$$

$$\text{Maxpooling} \rightarrow X_{ij}^{[l]} = \frac{1}{MN} \Sigma_{N}^{M} \Sigma_{N}^{M} X_{iM+m, jN+n}^{[l-1]}, \tag{2}$$

$$\rightarrow \text{sigmoid}\,(x) = \frac{1}{1 + e^{-x}}, \tag{3}$$

$$\text{SoftMax} \rightarrow S(y)_i = \frac{\exp(y_i)}{\Sigma_{j=1}^{n} \exp(y_i)}. \tag{4}$$

15. Define function for training and evaluation model.
16. End of algorithm.

4 Result and Analysis

Compared to other models, the suggested model is accurate. As opposed to other techniques like Recurrent Neural Networks (RNNs), CNN is more accurate. The best substitute for RNN models is perhaps CNN. The proposed solution makes use of Pickel, a CNN. The model's accuracy is shown below. The graphical representation of training accuracy and training loss is shown in Fig. 3. Confusion matrix is defined in Table 2. The accuracy of the model was 91%.

Confusion Matrix Table:

Figures/Results:

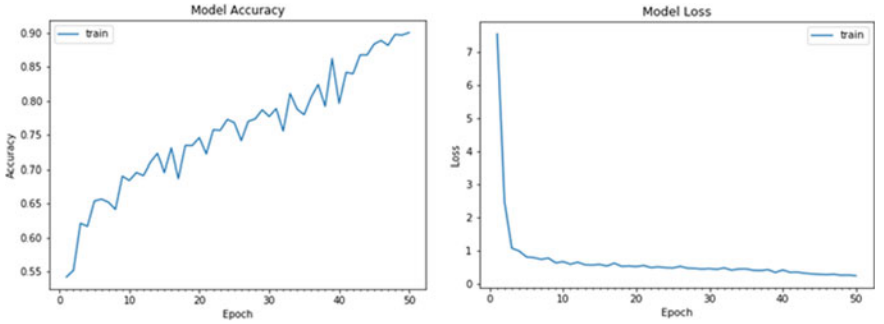


Fig. 3 Training accuracy and loss of the ML code

Table 2 Confusion matrix table

-	Precision	Recall	F1-score	Support
Healthy cow	0.83	0.74	0.78	66
Lumpy cows	0.71	0.81	0.76	52
Accuracy			0.72	118
Macro avg.	0.77	0.78	0.77	118
Weighted avg.	0.78	0.77	0.77	118

The degree to which a model accurately predicts the goal variable from the training set of data is referred to as training accuracy. A machine learning model's performance during a training period is gauged by its epoch loss (also known as training loss). Usually, it is determined by averaging the losses of all the training samples processed throughout the epoch.

5 Conclusion

In this research, we devised a technique to identify whether cattle are infected with the lumpy virus earlier. The individuals are assisted by this suggested solution, which uses CNN, TensorFlow, Pickel, and Firebase as backend measure to detect LSDV virus. With the addition of additional libraries for picture noise cancellation, we can also leverage thermal detection to measure temperature and image detection accuracy can be improved and use the results for further research.

References

Abutarbush SM, Ababneh MM, Al Zoubi IG, Al Sheyab OM, Al Zoubi MG, Alekish MO, Al Gharabat RJ (2013) Lumpy skin disease in Jordan: disease emergence, clinical signs, complications and preliminary-associated economic losses. Transbound Emerg Dis 62:549–554. https://doi.org/10.1111/tbed.12177

Albiol A, Torres L, Delp EJ (2001) An unsupervised color image segmentation algorithm for face detection applications. In: Proceedings 2001 international conference on image processing (Cat. No.01CH37205), vol 2. Thessaloniki, Greece, pp 681–684. https://doi.org/10.1109/ICIP.2001.958585.

Alemayehu G, Zewde G, Admassu B (2013) Risk assessments of lumpy skin diseases in Borena bull market chain and its implication for livelihoods and international trade. Trop Anim Health Prod 45:1153–1159. https://doi.org/10.1007/s11250-012-0340-9

Al-Salihi KA, Hassan IQ (2015) Lumpy skin disease in Iraq: study of the disease emergence. Transbound Emerg Dis 62:457–462. https://doi.org/10.1111/tbed.12386

Annandale CH, Irons PC, Bagla VP, Osuagwuh UI, Venter EH (2010) Sites of persistence of lumpy skin disease virus in the genital tract of experimentally infected bulls. Reprod Domest Anim 45:250–255. https://doi.org/10.1111/j.1439-0531.2008.01274.x

Carn V, Kitching R (1995) An investigation of possible routes of transmission of lumpy skin disease virus (Neethling). Epidemiol Infect 114(1):219–226. https://doi.org/10.1017/S0950268800052067

Dhillon A, Verma GK (2020) Convolutional neural network: a review of models, methodologies and applications to object detection. Prog Artif Intell 9:85–112. https://doi.org/10.1007/s13748-019-00203-0

Kumar N, Chander Y, Kumar R, Khandelwal N, Riyesh T, Chaudhary K et al (2021) Isolation and characterization of lumpy skin disease virus from cattle in India. PLoS ONE 16(1):e0241022. https://doi.org/10.1371/journal.pone.0241022

Magori-Cohen R, Louzoun Y, Herziger Y et al (2012) Mathematical modelling and evaluation of the different routes of transmission of lumpy skin disease virus. Vet Res 43:1. https://doi.org/10.1186/1297-9716-43-1

Mesfin Z (2018) Lumpy skin disease in Ethiopia 6:50–58. https://doi.org/10.22192/ijarbs

Mulatu E, Feyisa A (2018) Review: lumpy skin disease. J Veterinary Sci Technol 9:3

Roche X, Rozstalnyy A, TagoPacheco D, Pittiglio C, Kamata A, Beltran Alcrudo D, Bisht K, Karki S, Kayamori J, Larfaoui F, Raizman E, VonDobschuetz, S, Dhingra MS, Sumption K (2020) Introduction and spread of lumpy skin disease in South, East and Southeast Asia: qualitative risk assessment and management. FAO animal production and health, Paper 183. Rome, FAO. https://doi.org/10.4060/cb1892en

Wang Y, Zhang J, Cao Y, Wang Z (2017) A deep CNN method for underwater image enhancement. In: 2017 IEEE international conference on image processing (ICIP), Beijing, China, pp 1382–1386. https://doi.org/10.1109/ICIP.2017.8296508

Zeynalova S, Asadov K, Guliyev F, Vatani M, Aliyev V (2016) Epizootology and molecular diagnosis of lumpy skin disease among livestock in Azerbaijan. Front Microbiol 7:1022. https://doi.org/10.3389/fmicb.2016.01022

Internet of Healthcare Things-Enabled Open-Source Non-invasive Wearable Sensor Architecture for Incessant Real-Time Pneumonia Patient Monitoring

K. M. Abubeker, S. Baskar, and Michaelraj Kingston Roberts

Abstract The elderly, the chronically sick, and those who require constant monitoring benefit greatly from integrating IoT features into medical equipment, since it improves service quality and efficiency. Patients can have better outcomes in acquiring and transmitting vital healthcare data such as heart rate, SpO2, and body temperature using modern sensors and graphical processing unit (GPU). This paper presented the Medical and Healthcare IoT (MH-IoT), an IoT-based novel architecture built on open-source software and hardware for healthcare and ambient-assisted living. The MH-IoT framework incorporates non-invasive MLX90614D and MAX30102 sensors to enable uninterrupted health monitoring. Compared with conventional IoT-based healthcare facilities, the MH-IoT framework ensures and preserves patient safety, keeps connections alive when it matters most, and cuts down on human error. As a future-proof IoT solution, the MH-IoT cloud platform enables global device connectivity related to pregnancies and critical care at home for isolated patients in the İntensive Care Unit (ICU). Different test cases are executed, and comparisons are made with state-of-the-art devices to evaluate the efficacy of the developed wearable sensor platform. Since it uses open-source software and hardware, the proposed GPU-based MH-IoT technology is an excellent option for healthcare connectivity.

Keywords Healthcare · Internet of Things · Internet of vision · IoMT · Photoplethysmography · SpO2 · Wearable device

K. M. Abubeker (✉) · S. Baskar
Faculty of Engineering, Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore, India
e-mail: kmabubeker82@gmail.com; kmabubeker@ieee.org

K. M. Abubeker
Amal jyothi College of Engineering, Kanjirapally, Kerala, India

M. K. Roberts
Department of ECE, Sri Eshwar College of Engineering, Coimbatore, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_22

1 Introduction

Real-time health monitoring facilitated by the Internet of Things is feasible to offer outstanding care and keep patients safe and healthy by utilizing recent technological advances and intelligent sensors. The advent of 5G technology proliferation of mobile devices allows patients in rural communities to consult with physicians at urban specialty hospitals from the comfort of their homes. Undeniably, the continuing Covid'19 epidemic has increased the need for real-time remote care solutions (Mehrdad 2021). However, the healthcare sector is typically reluctant to embrace new technology, evaluations and a recent Covid'19 scenario show tremendous possibilities for IoT-based intelligent healthcare equipment and CT-based lung analysis (Al Bassam et al. 2021; Bushara et al. 2023). Patients and doctors from the poorest and most distant areas now have greater access to quality healthcare. IoT-enabled medical gadgets offer vital data to help healthcare professionals do their work smartly (Jain et al. 2021). Innovative technology in healthcare will enable patients to better monitor and treat their health issues and get assistance promptly if anything goes wrong (Kumar et al. 2020). It also enables healthcare practitioners to collect data remotely, allowing for observation and treatment that was previously merely feasible in a hospital environment.

Internet of Medical Things (IoMT) is limited to equipment used for medical purposes, such as smart sensors and artificial intelligence (AI) linked through Wi-Fi or Bluetooth to enable a machine to machine communication (Sun et al. 2020; Monfredi et al. 2023). Additionally, these IoMT devices connect to cloud services like Google Firebase and Amazon Web Services (AWS), which enable the storage and analysis of recorded data (Webster et al. 2022; Abubeker et al. 2022). The Internet of Healthcare Things (IoHT) defines individually identifiable medical equipment linked to the Internet and communicating with one other to enable asset tracking and location (Sarosh et al. 2021). In addition to medicines, wearable sensors such as, blood glucose, SpO2, heart rate, etc., feed physicians' findings using intelligent sensors and real-time data streaming (Karmakar et al. 2020; Tham et al. 2020). The proposed MH-IoT architecture, enhances patient outcomes and safeguards and preserves patient safety, maintains connectivity, and eliminates human error. The proposed architecture of MH-IoT is discussed in Chap. 2 with a thorough layered foundation. The third chapter covers the deployment of open-source software for the MLX90614D and MAX30102 sensors, involved as a curated research project deployed in on the Raspberry Pi 4B System on Module (SoM) computer. Furthermore, the system has been meticulously designed and thoroughly analysed, considering the well-being of pregnant women amidst the challenges posed by the COVID-19 epidemic. Finally, Sect. 4 summarizes the experiment's conclusion and findings.

2 Medical and Healthcare–IoT

Figure 1 represents a framework of medical and healthcare Internet of Things (MH-IoT) system; a cloud based real-time patient monitoring technology that allows physicians to evaluate patients at any stage in the patient's cycle. The proposed MH-IoT framework is illustrated in Fig. 2, it has a four tier architecture: IoT end layer, communication network layer, cloud layer, and front end layer. These layers work independently and are linked for wireless patient monitoring. The end layer collects non-invasive health data such as temperature, heart rate, and blood oxygen saturation (SpO2). Based on application design, IoT core receives data from linked sensor modules through a powerful Raspberry Pi 4B GPU hardware. This layer aims to detect aberrant patient symptoms and gathers sensor data to predict and monitor the patient's condition. The second layer is the communication network layer, including Bluetooth, Wi-Fi, 4G, and 5G. These low-power protocols are ideal for wireless networks. The Wi-Fi routers function as multi-thing gateways. This layer uses the NVIDIA Jetson Nano GPU system to interact with the sensor devices and create a local server unit. The firebase cloud is intended to support any infrastructure, application, and safety linked to the Internet. It secures all data via firewall and password protection. The application layer is the highest-level layer in the MH-IoT architectural hierarchy. Monitoring systems, electronic medical records, remote diagnosis systems, telemedicine systems, and healthcare professionals are the technologies that fall under this category. One of the primary functions of this layer is to accept real-time data from the cloud while maintaining ownership and credibility for the information received.

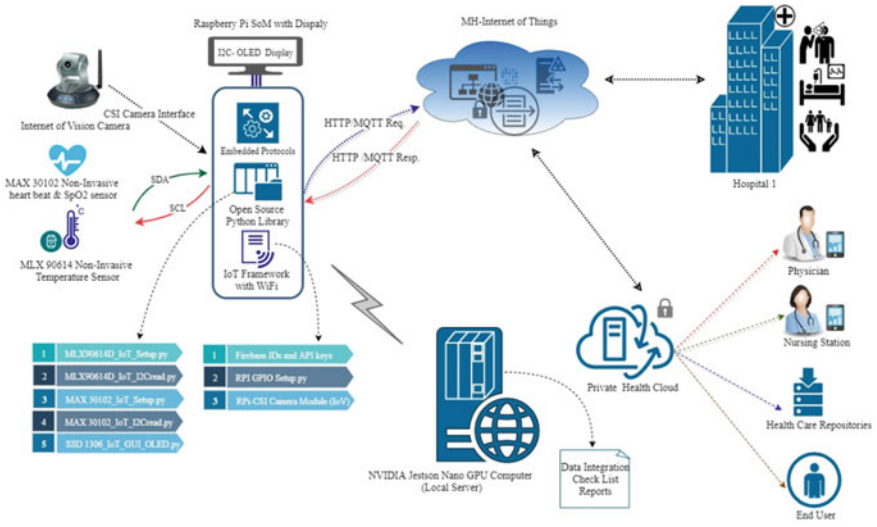


Fig. 1 Illustration of open-source software framework for Medical and Healthcare Internet of Things (MH-IoT) using non-invasive sensors

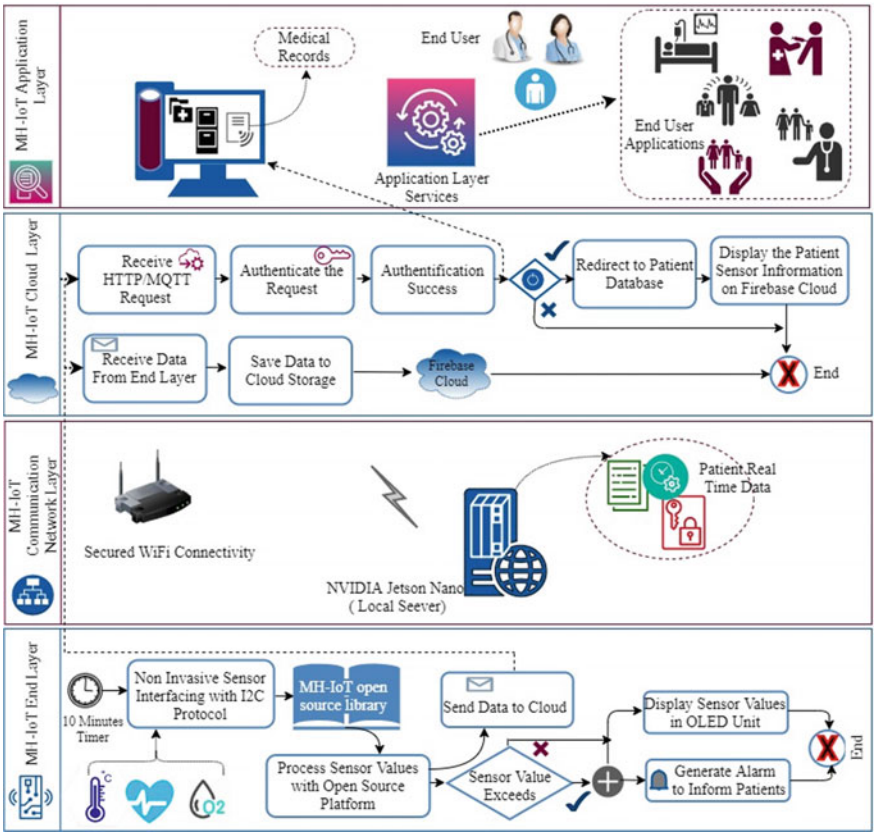


Fig. 2 Medical and healthcare IoT system modeling

3 Deployment of an Open-Source Software for MLX90614D and MAX 30102 in MH-IoT Framework

A number of techniques to measure the temperature of the human body are now available, utilizing both invasive and non-invasive approaches (Caizzone et al. 2019; Hina et al. 2020). This section discusses the development of an IoT-based real-time temperature monitoring system by utilizing open-source programming with non-invasive integrated sensor MLX90614D. MLX is a high-precision sensor with system-management bus (SMBus) that is compatible with the digital interface, ideal for GPU computers and SoM with IoT connectivity. At normal temperatures, most of the radiation falls in the IR range. MLX IR sensor works on the principle of Stefan–Boltzmann law, stated that total energy radiated per unit area of blackbody is proportional to the fourth power of its absolute temperature $\mathbb{T}$.

$$\mathbb{E} = \sigma \mathbb{T}^4 \tag{1}$$

Here, σ is the Stefan–Boltzmann constant (What is the Stefan-Boltzmann constant 2022) derived as:

$$\sigma = 2\pi^4 K^4 / 15c^2 h^3 \tag{2}$$

In Eq. (2), K is the Boltzmann constant c is speed of light in vacuum, and h is the Planks constant. It is possible to compute the absolute power of energy emitted by an object with unit area $\mathbb{A}$ as follows:

$$\mathcal{P} = \mathbb{A}\mathcal{E}\sigma\mathbb{T}^4 \tag{3}$$

Here, $\mathcal{E}$ is emissivity of the object with a value of $\mathcal{E} < 1$ for gray bodies and $\mathcal{E} = 1$ for black bodies. The MLX sensor provides high medical accuracy and resolution, consequently regulate the measurements and computations of the object temperature $\mathbb{T}_{\text{obj}}$ and ambient temperatures $\mathbb{T}_{\text{amb}}$. A 10-bit PWM and SMBus output are set to constantly communicate the temperature range of $-20°$ to $120°$ with a precision of 0.14 under isothermal and thermal equilibrium conditions. The IR radiation generated is represented as,

$$\mathbb{v}_{\text{IR}}(\mathbb{T}_{\text{obj}}, \mathbb{T}_{\text{amb}}) = \mathbb{A}\left(\mathbb{T}^4_{\text{obj}} - \mathbb{T}^4_{\text{amb}}\right) \tag{4}$$

where $\mathbb{T}_{\text{amb}}$ and $\mathbb{T}_{\text{obj}}$ is represented in kelvin. The internal DSP can compute the appropriate ambient and object temperatures. The RAM conversions are straightforward to use in the following relationship:

$$\mathbb{T}_{\text{amb}(K)} = \mathbb{T}_{\text{amb}(K)} * 0.02, \mathbb{T}_{\text{obj}(K)} = \mathbb{T}_{\text{obj}(K)} * 0.02, \tag{5}$$

the Raspberry Pi 4B computer reads the $\mathbb{T}_{\text{obj}}$and $\mathbb{T}_{\text{amb}}$values via serial data (SDA) and serial clock (SCL) of inter integrated circuit (I2C) communication protocol.

In Fig. 3, RPi 4B is configured as master and MLX 90614D is set as slave device via the SMBus interface. In general, the RPi starts the transmission of data by choosing the slave address (SA). The communications often have to start from zero SA with a low read/write signal. Once each 8 bits has been received, the slave transmits a signal ACK or NACK once the address has been recognized. If the slave sends NACK after packet error code (PEC), the master stops and repeats the message and master terminates the connection.

In Fig. 4, a wearable IoT-based thermal sensor using MLX90614D has achieved good efficiency, low error rate, and adaptability in terms of technology and cost. It is ideal for real-time medical IoT applications because of its unique characteristics, which include a very fast stabilization time, excellent memory reliability due to an integrated error correction process, and an expanded resolution range. The software sleep mode feature reduces the power consumption of the device, making it more appropriate for wearable technology. When compared with portable non-contact thermometers accessible in hospitals and public locations, MH-IoT wearable device

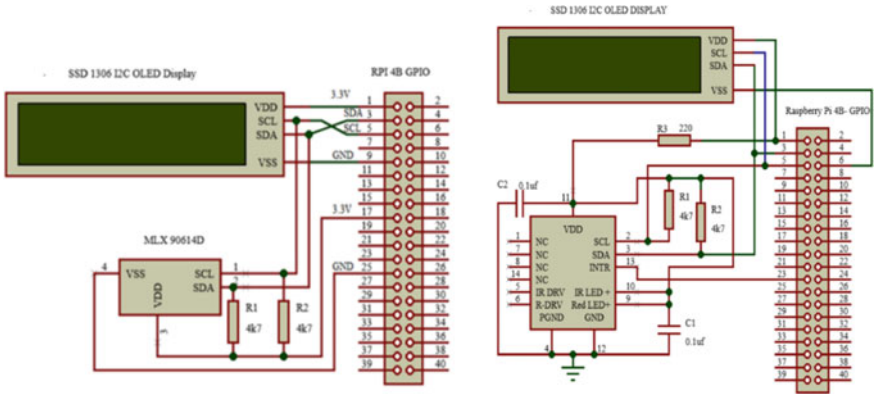


Fig. 3 MLX 90614 IR thermal sensor and MAX 30102 interfacing with Raspberry Pi 4B

eliminates the risk factor and improved the real-time performance. The proposed approach has the benefit of being an IoT-based, non-contact fever screening with a low error rate.

The MAX30102's SpO2 subsystem consists of an ambient light canceling (ALC) and 16-bit sigma-delta analog to digital converter (ADC) with 16-bit resolution. The discrete-time filter has been selected to eliminate interference from 50 Hz/60 Hz and residual low-frequency environmental noise. In SpO2 mode, following each data sample, the interrupt (INTR) triggered consisting of one IR and one red data. A data sample is composed of an IR data point in the cardiac rate mode. The hardware configuration of MAX30102 and GPIO pins of RPi 4B is shown in Fig. 3. The wearable model improves sensor readings by reducing skin motion noise and PPG channel aberrations. When compared with current techniques, the highest average SpO2 value was 97.2% ± 0.9% and 92.2% ± 1.1% for finger tips and inner wrist, respectively.

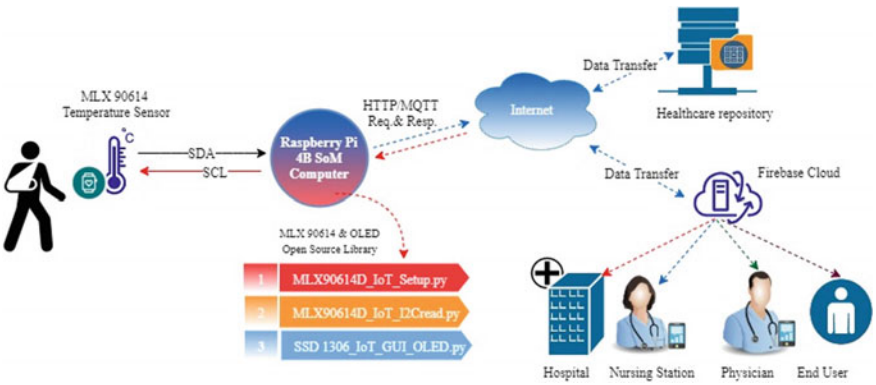


Fig. 4 Architecture of IoT-based human body temperature measurement system

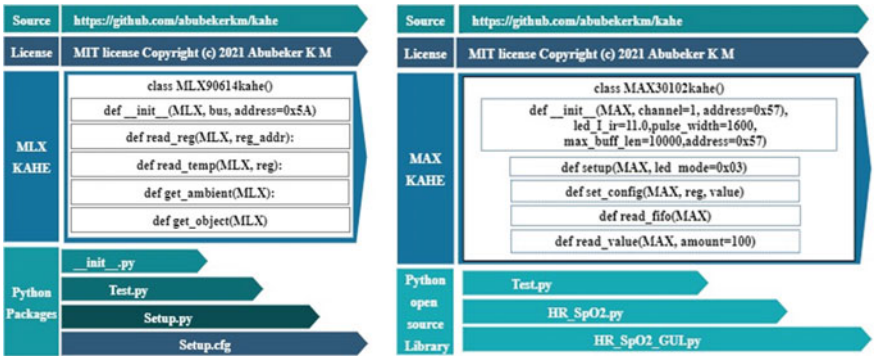


Fig. 5 Illustration of MLX 90614 and MAX 30102 open-source library

3.1 Open-Source Library for MLX90614 and MAX 30102

The open source library for the Melexis MLX 90614 IR temperature sensor and MAX 30102 have developed in Python Integrated Development and Learning Environment (IDLE) under the MIT (Massachusetts Institute of Technology) license (Copyright(c) 2021 Abubeker K M). In Fig. 5, README.rst is a file at the start of Python's documentation ecosystem that is structured using ReStructured Text (ReST). ReST is more flexible with native table support and automated content creation.

İn Fig. 5, Python's regular packages are usually implemented as a directory including a __init__.py file and objects are tied to names inside the namespace of the package. 'Setup.py' script, designed to create, distribute and install modules. This script sets the package's name and version, authors' names, e-mail Ids, and other meta information like details of the open-source license and package data. The project will not function unless packages specified in setup.py are installed using the command 'python setup.py' on root directory of MLX sensor. Setup tools enables 'setup.cfg' configuration files to define the metadata of the package provided for the setup () function. The 'setup.cfg' file contains default configuration command and the complex Python packages that are built and distributed. This is very useful if the process is used to build and distribute complex Python packages to the setup.py script. This makes the distribution of the project independent and provides package transparency.

In addition to tracking users' physical activity, wearable gadgets may also be used to care for the elderly, newborns, and patients with pandemic and epidemic diseases. The healthcare industry has encountered great challenges after the Covid'19 pandemic, and hospitals are constantly overflowing with people in need of urgent attention. With MH-IoT architecture, clinicians can easily monitor and care for patients under ICU or quarantine without being exposed to infection risk. Patients can watch on-demand wellness videos for aftercare, prevention, and precautions at any time. Aftercare, preventions, and interaction with physicians in urgent situations can be possible by the proposed framework. Thus, the MH-IoT architecture

with open-source platforms simplifies and streamlines the healthcare sector, even for doctors, patients, and their families.

4 Conclusion

Many individuals lose their lives every day because they either do not get timely medical care or wait too long to receive treatment. There is currently no practical alternative to Intensive Care Unit standards of care and cleanliness combined with remote patient monitoring for patients requiring critical care at home. In ICUs, keeping track of patients' vital signs and relaying that information to doctors is essential. Remote health monitoring technologies are used in home care, clinical settings, and hospitals to monitor an individual's vital signs and report any abnormalities to the patient and physician in real time, which reduces the risk of pandemic and epidemic infections and saves professionals time. The proposed framework utilises a Raspberry Pi SoM computer, MLX90614D and MAX30102 biosensors, and a Firebase cloud infrastructure with internet connectivity. The firebase cloud gathers, analyzes, stores, and makes available real-time health data and visualizations from patients in the ICU pneumonia and Covid'19 who are physically separated from medical care. This wearable device considers the health of patients with pandemic and epidemic diseases, including viral fever, influenza, and Zika virus, as well as pregnant women, without involving them in the traditional medical institution's approach. Therefore, the open-source MH-IoT architecture makes the healthcare system easier for physicians and patients.

References

Abubeker KM, Joshy A, George AT, Gopika G (2022) Internet of Healthcare Things (IoHT) enabled incessant real time patient monitoring system using non-invasive sensors. In: 2022 10th International conference on reliability, Infocom technologies and optimization (trends and future directions) (ICRITO), Noida, India

Al Bassam N, Hussain SA, Al Qaraghuli A, Khan J, Sumesh EP, Lavanya V (2021) IoT based wearable device to monitor the signs of quarantined remote patients of COVID-19. Inf Med Unlocked 24:100588. ISSN 2352-9148

Bushara AR, Vinod Kumar RS, Kumar SS (2023) LCD-capsule network for the detection and classification of lung cancer on computed tomography images. Multimed Tools Appl

Caizzone A, Boukhayma A, Enz C (2019) A 2.6 μW monolithic CMOS photoplethysmographic (PPG) sensor operating with 2 μW LED power for continuous health monitoring. IEEE Trans Biomed Circ Syst, pp 1–1

Hina A, Saadeh W (2020) A 186μW glucose monitoring SoC using near-infrared photoplethysmography. In: 2020 IEEE Asian solid-state circuits conference (A-SSCC), pp 1–4

Jain S, Nehra M, Kumar R, Dilbaghi N, Hu T, Kumar S, Kaushik A, Li CZ (2021) Internet of medical things (IoMT)-integrated biosensors for point-of-care testing of infectious diseases. Biosens Bioelectron 1(179):113074

Karmakar S, Kundu SK, Bandyopadhyay SK, Gangopadhyay M, Taki GS (2020) Importance of transition metal modified graphene-based non-enzymatic blood glucose sensors. In: 2020 4th International conference on electronics, materials engineering & nano-technology (IEMENTech), pp 1–4

Kumar S, Buckley JL, Barton J, Pigeon M, Newberry R, Rodencal M, Hajzeraj A, Hannon T, Rogers K, Casey D, O'Sullivan D (2020) A wristwatch-based wireless sensor platform for IoT health monitoring applications. Sensors 20(6):1675

Mehrdad S, Wang Y, Atashzar SF (2021) Perspective: wearable internet of medical things for remote tracking of symptoms, prediction of health anomalies, implementation of preventative measures, and control of virus spread during the era of COVID-19. Front Robot

Monfredi OJ, Moore CC, Sullivan BA, Keim-Malpass J, Fairchild KD, Loftus TJ, Bihorac A, Krahn KN, Dubrawski A, Lake DE, Moorman JR, Clermont G (2023) Continuous ECG monitoring should be the heart of bedside AI-based predictive analytics monitoring for early detection of clinical deterioration. J Electrocardiol 76:35–38

Sarosh P, Parah SA, Bhat GM, Heidari AA, Muhammad K (2021) Secret sharing-based personal health records management for the Internet of Health Things. Sustain Cities Soc 74:103129

Sun L, Jiang X, Ren H, Guo Y (2020) Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application. IEEE Access 8:101079–101092

Tham OY, Markom MA, Bakar AHA, Tan ESMM, Markom AM (2020) IoT health monitoring device of oxygen saturation (SpO2) and heart rate level. In: 2020 1st International conference on information technology, advanced mechanical and electrical engineering (ICITAMEE), pp 128–133

Webster CS, Scheeren TW, Wan YI (2022) Patient monitoring, wearable devices, and the healthcare information ecosystem. Br J Anaesth 128(5):756–758

What is the Stefan-Boltzmann constant? (2022, August 1). Retrieved from https://www.techtarget.com/whatis/definition/Stefan-Boltzmann-constant.

VANET Security Optimization with Blowfish Algorithm and Adversarial Transfer Learning

Richa Singh and Deepti Kakkar

Abstract Transportation systems have taken new heights ever since they are combined to wireless technologies. With network protocols being developed and optimized a whole new era of intelligent vehicles is here. However, the issue of security sustains as attacks and malicious elements in networks that can harm the integrity of whole system have grown too. In this paper, we have presented a framework for structuring the vehicular network carrying sensitive data in a open source environment. The unique vehicular identification number is generated by our pre-processing algorithm, followed by encryption provided by Blowfish algorithm. To optimize the data security, we will incorporate the adversarial transfer learning model as a defense mechanism against all known as well unknown attacks. The entire works shows promising results with robustness against cyber-attacks and the scalability rankings to be deployed over a large-scale networks. With our framework we have provided resultant model that have been proven to be more stable and immune to VANET attacks. We have analyzed our work over parameters like maximum stored keys, encryption time, battery capacity required, and memory required, it's been observed that with optimization, we have provided better results.

Keywords Adversial learning · Blowfish · Cryptography · Machine learning · VANETs

R. Singh (✉) · D. Kakkar (✉)
Department of Electronics and Communication Engineering, Dr. BR Ambedkar National Institute of Technology Jalandhar, Jalandhar, Punjab, India
e-mail: richas.ec.21@nitj.ac

D. Kakkar
e-mail: kakkard@nitj.ac

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_23

1 Introduction

Vehicular Ad Hoc networks (VANET) are one of the most popular wireless technologies being utilized in the fields of intelligent transportation systems (ITSs) (Boucetta and Johanyák 2022). The VANETs can provide enhancement in driving assistance technologies, hands on information regarding real time variable scenarios over a road track and the most important of all the safety of the user (Hu et al. 2020; Wang et al. 2020). VANET in general is build-up of a number of mobile nodes (vehicles), along with a roadside unit (RSU) and a trusted authority managing it all over a wireless communication medium. A VANET vehicle would be equipped with a on-board unit (OBU), OBUs are responsible for vehicle-to-vehicle (V2V) communications and vehicle-to-infrastructure (V2I) communications with the help of RSUs (Lin et al. 2022). The VANET set up works with a dedicated short range communication (DSRC) radio (Wei et al. 2023) and IEEE 1609 wireless access for vehicular network (WAVE) standards that are essentially used for V2V and V2I communications.

The VANETs in themselves possess some unique features like dynamic network topology, high mobility nodes all over the network, distributed networking, and a large-scale node placement. These features have made VANETs the most versatile as well as challenging when it comes to the security in VANETs. The task of securing the data has to counter numerous internal and external attacks in order to keep the data intact (Funderburg et al. 2021).

In order to preserve the data security in VANETs, most of the researchers have presented cryptography as a potential solution where each vehicle can utilize a pair of keys (private and public keys) along with a unique signature. These schemes have their own merits and demerits, however, moving toward the advanced mechanisms for data protection there is a need to incorporate machine learning models to provide a model that can not only provide the encryption but also can learn to protect the data from internal as well as external attacks (Debicha et al. 2023).

Algorithms used for encryption and decryption of data are broadly classified into two categories: first, Symmetric Key Cryptography, here there's a single key that is utilized for both encryption and decryption of the data. Second, Asymmetric Key Cryptography, here there would be two different keys. Unique keys for encryption and decryption of data (Fig. 1).

Symmetric Key Cryptography can be further classified in two categories based on their operation, Stream Cipher and Block cipher. Encrypting a digital data one bit at a time is known as stream cipher, while block cipher will treat the entire block of plain data as a whole to generate the encrypted value. Asymmetric encryption has its own merits but in order to provide an optimization layer and maintain a proper RSU to OBU communication with proper synchronization symmetric encryption algorithms have come out to be more efficient. As for our work we are required to generate a unique vehicle identification number (UVID), which can be encrypted to provide secure data. This secure data still can have intrusions dwelled over it, to overcome

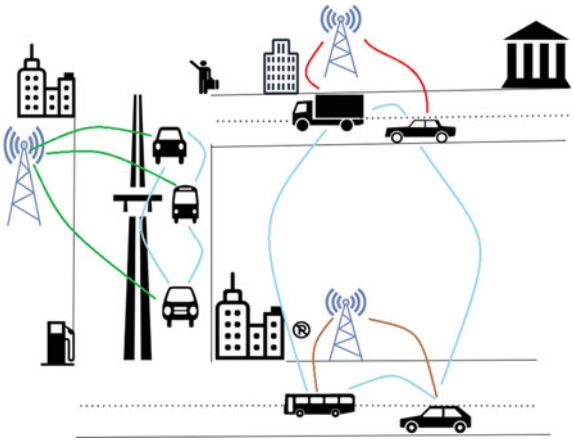


Fig. 1 Basic VANET architecture

this we would utilize a learning model to train our entire model and overcome all known as well as unique attacks.

The advancements of machine learning (ML) in recent years have provided a wide range of its applications and a vast scope of its further growth. On split thoughts, it also raises challenges and issues, peculiarly if we take a dive into the domains of privacy, security, and data protection.

ML has developed its scope in applications related to fields of finances, forecasting, medical diagnosis, and data processing related applications like cloud computing, its transmission, etc. However, the ML domain often finds itself standing in opposition to encrypted data sets.

Recently there have been studies and implementations regarding encryption algorithms which can allow computations over an encrypted data set via a server, which shows that a server can potentially utilize the machine learning algorithms to be a third party and perform computations over encrypted data set, causing it to change and hence, was not going to provide the original information at the receiver end with decryption action.

Existing algorithms only deal with the encryption and decryption of resultant outputs, which clearly presents a limit in their application as well as optimization in security domains. The model in use for optimization and encryption should work on providing encrypted results to its client for the decryption, potentially with no knowledge about any prior computation being performed over it, and the information should be decrypted to provide the original message to the end user. Clearly, we need to provide some way to ensure that the data encrypted will have no exterior computations corrupting it.

As its already clear that there is a pull and push with machine learning applications in attacks and defenses over data in a volatile environment. The realm of adversarial learning comes out as a solution. Adversarial learning is a mechanism where the model can learn numerous methods of attack that can be potentially possible over the model and adapt to tackle all those attacks. As new standards of attacks can take advantage of the fact that the input output relationship at a ML model is discontinuous, the completely unpredictable changes can change the output of a model to an extreme level. These types of perturbations at input side are called adversarial cases. Adversarial learning has shown promising results when exposed to such inputs which are relatively new to the entire model.

This paper will focus on how VANETs are structured with a overview of their architecture, followed by the types of attacks that can cause the system parameters to fail to protect the information in system. Then, there will a outlook presented for the cryptography algorithms existing. Further, there will be a detailed discussion of generating the UVID for all nodes with encryption process. Finally, the detailed process discussion of how we can use ML applications to optimize the entire model for an advance security model for VANETs.

1.1 Problem Statement

1. To carefully choose a cryptography technique of efficiency and security enhancement.
2. Constructing a model to optimize the security and develop the model to withstand the VANETs volatile nature.
3. Finally, the encryption and optimization should be scalable, reliable, and work toward handling a vast range of data.

1.2 Research Contribution

The noteworthy contributions from our work proposed are as follows:

1. Creating a standard structure flow for generation of data that can pose as a unique vehicular identity.
2. Structuring the entire model with adversarial transfer learning.
3. Frame work toward a much secure and robust VANET security mechanism.

2 Related Works

Boucetta and Johanyák (2022) have shown how the extensive demand of intelligent vehicles have led to VANETs becoming an increasingly expanding network. They have provided a study over security attacks in software defined networks over an open network and how these attacks are capable of threatening all the aspects of a VANET security system.

Hu et al. (2020) discussed the idea of two aggregated signature schemes for the security analysis in Ad Hoc networks. Their work provides an insight over how to design a security signature key which can be effective in case of handling some of the existing security threats. However, the VANETs are a special kind of network to be operated under open networks where one can't possibly predict the threats. Wang et al. (2020) provided a context aware and environment adaptive security framework based on the Markov chain. With the help of extensive calculation they have shown a way to combine environmental variables with other variables. Their work was able to show the current VANET security state.

Lin et al. (2022) have discussed an efficient conditional privacy preserving blockchain authentication for VANET. Their work discusses the requirement of improving the existing security mechanisms, with the help of key derivation and signatures of knowledge bundling up with smart contracts.

Wei et al. (2023) have provided a privacy preserving authentication key agreement scheme. They have utilized multi-trusted-authority model to form a symmetric cryptographic scheme, while introducing a fog computing concept to address the shortcomings of cloud computing and fog computing techniques.

Lee et al. (2021) have presented a framework for pairing free signatures to build up a insider attack resistance for VANETs. Their framework shows prevention against forged identities with malicious vehicles inside the VANET communication network.

Mees et al. (2023) have demonstrated the multi adversarial detection of evasion attack against the intrusion. They have shown very effectively how we can utilize the machine learning and deep learning techniques to form up an advance learning system which can solidify the network security further.

Xu et al. (2023) have provided a new dimension to securing V2V and V2I communications in an open network environment for VANETs with the help of tree-based authentication key. Their proposed work utilizes two scenarios, an identified vehicle joining and leaving the network, while updating the keys at regular intervals. However, the optimization part is being missed here.

Li et al. (2022) have shown a comprehensive implementation of semantic aware search over dynamic spatial data in VANETs. Their work have provided an insight for the spatial keyword searches while striking the tradeoff between privacy preserving and efficiency of search algorithm. The R-tree structure provides keywords and dynamic updates.

Almomani et al. (2022) have presented a detailed scheme of avoiding jammers and efficient localization in Ad Hoc networks. Their work has demonstrated the

estimation of jammer position, how the network can handle such intrusions and avoid it to save the users from Distributed Denial of Service attacks (DDoS).

Corona et al. (2013) have discussed the study of intrusion detection systems with issues to tackle the adversarial attacks over an open area network. Their work takes up a dive toward considering general taxonomy of attack against IDs and explored the solutions for how efficiently these attacks can be dealt with.

3 VANETs Threats, Challenges, and Requirements

VANETs are distributed, self-organized networks built up by many high-speed vehicles. Due to the unique features of VANETs, such as the high mobility of the nodes and the large scale of the network, security and privacy in these networks is particularly important as well as challenging. Lack of authenticated information shared in the network may lead to malicious attacks and service abuses, which could pose threat to drivers.

In order to preserve the sensitive information in the volatile open networks, we need to understand the requirement of deploying encryption and optimization techniques:

1. Integrated messages
2. Source authentication
3. Consistency in information
4. Independent of devices
5. Vehicle anonymity
6. Confidentiality and non-repudiation
7. Availability and scalability
8. Trusted architecture
9. Effective and efficient communication.
10. Robust design
11. Authorities should be able to reveal real identities (Table 1).

Table 1 Basic types of attacks in VANETs

Types	Purpose
Bogus information	To disturb public order and for illegal purposes
Denial of service	To jam channel or consume limited resources
Impersonate	Inserting malicious information
Eavesdropping	To collect specific information
Message suspension	Delays in collision reports and public access to information
Hardware tampering	To disturb the device functioning

4 Proposed Framework

The proposed framework starts with using the basic fundamental information that every vehicle possesses, i.e., its Government allotted vehicle identity. These IDs consist of alphabets and integers to register a vehicle. Proposed algorithm will use this ID for the initial process of generating a frame, this frame will be used for initial authentication or registering the vehicle to RSU and then will be used further for key generation. As the very first step of security authentication the base station will only accept the automobile with an authentication code matching the available one's in the data set, otherwise the vehicle will remain unconnected. Since the frame generation with the help of automobile number plates is totally a random process, this results in key generation to be even more random, robust, and secure.

4.1 Frame Generation

The frame generation process will provide an authentication code for any vehicle. Frame structure will contain the authentication number, followed by a randomly selected frame ID, which we can utilize for further sensitive information identification once the authentication is done, then a transmission mode activation number to define which type of communication is under way, e.g., 0 for V2V and 1 for V2I, and a frame number. These frames will be encrypted using Blowfish encryption algorithm for providing the very first layer of security.

4.2 Encryption of Data

Blowfish encryption is a block cipher with 64 bit size technique which contains a unique feature of having a variable-length key, which varies from 32 to 448 bits.

The encryption and decryption (reverse procedure) consists of two phases, first key initialization and then the cryptography. Key initialization will use P-array with 18 of 32 bit sub keys and a S-box which contains 256 entries. The process of generating the sub array keys also depends upon the user key, which enhances the complexity of the encryption process.

Later on in the encryption process, the algorithm will use updated keys instead of user key. The data initially gets divided into 32-bit input, and further as four eight-bit 4 set of data serving as the inputs to the S-boxes. The resultant data are added and XOR-ed to generate the final 32-bit resultant. Blowfish will add the security to its data by iterating the encryption function for 16 rounds. Decryption for Blowfish works in the similar but reverse order.

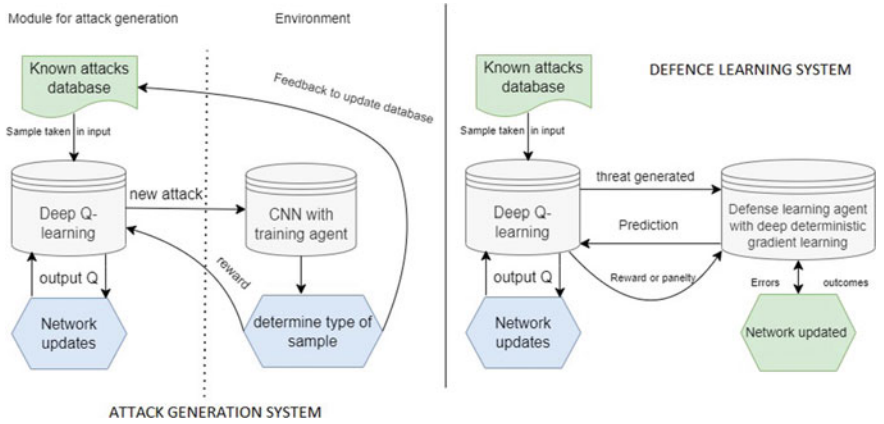


Fig. 2 Attack generation and defense leaning system

4.3 Adversarial Reinforcement Learning

Now that the data we intended to protect is being generated, next step is to add the reinforcement learning to our model. For this, the core of our model includes two secondary systems, (i) generation of attack (ii) adversarial reinforcement learning mechanism. The subsystem for attack generation uses convolutional neural networks to build up the possible attacks over the encrypted data, resulting in our sample attack system. The adversarial reinforcement learning includes two learning agents, a Q-learning-based training agent and a deep deterministic learning gradient defense agent.

The attack simulation system: This subsystem will train the Q-learning agent to generate attacks being model free, reinforcement learning agent it will be capable of determining the best attack possible over the data set created. The CNN environment will train over this examples (Fig. 2).

These set of examples will form our adversarial learning data set. The Q value in Q-learning will be updated based on (1) the state of the agent $= T$ and, (2) the action taken up by the agent $= A$. the relationship of update can be represented in Eq. (1). Where T' represents the next state and A' represents the next action, R is the reward for the state and α, δ are the attenuation factors.

Algorithm 1 Adversarial learning algorithm

```
1. initialize the state S with a value 0
2. Initialize the two agents DQL and DPG for i = 1 to max range do
    randonmly select one sample from dataset
end
for i = N do
    take features of current sample

    if features =0 then
        Break
    end
end
3. with a probability greater than zero, select a random a
or setup a with maximum available or repeated value
change the features as per sample and take a course of action
Compare with label

while matrix → 1 do
    if set the then
        r = -1
    end
    if set the then
        r = +1
    end
    else
        set the
    end
    r =0 return r
end
```

$$Q(T, A) \leftarrow Q(T, A) + \alpha\left[R + \delta\left(\max\ Q\left(T', A'\right)\right)Q(T, A)\right] \tag{1}$$

The deep Q-learning network consists of two neural structures for attack evaluation and target building. The attack generation system will be updated much faster as it will work over the delta from one value to another.

$$\Delta = R^i + \delta(\max Q\left(A', \theta^-\right) * \max\ Q\left(A', \theta\right)) \tag{2}$$

Theta and theta' are our parameters for target and evaluation network respectively to provide us delta updates.

The defense simulation system: The defense agent is then going to parallelly tackle the attacks simulations. The Q-learning attack agent with its targeted value will first train the very initial defense agent to form an arbitrary layer of security. With the help of deep deterministic learning policy, the defense agent will start to determine the attack over the encrypted data categorizing them into general noise variations or an actual attack over the system. To improve the efficiency of the framework the model needs to be continuously trained and updated for updating its defense mechanism. As the reinforcement learning fundamentals theorize, the learning agent will learn

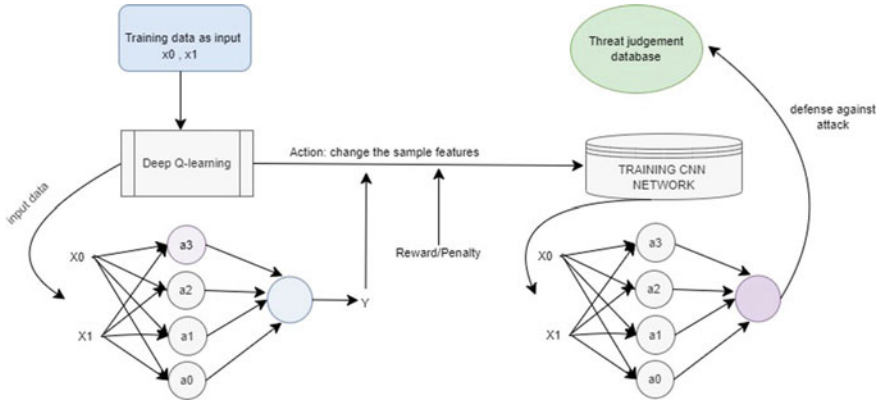


Fig. 3 Adversarial learning system

with its previous events as well as with the help of delta it finds while dealing with a completely new attack over the system (Fig. 3).

5 Simulation and Results

In this section we first represent the results that we obtained using our improvised vehicular ID generation and encryption process, based on performance parameters like, maximum number of keys an algorithm can store, the memory storage space it would require and encryption time. Table 2 represents the data analysis for proposed scheme and Fig. 5 represents the comparative encryption time study with other available encryption schemes (Fig. 4).

For case study we take samples from a data set to train the model and hence detect it against the Distributed Denial of Service (DDoS) where a selected data set is modified with all zeros to make it seem like a null request to RSU. We take initial weights as null and train the model to get the weights modified.

The CNN weights are W^i with X0 and X1 and our inputs derived from a data set. Before training the parameters of DQN are all zeros. Simple output relation can be defined as:

Table 2 Performance evaluation of proposed scheme for encryption

Parameters	Proposed scheme
Max-keys stored	None
Internal state memory required	Maximum of 50 KB
Battery capacity required	4–9%
Encryption time	2.11 s for 7 set of data input

Vehicle number	Unique VID	Binary	encrypted data (hex)
RJ19SA1884	51.453	110011.011	0x445bb2a71f95dd3f
PB01CA2009	22.216	10110.0011	0xa88a7cf1655bd79d
KA21CS1990	37.815	100101.11	0x19c6bbfec3de90ee
RJ14SE1456	54.546	110110.1	0x141356eb3142f729
AZ22DE3450	40.022	101000	0xf4a511035360aaa0
WB33GV2300	59.981	111011.111	0xf6794740bc0ad112
MY11VF9000	68.142	1000100	0x9e27df23e313fb99

Fig. 4 Data generation and encryption

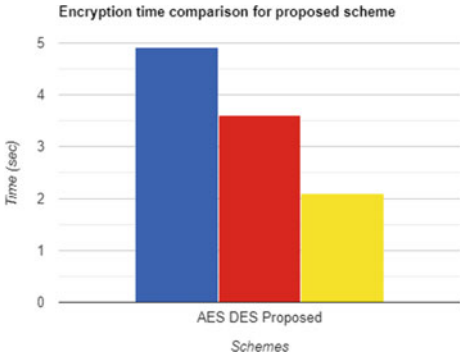


Fig. 5 Encryption time comparison

$$\text{output} = W^i * X^j + b \tag{3}$$

where i varies from 0 to 3 and $j = 0, 1$ as we divide the data into two equal parts. As we worked around with sample values, the weights get updated:

Weights	4 rounds	7 rounds
w0	0.041	0.059
w1	0.119	0.112
w2	0.043	0.045
w3	0.009	0.012

As it is quite evident that the weight are getting stable after every round and hence, the algorithm will learn to tackle up the attack over its course as well as delta is updated.

6 Conclusion

With our proposed framework we can conclude that with encryption and optimization we can construct a better, robust, and stable model. The proposed framework can be implemented over any environment and is flexible to train and learn new attacks

that can be known or unknown to the user. We have analyzed this framework over encryption time, maximum keys stored, internal state memory, and battery capacity as well as training the model to tackle DDoS attack. For future works we can train the model over multiple unknown environments where a far better delta can be observed.

References

Almomani I, Ahmed M, Kosmanos D, Alkhayer A, Maglaras L (2022) An efficient localization and avoidance method of jammers in vehicular ad hoc networks. IEEE Access 10:131640–131655. https://doi.org/10.1109/ACCESS.2022.3229623

Boucetta SI, Johanyák ZC (2022) Survey on security attacks in software defined VANETs. In: 2022 IEEE 16th international symposium on applied computational intelligence and informatics (SACI), Timisoara, Romania, pp 000185–000190. https://doi.org/10.1109/SACI55618.2022.9919595

Corona I, Giacinto G, Roli F (2013) Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Inf Sci 239:201–25. https://doi.org/10.1016/j.ins.2013.03.022, ISSN 0020-0255

Debicha I, Bauwens R, Debatty T, Dricot JM, Kenaza T, Mees W (2023) TAD: transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. Fut Gener Comput Syst 138:185–197. https://doi.org/10.1016/j.future.2022.08.011, ISSN 0167-739X

Funderburg LE, Ren H, Lee I-Y (2021) Pairing-free signatures with insider- attack resistance for vehicular ad-hoc networks (VANETs). IEEE Access 9:159587–159597. https://doi.org/10.1109/ACCESS.2021.3131189

Hu X, Tan W, Ma C (2020) Comment and improvement on two aggregate signature schemes for smart grid and VANET in the learning of network security. In: 2020 International conference on information science and education (ICISE-IE), Sanya, China, pp 338–341. https://doi.org/10.1109/ICISE51755.2020.00081

Kakkar D, Kaur G, Tirkey A (2022) Multilevel distributed energy efficient clustering protocol for relay node selection in three-tiered architecture. In: Tripathi SL, Ganguli S, Magradze T, Kumar A (eds)/Intelligent green technologies for sustainable smart cities. https://doi.org/10.1002/9781119816096.ch13

Kaur G, Kakkar D, Singh D (2022) Cluster optimization using meta- heuristic JAYA algorithm for secure VANETs. In: Rawat R, Bhardwaj P, Kaur U, Telang S, Chouhan M, Sankaran KS (eds) Autonomous vehicles, vol 2. https://doi.org/10.1002/9781394152636.ch9

Li J, Ma J, Miao Y, Yang F, Liu X, Choo K-KR (2021) Secure semantic-aware search over dynamic spatial data in VANETs. IEEE Trans Veh Technol 70(9):8912–8925. https://doi.org/10.1109/TVT.2021.3098177

Lin C, Huang X, He D (2022) EBCPA: efficient blockchain-based conditional privacy-preserving authentication for VANETs. IEEE Trans Depend Secure Comput. https://doi.org/10.1109/TDSC.2022.3164740

Wang J, Chen H, Sun Z (2020) Context-aware quantification for VANET security: a Markov chain-based scheme. IEEE Access 8:173618–173626. https://doi.org/10.1109/ACCESS.2020.3017557

Wei L, Cui J, Zhong H, Xu Y, Liu L (2021) Proven secure tree-based authenticated key agreement for securing V2V and V2I communications in VANETs. IEEE Trans Mob Comput 21(9):3280–3297. https://doi.org/10.1109/TMC.2021.3056712

Wei L, Cui J, Zhong H, Bolodurina I, Liu L (2023) A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-TA model for fog-based VANETs. IEEE Trans Depend Secure Comput 20(1):422–436. https://doi.org/10.1109/TDSC.2021.3135016

LoRa-IoT-Based Smart Weather Data Acquisition and Prediction for PV Plant Using Machine Learning

S. Ramalingam, M. Nagabushanam, H. S. Gururaja, and K. Baskaran

Abstract The use of renewable energy plants is in high demand worldwide. Reducing air pollution, greenhouse gas emissions from fossil fuels, economic development, and environmental protection are the primary reasons for the increased installation of renewable energy PV power plants but require efficient monitoring systems for better production. Commercial solar monitoring systems are relatively costly. Conventional monitoring devices are prohibitively expensive due to human error and the difficulty of PV plant monitoring. This research suggests a real-time IoT LoRa network that uses IoT technology in the communication layer and has the advantage of covering long distances and consuming little power. Our system has entirely collected the weather and climatic parameter data such as air temperature, PV temperature, and irradiation. The data were stored in the IoT cloud and visualized by IoT Blynk Application. An efficient data prediction plays a crucial role in PV plant monitoring. This work proposes an ensemble learning algorithm for efficient data prediction for PV plant applications. The proposed model's determination coefficient (R^2), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) values are 1, 0.010988, 0.10482, and 0.075023, respectively.

Keywords Long Range Radio (LoRa) · Internet of Things (IoT) · Machine learning · Photovoltaic plant · Ensemble algorithm · And support vector machine (SVM)

S. Ramalingam (✉)
Department of ECE, Sri Eshwar College of Engineering, Coimbatore, India
e-mail: ramalingam.s@sece.ac.in

M. Nagabushanam
Department of ECE, M. S. Ramaiah Institute of Technology, Belagavi, India

H. S. Gururaja
Department of ISE, B. M. S. College of Engineering, Bangalore, India
e-mail: gururajhs.ise@bmsce.ac.in

K. Baskaran
Department of EEE, Alagappa Chettiar Government College of Engineering and Technology, Karaikudi, India
e-mail: drbaskaran@gct.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_24

1 Introduction

In our daily lives, power is rapidly increasing. However, the use of renewable energy has increased. Environmental improvements have increased renewable energy usage. PV power plant is a mature technology due to its high energy density, supply security, and over 60% sustainability. In PV power plant surveillance, PV temperature and air temperature measurements must be performed, evaluated, and correlated with actual measured data to ensure its performance (Babu et al. 2018). Every week or month, various parameters such as temperature, power, voltage, and current must be checked manually to ensure system performance. IoT-enabled weather stations and machine learning-based forecasting systems implementing IoT-based irrigation systems will significantly change implementation and forecasting efforts (Beránek et al. 2018).

Consequently, it is critical to verify the timeline and requires much staffing. Due to this, the development of online monitoring systems is becoming increasingly important as the global use of distributed and renewable solar energy resources increases. Large power plants commonly use web-based monitoring and data collection frameworks. These frameworks are impractical for micro and small solar power plants due to their complex monitoring and relatively high costs. As a result, this work proposed an Internet of Things (IoT) based PV power plant monitoring system to overcome this limitation and promote the popularization of micro and micro power generation systems (Made and Singh 2017; Failed 2022; Ramalingam et al. 2020).

Solar cells are a new technology with promising future applications in renewable energy. Several advantages, such as an easy installation process, more reliability, no fuel costs, low maintenance costs, and no noise due to the absence of moving parts, have enabled rapid growth. Its goal is to modify, reduce system design costs, and make web-based tracking frameworks more practical and data processing more efficient. Internet of Things (IoT) cloud systems and hardware monitoring systems can be well developed by leveraging IoT concepts (Failed 2023). The proposed IoT system seeks to address the shortcomings of traditional online monitoring devices used in centralized factories. High-cost data loggers and sensor modules, proprietary software control and data storage, limited sensor connectivity, and maintenance limited to the factory manufacturer are the only links for long communication distances. It will be realized. LoRaWAN is ideal for IoT solutions for precision agriculture due to its long-range communication and low power consumption (Behera MK et al. 2018; Beránek et al. 2018). This paper uses LoRa (remote) technology to monitor photovoltaic power plants remotely and uses machine learning for data prediction and analysis. LoRa (protocol) technology is used in this proposed system to control photovoltaic cells to detect and analyze data. In addition, the Blynk application program transfers data to a web server for user control and system performance analysis.

This research work proposes an intelligent system for monitoring and scheduling precision irrigation using IoT, a low-power, long-range wireless sensor network (LoRa), and machine learning. The following are the significant contributions of this research. The main contributions of the present article are organized as follows: Our

proposed 'Smart weather monitoring system' provided better performance unlike conventional weather monitoring instruments is very complex.

- Long-distance weather data monitoring system. It comprises a sensor array with a long-range wireless communication module communicating climatic conditions to a weather station.
- The low cost LoRA-IoT based weather station to monitor the weather conditions.
- The best machine learning algorithm for predicting when to begin irrigation.
- Data storage and weather monitoring via IoT. The data from the weather station are sent to the IoT cloud for storage and monitoring.

2 Literature Survey

Chiandone et al. (2014) proposed WSN technology. Recently, various PV module panel control devices have been proposed and developed. The proposed technology is a fixed oscillator operating in the 2.4 GHz band that measures device temperature, illumination, current, voltage, and location. Its primary purpose is to investigate the causes of inefficiency based on the data. Froiz-Míguez et al. (2018) had proposed another monitoring and control system for photovoltaic power generation based on GSM technology. This system mainly uses GSM technology to record PV plant data such as power, current, and voltage and to report it periodically. Gagliarducci et al. (2007) proposed Zigbee and 4G communication-based tracking system. This system directly uploads the incoming data to the cloud platform through 4G technology. Kekre et al. (2017) proposed another low-cost IoT-based solar monitoring system has been. In this system, GPRS module and microcontroller are used to collect and send data to the cloud using Ethernet.

Lobaccaro et al. (2016) have been proposed one more system using NodeMCU as a microcontroller to collect data and using the graphical user interface to monitor the data. The first-level sensor network node of the system is used to monitor the current and voltage of all PV modules. The system's second-level sensor network node monitors ambient temperature and solar irradiance and uses Zigbee technology for data transmission. López-Vargas et al. (2019) proposed embedded systems network and sensor signal conditioning applied to decentralized photovoltaic plants based on ESP32 and ESP8266. In his proposed work, solar irradiance, wind speed, humidity, temperature, and DC generation are measured and transmitted to the local host through Wi-Fi.

Naveen Kumar et al. (2022) proposed a real-time monitoring system for a utility-scale photovoltaic power plant system in which precision time protocol (PTP) was utilized to integrate time-tamps of the nodes, which makes a wireless sensor network. It consists of sensors for computing rainfall index, ambient temperature, irradiance, and wind speed. Rusyd et al. (2016) proposed the evaluation of an inexpensive data collection system for the "PROPRE MA" system. This project combines Arduino with Raspberry Pi for monitoring several sensor parametric values like environment temperature. These files are on an SD card and can be retrieved through online. Wang

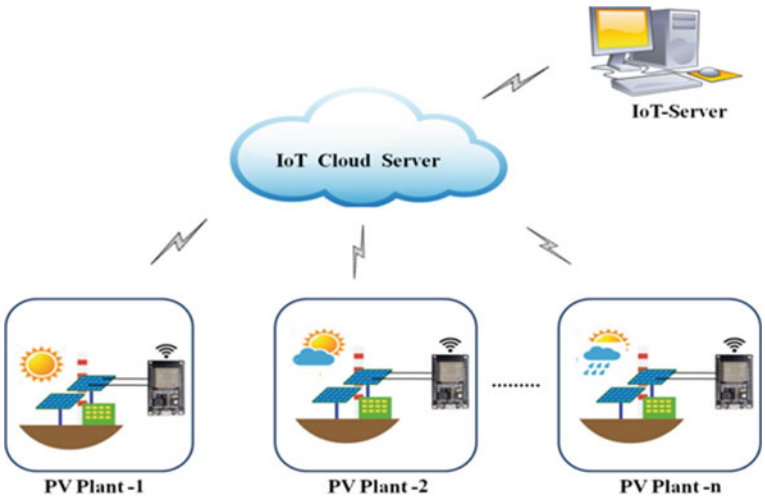


Fig. 1 Architecture of proposed LoRa-IoT based PV plant Monitoring system

et al. (2018) developed photovoltaic (PV) panel-performance modeling. This system uses MATLAB to model large dynamic PV systems, and these modelings are further used to investigate system performance (Fig. 1).

2.1 Comparison Between Existing Solutions

See Table 1.

Table 1 Comparison between existing solutions

Wireless technology	Operating frequency	Range	Data range	Power usage	Topology
Bluetooth	2.4 GHz	100 m	1 Mbps	10–500 mW	Point-to-point
ZigBee	2.4 GHz	300 m	250 kbps	1 mW	Mesh
Wi-Fi	2.4,3.6,5 and 60 GHz	150 m	6 Mbps–6.75 Gbps	1 W	Star
GSM	700–2600 MHz	10–30 km	35 Kbps–1 Gbps	1–5W	Star
Sigfox	868 MHz	20–50 km	100 bps	160 mW	Star
LoRa	433,868 MHz	10-30 km	0.3–5 kbps	100 mW	Star, Mesh

3 Proposed Methodology

The main goal of our proposed research is to use LoRa-based PV plant monitoring and data acquisition system using IoT. In our system, PV electrical and climatic parameter values are assembled to predict PV working conditions, and these data are transmitted to the gateways. The system's main components are Heltec Wi-Fi LoRa 32 (V2) and IoT Development Board powered by an ESP32 dual-core microcontroller (MCU). It has a micro-USB connector that connects to Wi-Fi, LoRa, Bluetooth, an onboard OLED display, and various sensor nodes. Various sensors used are pyranometer which converts the total received solar radiation into a measurable electrical signal, a temperature/humidity sensor which measures temperature and humidity by converting them into an electrical signal, and a PT 100 sensor which measures surface temperature by using a resistance change to denote the temperature value. A pyranometer is a sensor that uses the Seebeck effect or pyroelectric effect to convert the global solar radiation it receives into an electrical signal. Pyranometers measure the temperature difference between a transparent and a dark surface. Throughout the systems process, we can continuously measure and obtain the photovoltaic solar panel's current, voltage, solar radiation, temperature, and humidity. The generated data are processed and sent to the receiver system, where the Wi-Fi module integrated NodeMCU Esp8266 is connected to the LoRa module. The received packets are sent to the network application over the IoT through NodeMCU. After successfully collecting the data on the cloud platform, the ensemble data prediction algorithm has been proposed for efficient data prediction (Fig. 2).

The ESP 32 is used in the current weather monitoring system because it has these advantages over other generations of microcontrollers, typically requiring an

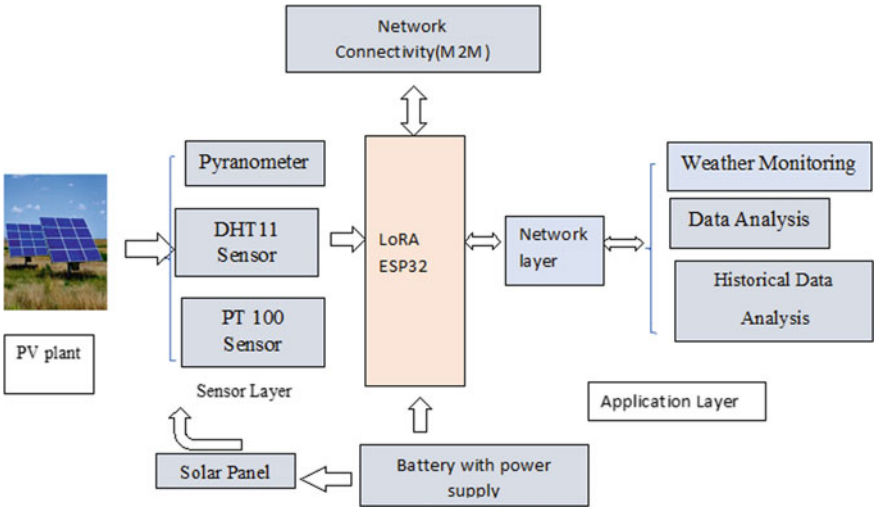


Fig. 2 Representation of photovoltaic power plants block diagram

additional Ethernet shield. The ESP32 has built-in Wi-Fi. As a result, the Esp32 is better suited for IoT projects. The Blynk app can visualize the predicted data. Blynk was used to collect all data. First, users must create an account. Blynk will then send the authentication token to the registered email address. After sending the authentication token, the user can use the application to create their data, such as graphs, display values, buttons, tables, etc. Finally, a remote base station has been developed to host the web server, and the user can access the web application through the remote server from any browser.

The Heltec Wi-Fi LoRa-esp32 (V2) system has an inbuilt Wi-Fi with a microcontroller that gets the power supply directly from the solar panels, and a 12 V battery is also connected as a backup. The function of sensor modules such as pyranometer, DHT11, and PT100 was controlled by the microcontroller. The PT100 temperature sensor measures real-time PV temperature from 0 to 100 °C by having a default resistance value of 100 Ω at 0 and 138.4 Ω at 100 °C, respectively. The DHT11 sensor is used to monitor the environmental temperature and relative humidity. The dome on a pyranometer acts as a radiation filter that blocks thermal radiation, ensuring the pyranometer is sensitive to solar radiation with wavelengths between roughly 0.3–3 * 10^{-6} m. Sensor data were collected and displayed on the OLED display. The ESP32 is installed externally near the PV module field site to minimize cable resistance losses and perform environmental variable measurements near the PV plant.

(b) The microcontroller is also responsible for transmitting data to the transmitter LoRa module. Then, the data are transferred from the transmitter LoRa to the receiver gateway through antennas. LoRa transmits over license-free megahertz radio frequency bands such as 169, 433, 868, and 915 MHz. In this system, 433 MHz frequency is used to transmit the data. (c) Using NodeMCU as a microcontroller, the data are transmitted to the IoT cloud platform via Wi-Fi. IoT developed an innovative solution for the proposed system because of its scalability. The IoT platform has the potential to support an increasing number of connected devices, users, application features, and analytics capabilities without any reduction in the quality of service. (d) Now the most significant role takes place. The proposed system considers error-prone environmental parameter data for monitoring PV plants, so machine learning (ML) algorithms should be applied to its dataset to obtain error-free values. An ensemble machine learning algorithm is applied in the proposed system for data prediction and analysis.

3.1 Data Prediction Using Proposed Ensemble Algorithm

Various ML algorithms such as linear regression, regression analysis, SVM, and ensemble algorithms are tested in the proposed system's data analysis and analytics part. In order to achieve better measurement performance, multiple model measurements are integrated so that the system provides the lowest error in four kinds of

measurement analysis algorithmic calculations. Among them, the ensemble algorithm has much efficient data and accurate predictions. Generally, the ensemble algorithmic technique generates multiple models and combines them to produce improved results. This ensemble technique mainly has three methods: boosting, bagging, and stacking. In the proposed system, boosting and bagging ensemble techniques were performed. The principle of bagging is based upon making the training data available to an iterative learning process. Each model of bagging learns the error caused by the preceding model independently in parallel by using different subsets of the training model and combining them to decide the model's average. As a result, it increases system stability and accuracy by decreasing variance and avoiding overfitting. Boosting works are completely different from the bagging technique. It approaches building a strong classifier from the group of weak classifiers. In boosting initially, a model is built from the training data, then the next model is built, which tries to make corrections in the initial model if any error is found. This procedure is carried out repeatedly in series, and models are added until the entire training data are predicted correctly or until the maximum number of models is added. Predicted data of tested algorithms are tabulated below in the Result and Analysis section.

$$\overline{Y} = \frac{1}{N} \sum_{i=1}^{N} f_i(X), \tag{1}$$

where each tree model fi is trained on bootstrap data fi. Ensemble algorithms combining multiple base student predictions are known as boosting algorithms (Wang et al. 2018). Boosting was first proposed to improve the performance of weak classifiers, which exceeded random guesses. On benchmark problems, boosting algorithms have forecasting results comparable to state-of-the-art support vector machine methods and have been used in many successful approaches to solving data mining challenges (Tiago et al. 2015). Boosting can be assumed as a gradient descent algorithm in feature space. The authors (Wolff et al. 2016) contribute to exploring connections between boosting algorithms and statistical inference frameworks. A general framework is proposed where the boosting algorithm is viewed as a steep descent minimization numerical optimization and opens the door to applications beyond classification. More information about this source text additional information about the translation required source text post comments side panel.

$$f(X) = \sum_{i=1}^{m} f_i(X). \tag{2}$$

4 Results and Discussion

The below result occurs when using a small unit sensors network, LoRa module, and microcontroller of the proposed system. Figure 3 depicts the outcome of the sensor network on the OLED display. The Blynk mobile app also displays data on Blynk's web page that users can analyze for monitoring the temperature (Celsius), humidity, and airflow (ppm), which can be accessed from anywhere. The experiments were carried out with an input dataset of 2340 samples that included three variables: room temperature, irradiance, and surface temperature. The data computation device transmits the acquired data to the LoRa network via the LoRa wireless communication connection and connects to the network via Wi-Fi (IEEE 802.11). A network is an intermediary between devices and computing systems. Transport and Application layer protocols help IoT end devices and cloud services communicate with each other. Device-to-Cloud data transmission occurs via the transport layer's User Datagram Protocol (UDP) (Table 2).

Message Queuing Telemetry Transport (MQTT) of the application layer takes care of sending responses and commands to the end devices. Through a web application, all sensed data collected by the network can be retrieved and viewed in real time or explored in the pipeline of cloud data servers. Web applications can be accessed from any monitor.

4.1 Data Prediction Analysis for Performance Evaluation

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(X_i - \widetilde{X_i})^2}, \tag{3}$$

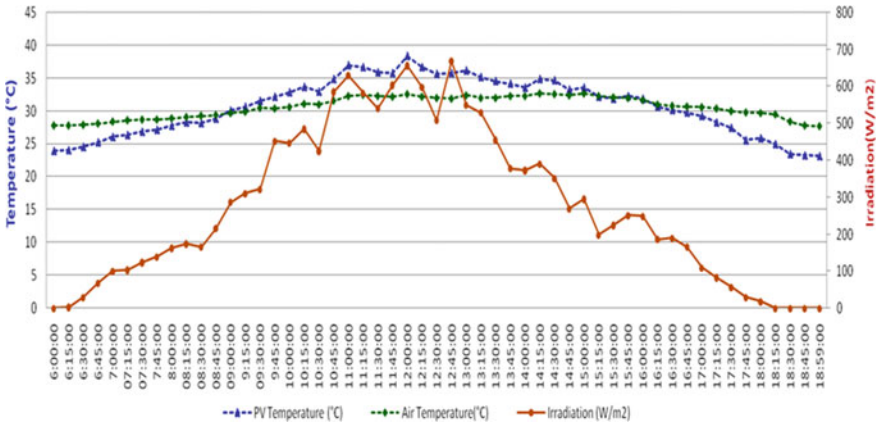


Fig. 3 Weather condition recorded on the lowest weather day—22/3/2022

Table 2 Recorded weather condition value

Date-time	Air temperature	Irradiation	Surface temperature
7/15/2019 06:00:00	28.7	0	24.5
7/15/2019 06:01:00	28.7	0	24.5
7/15/2019 06:02:00	28.7	0	24.5
7/15/2019 06:03:00	28.7	0	24.5
7/15/2019 06:04:00	28.7	0	24.5
7/15/2019 06:05:00	28.7	0	24.5
7/15/2019 06:06:00	28.8	0	24.5
7/15/2019 06:07:00	28.8	2.9	24.6
7/15/2019 06:08:00	28.8	2.9	24.6
7/15/2019 06:09:00	28.8	5.9	24.6
7/15/2019 06:10:00	28.8	5.9	24.7
7/15/2019 06:11:00	28.8	5.9	24.7

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{M}\left|X_i - \widetilde{X}_i\right|, \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(X_i - \tilde{X}_i)^2}{\sum_{i=1}^{n}(X_i - \tilde{X}_i)^2}, \tag{5}$$

where $\tilde{X}_i$ is the prediction and X_i is the real value (Tables 3, 4, and 5; Figs. 4 and 5).

Coefficient of determination (R2) of the proposed model is 85.75%, 98.87%, and 2% on the average, lowest, and highest weather days, respectively. In comparison

Table 3 Low power generation—weather day-air temperature

Algorithm		RMSE	R^2	MSE	MAE
Linear regression	Linear	1.5162	0.11	2.2989	1.3289
	RL	1.5199	0.11	2.31	1.32
	SWL	1.5162	0.11	2.2989	1.3289
Regression tree	FT	0.10839	1	0.011749	0.078421
	MT	0.14578	0.99	0.021252	0.10383
	CT	0.27932	0.97	0.078018	0.21422
SVM	Linear	1.6592	−0.06	2.7529	1.2889
	Quadratic	0.47325	0.91	0.22396	0.38981
	Cubic	1.6506	−0.05	2.7244	1.0119
Ensemble	Boosted	1.3067	0.34	1.7074	1.2988
	Bagged	0.10482	1	0.010988	0.075023

Table 4 Low power generation—weather day-solar irradiation

Algorithm		RMSE	R^2	MSE	MAE
LR	Linear	199.01	0.02	39.605	171.32
	RL	199.11	0.01	39.645	171.02
	SWL	199.01	0.02	39.605	171.32
Regression tree	FT	12.874	1	165.74	7.971
	MT	18.845	0.99	355.15	12.852
	CT	32.92	0.97	1083.8	24.655
SVM	Linear	201.54	−0.01	40.618	171.39
	Quadratic	78.584	0.85	6175.5	64.699
	Cubic	154.78	0.40	23.956	127.89
Ensemble	Boosted	24.946	0.98	622.31	15.558
	Bagged	12.373	1	153.09	7.8151

Table 5 Low power generation—weather day-surface temperature

Algorithm	Types	RMSE	R^2	MSE	MAE
LR	Linear	4.2133	0.00	17.752	3.6742
	RL	4.215	−0.00	17.766	3.6639
	SWL	4.2137	0.00	17.755	3.6814
Regression Tree	FT	0.22819	1	0.05207	0.15766
	MT	0.36862	0.99	0.13588	0.26707
	CT	0.78473	0.97	0.61579	0.64723
SVM	Linear	4.2699	−0.03	18.232	3.6244
	Quadratic	1.0188	0.94	1.038	0.81251
	Cubic	2.5382	0.64	6.4423	1.9443
Ensemble	Boosted	1.3664	0.89	1.8669	1.308
	Bagged	0.19821	1	0.039288	0.13962

with LR, RT, and SVM models, it gives extremely accurate values. Figure 6 portrays a temperature comparison for various RMSE-based algorithms. The lowest percent RMSEs suggested by the model for the average, lowest, and highest forecast days are 4.21%, 0.22%, and 4.26%, respectively. The r-set algorithm is designed to improve the random forest algorithm's performance. This means it has a minimum error and a low variance and is optimum for forecasting.

The proposed ensemble algorithm method demonstrates that various learning classifiers perform better based on performance quality factors. As shown in Ramalingam et al. (2020), the proposed ensemble model achieved the lowest surface temperature RMSE ratio of 0.19 percent compared to other models on three different weather days. The lowest error values demonstrate the highest forecasting accuracy. So the

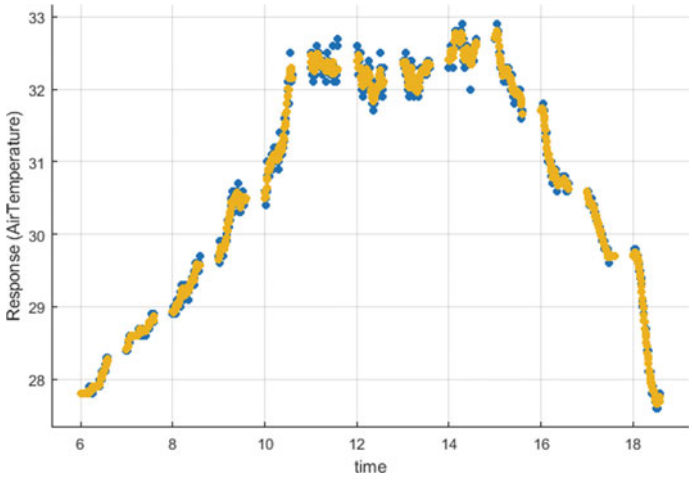


Fig. 4 Prediction analysis for air temperature

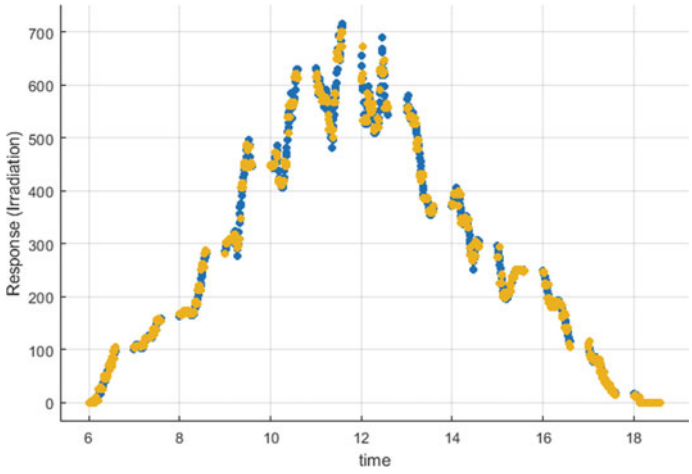


Fig. 5 Proposed work data prediction model for irradiation

suggested system uses ensemble algorithms over existing algorithms. The ensemble algorithm optimizes the RF parameters by minimizing the RMSE.

This improves the proposed ensemble algorithm's prediction results. It has a low error rate and is ideal for forecasting. As a result, the proposed method yields the lowest RMSE and MAE values.

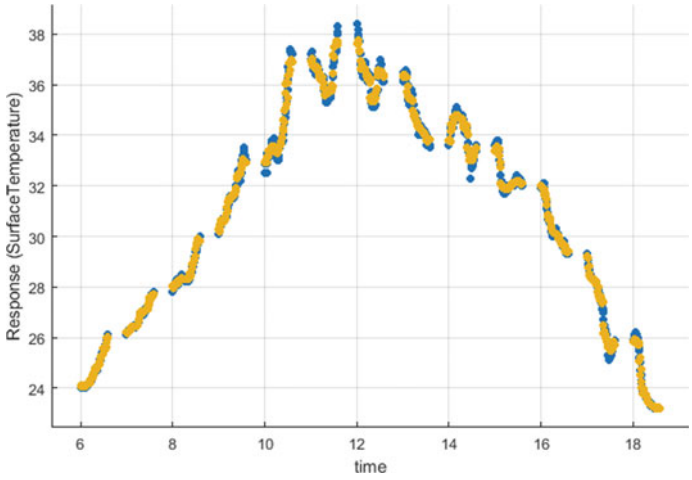


Fig. 6 Prediction analysis for surface temperature

5 Conclusion

In this model, an IoT-based system was developed for real-time monitoring of PV system performance. This system stands out because it uses open traffic to measure all essential meteorological values via wireless instantaneous network connectivity with LoRa as the file transfer module. Additionally, it moves data locally and online. Additionally, it moves data locally and online from PV plant and calculating the PV yield. Moreover, explanations of the effectiveness of the proposed system may be based on real-world examples. By coupling the proposed system with mechanical engineering techniques, electrical detection based on gas data and automatic fault detection will optimize electrical processes and improve photovoltaic systems.

The monitoring system was tested in an outdoor environment, and the sensor data were successfully updated. Data will be analyzed in various ways and may be shared with other people or users. The project could be used to monitor developing cities and industrial areas, particularly for pollution monitoring. The system could also offer authorities a low-cost and efficient way to protect public health from pollution. It is also suitable for future continuous environmental monitoring.

References

Babu RL, Rambabu D, Naidu AR et al (2018) IoT-enabled solar power monitoring system. Int J Eng Technol 7(3, 12):526–530. https://doi.org/10.14419/ijet.v7i3.12.16172

Behera MK, Majumder et al (2018) Solar photovoltaic power forecasting using optimized modified extreme learning machine technique. Eng Sci Technol Int J 21:428–438. https://doi.org/10.1016/j.jestch.2018.04.013

Beránek V, Olsan T, Libra M et al (2018) New monitoring system for photovoltaic power plants, management. Energies 11:2495. https://doi.org/10.3390/en11102495

Chiandone M, Sulligoi G et al (2014) Hierarchical voltage regulation of transmission systems with renewable power plants: an overview of the Italian case. In: Proceedings of the 3rd renewable power generation conference, Naples, Italy, pp 24–25. https://doi.org/10.1049/cp.2014.0861

Dhanasekaran S, Gomathi P, Maximus AR, Krishnan T, Kannan B (2022) Solar tree based smart city street light control system using IoT BLYNK platform. In: 2022 International conference on automation, computing and renewable systems (ICACRS), Pudukkottai, India, pp 284–290. https://doi.org/10.1109/ICACRS55517.2022.10029107

Froiz-Míguez I, Fernández-Caramés TM et al (2018) Implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes. Sensors 18:2660. https://doi.org/10.3390/s18082660

Gagliarducci M et al (2007) GSM-based monitoring and control of photovoltaic power generation. Meas J Int Meas Confed 40(3):314–321. https://doi.org/10.1016/j.measurement.2006.05.018

Kekre A, Gawler SK et al (2017) Solar photovoltaic remote monitoring system using IoT. In: International conference on recent innovations in signal processing and embedded systems (RISE), IEEE, pp 619–623. https://doi.org/10.1109/RISE.2017.8378227

Kumar BS, Ramalingam S, Divya V, Amruthavarshini S, Dhivyashree S (2023) LoRa—IoT based industrial automation motor speed control monitoring system. In: 2023 International conference on intelligent data communication technologies and Internet of Things (IDCIoT), Bengaluru, India, pp 11–15. https://doi.org/10.1109/IDCIoT56793.2023.10053525

Lobaccaro G, Carlucci et al (2016) A review of systems and technologies for smart homes and smart grids. Energies 9:348. https://doi.org/10.3390/en9050348

López-Vargas A, Fuentes M et al (2019) IoT application for real-time monitoring of solar home systems based on arduino™ with 3G connectivity. IEEE Sens J 19(2):679–691. https://doi.org/10.1109/JSEN.2018.2876635

Made SR, Singh S (2017) Monitoring system for photovoltaic plants: a review. Renew Sustain Energy Rev 67:1180–1207. https://doi.org/10.1016/j.rser.2016.09.088

Naveen Kumar GN, Navalgund et al (2022) LoRa based photovoltaic solar energy monitoring system. Int J Eng Res Technol (IJERT) 11(3). https://doi.org/10.17577/IJERTV11IS030147

Ramalingam et al (2020) An efficient data prediction model using hybrid Harris Hawk optimization with random forest algorithm in wireless sensor network. J Intell Fuzzy Syst. https://doi.org/10.3233/JIFS-201921

Rusydi MI et al (2016) Real-time measurement of grid connected solar panels based on wireless sensor network. In: International conference on sustainable energy engineering and application, pp 95–99. https://doi.org/10.1109/ICSEEA.2016.7873574

Tiago M et al (2015) An Intelligent power outlet system for the smart home of the Internet of Things. Int J Distrib Sens Netw 1:1–11. https://doi.org/10.1155/2015/214805

Wang F, Zhen Z, Liu C, Mi Z et al (2018) Image phase shift invariance based cloud motion displacement vector calculation method for ultra-short-term solar P.V. power forecasting. Energy Convers 157:123–135. https://doi.org/10.1016/j.enconman.2017.11.080

Wolff B, Lorenz E, Kramer O (2016) Statistical learning for short-term photovoltaic power predictions. In: Lässig J, Kersting K, Morik K (eds) Computational sustainability. Studies in computational intelligence, vol 645. Springer, Cham. https://doi.org/10.1007/978-3-319-31858-5_3

Entity Perception Using Remotely Piloted Aerial Vehicle

J. Jeba Emilyn, S. Sri Chandrika, T. Susma, S. Vinisha, and S. V. Yesvantini

Abstract Remotely piloted aerial vehicle (RPAV), popularly called a drone, unmanned aerial vehicle (UAV) is an airborne machine, or an aircraft operated remotely by means of a human operator or autonomously by means of an onboard laptop. The usage of unmanned aerial motors is developing hastily throughout many civil utility domain names, such as actual-time ascertainment offering wireless insurance, remote sensing, object monitoring, object detection, exploration and extrication, transport of goods, security and surveillance, precision agriculture and civil infrastructure inspection. Remotely piloted aerial vehicle (RPAV) is swiftly gaining recognition. Due to their many desired functions, UAVs could soon be an inevitable part of many fields. This hooked up use of UAVs has given upward thrust to the concept of multi-UAV systems, normally referred to as UAV swarms, wherein several UAVs starting from two to hundreds must collaborate with every different and carry out a not unusual, shared challenge or tasks. As a result, it is vital to have a green inter-UAV collision avoidance mechanism. Object detection is one of the maximum full-size elements of pc imaginative and prescient. It has to do with image popularity in addition to digital image partition. Entity detection targets things in the pictures or movies, permitting them to be trailed and tailed, not like image popularity, which

J. J. Emilyn
Department of Information Technology, Sona College of Technology, Tamil Nadu, Salem 636005, India
e-mail: jebaemilynj@sonatech.ac.in

S. S. Chandrika (✉) · T. Susma · S. Vinisha · S. V. Yesvantini
B. Tech. Information Technology, Sona College of Technology, Tamil Nadu, Salem 636005, India
e-mail: srichandrika.19it@sonatech.ac.in

T. Susma
e-mail: susma.19it@sonatech.ac.in

S. Vinisha
e-mail: vinisha.19it@sonatech.ac.in

S. V. Yesvantini
e-mail: yasvantini.19it@sonatech.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_25

produces a classification label for each identified entity, and image segmentation, which results in creating a pixel degree expertise of gadgets in the image.

Keywords Entity perception of RPAV images · Reinforcement learning · RPAV applications

1 Introduction

Remotely piloted aerial vehicle (RPAV), commonly known as a drone, is an aircraft operated ever by a mortal driver or autonomously by an onboard computer. The use of unmanned aerial vehicles is growing fleetly across numerous civil operation disciplines, including actual-time ascertainment, furnishing cellular content, remote sensing, object shadowing, object discovery, hunt and deliverance, delivery of appurtenances, security and inspection and civil structure examination. Unmanned aerial vehicles (UAV) are fleetly gaining fashionability. Due to their numerous asked features, soon, UAVs will be an ineluctable part of numerous fields. This mounted use of UAVs has given rise to the idea of multi-UAV systems, generally known as UAV masses, where several UAVs ranging from two to hundred must unite with each other and perform a commonly participated task or tasks. Therefore, it's essential to have an effective inter-UAV collision avoidance medium. Object discovery is one of the most significant aspects of computer vision. It has to do with picture recognition as well as picture segmentation. Object discovery locates effects inside pictures or vids, allowing them to be trailed and tallied, unlike image recognition, which produces a bracket marker for each honored entity, and digital image partition, which generates a pixel position knowing of features in the picture. Entity discovery from an upstanding view can be used to check the condition of target installation itself and find the threat factor around the installation to achieve the purpose of replacing homemade line examination. Because the flight of drones isn't affected by terrain, etc., and it's further timesaving and labor-saving than homemade examination, it has developed fleetly in recent times. It is a low-cost platform to collect spatiotemporal data for broad operations that are involved in numerous aspects of diurnal life, searching, intelligent monitoring, logistics transportation, path planning and navigation. The main idea of object discovery using drones is to track and describe objects, discovery of small images by UAV equipped with vision ways and path planning. The objects discovery module, fine tuning the pre-trained models for object discovery of the Vis-drone dataset and enhancing the performance by reducing the error rates using the residual blocks.

2 Related Works

Nousi et al. (2019), proposed a real-time object detection from the video footage from the vantage point of embedded remotely piloted aircraft vehicle. The proposed model was based on SSD and YOLO. The methods provide a correlation-based tracking algorithm. They evaluated based on another single-class problem. Real-time object tracking was proposed with target tracking of 2D visual using correlation filter.

Pham et al. (2017), proposed a method for monitoring UAV groups for covering and tracking wildfire. The model proposed along with collision prevention and protection. The model proposed a distributed framework for the continuous monitoring of wildfire in an open space. Huang et al. (2022) (Saravanan et al. 2019), proposed a dissemination of data in an aircraft vehicle by standardized optimization and bringing the effective action of dueling double deep Q network (double DDQN) which gives higher-level performance of the algorithm.

Jana and Biswas (Saravanan et al. 2019), proposed a technique for enhancing computational and processing speed. The technique successfully locates the objects in the video recordings and helps in detection and classification of items ranging from various instances of single objects to various instances of various objects; the proposed method uses both YOLO ver2 and YOLO 9000.

Baojun et al. (2019), suggested a top-to-down and bottom-to-up feature pyramid network (TDBU-FPN), which blends the anchor creation at various aspect ratios with multi-scale feature representation. Pattern analysis, statistical modeling and other metrics are used to assess the proposed approach. Tang et al. (2017), introduced the traditional methods for object detection and explained how they relate to and differ from the deep learning methods for object recognition. The study concentrates on the framework design and the models' underlying principles in the description of the approaches.

Hasan et al. (Masita et al. 2020), researched convolutional object detection based on machine learning. This entailed creating features to describe the properties of the objects, then integrating those features with classifiers using deep learning (DL) applications. Saravanan et al. (2019), proposed a feature extraction approach which is neither vector nor scalar called hierarchical multi-scale symbolic dynamic entropy is developed, which is used to test the changeable or dynamic characteristic of the vibration data at different layers and scales.

Mohanraj et al. (2020), presented a deep learning hybrid model, which is presented to foretell and target the vaccination coverage in the lower immunization areas. Saraswathi et al. (2021), proposed a systematic many-sided definition correlation-based online social guidance system. This model uses convolutional neural network (CNN) to train and foretell all topics the user is interested in.

Mane and Mangale (2018), proposed robust detection of objects in occlusion environment. This approach uses detection of objects in motion with the help of novel CNN-based entity tracking. Faza and Darma (2020), proposed the usage of state-of-the-art of object detection to control a drone. It uses SSD to detect objects. Fradi et al. (2018), proposed a color-based detection framework to achieve the navigation

functionality of drones. This method calculates on videos captured by the aircraft vehicle.

Hossain and Lee (2019), proposed the efficient methodology on deep leaning framework which uses hypothesis tracking method and GPU-based algorithm for demonstration of mini multi-rotor drones. Saetchnikov et al. (2021), proposed a comparative study on the efficacy of various deep neural networks for identification of the entities with familiar patterns. They used YOLO ver3 for better accuracy rate along with RCNN, faster RCNN and SSD architecture. Bindemann et al. (2017), suggested human detection from aerial vehicle captured pictures even in dismal surroundings by optimizing sub-optimal quality of the footage.

3 Proposed Methodology

The proffered methodology consists of the following phases of image processing, point birth, image bracket, object localization, object discovery, object segmentation and object labeling. The image preprocessing is an essential step that is used to clean image data before the image is used in the computer vision model. The point birth in image processing is the process to prize or transfigure the raw data into numerical features. The point birth process increases the delicacy of learned models by rooting the features from the input data (image).

Image classification is the process of predicting the category of specific object in each visual input image.

- Input: Image.
- Output: Image with label for each object.

Object localization alludes to relating the position of one or further realities in an input and drawing pullulating box around the particular object extent. Object discovery combines these dual tasks and localizes and classifies one or more entities in an image. It will detect the presence of entities in an image with the bounding box and types or classes of the located entities in an image. Object segmentation is also called "semantic segmentation" or "object case segmentation" in which the cases of honored objects are indicated by pressing the specific pixels of the object, rather of a coarse bounding box.

$$\text{Objectness}_{\text{score}}(\text{IoU}) = \{\text{Positive} \rightarrow \text{IoU} > 0.7\}$$
$$\text{Objectness}_{\text{score}}(\text{IoU}) = \{\text{Positive} \rightarrow 0.5 < \text{IoU} <= 0.7\}$$
$$\text{Objectness}_{\text{score}}(\text{IoU}) = \{\text{Negative} \rightarrow \text{IoU} < 0.3\}$$
$$\text{Objectness}_{\text{score}}(\text{IoU}) = \{\text{Not Negative / Positive} \rightarrow 0.3 <= \text{IoU} <= 0.5\}$$

4 Architecture of Object Detection

The design of the object detection from unmanned aerial vehicle in Fig. 1 shows a region-based two-stage (faster RCNN) architecture.

4.1 Object Detection Multi-class Classification

Multi-class classification of the object detection involves in extracting features from the input image using CONV layers which uses RPN (Region Proposal Network) for each spatial location it generates object classification and bound boxing regressor from the projected region proposals in the feature maps ROI pooling is done and the similar regions are classified according to multi-class classification and the object will be detected and classified.

4.2 Objective Function

The algorithm is region-based on two-stage (faster RCNN) architecture. Initially the model gets the region of interest by a region proposal network, then it combines the convolutional layers created with the various filters and the proposal of regions with bounding boxes classifies the object to its corresponding class. Although it is a two-stage process, this increases the stability of the model to find the class of the object efficiently. The resizing phase of the model creates a different version of the particular object (region proposed).

5 Test Result

5.1 Experimental Settings

The object detection algorithm uses modified RESNET architecture and faster RCNN approach. The proposed system was implemented in Python programming language and used Anaconda as running platform/environment for the implementation. Pandas Python library was used for data manipulation and analysis. It stores and manipulates tabular data. Scikit-learn is a machine learning library used for the preprocessing functions and efficient tools for the machine learning models. Numpy tool is used for the efficient computing of data. The Vis drone dataset was used for object detection from UAV view. The pretrained coco dataset model was used.

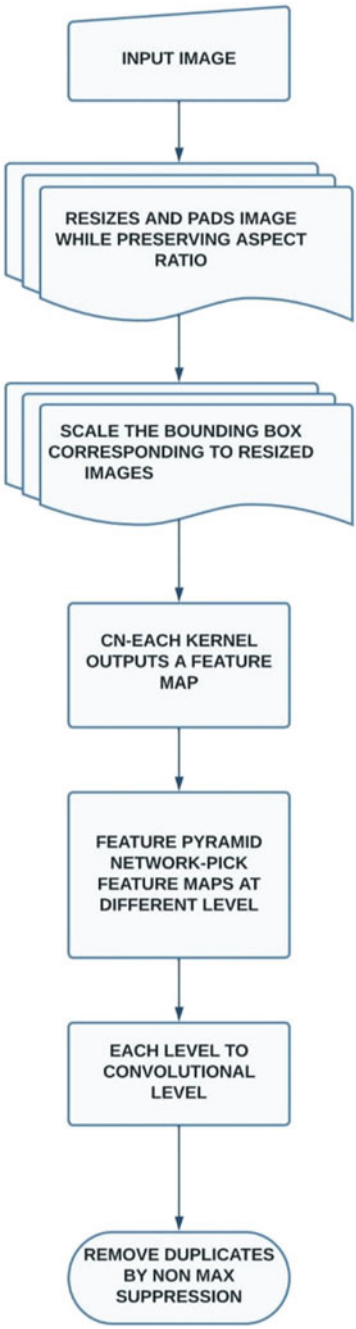


Fig. 1 Architecture of object detection

Table 1 Table captions should be placed above the tables

Algorithm	Accuracy	Mean average precision (mAP)
YOLO	Moderate	0.505126598721
SSD	Low	0.312987654321
Retina Net	Low	0.391234567891
RESNET	High	0.84438122449
Proposed methodoloy (faster RCNN)	High	0.878706131123

5.2 Performance Evaluation Metrics

See Table 1.

5.3 Q-Learning Approaches

Figure 2 shows the Q-learning-based approach to indicate the relationship between the episode of the current state of the agent and the reward based on the finding the best course of action by selecting the one with the greater Q-value in RESNET model. The outcome of the demonstration will be based on comparing various algorithms with our proposed method by mainly focusing on the algorithm with high mean average precision. Q-learning-based approach can generally be solved by using one of the most popular and efficient equation known as Bellman equation, where the s denotes the current state of the agent that is present in the environment, s' denotes the transition of state from the action (a), V represents the state which helps to find the trajectory pathway and finally the R denotes the reward which the agent gets after the performance of certain action. We can see that in Fig. 3, we can reach the high reward between the episodes 0–10 but after that the reward graph gradually decreases only providing high reward for the trained data. The classifier we used improved the accuracy in the Vis drone and COCO demo dataset. Every single data has gone under proper training. Our proposed modified RESNET and faster RCNN renders a higher percent in terms of mean average precision, thereby increasing the accuracy, while out proposed approach of implementing the modified RESNET and faster RCNN provided the consistency in the reward graph between those episodes where the mean average precision actually increased from 0.84438122449 to 0.878706131123. Hence it identifies the objects faster and with higher accuracy rate.

Figure 3 shows object detection in low illumination, a random street. The design of the object detection from unmanned aircraft vehicle in figure follows a region-based two-stage (faster RCNN) architecture. Initially the model gets the region of interest by a region proposal network then it combines the convolution layers created with the various filters and the proposal of regions with bounding boxes classifies the object to its corresponding class. Although it's a two-stage process, this increases the stability of the model to find the class of the object efficiently.

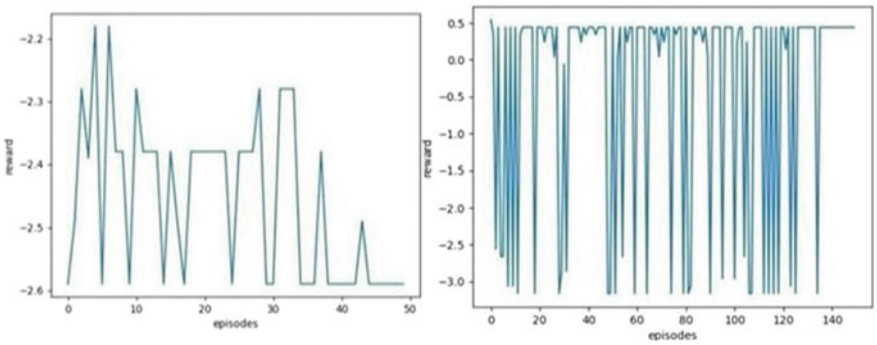


Fig. 2 Q-learning approach reward and model-based reward graphs

Fig. 3 Sample output image of entity perception that detected persons, animals, boats and detected the objects in the random street

The resizing phase of the model creates a different version of the object. It is the anticipated return for an agent who starts from a state and takes an action, then acts according to policy for the rest of his life. Since a state may have multiple acts, it will have multiple Q values. The Q-table provided the following syntax:

$$(\text{State}) - > (\text{action}, \text{reward})$$

6 Discussion

Our proposed methodology picks out the target features by using RPN. The inessential and less prominent features are not taken in accordance since we are using RPN which provides the reduction in the feature spacing. With both high and low illumination, our method can detect objects and can also provide faster and more accurate output. We achieved a mean average precision of 0.878706131123. Most of the object detection methods that are incorporated using various algorithms either provide accuracy of faster detection rate but our work was purposely done based on dual prominent factors which results in better accuracy and faster detection of entity. The accuracy rate is solely based on the reward graph, modified RESNET base architecture and faster RCNN. Presently, there are few piffling hindrances of our work that the gained delicacy is varied by 0.4 from the benchmark of the prerecorded performances in Q-learning graph of RESNET. Our plan is to probe these furthermore by merging other objects discovery of algorithms for a bracket. Either we can extend this further by merging two completely different algorithms or merging similar algorithms.

7 Conclusion

The goal of the system was to automate the UAV applications. The proposed method for the entity perception using UAV learns in a better rate than the previous approach. The model-based reinforcement learning used reduces the overhead of the traditional Q-table, by avoiding collision and can be used for large number of UAV. The state space, initial, final coordinate and number of episodes can be given by the user. So, it learns based on the final coordinates specified along the slope angle. The proposed approach finds the highest reward at the earliest than the previous methods. The method for object detection used RESNET as base architecture and the faster RCNN approach to classify objects. Thus, the different version fed into the convolution layers helps the learner to classify the objects on low illumination, darkness and extra brightness as well. Hence the mean average precision obtained shows the modified RESNET architecture was improved by including the different version and angle of objects at the top of the architecture.

References

Baojun Z, Boya Z, Linbo T (2019) Multi-scale object detection by top- down and bottom- up feature pyramid network. J Syst Eng

Bindemann M, Fysh MC, Sage SS, Douglas K, Tummon HM (2017) Person identification from aerial footage by a remote-controlled drone. Sci Rep 7(1):13629

Faza A, Darma S (2020) Implementation of single shot detector for object finding in drone platform. J Phys Conf Ser 1528, Article 012005. https://doi.org/10.1088/1742-6596/1528/1/012005

Fradi H, Bracco L, Canino F, Dugelay JL (2018) Autonomous person detection and tracking framework using unmanned aerial vehicles (UAVs). In: Proceedings of the 2018 26th European signal processing conference (EUSIPCO), pp 1047–1051. https://doi.org/10.23919/EUSIPCO.2018.8553010

Hossain S, Lee D (2019) Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. Multidisc Digit Publish Inst 19(15):3371. https://doi.org/10.3390/s19153371

Huang F, Li G, Tian S, Chen J, Fan G, Chang J (2022) Safe navigation for UAV—enabled data dissemination by deep reinforcement learning in unknown environments. China Commun 19(1):202–217

Jana AP, Biswas A (2018) YOLO based detection and classification of objects in video records. In: International conference on recent trends in electronics, information & communication technology (RTEICT)

Mane S, Mangale S (2018) Moving object detection and tracking using convolutional neural networks. In: Second international conference on intelligent computing and control systems (ICICCS), pp 1809–1813. https://doi.org/10.1109/ICCONS.2018.8662921

Masita KL, Hasan AN, Shongwe T (2020) Deep learning in object detection: a review. In: International conference on artificial intelligence, big data, computing and data communication systems

Mohanraj G, Mohanraj V, Senthilkumar J, Suresh Y (2020) A hybrid deep learning model for predicting and targeting the less immunized area to improve children vaccination rate. Intell Data Anal 24(6):1385–1402

Nousi P, Mademlis I, Karakostas I (2019) Embedded UAV real-time visual object detection and tracking. In: IEEE international conference on real- time computing and robotics (RCAR)

Pham HX, La HM, Feil-Seifer D (2017) Control framework of UAV for dynamic wildfire tracking. In: IEEE international workshop on intelligent robots and systems (IROS)

Saetchnikov IV, Tcherniavskaia EA, Skakun VV (2021) Object detection for unmanned aerial vehicle camera via convolutional neural networks. IEEE J Miniaturization Air Space Syst 2(2):98–103. https://doi.org/10.1109/JMASS.2020.3040976

Saraswathi K, Mohanraj V, Suresh Y, Senthilkumar J (2021) A hybrid multi feature semantic similarity based online social recommendation system using CNN. Int J Uncertainty Fuzziess Knowl Based Syst

Saravanan B, Mohanraj V, SenthilKumar J (2019) A fuzzy entropy technique for dimensionality reduction in recommender systems using deep learning. Soft Comput Springer 23(8):2575–2583 (SCI—Journal, Impact Factor: 2.367)

Tang C, Feng Y-S, Yang X, Zheng C, Zhou Y (2017) The object detection based on deep learning. In: 4th International conference on information science and control engineering (ICISCE)

RF-MEMS SPDT Capacitive Switch: Accelerating the Performance in B5G Applications

Raj Kumari and Mahesh Angira

Abstract The wideband high-frequency components and hardware are the need of future communication. The switch is one of the basic integral components found in every communication system and plays a vital role in making the communication process simple by adding multi-functionality. This work reports a single pole double throw (SPDT) switch for B5G applications. The design incorporates low-loss micro-electromechanical single pole single throw (SPST) capacitive shunt switches separated by a quarter wavelength distance over a coplanar transmission line. The simulated SPDT design showed minimum insertion losses of −0.2 dB at 26 GHz and maximum isolation of −65 dB at 20 GHz. The RF response of the switch is found satisfactory over the frequency range 15–29 GHz resulting in a bandwidth of 14 GHz. For bandwidth calculation insertion loss, below 1 dB and isolation above 20 dB are computed as per the specifications of B5G.

Keywords Capacitive switch · FEM · RF-MEMS · SPDT · SPST

1 Introduction

Integrating a large number of cutting-edge wireless technologies into future B5G (beyond the fifth generation) wireless systems is an unswerving trend (Huo et al. 2018, 2019). This needs devices and components which are easy to integrate and consumes minimum power in order to enhance portability. The micro-electromechanical systems (MEMS) technology is one of the key fields that follow

R. Kumari (✉) · M. Angira
Department of Electronics and Communication Engineering, NIT Hamirpur, Hamirpur, Himachal Pradesh 177005, India
e-mail: kumari09raj@gmail.com

M. Angira
e-mail: mahesh_angira@nith.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_26

this trend and has been employed in many 5G/B5G applications (Iannacci and Tagliapietra 2022). The MEMS devices do not consume any DC power and are compatible with basic integration technologies. These (sensors, actuators, and switches) are utilized in various fields such as communication, electronics, mechanics, biomedical and automotive (Tilmans et al. 2003). The highly fast technology with low power has become an integral part of everyday life and lead to switching over high-frequency technologies (B5G and 6G) (Rao et al. 2020). RF-MEMS switches perform very well at higher frequencies. The traditional switching devices evolved around solid state devices such as FET, PIN diode and coaxial devices, which require considerable power and area for biasing (Tang et al. 2005; Kumari and Angira 2022a, b; Angira and Rangra 2015). Conventional switches provide ample functionality at the cost of transmission loss and narrow bandwidth. The RF-MEMS technology is one that addresses many of the limitations of conventional switches. The MEMS switches are most preferred and widely used at mmWave frequencies because of their near-to-zero power consumption, low losses and high isolation (Rebeiz and Muldavin 2001; Chae et al. 2020; Lysenko et al. 2018). They can function across a large bandwidth, resulting in fewer pieces of gear being needed. For tunable and reconfigurable mmWave systems, much recent research recommends RF-MEMS switches (Iannacci 2017; Angira et al. 2019; Angira and Rangra 2016; Huo et al. 2017). Depending on the mode of actuation, the device's geometric configuration and the type of contact established between the transmission line and the switching element, MEMS switches can be categorized into various topologies. The most common types are capacitive shunt switches and ohmic switches (Rebeiz and Muldavin 2001; Bansal et al. 2021). Generally, shunt capacitive switches are preferable for low-power and low-loss applications at higher frequencies (>10 GHz). However, structural modifications and optimizations are always needed to fit a particular application. This work is mainly focused on optimizing the switch performance for B5G applications and then designing a single pole double throw (SPDT) switch. SPDT is an integral part of numerous millimetre and microwave transceiver systems and front ends, used to route radio frequency (RF) signals through various amplifiers, filters block for wideband coverage, selecting different transmitter, receiver channels and building phase shifting circuits for phase arrays (Pacheco et al. 2001; Malczewski and Pillans 2004). This paper discusses the design and finite element method (FEM) simulation of a broadband SPDT capacitive shunt switch. The SPDT device configuration utilizes a shunt-shunt capacitive switch topology for broadband operations covering 15–29 GHz. The isolation more than 20 dB and transmission loss less than 1 dB has been achieved. With a brief introduction in this section, Sect. 2 discusses about the structure and working of the proposed device. Section 3 contains the discussion on simulation results and thereafter, the conclusion is presented in Sect. 4.

2 Switch Circuit Design and Working

The SPDT design consists of a CPW tee with outputs at coplanar arm through RF-MEMS capacitive shunt switches and input on the third arm. This design is shown in Fig. 1.

The switches are placed at a distance of quarter wavelength from the centre of the input port. This has been done to convert RF virtual short circuit to an open to a tee junction such that RF signals can be transferred selectively from any of the output ports. The SPST capacitive shunt switch tuned at 20 GHz has been utilized in the SPDT design. This tuning is done by realizing the switch in terms of its R-L-C model and then calculating its capacitances and inductance (Kumari and Angira 2022c). The following mathematical equations can be used to approximate the scattering parameters and up/down-state capacitances.

$$S11 = -20\text{Log}\left|\frac{-Z_0}{2Z_b + Z_0}\right|, \tag{1}$$

$$S21 = -20\text{Log}\left|\frac{2Z_b}{2Z_b + Z_0}\right|. \tag{2}$$

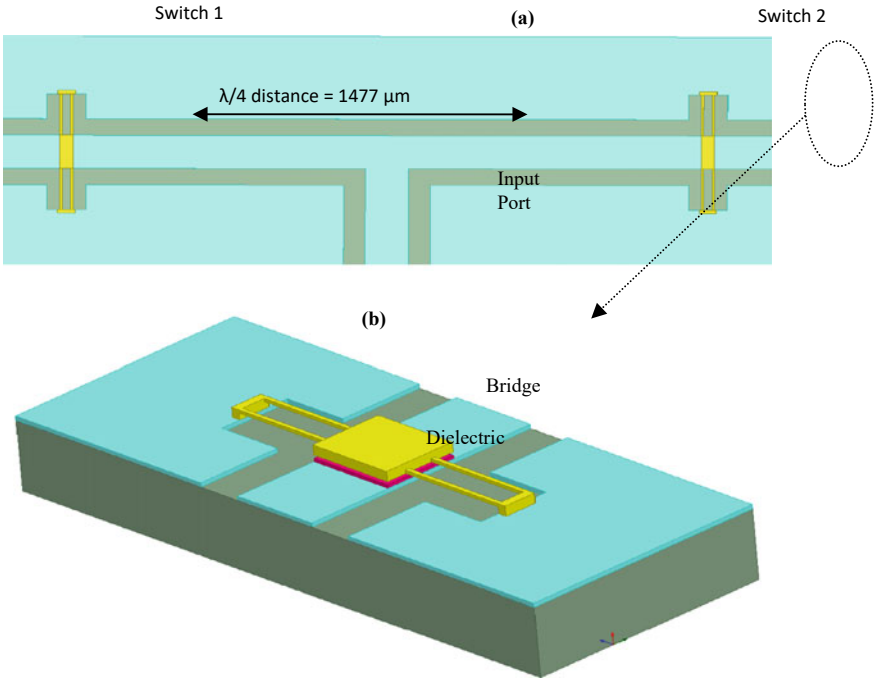


Fig. 1 **a** Simulated SPDT switch circuit design, **b** 3D capacitive shunt switch

$S11$ and $S12$ give transmission losses in terms of return loss, insertion and isolation corresponding to on/off-state of switch. Further, Z_0 is the characteristic impedance of the transmission line and Z_b is bridge impedance given as

$$Z_b = R_t + j\left(\omega L_b - \frac{1}{\omega C_b}\right). \tag{3}$$

The $R_{t,}$ L_b and C_b are resistance, inductance and capacitance of structure used in switch design. The bridge capacitance C_b varies as per the on/off-state of switch. In up-state capacitance is given as

$$C_{\text{up}} = \frac{\varepsilon_o A}{g1 + \frac{t_d}{\varepsilon_r}} = \frac{\varepsilon_o A}{g1}. \tag{4}$$

In down-state capacitance is

$$C_d = \frac{\varepsilon_o \varepsilon_r A}{t_d}. \tag{5}$$

ε_o is the dielectric permittivity in vacuum, A represent conductor's overlap area, g1 is separation of bridge from dielectric, and t_d gives height of dielectric (Angira and Rangra 2015). Switch structure analysed in Kumari and Angira (2022c) with uniform meandering and structural optimization is placed at a distance of 1477 μm from centre for operating frequency 20 GHz. The design parameters used are tabulated in Table 1.

The SPDT switch operates in same manner as a single pole single throw (SPST). When switch is in off-state, both the SPSTs remain in down-state. Biasing voltage is applied across the switch electrodes, and thus, capacitance between signal conductor and switch is large enough to provide conductive path to the signal towards ground. In up-state one SPST switch is made on by simply removing the biasing voltage. The capacitance between signal conductor and bridge reduces and thus signal only have single path towards the output port.

Table 1 Simulated SPDT switch dimensional parameters

	Width (μm)	Length (μm)	Height (μm)	Materials
CPW ground	500	3554	3	Gold
CPW signal	200	3554	3	Gold
Dielectric layer	60	200	0.25	SiO_2
Bridge layer	60	200	3	Gold
Bridge arm	10	250	2	Gold

3 Results and Discussion

The proposed switch design is focused on its applications in 5G/B5G which requires good RF response as per the specifications (isolation > 20 dB and insertion loss < 1 dB) (Iannacci and Tagliapietra 2022; Tagliapietra and Iannacci 2021). The RF performance of switch has been analysed on high-frequency structure simulator (HFSS), a finite element method (FEM) tool. Initially, the SPST switch has been analysed, and performance is shown in Figs. 2 and 3.

The performance plots show that the transmission loss is below 0.3 dB for a wide range from 1 to 30 GHz and maximum isolation of 56 dB is obtained at 20 GHz. The RF response is the most important figure of merits of a switching device employed in communication applications and, as the 5G and beyond 5G technologies are focused on best quality of signal transmission with enhanced speed and coverage so, good electromagnetic characteristics in terms of minimum insertion hand maximum isolation are of most concern over a wide band of mmWave range. The performance characteristics show that switch ensures minimum signal loss during transmission and good isolation when transmission is stopped. Along with this the device will require small voltage to operate as the signal central actuating electrode is used to actuate the moving membrane of switch. The size of the switch is also very compact as can be seen in Fig. 1, and the common area of central signal line has been used for AC and DC signals. Furthermore, SPDT is simulated using the two SPSTs on a coplanar arm and their transmission losses has been analysed. The up-state response of the SPDT is shown as below.

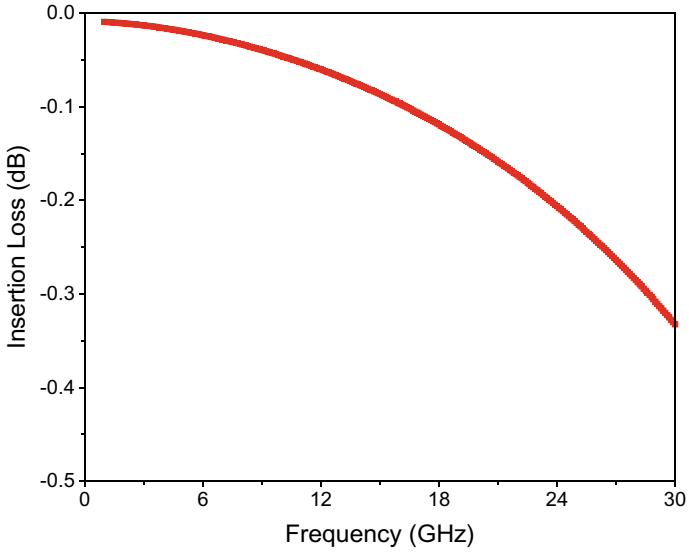


Fig. 2 Up-state performance of simulated SPST switch

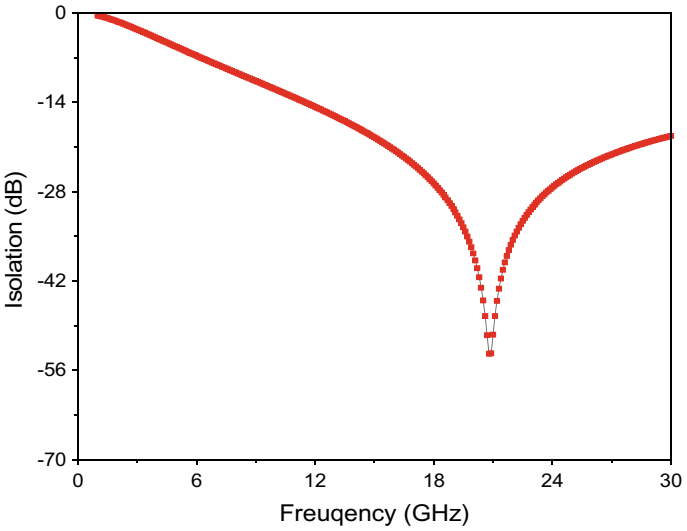


Fig. 3 Down-state performance of simulated SPST switch

The response in Fig. 4 shows that for frequencies from 15 to 29 GHz insertion losses are below 1 dB as per the specifications of 5G/B5G. Furthermore, a minimum insertion loss of −0.2 dB is observed around 26 GHz. In on-state, the capacitance is calculated as 34.69 fF using Eq. (4). The isolation response is shown between input port and switch 2 which is in down-state or in off-state as in Fig. 5.

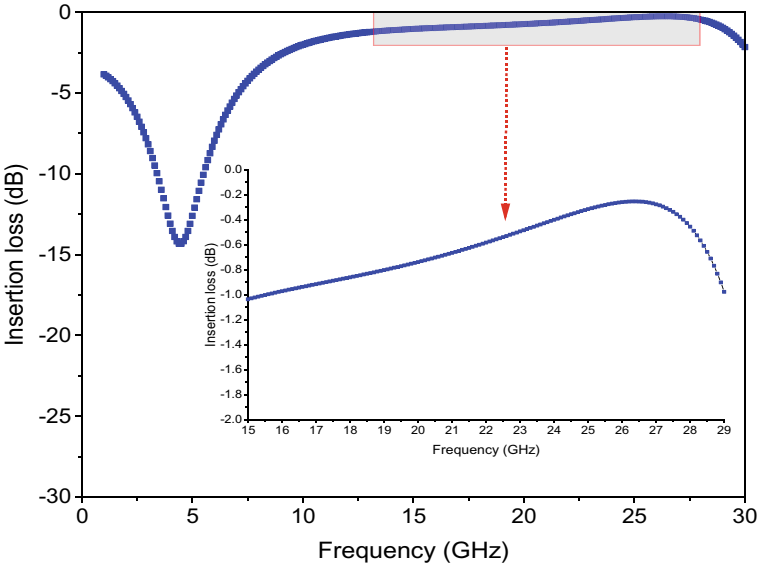


Fig. 4 Insertion loss response of simulated SPDT switch when switch 1 is in up-state

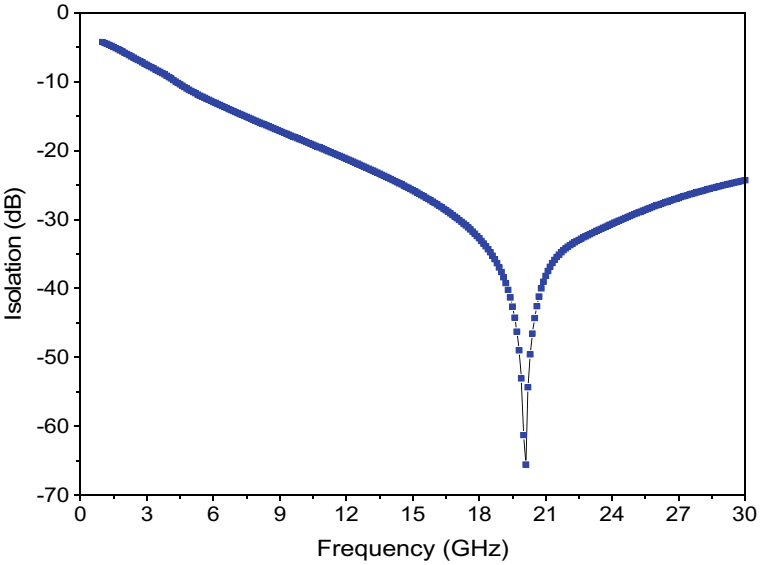


Fig. 5 Isolation response of down-state switch 2 and input port of simulated SPDT switch

The switch is tuned around 20 GHz, where maximum isolation of -65 dB is achieved. Moreover, isolation > 20 dB is widely obtained over range from 11 to 30 GHz. The capacitance is reduced to 1.69 pF in off-state. Thus, along with good RF performance a wide bandwidth of 14 GHz is covered with the proposed SPDT switch design. The SPDT design will also reflect the advantages of SPST switch which is used for designing such as small size, low operating voltage and light weight due to common AC/DC signals area.

4 Conclusion

In this paper low loss, high isolation capacitive shunt switch has been used in SPDT circuit design operable at high-frequency mmWave ranges with compact size and light weight. The magic of quarter wavelength has been utilized to design a simple SPDT transceiver switch using SPSTs. The electromagnetic performance analysis is carried out in order to measure the transmission losses in on/off switch states. The performance is well within the limits specified for 5G/B5G frequencies. The proposed device is thus found fit for a wide application area in the current technology paradigms such as Internet of things (IoT), tactile Internet (TI), Internet of everything (IoE) and augmented reality/virtual reality (AR/VR) which operates over high frequencies above sub-6 GHz ranges. The SPDT switches also employed extensively in satellite communication, drones and broadcasting applications.

References

Angira M, Rangra K (2015) Design and investigation of a low insertion loss, broadband, enhanced self and hold down power RF-MEMS switch. Microsyst Technol 21:1173–1178. https://doi.org/10.1007/s00542-014-2188-6

Angira M, Rangra KJ (2016) A novel design for low insertion loss, multi-band RF-MEMS switch with low pull-in voltage. Eng Sci Technol an Int J. 19:171–177. https://doi.org/10.1016/j.jestch.2015.07.001

Angira M, Bansal D, Kumar P et al (2019) A novel capacitive RF-MEMS switch for multi-frequency operation. Superlattices Microstruct 133:1–7. https://doi.org/10.1016/j.spmi.2019.106204

Bansal D, Kumar P, Kumar A (2021) Improvement of RF MEMS devices by spring constant scaling laws. J Comput Electron 20:1006–1011. https://doi.org/10.1007/s10825-021-01657-z

Chae U, Yu HY, Lee C, Cho IJ (2020) A hybrid RF MEMS switch actuated by the combination of bidirectional thermal actuations and electrostatic holding. IEEE Trans Microw Theory Tech 68:3461–3470. https://doi.org/10.1109/TMTT.2020.3003553

Huo Y, Dong X, Xu W (2017) 5G cellular user equipment: from theory to practical hardware design. IEEE Access 5:13992–14010. https://doi.org/10.1109/ACCESS.2017.2727550

Huo Y, Dong X, Xu W, Yuen M (2019) Enabling multi-functional 5G and beyond user equipment: a survey and tutorial. IEEE Access 7:116975–117008. https://doi.org/10.1109/ACCESS.2019.2936291

Huo Y, Dong X, Xu W, Yuen M (2018) Cellular and WiFi co-design for 5G user equipment. In: IEEE 5G world forum, 5GWF 2018—conference proceedings, pp 256–261. https://doi.org/10.1109/5GWF.2018.8517059

Iannacci J (2017) RF-MEMS for high-performance and widely reconfigurable passive components—A review with focus on future telecommunications, Internet of Things (IoT) and 5G applications. J. King Saud Univ 29:436–443

Iannacci J, Tagliapietra G (2022) Getting ready for beyond-5G, super-IoT and 6G at hardware passive components level: a multi-state RF-MEMS monolithic step attenuator analyzed up to 60 GHz. Microsyst Technol 28:1235–1240. https://doi.org/10.1007/s00542-022-05285-w

Kumari R, Angira M (2022c) Investigation on switching structure geometries and their impact on electromagnetic response of RF-MEMS capacitive switch. In: 2nd Asian conference on innovation in technology (ASIANCON), pp 1–6. https://doi.org/10.1109/ASIANCON55314.2022.9909237

Kumari R, Angira M (2022a) Analysis on maximizing the performance of RF-MEMS switches across multiple domains through appropriate bridge material selection. Microsyst Technol 28:1179–1190. https://doi.org/10.1007/s00542-022-05277-w

Kumari R, Angira M (2022b) FEM simulation and material selection for enhancing the performance of a RF-MEMS capacitive switch. J Comput Electron. https://doi.org/10.1007/s10825-022-01905-w

Lysenko IE, Tkachenko AV, Sherova EV, Nikitin AV (2018) Analytical approach in the development of RF MEMS switches. Electronics 7:1–23. https://doi.org/10.3390/electronics7120415

Malczewski A, Pillans B (2004) Single-pole double-throw switches using capacitive MEMS switches. Int J RF Microwave Comput Aided Eng 14:383–387

Pacheco SP, Peroulis D, Katehi LP (2001) MEMS single-pole double-throw (SPDT) X and K-band switching circuits. IEEE MTT-S, Phoenix AZ USA, pp 321–324

Rao KS, Vasantha K, Sravani KG (2020) Design and simulation of capacitive SPDT RF-MEMS switch to improve its isolation. Microsyst Technol 26:2553–2560. https://doi.org/10.1007/s00542-020-04795-9

Rebeiz GM, Muldavin JB (2001) RF MEMS switches and switch circuits. IEEE Microw Mag 2:59–71. https://doi.org/10.1109/6668.969936

Tagliapietra G, Iannacci J (2021) A comprehensive overview of recent developments in RF-MEMS technology-based high-performance passive components for applications in the 5G and future

telecommunications scenarios. Facta Univ Ser Electron Energ. 34:333–366. https://doi.org/10.2298/fuee2103333t

Tang M, Yu AB, Liu AQ et al (2005) High isolation X-band MEMS capacitive switches. Sens Actuators A Phys 120:241–248. https://doi.org/10.1016/j.sna.2004.11.026

Tilmans HA, De Raedt W, Beyne E (2003) MEMS for wireless communications: "From RF-MEMS components to RF-MEMS-SiP." J Micromechanics Microeng 13(4). https://doi.org/10.1088/0960-1317/13/4/323

Investigation of Bulk, Electronic and Transport Properties of Armchair Silicene Nanoribbon as Liquefied Petroleum Gas Combustion Indicator: A DFT Study

Jaskaran Singh Phull, Harmandar Kaur, Paramjot Singh, Gurleen Kaur Walia, and Deep Kamal Kaur Randhawa

Abstract Liquefied petroleum gas is highly combustible and has propane and butane as its main constituents. It is a clean fuel gas used widely in homes, industries and vehicles. However, accidental gas leaks can lead to explosions followed by fire, risking lives and causing material damage. Liquefied petroleum gas is highly inflammable and opposed to natural gas, is heavier than air and hence does not disperse easily. In this work, the first-principle method based on density functional theory is applied to investigate the adsorption behaviour of combustion gas constituents on armchair silicene nanoribbon. The structural, electronic and transport properties are determined for the considered complex of the armchair silicene nanoribbon adsorbed primary LPG gas molecules that are butane and propane. Additionally, incomplete combustion wherein the toxic gas carbon monoxide is a by-product of the combustion process is also studied for gas sensing using pristine armchair silicene nanoribbon.

Keywords Combustible gas · Armchair silicene nanoribbon · Density functional theory · Sensing

J. S. Phull · H. Kaur (✉) · D. K. K. Randhawa
Department of Engineering & Technology, GNDU, Regional Campus, Jalandhar, India
e-mail: harmandar.ecejal@gndu.ac.in

P. Singh
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G1M8, Canada

G. K. Walia
School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_27

1 Introduction

Two-dimensional (2D) materials have gained considerable attention in several imperative domains owing to their unique properties (Singh et al. 2021a, 2022). These materials are emerging contenders for gas sensing in extensive avenues for instance industry, environmental protection, health, safety, amenity, etc. (Tarun et al. 2021; Singh et al. 2021b; Chaves et al. 2020; Yamazoe 2005). Silicene, an allotrope of silicon (Akbari et al. 2016), is a two-dimensional material structurally similar to graphene (Paola et al. 2012) and has a zero bandgap with a periodic buckling arrangement attributed by the Pseudo Jahn Teller distortions (Akbari et al. 2016). Various bandgap engineering methods are applied to obtain bandgap opening (Patrick et al. 2012). Creating quantum confinement by constraining silicene to one-dimensional material like a nanoribbon is vital for introducing a bandgap opening in silicene. Also, silicon atoms adopt sp3 hybridization in silicene (Akbari et al. 2016) making them chemically active on the surface under external chemical functionalisation.

Takeda and Shiraishi (1994) were the first to report and study silicene in 1994. Guzmán-Verriand and Voon (2007) named it thus in 2007. Ferromagnetism (Wang et al. 2012), superconductivity (Chen and Wua 2013), half-metallicity (Zheng and Zhang 2012) and the quantum Hall effect (Liu et al. 2011) are all features of the two-dimensional structure.

Silicene is not totally planar in two dimensions; like graphene, it is made up of a hexagonal pattern of atoms, but the hexagonal rings of silicene include puckering distortions due to armchair configuration. Because of its buckled honeycomb structure, silicene has a high chemical reactivity. A semiconductor is defined by a nonzero bandgap, which permits semiconductors to be employed in electronic components such as diodes and transistors. Because silicene is made of silicon, it can be manufactured using current semiconductor manufacturing techniques. Its edges do not show any signs of oxygen reactivity, indicating that it will be stable enough for usage in electronics (Kara et al. 2012).

Silicene offers a number of advantages, including stability, tuneability, lack of oxygen edge reactivity and compatibility with silicon-based electronics, which makes it a suitable choice for most gas sensing applications (Prasongkit et al. 2015). The edges of silicene can be zigzag or armchair based on quantum confinement. Armchair silicene is favoured over zigzag silicene because the former has semiconducting characteristics and is preferred for gas molecule sensing due to its adsorption sensitivity compared to its zigzag sibling (Cahangirov et al. 2016; Song et al. 2013).

Silicene has been analyzed in literature for gas sensing for the detection of various gases such as NO, NO_2, NH_3, CO, SO_2, H_2S and CO_2 (Liu and Zhou 2018; Kaur et al. 2020; Tarun et al. 2021). However, to the best of our knowledge, silicene nanoribbon (SiNR) remains unexplored for LPG fuel gas sensing. Owing to the vulnerability of voluntary existence of highly flammable fuel gas in homes, industries and LPG-fuelled vehicles, the sensitivity of a gas sensor is highly crucial (Aghaei et al. 2016a). Hence, even an individual primary gas molecule constituent of LPG should be detected. The adsorbate gas molecule should induce noticeable changes

in the structural, electronic and the transport characteristics of the pristine armchair silicene nanoribbon (ASiNR) for ensuring high sensitivity.

In this work, primary LPG constituent gas molecules are investigated as adsorbates on pristine ASiNR with regard to their structural, electronic and transport properties. LPG is a clean fuel gas, commonly used in domestic, vehicular and industrial environments. It has been used as a fuel in vehicles since 1900s (Werpy et al. 2010) since it has lower emissions compared to gasoline (Gummus 2011). LPG is a fuel gas composed of primary constituents of propane and butane in majority and ethane or pentane in negligible traces (El-Morsi 2015; Setiyo et al. 2017). In this paper, first principles' study method based on density functional theory (DFT) (Hohenberg and Kohn 1964; Kohn and Sham 1965; Brandbyge et al. 2002; Kresse and Hafner 1993, 1994; Kresse and Furthmüller 1996a, b; https://www3.epa.gov/ttnchie1/old/ap42/ch01/s05/final/c01s05_oct1996.pdf) is employed for the investigation of ASiNR-adsorbate complex for LPG fuel gas sensing. Furthermore, the incomplete combustion condition, which is responsible for the generation of a toxic by-product, that is carbon monoxide, is also accounted for. In literature, few works are reported pertaining to propane and butane adsorbate gas sensing (Rashid et al. 2018, 2020; Bhuvaneswari et al. 2020). The structural (bond lengths, adsorption distances, buckling height and lattice constants before and after relaxation), electronic (adsorption energy, charge transfer using Mulliken population (Kresse and Furthmüller 1996b), bandgap energy and density of states) and transport properties (transmission spectra, transmission eigenstates and current voltage characteristics using Landauer–Buttiker formula) are determined (Mulliken 1955; Datta 1997; Abadir et al. 2009).

2 Computational Details

The first principles' calculations are carried out within the scope of density functional theory (DFT) to examine the adsorption behaviour of primary LPG gas molecules on pristine ASiNR. This assessment for the sensing application of pristine AsiNR towards the said gases is accomplished with the help of the Quantum Atomistix Tool Kit package (Synopsis: Quantumwise Atomistix Tool Kitversion 2018). To obtain a nanoribbon with a certain width n, the neighbour honeycomb structure is slit in the direction that is parallel to the nearest neighbour bonds. Both the structural and electronic calculations are performed on pristine ASiNR with width n = 6 (AsiNR-6). Generalized gradient approximation (GGA) (vanMourik et al. 2011; Becke 1988; Perdew et al. 1992, 1996a; Perdew 1991; Lee et al. 1988; Gritsenko et al. 1997; Rappoport et al. 2009; Solomon 2013) with Perdew–Burke–Ernzerhof (PBE) function (Perdew et al. 1996b) represents the exchange correlation potential with double zeta-polarized (DZP) basis set (Abadir et al. 2009) with pseudo-dojo pseudo-potential set. However, GGA may underestimate the bandgap of semiconductors, but it is observed to give corrected band order for silicon-based systems. The computational efficiency persists since the PBE exchange function has its correction terms constructed solely based on the exact conditions. Force and stress tolerance

drop below 0.05 eV/Å and 0.01 Gpa, respectively, in the optimized structure. With the Monkhorst–Pack grid (Monkhorst and Pack 1976), the k-point sampling of 1 × 1 × 12 is used in the Brillouin zone, for geometry optimization, and 1 × 1 × 99 k-points for electronic and transport calculations. The I–V relation of the ASiNR with adsorbed combustible gases is computed by the non-equilibrium Green's function (NEGF). About 300 K value of the electron temperature is used (Yamacli 2014; Aghaei et al. 2016b).

3 Results and Discussion

ASiNR has a buckled structure with a buckling height of 0.44 Å, a lattice constant value of 3.88 Å and a bond length of 2.29 Å between Si and Si molecules (Wei et al. 2017) and is being studied widely for pertinent gas sensing. It has highly reactive edges that are passivated by hydrogen atoms (Wang et al. 2015). The nanoribbon surface is exposed to the primary LPG constituting gas molecules, i.e. propane and butane. Furthermore, the incomplete combustion condition is also accounted for that is responsible for the generation of a toxic by-product that is carbon monoxide. The adsorption of LPG fuel gas molecules affects the bulk, electronic and transport behaviour of the ASiNR. Individual gas molecule adsorbed on 6-ASiNR is investigated for detection that corresponds to extremely low ambient gas densities. The stability of the relaxed structure is evaluated by computing the adsorption energy (Ead) given as (Lu et al. 2012; Pan and Zhu 1998):

$$E_{\text{ad}} = EASiNR + \text{Molecule} - EASiNR - E\text{Molecule},$$

where EASiNR + Molecule, EASiNR and EMolecule are the energies associated with the complex, ASiNR and the gas molecules, respectively. Figure 1 presents the structural distinctions of the unoptimized and optimized structural geometries of the combustion gas molecules adsorbed pristine ASiNR. It is observed in Fig. 1 that upon relaxation, the ASiNR shows significant variation in its Si–Si bond lengths owing to the influence of the gas molecule adsorbates. In addition, the buckling height and the lattice constant of the nanoribbon are also changed from 0.44 Å and3.88 Å to less than 0.5 Å and 3.91 Å, respectively. Propane molecule after relaxation sits at a distance 3.66 Å away from the surface while its orientation remains almost the same.

Butane molecule has a molecule-sheet distance of 3.51 Å after relaxation which is evident from Fig. 1b. The orientation of the molecules is the same although they are displaced by accountable distance from their original positions. For both propane and butane adsorbates upon adsorption on the base material, their bond lengths do not exhibit much variation. The presence of gas adsorbent molecules, i.e. CO, propane (C_3H_8) and butane (C_4H_{10}) together on the pristine ASiNR has slight effects on the lattice constant and Si–Si bond length which changes from 3.88 Å and 2.29 Å to 3.86 Å and 2.32 Å, respectively. After relaxation, the three gas adsorbents of CO, C_3H_8 and C_4H_{10} stabilize at a distance of 2.20 Å, 3.33 Å and 3.45 Å from the

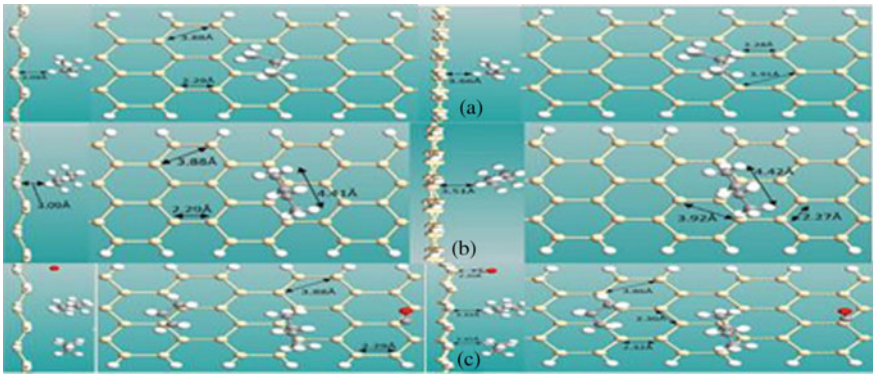


Fig. 1 Unoptimized (left) and optimized (right) configurations of ASiNR on adsorption of **a** propane, **b** butane, **c** propane and butane with carbon monoxide (carbon atom: grey; hydrogen atom: white)

surface, respectively. CO is closest to the adsorbent surface exhibiting a possibility of bond formation, depending on the orientation of the molecule. The orientation of the molecules remains unchanged although they are displaced by some distance from their original positions.

4 Electronic Properties

4.1 Charge Transfer, Charge Density and Adsorption Energy

Sensing process requires the adsorption of the adsorbate on the adsorbent material. Upon relaxation, the considered molecules adopt various geometries. For propane and butane, the electronic properties are investigated upon their adsorption on pristine ASiNR. This has been extended to include the incomplete combustion condition by incorporating carbon monoxide with the constituent LPG fuel gas molecules. The analysis covers the determination of the charge transfer, bandgap, adsorption energy and electronic charge density. The bandgap energies E_g (eV), adsorption energies E_{ad} (eV), charge transfer Δq (e) by Mulliken analysis (Khaliullin et al. 2008) and binding distances D (Å) of the afore-mentioned molecules on ASiNR adsorbent are listed in Table 1.

The adsorption energies and the charge transfer values of propane and butane show favourable results be fitting fuel gas sensing. The adsorption energies E_{ad} (eV) of C_3H_8, C_4H_{10} and CO, C_3H_8 and C_4H_{10} gas molecules taken together on ASiNR are -1.17 eV, -1.31 eV and -3.30 eV, respectively. The higher negative value of adsorption energy signifies strong adsorption. This might lead to a chemical bond

Table 1 Observed electronic bandgap (E_g), adsorption energy (E_{ad}), charge transfer (Δq) and binding distance (D) for ASiNR adsorbent gas molecules

Combustion condition	Adsorbate molecule(s)	E_g (eV)	Δq (e)	E_{ad} (eV)	D (Å)
LPG fuel gas combustion	C_3H_8	0.5056	0.124	−1.17	3.66
	C_4H_{10}	0.5085	0.164	−1.31	3.51
LPG fuel gas incomplete combustion	CO	0.4494	0.072, 0.119, 0.143	−3.30	2.20
	C_3H_8				3.33
	C_4H_{10}				3.45

formation between the base material and adsorbate molecule owing to chemisorption process. Strong bond formation corresponds to the long recovery time sowing to the requirement of desorption process to render the reusability of the sensing system. The application of electric field and UV irradiation process are generally used for desorption processes. The van der Waal interactions are indicated by the low adsorption energies leading to physisorption. The charge transfer values (Δq) due to adsorption of C_3H_8, C_4H_{10} and CO, C_3H_8 and C_4H_{10} gas molecules taken together on pristine ASiNR are calculated by Mulliken population analysis and are 0.124, 0.164 and 0.332, respectively.

Also, the electrical conductivity (σ) of a material is related to its band gap as (Osborn and Farajian 2014; Li 2012):

$$\sigma \alpha e\left(-E_g/2kBT\right), \tag{2}$$

where T is the temperature in Kelvin, kB is the Boltzmann constant and E_g is the band gap.

The electronic charge density investigates the orbital overlap. Large charge density corresponds to the orbital overlap that forms bonding molecular orbital. The electron charge density in Fig. 2c presents the obtained orbital mixing for the case of carbon monoxide adsorbed on ASiNR.

4.2 Band Structure

The adsorption of different molecules on silicene nanoribbon engineers its bandgap. Generally, it is the permissible energy levels that can be occupied by the electrons. The variations in the bandgap values for the different adsorbate molecules are a preliminary qualifier for fuel gas sensing. The band structure of pristine ASiNR is shown in Fig. 3a and those obtained with constituent fuel gas molecules on ASiNR are shown in Figs. 3b–d. Figure 3d represents the incomplete combustion of the LPG fuel gas.

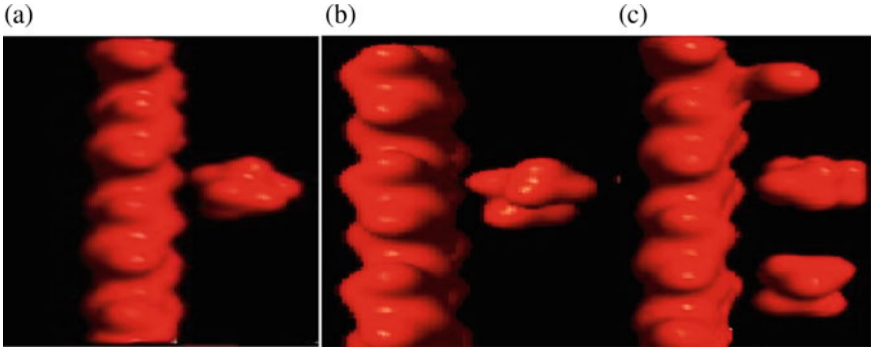


Fig. 2 Electron charge density of ASiNR on adsorption of **a** propane, **b** butane, **c** propane, butane with carbon monoxide

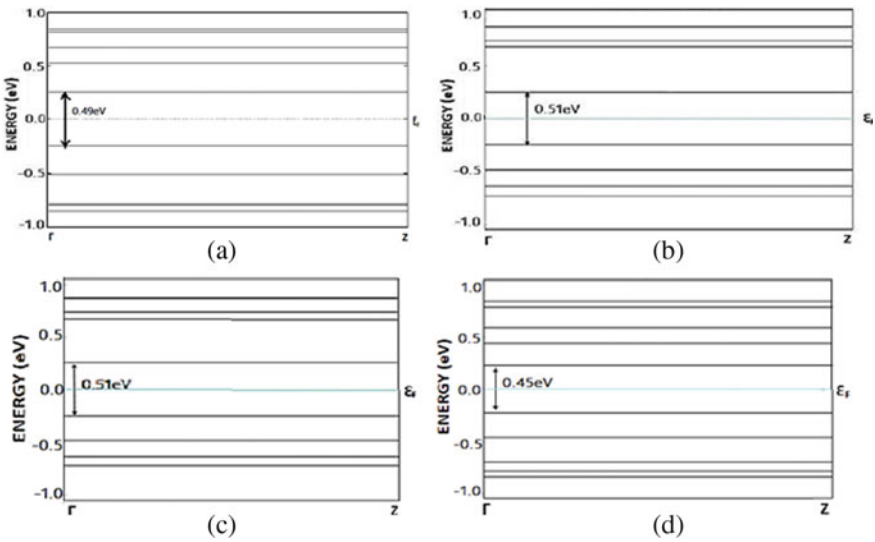


Fig. 3 Calculated band structure of **a** pristine ASiNR and ASiNR on adsorption of **b** propane **c** butane **d** propane, butane with carbon monoxide

The bandgap energy as presented in Table 1 is 0.5056 eV and is 0.5085 eV for propane and butane gas molecules, respectively. The complex remains intrinsic in nature after adsorption as is clear from the position of conduction band minima and valence band maxima with respect to the Fermi level in Fig. 3. The bandgap value for the adsorbent pristine ASiNR is 0.4901 eV as is shown in Fig. 3a. The adsorption of LPG gas molecules in presence of carbon monoxide is also calculated and it generates a reasonable bandgap opening with a corresponding energy of 0.4495 eV, offering the scope for the simultaneous sensing of the combustible LPG gas molecules that is shown in Fig. 3d. Compared to the bandgap value of pristine ASiNR, the bandgap

value for the incomplete combustion case as is shown in Fig. 3d has a significant variation.

The DOS is mathematically expressed as (Poole and Owens 2003):

$$D(E) = d\,N(E)/dE, \tag{3}$$

where N denotes the electronic states having energy E_{in} range dE. It is observed from the DOS analysis that both ASiNR-propane and ASiNR-butane complexes show semiconducting nature in Figs. 4a, b, respectively. In case of incomplete combustion having propane, butane fuel gas constituent molecules with carbon monoxide, density of states obtained is shown in Fig. 4c, that is high around the Fermi level and absent at the Fermi level, which denotes that this complex is semiconducting in nature.

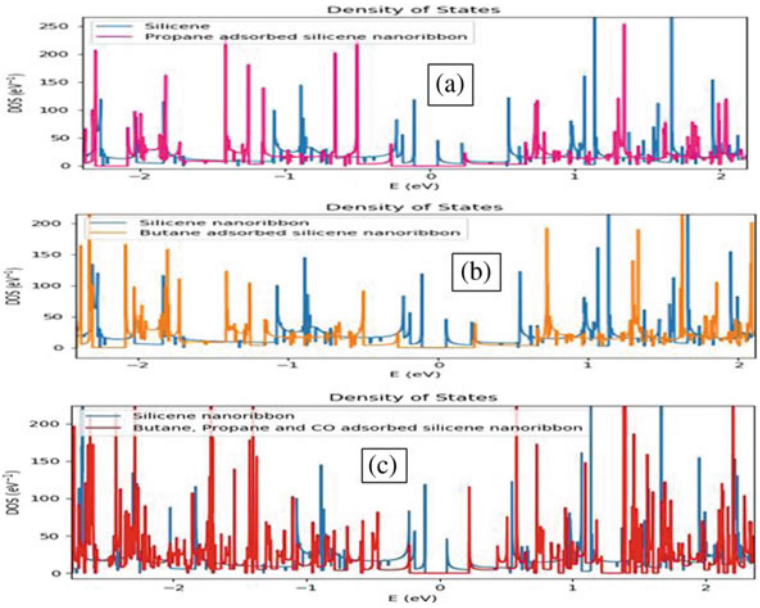


Fig. 4 Density of states of ASiNR on adsorption of **a** propane, **b** butane, **c** propane, butane with carbon monoxide

5 Device Characteristics

5.1 Transport Properties

The bulk properties are calculated by applying DFT method on the considered nanoribbon; however, to evaluate the transport properties, the nanoribbon is transformed into a simple device. Here, a two-probe model comprising right electrode, left electrode and scattering region is used to calculate the transport properties of the gas adsorbed nanostructure obtained.

I–V Characteristics. The transport properties are studied in the device configuration via two-probe system with right electrode, left electrode and the scattering region, for the analysis of the I–V curve, transmission spectrum and transmission eigenstates. The electrical conductivity of the adsorbent material changes in the presence of the adsorbate molecules. This change in the conductivity forms the basis for gas sensing application. The I–V characteristic curve of the primary LPG combustible gas molecules adsorbed ASiNR complex is shown in Fig. 5 and is extended to present the incomplete combustion. As is evident from these results, the I–V curve holds compliance with the nature of bond formation and hence the obtained adsorption energies.

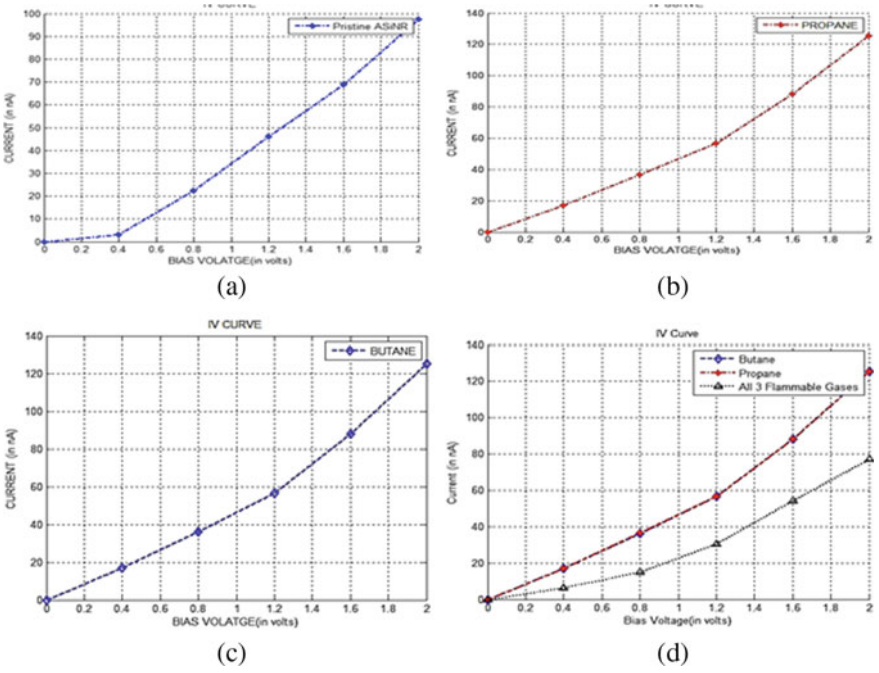


Fig. 5 I–V characteristics of ASiNR on adsorption of **a** pristine ASiNR, **b** propane, **c** butane molecule, **d** propane, butane and all three gases propane, butane and carbon monoxide

The current is calculated using the Landauer–Buttiker formalism (Brocks 2005; Nemnes and Racec 2004):

$$I = \int_{\mu L}^{\mu R} T(E, Vb)\mathrm{d}E, \tag{4}$$

where μR and μL are the chemical potentials of the right and left electrodes, $T(E, V_b)$ is the electronic transport probability and V_b is the applied voltage.

LPG fuel gas adsorbed pristine ASiNR based on single molecule detection, and for all the adsorbates, the I–V characteristics in Fig. 5 show an increase in the conduction with a corresponding increase in the applied bias. Figure 5d includes the I–V characteristic of incomplete combustion having the constituent fuel gas molecules with carbon monoxide. The increase in the current values upon adsorption indicates the increased sensitivity. The adsorbate molecule causes a change in conductivity by behaving as a donor or an acceptor. Both propane and butane gases exhibit similar I–V characteristics. However, as shown in Figs. 5d, a, reduction in current value is observed opposed to individual LPG constituent gas adsorbent molecules for the same applied bias. Also, owing to scattering process by the joint contribution of all three gas molecules in case of incomplete combustion, the current value lowers compared to individual gas molecule adsorption onto ASiNR.

Transmission Spectrum. The transmission spectrum of pristine ASiNR with the adsorption of the LPG gas molecules when the applied voltage is 0 V is depicted in Fig. 6. For a given energy value, say at Fermi level the transmission value is different for different adsorbate molecules, implying that each gas molecule causes varied transmission through the device for the same energy level. This aids in sensing the individual gases uniquely. The similarity in the DOS peaks and the transmission spectrum elucidates the correspondence in the energy levels. It can be observed that for propane and butane, the value is approximately the same. Henceforth, the external bias is applied with a range of voltage values varying from 0 to 2 V which supports an enhancement in the transmission process. In addition, the case of incomplete combustion having propane, butane with carbon monoxide molecule has 0.195 transmission value at zero applied bias.

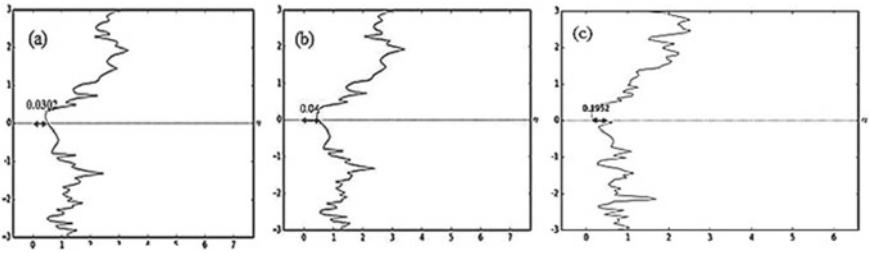


Fig. 6 Transmission spectrum of ASiNR on adsorption of **a** propane, **b** butane, **c** propane, butane with carbon monoxide molecules at zero applied bias

Transmission Eigen States. The eigen functions of the transmission matrix in real space signify a specific eigenstate. The transmission probability is achieved by diagonalizing the transmission matrix that gives the eigenvalues. The transmission eigenstate is a complex three-dimensional wave function that describes the electron transport, which is represented using isosurface wherein the magnitude of eigenstate is given by isovalue and phase of the eigenstate is shown by the colour map. The channels in the molecule are equal to the number of eigen values obtained and the channel strength is given by the magnitude of the eigen value (Compernolle et al. 2003; Lee 2000; Magnus 2007). An isovalue of 0.08 is considered as the absolute value of the wave function.

Figure 7 shows that the eigenstates are delocalized implying a finite probability of transmission of electrons from the left electrode to the right electrode. The colour bar shows the phase of wave functions. The eigenstates have higher amplitude in Fig. 7c to the left side. The eigenstates on this side comprise the incident and the reflected states that can cause superposition effects of the wave functions forming constructive and destructive interference. A transmission eigenstate is a superposition of different states that transit form left electrode to right electrode and/or vice versa as per the associated phase values. The complete delocalization of the eigenstates denotes the ease of forming a transmission channel from one electrode to the other which is shown in Figs. 7a and 8b. However, partial localization due to destructive interference is observed in Fig. 7c which indicates partial inhibition of the carrier transport across the scattering region.

Furthermore, it is known that for sensing application, the membrane surface plays a pivotal role of distinguishing the adsorbates. Hence, the bulk and transport properties evaluated for the adsorption of the primary LPG constituent gas molecules, i.e. propane, butane and with carbon monoxide (incomplete combustion) on ASiNR adsorbent show significant variations in the presence of the adsorbate molecules. Therefore, the variations in the structural (bond lengths, adsorption distances, buckling height and lattice constants), electronic (adsorption energy, charge transfer using Mulliken population, bandgap energy and density of states) and transport characteristics (current–voltage characteristics using Landauer–Buttiker formula, transmission spectra and transmission eigenstates) of the pristine ASiNR adsorbed combustible

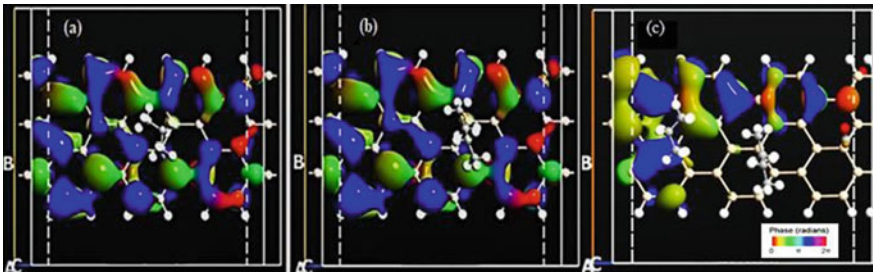


Fig. 7 Transmission eigenstates on adsorption of **a** propane, **b** butane, **c** propane, butane with carbon monoxide molecule at zero applied bias

gas molecule complex signify its potential role in sensing LPG fuel gas for combustion. Furthermore, the pristine ASiNR sensor can identify incomplete combustion of the LPG fuel gas, and thus, it can be further extended for calibrating efficient fuel utilization.

6 Conclusion

The accidental leak of combustible gases can be highly hazardous for the domestic, vehicular and industrial environments. Generally, the exposure to the combustible gases in domestic settlements, LPG fuel-based vehicles and industries, is dangerous owing to its high flammability primary constituent gases of propane and butane. The adsorption of the primary LPG combustible gas molecules on ASiNR exhibits significant variations in its electronic and transport properties thus befitting combustible gas sensing application. The structural variations pertaining to bond length variations, adsorption distance, buckling height and lattice constants are studied. The electronic properties are investigated from the adsorption energy calculations, charge transfer calculations using Mulliken population, bandgap energy obtained from the band structure analysis. Furthermore, the transport properties that include the I–V characteristics using Landauer–Buttiker formalism, transmission spectra and transmission eigen state calculations are examined. Extensive analysis of the structural, electronic and transport properties elucidates that combustible gas primary constituent fuel gas molecules propane and butane are physisorbed, thus ensuring ASiNR a suitable combustible gas sensing candidate. The study of the considered complex ensures a single molecule resolution combustible LPG fuel gas sensing suitable for the calibration in low gas density scenarios. A significant variation in the structural, electronic as well as transport properties results upon the adsorption of the primary constituent molecules of liquefied petroleum gas on ASiNR, explicitly indicating its veritable application for LPG combustion gas sensing including incomplete combustion sensing that can aid to calibrate fuel utilization efficiency.

References

Abadir GB, Walus K, Pulfrey DL (2009) Basis set choice for DFT/NEGF simulations of carbon nanotubes. J Comp Electron 8(1):1–9. https://doi.org/10.1007/s10825-009-0263-5

Aghaei SM, Monshi MM, Calizo I (2016) A theoretical study of gas adsorption on silicene nanoribbons and its application in a highly sensitive molecule sensor. RSC Adv 6:94417–94428. https://doi.org/10.1039/C6RA21293J

Aghaei SM, Monshi MM, Torres I, Calizo I (2016) Edge functionalization and doping effects on the stability, electronic and magnetic properties of silicene nanoribbons. RSC Adv 6(21):17046–17058. https://doi.org/10.1039/C5RA26107D

Akbari E, Buntat Z, Afroozeh A, Pourmand SE, Farhangd Y, Sanatie P (2016) Silicene and graphene nanomaterials in gas sensing mechanism. RSC Adv 6:81647–81653. https://doi.org/10.1039/C6RA16736E

Becke AD (1988) Density functional exchange energy approximation with correct asymptotic behavior. Phys Rev A 38(6):3098. https://doi.org/10.1103/PhysRevA.38.3098

Bhuvaneswari R, Nagarajan V, Chandiramouli R (2020) Physisorption of propane and butane vapors on novel Kago meantimonenesheets—A first-principles perception. Chem Phys Lett 754:137693

Brandbyge M, Mozos JL, Ordejón P, Taylor J, Stokbro K (2002) Density-functional method for non-equilibrium electron transport. Phys Rev B 65(16):165401.https://doi.org/10.1103/PhysRevB.65.16540

Brocks G (2005) Electron transport at the nanoscale lecture notes, preliminary version. http://www.acmm.nl/molsim/han/2005/lnotes.pdf

Cahangirov S, Sahin H, Le Lay G, Rubio A (2016) Introduction to the physics of silicene and other 2D materials. Springer

Chaves A, Azadani JG, Alsalman H et al (2020) Bandgap engineering of two-dimensional semiconductor materials. Npj 2D Mater Appl 4:29. https://doi.org/10.1038/s41699-020-00162-4

Chen L, Feng B, Wua K (2013) Appl Phys Lett 102:081602

Compernolle S, Chibotaru L, Ceulemans A (2003) Eigen states and transmission coefficients of finite-sized carbon nanotubes. J Chem Phys 119(5):2854. https://doi.org/10.1063/1.1587691

Datta S (1997) Electronic transport in mesoscopic systems. In: Ahmad H, Broers A, Pepper M(eds). Cambridge University Press, New York, pp 57–111. ISBN 978-0-521-59943-6

El-Morsi M (2015) Energy and exergy analysis of LPG (liquefied petroleum gas) as a drop in replacement for R134a in domestic refrigerators. Energy 86:344–353

Gritsenko OV, Schipper PRT, Baerends EJ (1997) Exchange and correlation energy in density functional theory : Comparison of accurate density functional theory quantities with traditional Hartree–Fock based ones and generalized gradient approximations for the molecules Li2, N2, F2. J Chem Phys 107(13):5007–5015

Gummus M (2011) Effects of volumetric efficiency on the performance and emissions characteristics of a dual fueled (gasoline and LPG) spark ignition engine. Fuel Process Technol 92(10):1862–1867

Guzmán-Verri GG, Voon LL (2007) Phys Rev B 76:075131

Hohenberg P, Kohn W (1964) Inhomogeneous electrongas. Phys Rev 136(3B):864. https://doi.org/10.1103/PhysRev.136.B864

https://www3.epa.gov/ttnchie1/old/ap42/ch01/s05/final/c01s05_oct1996.pdf

Kara A, Enriquez H, Seitsonen AP, Voon LL, Vizzini S, Aufray B, Oughaddou H (2012) A review on silicene—New candidate for electronics. Surf Sci Rep 67(1):1–18. https://doi.org/10.1016/j.surfrep.2011.10.001

Kaur H et al (2020) First principles study of sarin nerve gas adsorption on graphene nanoribbon with single molecule resolution. Mater Today Proc 28:1985–1989. https://doi.org/10.1016/j.matpr.2020.05.574

Khaliullin RZ, Bell AT, Head-Gordon MJ (2008) Analysis of charge transfer effects in molecular complexes based on absolutely localized molecular orbitals. J Chem Phys 128(18):184112. https://doi.org/10.1063/1.2912041

Kohn W, Sham LJ (1965) Self-consistent equations including exchange and correlation effects. Phys Rev 140(4A):A1133

Kresse G, Furthmüller J (1996a) Efficiency of abinitio total-energy calculations for metals and semiconductors using a plane wave basis set. Comput Mater Sci 6(1):15–50

Kresse G, Furthmüller J (1996b) Efficient iteratives schemes for ab initio total-energy calculations using a plane wave basis set. Phys Rev B 54:11169. https://doi.org/10.1103/PhysRevB.54.11169

Kresse G, Hafner J (1993) Ab initiomolecular dynamics for liquidmetals. Phys Rev B 47(1):558

Kresse G, Hafner J (1994) Abinitiomolecular-dynamics simulation of the liquid-metal-amorphous-semiconductor transition in germanium. Phys Rev B 49(20):14251

Lee D-H (2000) Quant Mech. http://hitoshi.berkeley.edu/221B-S01/221A.pdf
Lee C, Yang W, Parr RG (1988) Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. Phys Rev B 37(2):785. https://doi.org/10.1103/PhysRevB.37.785
Li SS (2012) Semiconductor physical electronics. Springer, Berlin
Liu B, Zhou K (2018) Recent progress on graphene analogous 2D nanomaterials: properties, modelling and applications. Progr Mater Sci. https://doi.org/10.1016/j.pmatsci.2018.09.004
Liu C, Feng W, Yao Y (2011) Phys Rev Lett 107:076802. https://doi.org/10.1103/physrevlett.107.076802
Lu W, Chen J, Li W, Yu J, Liu Z (2012) Adensity-functional theory study on the chemisorptions of $Ag(111)/O_2$ surface. Procedia Eng 31:671–675
Paulsson M, Brandbyge M (2007) Transmission eigenchannels from nonequilibrium Green's functions. Phys Rev B 76(11):115117. https://doi.org/10.1103/PhysRevB.76.115117
Monkhorst HJ, Pack JD (1976) Special points for brillouin-zone integration. Phys Rev B 13:125188–125192. https://doi.org/10.1103/PhysRevB.13.5188
Mulliken RS (1955) Electronic population analysis on LCAO-MO molecular wave functions. I. J Chem Phys 23(10):1833–1840. https://doi.org/10.1063/1.1741877
Nemnes GA, Racec PN (2004) Nano-transistors in the Landauer-Büttiker formalism. J Appl Phys 96:1. https://doi.org/10.1063/1.1748858
Osborn TH, Farajian AA (2014) Silicene nanoribbon as carbon monoxide nano sensor with molecular resolution. Nano Res 7:945–952. https://doi.org/10.1007/s12274-014-0454-7
Pan W, Zhu T (1998) First-principles study of the structural and electronic properties of ethylene adsorption on Si(100)-(2x1) surface. J Chem Phys 107:10. https://doi.org/10.1063/1.474753
Paola DP, Osamu K, Bruno O, Claudio Q, Tomonobu N, Masakazu A, Guy LL (2012) Multilayer silicene nanoribbons. Nano Lett 12(11):5500–5503. https://doi.org/10.1021/nl302598x
Patrick V, Paola DP, Claudio Q, Jose A, Emmanouil F, Maria C (2012) Silicene: compelling experimental evidence for graphene like two-dimensional silicon. Phys Rev Lett 108(15):155501. https://doi.org/10.1103/PhysRevLett.108.15
Perdew JP (1991) Electronic structure of solids. In: Ziescheand P, Eschrig H (eds) Akademie, Berlin, p 11
Perdew JP, Chevary JA, Vosko SH, Jackson KA, Pederson MR, Singh DJ, Fiolhais C (1992) Atoms, molecules, solids and surfaces: applications of the generalized gradient approximation for exchange and correlation. Phys Rev B 46:6671. https://doi.org/10.1103/PhysRevB.46.6671
Perdew JP, Burke K, Wang Y (1996) Erratum: generalized gradient approximation forth e exchange-correlation hole of a many-electron system. Phys Rev B 54:16533. https://doi.org/10.1103/PhysRevB.57.14999
Perdew JP, Burke K, Ernzerhof M (1996) Generalized gradient approximation made simple. Phys Rev Lett 77:3865–3868. https://doi.org/10.1103/PhysRevLett.77.386
Poole CP Jr, Owens FJ (2003) Introduction to nanotechnology. Wiley & Sons, Hoboken
Prasongkit J, Amorim RG, Chakraborty S, Ahuja R, Scheicher RH, Amornkitbamrung V (2015) Highly sensitive and selective gas detection based on silicene. J Phys Chem C 119(29):16934–16940. https://doi.org/10.1021/acs.jpcc.5b03635
Rappoport, Crawford NRM, Furche F, Burke K (2009) Which functional should I choose? In: Solomon EI, King RB, Scott RA (eds) Computational inorganic and bioinorganic chemistry. Hoboken Wiley & Sons, Inc., Wiley, Chichester
Rashid H, Koel A, Rang T (2018) Simulations of propane and butane gas sensor based on pristine armchair graphene nanoribbon. Published under license by IOP Publishing Ltd., IOP Conference Series: Materials Science and Engineering, vol 362, International Conference on Smart Engineering Materials (ICSEM 2018) 7–9 Mar 2018, Bucharest, Romania
Rashid MH, Koel A, Rang T (2020) Simulations of graphene nanoribbon field effect transistor for the detection of propane and butane gases: a first principles study. Nanomaterials (Basel, Switzerland) 10(1):98. https://doi.org/10.3390/nano10010098

Setiyo M, Soeparman S, Hamidi N, Wahyudi S (2017) Cooling effect characteristics of a ½ cycle refrigeration system on an LPG fuel system. Int J Refrig 82:227–237

Singh P, Abedini Sohi P, Kahrizi M (2021a) Finite element modelling of bandgap engineered graphene FET with the application in sensing methane thiol biomarker. Sensors 21:580. https://doi.org/10.3390/s21020580

Singh P, Sohi PA, Kahrizi M (2021b) Effect of point defects in armchair graphene nanoribbons for biosensing of Methane thiol biomarkers: a DFT study. In: 2021 IEEE 21st international conference on nanotechnology (NANO), pp 142–145. https://doi.org/10.1109/NANO51122.2021.9514321

Singh P, Abedini Sohi P, Kahrizi M (2022) Insilico design and analysis of Pt functionalized graphene-based FET sensor for COVID-19 biomarkers: a DFT coupled FEM study. Phys E Low-dimensional Syst Nanostruct 135:114972. https://doi.org/10.1016/j.physe.2021.114972. ISSN: 1386-9477

Solomon EI, Scott RA, King RB (2013) Computational inorganic and bioinorganic chemistry. Wiley & Sons

Song YL, Zhang S, Lu DB, Xu HR, Wang Z, Zhang Y, Lu ZW (2013) Band-gap modulations of armchair silicene nanoribbons by transverse electric fields. Eur Phys J B 86:488. https://doi.org/10.1140/epjb/e2013-31078-4

Synopsis: Quantumwise Atomistix Tool Kitversion2018.0. Available from: http://www.synopsis.com

Takeda K, Shiraishi K (1994) Phys Rev B Condens Matter Mater Phys 50:14916–14922

Tarun T, Singh P, Kaur H, Walia GK, Randhawa DKK, Choudhary BC (2021) Defective GaAs nanoribbon-based biosensor for lung cancer biomarkers: a DFT study. J Mol Model 27(9):270. https://doi.org/10.1007/s00894-021-04889-9. PMID: 34459994

vanMourik T, Bühl M, Gaigeot MP (2011) Density functional theory across chemistry, physics and biology. Philos Trans A Math Phys Eng Sci 372:20120488. https://doi.org/10.1098/rsta.2012.0488

Wang X, Li H, Wang (2012) Phys Chem Chem Phys 14(9):3031

Wang R, Xu MS, Pi XD (2015) Chemical modification of silicene. Chin Phys B 24(8):086807. https://doi.org/10.1088/1674-1056/24/8/086807

Wei W, Dai Y, Huang B (2017) Hydrogenation of Silicene on Ag(111) and formation of half-silicane. J Mater Chem A 34(5):18128–18137. https://doi.org/10.1039/C7TA03791K

Werpy MR, Burnham A, Bertram K (2010) Propane vehicles : status, challenges and opportunities. Center for Transportation Research Energy Systems Division, Argonne National Laboratory, Illinois

Yamacli S (2014) Comparison of the electronic transport properties of metallic graphene and silicene nanoribbons. J Nanoparticle Res 16(8):2576. https://doi.org/10.1007/s11051-014-2576-y

Yamazoe N (2005) Toward innovations of gas sensor technology. Sens Actuators, B Chem 108:2–14. https://doi.org/10.1016/j.snb.2004.12.075

Zheng B, Zhang C (2012) Nanoscale Res Lett 7:422

Recent Advancement of Artificial Intelligence in COVID-19: Prediction, Diagnosis, Monitoring, and Drug Development

Priya Rachel Bachan, Udit Narayan Bera, and Prakriti Kapoor

Abstract This paper intends to summarize the important applications of Artificial Intelligence and Machine Learning during the COVID-19 outbreak. In recent past, world has seen a massive pandemic due to COVID-19 virus outbreak, which affected millions of people across all nations. There have been tremendous implications and applications of Artificial Intelligence and Machine Learning, along with deep learning and Neural Networks, in the field of medical science, which has expanded the traditional dynamics of medical science and enhanced the decision-making as well as treatment delivery to the patients. Artificial Intelligence- and Machine Learning-based models have provided tools in numerous fronts, through which this pandemic can be overcome. In this paper, we have emphasized on the various important applications of Artificial Intelligence- and Machine Learning-based algorithms and models in the medical field for predicting the number of COVID-19-affected patients, diagnosing and detecting the virus, monitoring its spread as well as development of drugs and vaccines.

Keywords Artificial Intelligence · Machine Learning (ML) · Deep learning · AI and ML in COVID-19 · COVID-tech · AI in health care

P. R. Bachan (✉) · P. Kapoor
Computer Science and Engineering, Gyan Ganga Institute of Technology and Sciences, Jabalpur, India
e-mail: bachanpriya20@gmail.com

U. N. Bera
Electronics and Communication Engineering, Gyan Ganga Institute of Technology and Sciences, Jabalpur, India
e-mail: unbiitg@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_28

1 Introduction

The rapid spread of corona virus disease 2019 (COVID-19) caused a widespread distress and panic among people all over the world. By the end of 2019, there broke out a novel infectious corona virus disease that first spread from Wuhan, China, and then later spread out to 210 countries world-wide (Barlow 2004). COVID-19 is a deadly disease that targets the person's immunity and infects the respiratory tract and in most of the cases may even lead to death (Keni et al. 2020). Among many branches of Computer Science, Artificial Intelligence (AI) and Machine Learning (ML) are the top trending technologies, which have flourished remarkably in the last few years. The primary driving force for AI is to bring solutions to those problems which are not easily understood, extremely complex and require extraordinary levels of intelligence. Also, this technology has an extraordinaire in image processing (Dişken et al. 2017), speech recognition (Mei et al. 2021), and text to speech conversions (Columbus 2018), which assist in increasing productivity in health care and businesses. Pertaining to these, various AI and ML techniques leveraged the healthcare sector in prediction, diagnosis, and monitoring and drug development of COVID-19. Two popular screening applications of ML are Deep Convolutional Neural Network ResNet-101 by Ardakani et al. (2020) and Convolutional Neural Network Dark-CovidNet Architecture by Ozturk et al. (2020) which proposed optimal accuracies of around 98% and 99%, respectively.

There has been a tremendous growth in the research works in the field of AI in the last two decades, as is depicted in Fig. 1.

In this paper, we intend to review recent technological advancements of AI and ML in the medical field with regards to the COVID-19 pandemic and analyze their implications such as predicting the number of COVID-19-affected patients, diagnosing the disease, monitoring its spread as well as developing drugs and vaccines for the purpose of its cure and control. This paper outlines numerous fronts of AI and ML which have led to an enormous development of the healthcare industry, giving rise to expedite advancement of medical science. Also, we have analyzed the effectiveness of AI-based tools and algorithms during the COVID-19 outbreak and their effectiveness in handling extreme and risky events.

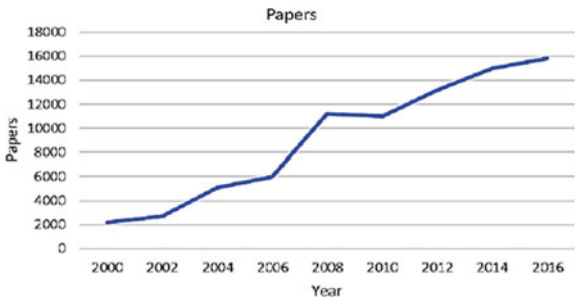


Fig. 1 Growth of annual research done in AI sector vs annual research done in AI sector in medical field (Mahajan et al. 2019)

2 Predicting Number of COVID-19 Patients and Its Growth

Artificial Intelligence along with Machine Learning algorithms is widely used all over the world to forecast disease, weather, stock market, and much more, to help enhance the decision-making and action-taking for the future course (Jiang et al. 2017). ML algorithms along with Neural Network models have been successful in predicting the number of affected patients of a particular disease in advance (Hu et al. 2002). Research (Rustam et al. 2020) confirmed that the accuracy of AI-based methods and tools for the multi-step forecasting of the course of the COVID-19 virus was high with estimated average errors of 1.64%, 2.27%, 2.14%, 2.08%, 0.73% for 6-step, 7-step, 8-step, 9-step, and 10-step forecasting respectively. In this section, the applications of AI- and ML-based forecasting mechanisms to predict the number of COVID-19-affected patients are discussed.

2.1 Supervised Machine Learning Models

Studies (Rustam et al. 2020; JHU 2020) have shown that the four forecasting models, namely linear regression (LR) model, least absolute shrinkage and selection operator (LASSO) model, support vector machine (SVM) model, and exponential smoothing (ES) model, have contributed to the current COVID-19 crisis. These supervised ML regression models were thoroughly trained by providing patient stats of COVID-19 obtained from John Hopkins University (JHU 2020). This training data was processed and divided as training parameters (85% of the records) and testing parameters (15% of the records). The research based on the time series datasets and the results claiming an accuracy of more than 90%, proved that the ML model forecasting can be very effective and beneficial for the decision-making to restrain the infections and consequences of the COVID-19 pandemic (Pahuja and Nagabhushan 2021). Previously, many related works have concluded that ML and AI techniques hold importance in predicting and diagnosing COVID-19. However, Muhammad et al. (2021) proposed a supervised Machine Learning being the only work so far done using epidemiology labeled datasets. The dataset contained demographic as well as clinical data, including 263,007 instances with 41 features. This study incorporated different supervised learning techniques for classification: (a) Logistic Regression, (b) Naïve Bayes Algorithm (c) Decision Tree Algorithm, and (d) Support Vector Machine (SVM) (Fig. 2).

2.2 Cloud-Based Machine Learning Models

With the emergence of advanced computing technologies, there has been a boost in the dimension of large datasets. Parallel to this, Cloud Computing offers flexible

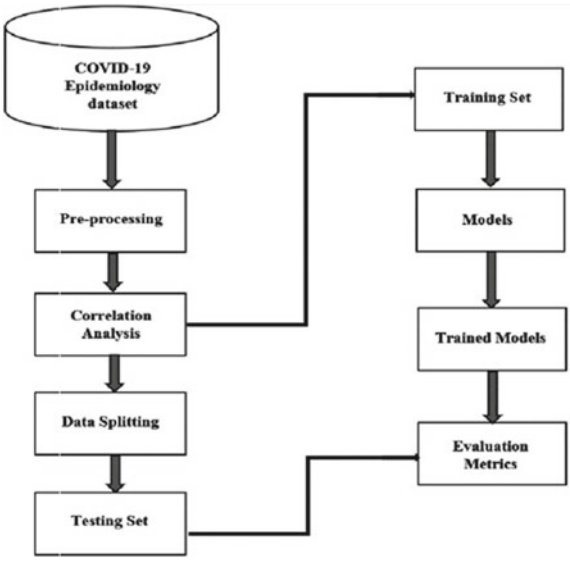


Fig. 2 Methodology to develop supervised learning model to predict COVID-19

and extensible storage resources, which are now widely adapted by many research communities.

Cloud Computing assists in promptly speed up the prediction process with high-speed computations and ciphering cloud data centers. Deployment of models on cloud data enters demonstrated in Fig. 1. The dataset was taken from 'Our World in Data: COVID-19 Dataset' (https://github.com/owid/covid-19data/tree/master/public/data/) and was continually updated from Situation Reports: WHO (https://www.who.int/emergencies/diseases/novel-coronavirus2019/situation-reports). Their system employed HealthFog framework (Tuli et al. 2020a) (a smart healthcare system based on deep learning which can automatically diagnose heart diseases in fog computing and IoT integrated environments) and FogBus (Tuli et al. 2019) (lightweight fog computing framework based on blockchain). These two frameworks were utilized to deploy collective learning approach to forecast variegated aspects of pandemic including the number of required staff to manage hospitals and affected patients. The study analyzed that the cost of regular tracking of patients was 1.2 USD per day (Tuli et al. 2020b) (Fig. 3).

Support Vector Machine was incorporated with cross-fold validation testing, rendering an accuracy of 98.15% with 15-fold validation (Ur et al. 2021). The study resulted in better efficiency and accuracy in comparison to the other state-of-the-art ML methodologies for predicting COVID-19. Further, cloud-based AI solutions were developed by Alibaba (Kose et al. 2021) to aid China during COVID-19 outbreak by predicting number, extent and areas of infected cases through surveillance mechanisms as well as track chain methods and the information was computed and evaluated through multi-edge and cloud layers. The study affirms that these solutions were employed with an accuracy of 98% in real time, within varied sectors in China.

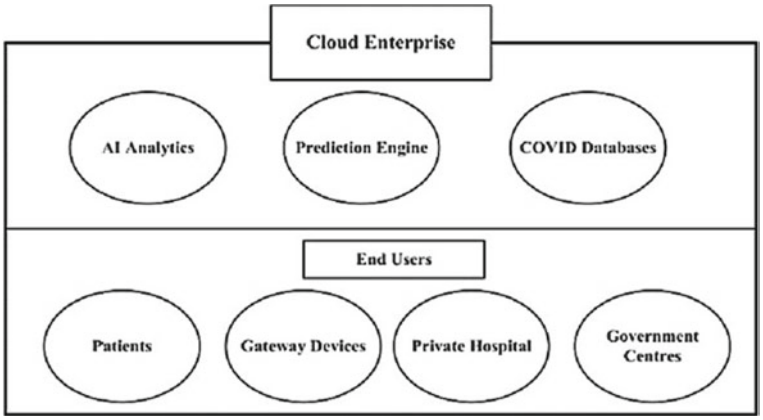


Fig. 3 Deployment of models on cloud data centers

2.3 Time Series Forecasting Model

The time series analysis is the best way for forecasting using ML where the problem is taken as regression, where the forecast is the target feature (dependent variable) and the time is the input feature (independent). Forecasting completely depends on the past trends and data. In case of respiratory disease such as COVID-19, where there is not much recovery time once the patient gets severely infected, forecasting the rate of increase or decrease in the number of infected cases, can greatly help in preparing the resources and measures to be taken in the future. One such time series forecasting model is autoregressive integrated moving average (ARIMA), which is used to predict future trends (https://www.kaggle.com/sudalairajkumar/covid19in-india). A unit root test called automatic Dickey–Fuller is used to estimate and monitor the continuity of the time series. Domenico et al. (Pinter et al. 2020) demonstrated that the ARIMA model has also been successful in predicting the prevalence and incidence of COVID-19 with a confidence interval of 95%, whose parameters were evaluated by employing partial autocorrelation (PACF) correlogram and autocorrelation function (ACF) graph.

2.4 Hybrid Machine Learning Models

The Hybrid Machine Learning methods of adaptive network-based fuzzy inference system (ANFIS) and multi-layered perceptron-imperialist competitive algorithm (MLP-ICA) have been proved to be successful in predicting the rate of infected patients and also the mortality rate with a higher level of accuracy, as suggested by a study conducted in Hungary (Prakash et al. 2020). In that study, the sample data was considered for two scenarios which were used to train both ANFIS and MLP-ICA;

for odd and even days. The prediction results were validated with promising results for a period of nine days, confirming its accuracy and determination coefficients ranging from 0.9963 to 0.9987, which wouldn't get hampered unless significant error occurs, since further research is required for improving the quality of procedure and better evaluation of models. Further, another research (Khalilpourazari and Hashemi Doulabi 2021) puts forward an optimized search methodology built around Reinforcement Learning (RL) and Evolutionary Computation (EC). This research proposes a hybrid algorithm to predict COVID-19 in Quebec, Canada (https://www.quebec.ca/en/health/healthissues/a-z/2019-coronavirus/). According to the results, the methodology predicted the peak of COVID-19 infection, and complex sensitivity analysis allows plotting future scenarios to help healthcare professionals and policymakers.

2.5 Regression and Classification Models

There are various other AI- and ML-based models that have caught the attention of researchers, namely K-nearest neighbor (KNN), Decision Tree Classifier (DTC), Logistic Regression, Linear Regression, SVM, etc. Random Forest Regressor and Random Forest Classifier are two such models which have beat the other AI- and ML-based prediction models with regard to accuracy and coefficient of determination of COVID-19. Research shows that they have outperformed the other models on the basis of coefficient of determination and accuracy (Vaishya et al. 2020) described in Fig. 4. Figure 5 shows the comparison among the accuracies of different models for COVID-19 data, built on COVID-19 dataset.

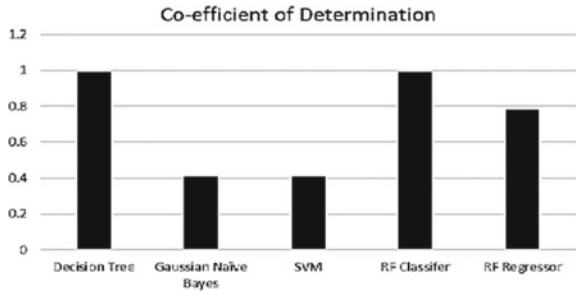


Fig. 4 Coefficient of determination of different models for COVID-19 data

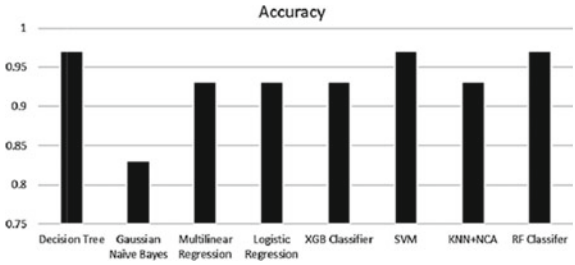


Fig. 5 Accuracy of different models for COVID-19 data

3 Diagnostics and Detection of COVID-19

Early detection and accurate diagnosis of COVID-19 helps the medical team with faster and steadier decision-making and taking appropriate actions accordingly. AI-and ML-based diagnostic mechanisms have boosted the efficiency of medical industry to provide faster, cost-effective, and precise treatment procedures to the affected patients, leading to better treatment and increased recovery rates.

3.1 Generative Adversarial Networks

Since, COVID-19 is a respiratory disease, the infection in lungs and nasal cavity could be determined generally by chest CT scans. Jin et al. (2020) in their research proposed one such model, which is based on Generative Adversarial Networks (GAN) and deep transfer learning model. The model consists of an actual dataset of 746 images, which is divided into training parameters (90%) and testing parameters (10%). After using GAN as an image augmenter, the 90% data is further divided into 80% for training purpose and the remaining 20% for validation purpose. ShuffleNet is the most accurate and optimal model for diagnosis of COVID-19 virus, since it resulted in 85.33% accuracy in overall performance metrics in terms of precision, recall, and F1 score as listed in Table 1.

GAN and deep transfer learning model for the diagnosis and detection of COVID-19 in insubstantial chest X-rays was proposed by Loey et al (2020). The research focused on gathering all the available chest X-ray images and produce new images using the GAN network to help detect the coronavirus. The research also claimed that

Table 1 Testing phase: comparison metrics for different models using GAN

Metric/model	Resnet50 (%)	ShuffleNet %()	MobileNet (%)
Precision	84.98	85.33	78.22
Recall	85.14	85.33	77.45
F1 score	85.06	85.33	77.83

the main deep transfer model which was selected was GoogleNet, since it achieves a testing accuracy of 100% and a validation accuracy of 99.9%.

3.2 Artificial Intelligence-Inspired Models

Huang et al. (2020) proposed a model called AIMDP or Artificial Intelligence-inspired model for COVID-19 diagnosis and prediction for patient response to treatment, which has two main functions; the diagnosis module, that predicts the number of COVID-19-affected patients and the prediction module that predicts the response ability of those patients toward the COVID-19 virus. They employed one of the most widely used deep learning libraries—TensorFlow framework, for the purpose of training the process. This research claimed that the accuracy of this model reached up to a 97.14%. Also, COVID-19 can be detected by respiratory tract infection or pneumonia. An AI system was built by Jin et al. to detect and analyze COVID-19. A total of 1136 training cases were used to train the system, out of which 723 were positive cases from five hospitals. A resulting system sensitivity of 0.974 was obtained. Chest CT images are a valuable feature that aids to detect the COVID-19 virus efficiently, and researchers have been trying to decompose these images using deep learning in order to detect the features of the virus. A team from Tianjin Medical University Cancer Institute and Hospital (Huang et al. 2020) collected samples of 180 patients for CT scan, having severe viral pneumonia way earlier than coronavirus epidemic, and 79 patients having confirmed COVID-19. The study required the training and testing parameters to be provided randomly to train the Convolutional Neural Networks (CNN)-based algorithm, which resulted in an accuracy of 89.5% to detect COVID-19 virus through chest CT.

3.3 Convolutional Neural Network (CNN) Models

Convolutional Neural Network is a class of Neural Networks in deep learning, which is generally used for image processing by resolving the images into sub-forms that could be processed more easily, keeping intact the important features useful for prediction. A CNN architecture was utilized by Kumar et al. (2022) which employed U-net. It was trained on interpreted dataset, gathered from 842 confirmed patients of coronavirus, and evaluated for lung opacity segmentation. The patients were made to undergo chest CT scans in the Tongji Hospital, China. Another neoteric research introduced a custom DL architecture called SARS-Net, which aided in detecting the irregularities present in the chest X-rays of patients, using a Computer-Aided Diagnosis system, integrating Graph Convolutional Networks and Convolutional Neural Networks (Wang et al. 2020). A dataset of 13,975 CXR images was taken, extending to 13,870 patients. This model was successful in achieving a 97.60% of accuracy as well as a 92.90% of sensitivity on the testing dataset. Image distribution

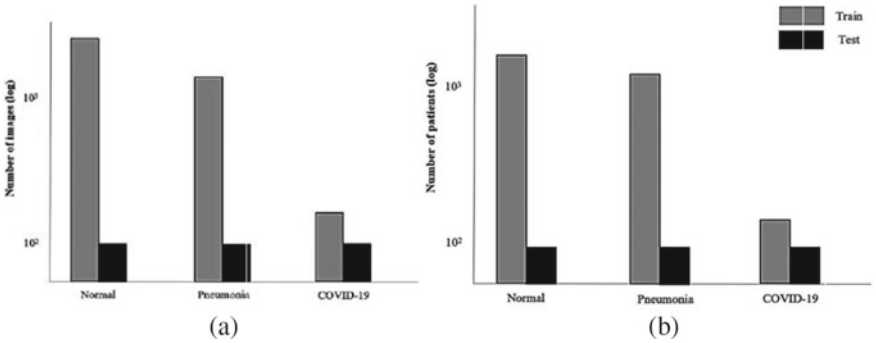


Fig. 6 **a** Image distribution (CXR images) of COVID-19 dataset according to infection type. **b** Number of patients of COVID-19 dataset according to infection type

(CXR images) of COVID-19 dataset according to infection type and number of patients of COVID-19 dataset according to infection type are depicted in Fig. 6a, b, respectively.

4 Monitoring the Spread of the Virus

AI and ML can efficiently build intelligent systems that can help to monitor the spread of COVID-19 virus among individuals. Such systems or tools can be effectively utilized to extract and track the visual features of the virus.

4.1 Contact Tracing and Remote Monitoring

Remote monitoring and contact tracing of the individuals can help to detect the infection levels by visually recognizing the 'hot spots' (Jumper et al. 2021). Contactless monitoring can be done by monitoring blood pressure, heart rate, oxygen saturation levels, and also using X-rays and MRI scans. The study (Agbehadji et al. 2020) employed deep DR, a DL-based method which comprehends the high-stage features of the medications, and was successful in producing a high-performance of 90.8% AUC. Various countries incorporated many centralized, decentralized, and hybrid AI and ML techniques in form of applications as mentioned in Table 2.

Table 2 Few contact tracing applications employed by countries

Country	AI/ML contact tracing app	Launched on
China	Conjunction with Alipay	Little info
Australia	COVIDSafe	April, 2020
Bulgaria	ViruSafe	May, 2020
Cyprus	CovTracer	May, 2020
India	AarogyaSetu	April, 2020
Italy	Immuni	May, 2020
Iran	Mask.ir	May, 2020

4.2 AI-Based Models for Case Detection and Monitoring

During the COVID-19 pandemic, China deployed AI-based temperature screening in public places, which aided in detecting symptoms in people and putting the suspected patients it no isolation (Petropoulos 2020). Also lately, thermal imaging cameras have been employed to quickly and accurately detect one of the major symptoms of the coronavirus—elevation in body temperatures. Intelligently, monitoring the key symptoms can help avoid severity of the disease and rapidly inform the concerned authorities to enhance adequate action-taking.

5 Development of Drugs and Vaccines

In addition to three pre-existing categories of vaccines (vaccines for pathogen, subunit vaccines and vaccines for nucleic acid), a newest category has been included, called COVID-19 vaccine. Understanding the structure and genetic sequence of a specific protein before acknowledging its unique 3D structure is critical for vaccine development. Since this procedure can be time-taking and complex, AI systems can be of a great help to simplify this process. An intelligent system, called ALPHAFOLD, was developed after years-long research on large genomic datasets to predict 3D models of the proteins, which are exceedingly accurate. A research team from a vaccination biotech company developed AI and cloud computing-based models for COVID-19 drug and vaccine development (Deoras 2020). They successfully modeled the COVID spike protein and design and developed a synthetic COVID-19 vaccine, named COVAX19. Researchers from MIT have implemented a novel approach, where they have applied optimized ML-based methodology that opts for amides (short organic compound fibers) that are expected to offer giant vaccine numbers. The 'OptiVax' style code incorporates ways of coming up with new peptide medicines, evaluating existing vaccines and increasing their composition (Beck et al. 2020). The researchers took 4690 samples of COVID-19 genomes to be evaluated and the removal of undesired mutation-rate peptides and genetic makeup of the different populations was specifically focused. A DL-based drug–target interaction model,

called molecule transformer-drug–target interaction (MT-DTI), was proposed by Beck et al. This model can effectively predict drugs for COVID-19 virus by utilizing SMILES strings and amino-acid sequences, to sort and classify target proteins with 3D crystal structures (Öztürk et al. 2018), and identify the commercially available drugs that could specifically target and destroy the viral components of the COVID-19 virus. The results led to the identification of six promising medicines; Remdesivir, Atazanavir, Efavirenz, Ritonavir, Dolutegravir, and Kaletra. A DTI model called DeepDTA, based on DL, was proposed, which was an end-to-end CNN-based model. This model can automatically detect useful features from raw protein sequences. For processing proteins and ligands' similarities, the model made use of Smith–Waterman (S–W) and PubChem Similarity algorithms, respectively (Shim et al. 2021). The study concluded that this DL-based model which used 1D representations of the drug and the target was a non-cognitive approach. For the purpose of reading and evaluating the representations from raw protein sequences, CNN blocks were employed, which later fed all this information to DeepDTA, a fully connected block.

6 Conclusion

This paper analyzed various possible applications of Machine Learning and Artificial Intelligence to predict number of COVID-19-affected patients and its growth, diagnostics and detection of virus as well as monitoring the speed of the virus. It is also concluded that these technologies are also helping in development of drugs and vaccines, which in turn can aid in stopping the spread of COVID-19. This survey offers a detailed overview of the existing as well as latest state-of-the-art methodologies of AI and ML with superlative accuracies, which have proven to be exceedingly efficient in boosting the potentiality of the healthcare industries as well as the concerned authorities for the purpose of detection, control, cure, and management of a pandemic such as substantial as the COVID-19.

References

Agbehadji IE, Awuzie BO, Ngowi AB, Millham RC (2020) Review of big data analytics, artificial intelligence and nature-inspired computing models towards accurate detection of COVID-19 pandemic cases and contact tracing. Int J Environ Res Public Health 17(15):5330

Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. Comput Biol Med 121

Barlow DH (2004) Anxiety and its disorders: the nature and treatment of anxiety and panic. Guilford press

Beck BR, Shin B, Choi Y, Park S, Kang K (2020)Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model. Comput Struct Biotechnol J 18:784–790

Columbus L (2018) 10 charts that will change your perspective on Artificial Intelligence's growth. Forbes. https://www.forbes.com/sites/louiscolumbus/2018/01/12/10-chartsthat-will-change-your-perspective-on-artificial-intelligences-growth/?sh=389af1974758. Last accessed 4 Sept 2022

Coronavirus diseases (COVID-19) in Quebec. https://www.quebec.ca/en/health/healthissues/a-z/2019-coronavirus/. Last accessed 02 Oct 2022

Deoras S, How ML is assisting in development of covid-19 vaccines. https://analyticsindiamag.com/how-ml-is-assisting-in-development-of-covid-19-vaccines/. Last accessed 28 Sept 2021

Dişken G, Tüfekçi Z, Saribulut L, Çevik U (2017) A review on feature extraction for speaker recognition under degraded conditions. IETE Tech Rev 34(3):321–332

Hu Z, Ge Q, Li S, Jin L, Xiong M (2020) Artificial intelligence forecasting of covid-19 in china. arXiv preprint arXiv:2002.07112

Huang L, Han R, Ai T, Yu P, Kang H, Tao Q, Xia L (2020):Serial quantitative chest CT assessment of COVID-19: a deep learning approach. Radiol Cardiothor Imag 2(2):e200075

JHU: John Hopkins University (2020) COVID-19 dashboard by the center for systems science and Engineering (CSSE) at Johns Hopkins University (JHU). https://www.coronavirus.jhu.edu/map.html. Last accessed 27 Apr 2021

Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, Wang Y, Dong Q, Shen H, Wang Y (2017) Artificial intelligence in healthcare: past, present and future. Stroke Vasc Neurol 2(4)

Jin S et al (2020) AI-assisted CT imaging analysis for COVID-19 screening: building and deploying a medical AI system in four weeks. medRxiv pp. 2020.03.19.20039354

Jumper J, Evans R, Pritzel A et al (2021) Highly accurate protein structure prediction with alphafold. Nature 596:583–589

Keni R, Alexander A, Nayak PG, Mudgal J, Nandakumar K (2020) COVID-19: emergence, spread, possible treatments, and global burden. Front Public Health 8:216

Khalilpourazari S, Hashemi Doulabi H (2021) Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec. Ann Oper Res, pp 1–45

Kose U, Grima S, Özen E, Boz H, Spiteri J, Thalassinos E (2021) Using artificial intelligence techniques for economic time series prediction. Contemp Issues Behav Fin 101:13e28

Kumar A, Tripathi AR, Satapathy SC, Zhang YD (2022) SARS-Net: COVID-19 detection from chest x-rays by combining graph convolutional network and convolutional neural network. Pattern Recogn 122:108255

Loey M, Smarandache F, Khalifa NEM (2020) Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning. Symmetry 12:651

Mahajan A, Vaidya T, Gupta A, Rane S, Gupta S (2019) Artificial intelligence in healthcare in developing nations: The beginning of a transformative journey. Cancer Res Stat Treat 2(2):182

Mei Y, Ye DP, Jiang SZ, Liu JR (2021) A particular character speech synthesis system based on deep learning. IETE Tech Rev 38(1):184–194

Muhammad LJ, Algehyne EA, Usman SS, Ahmad A, Chakraborty C, Mohammed IA (2021) Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset. SN Comput Sci 2:1–13

Our World in Data: COVID-19 Dataset. https://github.com/owid/covid-19data/tree/master/public/data/. Last accessed 03 Oct 2021

Öztürk H, Özgür A, Ozkirimli E (2018) DeepDTA: deep drug–target binding affinity prediction. Bioinformatics 34(17):i821–i829

Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Rajendra Acharya U (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med

Pahuja G, Nagabhushan TN (2021) A comparative study of existing machine learning approaches for Parkinson's disease detection. IETE J Res 67(1):4–14

Petropoulos G (2020) Innovation and competition policy. http://www.bruegel.org/2020/03/artificial-intelligence-in-the-fight-against-covid-19/. Last accessed 11 Sept 2021

Pinter G, Felde I, Mosavi A, Ghamisi P, Gloaguen R (2020) COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach. Mathematics 8(6):890

Prakash KB, Imambi SS, Ismail M, Kumar TP, Pawan YN (2020) Analysis, prediction and evaluation of covid-19 datasets using machine learning algorithms. Int J 8(5)

Rustam F, Reshi AA, Mehmood A, Ullah S, On BW, Aslam W, Choi GS (2020) COVID-19 future forecasting using supervised machine learning models. IEEE Access 8:101489–101499

Shim J, Hong ZY, Sohn I, Hwang C (2021) Prediction of drug–target binding affinity using similarity-based convolutional neural network. Sci Rep 11(1):4416

Situation Reports-WHO. https://www.who.int/emergencies/diseases/novel-coronavirus2019/situation-reports. Last accessed 03 Oct 2022

Sudalai Rajkumar, datasets. Available: https://www.kaggle.com/sudalairajkumar/covid19in-india. Last accessed 28 Apr 2021

Tuli S, Mahmud R, Tuli S, Buyya Fogbu R (2019) A blockchain-based lightweight framework for edge and fog computing. J Syst Softw

Tuli S, Basumatary N, Gill SS, Kahani M, Arya RC, Wander GS, Buyya R (2020a) HealthFog: an ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. Future Gener Comput Syst 104:187–200

Tuli S, Tuli S, Tuli R, Gill SS (2020b) Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing. Internet Things 11:100222

Ur A, Rahman S, Naseer I, Majeed R, Musleh D, Gollapalli MAS, Chabani S, Ibrahim N, Siddiqui SY, Khan MA (2021) Supervised machine learning-based prediction of covid-19. Comput Mater Continua 69:21–34

Vaishya R, Javaid M, Khan IH, Haleem A (2020) Artificial Intelligence (AI) applications for COVID-19 pandemic. Diab Metab Syndr 14(4):337–339

Wang L, Lin ZQ, Wong A (2020) Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Sci Rep 10(1):1–12

Analysis of Corner Truncated Rectangular Microstrip Patch Antenna for IoT Applications

Sonam Gour, Reena Sharma, and Gaurav Sharma

Abstract A miniaturized, rectangular, corner truncated microstrip patch antenna with FR4 substrate is proposed and simulated for the wearable sensor device IoT applications. The major challenges for developing any device for the healthcare system are sensor-feeding network and active worked element on the same base. The proposed antenna is feed by the microstrip patch line with the 50 Ω impedance. The upcoming resonance frequency is in the range of the ISM region or Wi-Fi technology. The resonating frequency of antenna is 2.44 GHz with the operating bandwidth of the device which is 2.41–2.47 GHz. Rectangle and circular shape slots are inserted in the design for enhancing the current path and direction of the electrical path. The simulated gain of antenna is 5.33 dB with the directivity of 4.76 dB. Small size and high performance are very desirable for the antenna design for the Internet of things (IoT) applications. The simulated result shows that it can be used for the IoT wearable sensor design units.

Keywords Microstrip patch antenna · IoT · Directivity · Return loss · Gain

1 Introduction

IoT enables different devices that are able to communicate, interpret, and sense the devices through various interconnects. IoT provides the different techniques for connecting the wearable antenna devices to the external connections. Device integration techniques require some more efficient techniques for proper functioning like miniaturization and biocompatibility (Soni and Gour 2021). Different types of miniaturization can be achieved by slot insertion and splitting the material parts by defective ground structure (Ghosh et al. 2011). The connections of the ground with the radiating rectangular patch are used for enhancing the bandwidth and radiation patterns (Rawat et al. 2021). The IoT-designed antenna is realized for the

S. Gour (✉) · R. Sharma · G. Sharma
Department of Computer Engineering, Poornima College of Engineering, Jaipur, India
e-mail: sonam.gour@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_29

2–4 GHz band applications, for the Wi-Fi applications and Bluetooth devices (Gupta et al. 2020). Smart sensing devices are connected for the wireless application for detecting the disease or any mishappening in the body. Use of IoT is increasing due to transmission of the lower amounts of data with low transmission rates, so the requirements of the bandwidth are reduced further, or narrow band antenna can be used for wide applications. The wearable sensors sense the unusual behavior of the connected tissue and generate the information to the connecting external device.

The use of antenna in the wearable device is increasing because with the help of the sensors it should be used for monitoring the home medical equipment like asthma, heart rate, and diabetes (Gour et al. 2022). It can be used for caring the patient and health monitoring. IoT systems are the combination of the communication system, antennas, sensors, and processors. IoT systems receive, process, and transmit the information to the Internet web. The integration of the IoT with the wearable device increases the automated process; further, it reduces the hardware and labor cost (Sabban 2022).

2 IoT and Sensors

The Internet of things (IoT) is new technique that uses for making our lives easier by providing communications between any electronic device and sensors. IoT uses smart sensors devices and the Internet for providing new solution of the problem related to business, government, public, and private sectors. IoT is an important invention that provides very intelligent system, smart devices and sensor. It is very useful in the developing regions for economic and industrial growth. The use of IoT is increasing in the retail and currency business. However, data and information security are a major concern and highly desirable. Besides many applications of Internet, it also opens various door to the hackers and cyber-attacks and puts the information in risk. IoT is very useful for providing the best solution to the information security issue and data rate. Different aspects of the IoT are shown in Fig. 1.

Use of IoT is increasing in the field of smart cities (Bingol et al. 2019), agriculture (Wenju et al. 2017) and parking system (Gotthard and Jankech 2018). Low data rates are used for the long-distance communications with the low power consumptions. The use of antenna for the widespread band is increasing due to the higher benefit of the band gaps. The insertion of the slots and structural modification in the design provide the enhancement in the bandwidth.

The structural of the antenna allows the users for switching the bandwidth between different frequency regions without changing the actual structure of the antenna fundamental structure of the antenna. Reconfigurability of the antenna is achieved by the tunable elements like capacitor and resistor. The use of antenna is increasing in the healthcare system. The inside sensor in the body senses the any tumor and increasing range of the sugar and blood pressure, and the data is analyzed by the

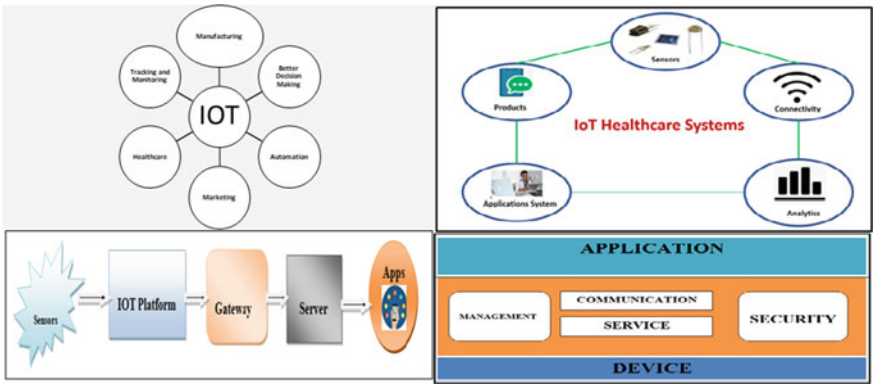


Fig. 1 Importance of Internet of things

external of the server. IoT also provides the connection between the sensor networks and nodes. A wireless sensor network is a topological arrangement of hundreds to thousands of sensor nodes that are placed either randomly or in accordance with some pre-defined scattering patterns over a geographic area of significance. A sensor is the networks that provide restrictions like limited power source, short communication radius, and limited processing capacity. It works efficiently with the limited region of interest.

3 Designing of Antenna

The basic antenna is used like loop, folded dipole, and dipole antenna. The influence of the metal is removed by using the patch antenna. When the metal antenna is close to the patch antennas back side, the performance of the antenna is increased but when the metal closes to the patch antenna front side, the performance of antenna drops. The performance of antenna is stable even when the metal closes on either side of the antenna. The schematic of antenna is represented in Fig. 2. The antenna includes the rectangular patch with the inset feeding techniques. FR4 material is used for the substrate with 0.035 height. Due to its easy availability and ease of fabrication process. Some miniaturization techniques are used for achieving the biocompatible behavior. Rectangular slots in the patch increase the current path and lowering the region of the return loss (Zhang et al. 2018). The design of the antenna as per calculation is represented in Fig. 2.

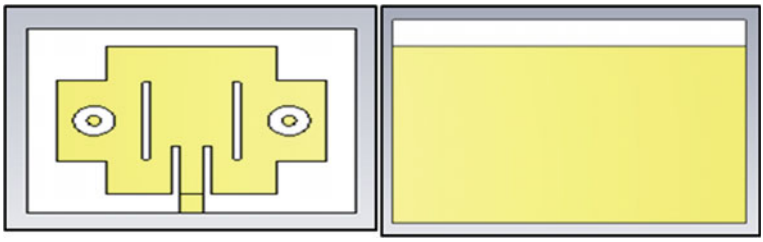


Fig. 2 Designed antenna and ground modification in antenna design for optimize the result

3.1 Mathematical Equation of the Antenna

The parameter is calculated by using the following formulas:

$$w = \frac{C_o}{2f_r\sqrt{\frac{\epsilon_r+1}{2}}} \tag{1}$$

$$\epsilon_{reff} = \frac{\epsilon_r+1}{2} + \frac{\epsilon_r-1}{2}\left(1+12\frac{h}{w}\right)^{-0.5} \tag{2}$$

$$L_{\text{eff}} = \frac{C_o}{2f_r\sqrt{\epsilon_{\text{reff}}}} \tag{3}$$

$$\Delta L = 0.412\frac{\left(\frac{w}{h}+0.264\right)(\epsilon_{\text{reff}}+0.3)}{(\epsilon_{\text{reff}}-0.258)\left(\frac{w}{h}+0.813\right)}$$

$$L = L_{\text{eff}} - 2\Delta L \tag{4}$$

CST design studio is used for simulating and designing the antenna. The antenna is designed and simulated by using the CST design studio suite. The simulated results show that it can be used for the IoT applications due to the operating range of the antenna (Table 1).

4 Results and Discussion

The achieved return loss of the antenna is −34.54 dB with the operating frequency of 2.41–2.47 GHz. The bandwidth of the simulated design is 60 MHz for the resonance frequency of 2.44 GHz. So, the miniaturized design can be used for the IoT applications as well as for the wearable design sensing device. The VSWR result in the figure shows that the designed antenna transmitted maximum amount of power

Table 1 Design parameters for the antenna design

Parameter	Dimension (mm)
Width of ground	47.2
Length of ground	38.36
Width of patch	38.56
Length of patch	30.78
Height of substrate	1.6
Height of ground	0.035
Length of inset feed	10.16
Width of inset feed	3.2

radiation, less than 1.5 result of the VSWR is required for transmitting more power and reflecting the lower value of the power (Fig. 3).

Directivity of any antenna is the ratio of radiation intensity in a given direction and radiation intensity in overall directions. All antennas are directed the signal to the specific direction. The directivity and gain of the simulated antenna are respectively 5.88 dB and 4.76 dB. Gain of any antenna is defined by the product of directivity and its efficiency. For the identical losses condition gain and directivity of the antenna should be the same. Gain is used for calculating the maximum radiation power when

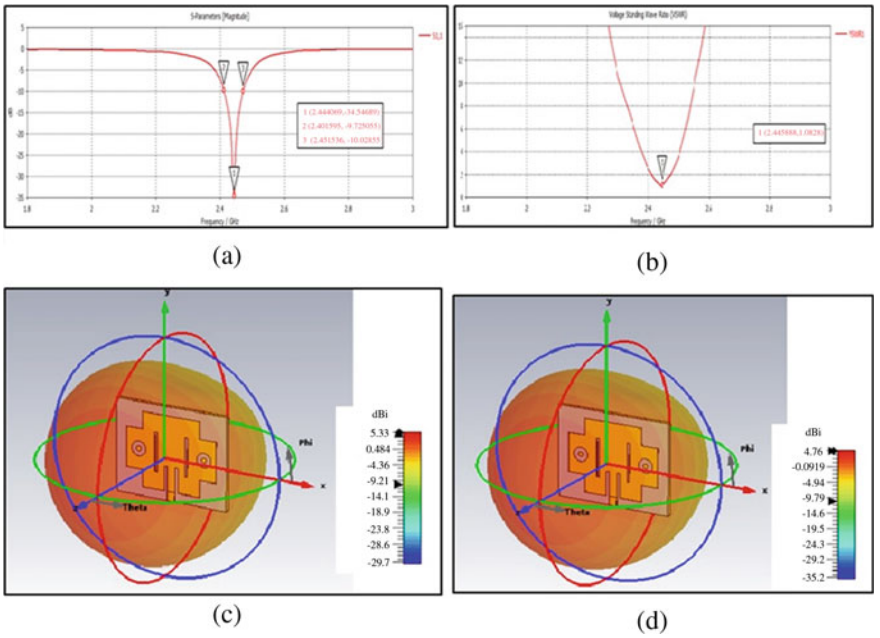


Fig. 3 Results of antenna **a** return loss, **b** VSWR, **c** directivity, **d** gain

any designed antenna power is known. The simulated result of gain and antenna shows that it can be used for the designing of any sensing device.

The radiation of the electrical field and magnetic field is measured by the simulation design and experimental process of the architecture. The value of the E-field, H-field, and surface current is shown in Fig. 4. The generated value of the E-field and H-field increases due to extra slot and cut structure in the design. The generated value of the E-field is 0.000684 J/m^3. The generated value of the magnetic field is 0.000798 J/m^3. The rectangular and circular slots are increased the surface current by some amount due to its better performance. The value of the surface current is 66 A/m (Table 2).

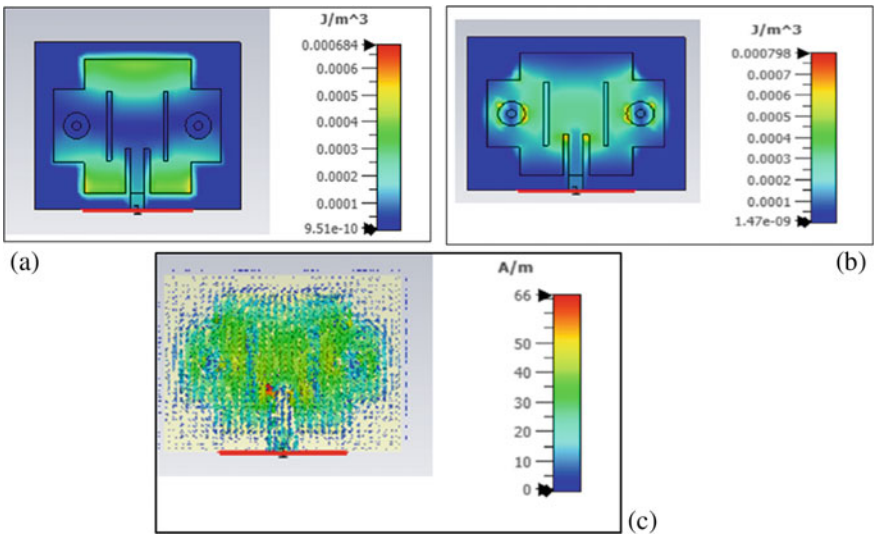


Fig. 4 E-field, H-field, and surface current of the antenna

Table 2 Comparison of existing antenna with proposed antenna

References	Design technique	Substrate	Bandwidth
Mao et al. (2018)	Microstrip line	Rogers RO 4003	3.4–3.6 GHz and 5.1–6.1 GHz
Diallo et al. (2018)	Microstrip line	Rogers RO 4003	915 MHz and 868 MHz
Hassan et al. (2017)	Microstrip line	FR4	8.9 GHz and 3.1–12GHz
Proposed	Microstrip line	FR4	2.41–2.47 GHz

5 Conclusion

The major goal of any antenna designing for the IoT applications is size miniaturization while maintaining the performance of the antenna. In this paper, we have successfully simulated the small-size antenna for IoT works. A miniaturized corner truncated antenna is designed and simulated. The dimension of the antenna is 47.2 mm * 38.36 mm. The antenna works for the Wi-Fi application due to its operating frequency which is 2.44 GHz. It is providing the 60 MHz bandwidth for working efficiently. The performance of the antenna is measured in terms of directivity, radiation pattern, electric field, and magnetic field. The proposed design of the antenna is well suited for the communications requirement for the small mobile sensor elements.

References

Bingol E, Kuzlu M, Pipattanasompom M (2019) A LoRa-based smart streetlighting system for smart cities. In: 2019 7th International Istanbul smart grids and cities congress and fair (ICSG)

Diallo K, NGom A, Diallo A, Ribero J-M, Dioum I, Ouya S (2018) Efficient dual-band pifa antenna for the internet of things (iot). In: 2018 IEEE conference on antenna measurements & applications (CAMA). IEEE, pp 1–4

Ghosh B, Haque SM, Mitra D (2011) Miniaturization of slot antennas using slit and strip loading. IEEE Trans Antennas Propag 59(10):3922–3927

Gotthard P, Jankech T (2018) Low-cost car park localization using RSSI in supervised LoRa mesh networks. In: 2018 15th Workshop on positioning, navigation and communications (WPNC)

Gour S, Chaudhary P, Rathi A (2022) Survey of microstrip antenna in nanotechnology using different nanostructures. In: Flexible electronics for electric vehicles. Lecture notes in electrical engineering, vol 863

Gupta A, Srivastava A, Anand R, Tomažič T (2020) Business application analytics and the internet of things: the connecting link. In: New age analytics. Apple Academic Press, pp 249–273

Hassan SA, Samsuzzaman M, Hossain MJ, Akhtaruzzaman M, Islam T (2017) Compact planar uwb antenna with 3.5/5.8 ghz dual bandnotched characteristics for iot application. In: 2017 IEEE international conference on telecommunications and photonics (ICTP). IEEE, pp 195–199

Mao Y, Guo S, Chen M (2018) Compact dual-band monopole antenna with defected ground plane for internet of things. IET Microw Antennas Propag 12(8):1332–1338

Rawat A, Tiwari A, Gour S, Joshi R (2021) Enhanced performance of metamaterials loaded substrate integrated waveguide antenna for multiband application. In: 2021 IEEE international conference on mobile networks and wireless communications

Sabban A (2022) Wearable circular polarized antennas for health care, 5G, energy harvesting, and IoT systems. Electronics

Soni GK, Gour S, Sharma A (2021) Design and analysis of multi-band fractal antennas for wireless communications. In: Emerging trends in data driven computing and communications

Wenju Z, Shengwei L, Jiwen H, Rongtao X, Hou L (2017) Design and implementation of smart irrigation system based on LoRa. In: 2017 IEEE globecom workshops (GC Wkshps)

Zhang Y, Liu C, Liu X, Zhang K, Yang X (2018) A wideband circularly polarized implantable antenna for 915-MHz ISM-band biotelemetry devices

Dual-Polarized Textile Antenna for Full-Duplex Wearable Applications

Neha Dalakoti and Priyanka Jain

Abstract In this article, an ultra-wideband, dual-port, textile antenna on jeans substrate has been designed and simulated for wearable and MIMO applications. The antenna consists of a partial ground plane on which a square shape has been cut out from the corner. It gives very wide bandwidth ranging from 1.58 GHz to 10.08 GHz and achieves isolation at least −11 dB throughout the operating band. The design has been done on 1.8 mm jeans substrate having dielectric constant 1.67. When ports are excited individually for both the polarizations, the gain of 2.73 dBi, 3.52 dBi, and 5.27 dBi is obtained at 3.5 GHz, 6 GHz, and 10 GHz, respectively. In addition to isolation between the ports, envelope correlation coefficient (ECC) and diversity gain have been calculated. The antenna is made out of a textile material, and as a result, it has a high level of structural flexibility and manufacturing precision. Using the Finite Integral Technique (FIT) numerical method, the provided two-port antenna structure is examined, investigated, and optimized.

Keywords Dual polarization · Textile antenna · Ultra-wide bandwidth · Wearable applications

1 Introduction

An important part of any wireless communication is the antenna, which is available in variety of shapes and sizes. Some of them are microstrip patch antenna, slot antenna, horn antenna, Vivaldi antenna, etc. Planar shape patch antenna is very much compatiple and suitable in wearable devices applications (Sanjaria et al. 2013). Microstrip patch antenna has been an area of interest for both the academia and industries for over more than seven decades. In recent years, the application of microstrip patch antenna in the Personal Area Network (PAN), or more specifically, the Wireless Body Area Network (WBAN), has received a lot of attention. Apart

N. Dalakoti · P. Jain (✉)
Delhi Technological University, New Delhi, India
e-mail: priyankajain@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_30

from this, microstrip patch antenna is highly useful in wearable biosensors, cameras, computers, healthcare, military, and a range of other wearable technologies. Such advancements arouse our interest in the creation and evolution of wearable antennas that are embedded in our clothing (Sundarsingh et al. 2014). Textile antennas are latest research interest because of its easy fabrication and good integration with planar circuits. These antennas are flexible, robust, highly efficient and possess low specific absorption ratio (SAR) (Mao et al. 2020). It is hugely desirable to have a transmitting and receiving antenna in a single antenna system for the most effective use of duplexing systems (Mao et al. 2020). In order to develop such antenna system, multiple-input multiple-output (MIMO) is used at both transmitter and receiver sides. MIMO antennas are better than single-input single-output (SISO) antennas in terms of channel capacity and transmitted power (Jensen and Wallace 2004; Choi et al. 2014). Channel capacity is directly proportional to the bandwidth of the system and it is improved with the use of ultra-wideband (UWB) antennas (Siddiqui et al. 2019). MIMO antennas sometimes use polarization diversity to double the channel capacity. Since these antennas use one radiating element and two ports, the isolation between the ports is very important to study. Multiple attempts have been made to design such antennas focusing bandwidth enhancement (Mao et al. 2020; Saxena et al. 2020), isolation improvement (Moradikordalivand et al. 2014; Chung and Yoon 2007; Luo et al. 2015), and gain improvement (Saxena et al. 2020). It has been reported that coupling degrades due to increased signal correlation between multiple radio signals (Singh et al. 2021). Numerous structures such as parasitic ring element (Moradikordalivand et al. 2014), extra ground wall (Chung and Yoon 2007), and approach for neutralization techniques on the ground plane (Luo et al. 2015) have been used between the ports to improve the isolation further.

In this paper, a dual-polarized, ultra-wideband, low profile, wearable, MIMO antenna has been designed and simulated in CST v.19. It has been designed on a general jeans substrate of dielectric constant 1.67 and height 1.8. Parameters like ECC and diversity gain have also been calculated to study isolation between the ports. The outline of the paper is as follows, Sect. 2 deals with antenna design, Sect. 3 includes results and discussion, and the last section summarizes the conclusion.

2 Antenna Design

2.1 Configuration

The design consists of a patch antenna on a jeans substrate of height 1.8 mm and dielectric constant 1.67 having one edge cut in the fashion of stairs. It has been feed from two orthogonal sides to have dual polarization in both the orthogonal planes. A partial ground has been used on the other side. The designed antenna has been shown in Fig. 1a with all the dimensions marked. Figure 1a shows the top view of the design which is very compact and uses no extra circuit for bandwidth

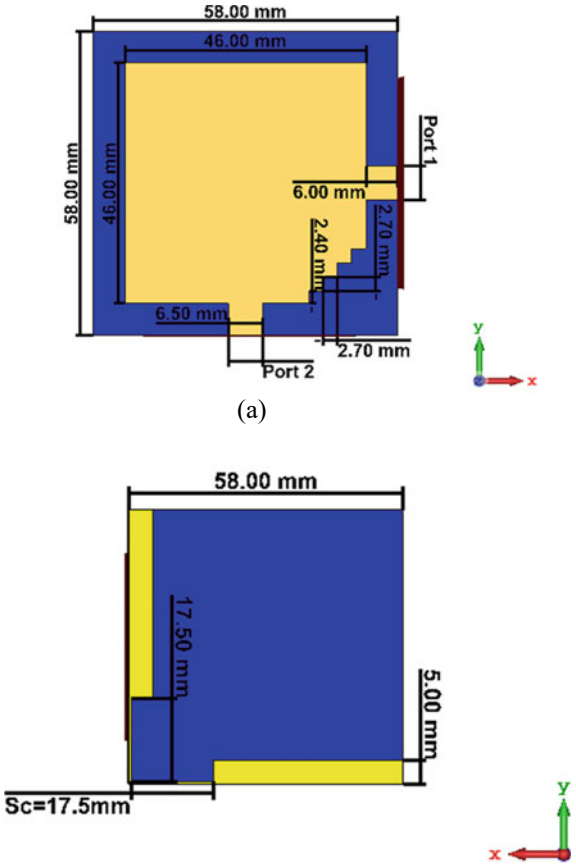


Fig. 1 Proposed design **a** top view and, **b** bottom view

enhancement. Figure 1b, on the other hand, shows the ground plane. Through a 6 mm long, 6.5 mm (50 Ω) broad microstrip line, the square shape patch is fed. The square patch has 58 mm side length. A square shape of sides 17.5 mm has been cut out from ground plane in order to enhance the bandwidth and improve the isolation between the ports. The antenna design and optimization are performed using the CST—Computer Simulation Technology based upon Finite Integration in Technique (FIT)

3 Results and Discussion

Ultra-wide bandwidth (UWB) of 145.8% ranging from 1.58 to 10.08 GHz has been obtained. Return losses (S11 and S22) are well below −10 dB, and also, the coupling coefficients (S12 and S21) are less than −11 dB throughout the bandwidth. The

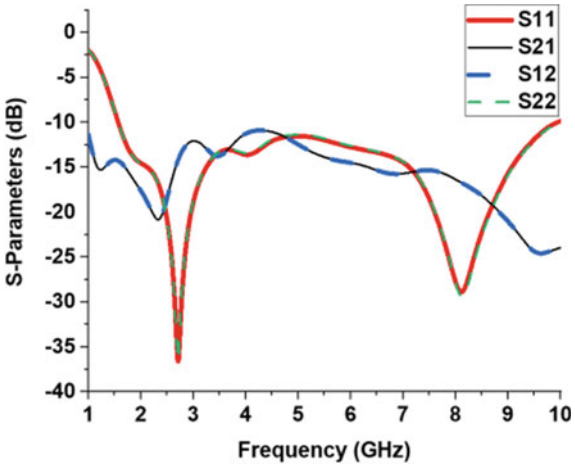


Fig. 2 S-parameters of the proposed antenna

design possesses good isolation in lower and upper bands which is less than -15 dB in frequencies ranging from 1.7 to 2.7 GHz and 6.4 to 10 GHz. Since the design is physically symmetrical, the value of S11 with S22 and S21 with S12 is almost identical. The parameter 'Sc' (square cut) has been optimized and simulated for 17, 17.5, 18, and 19 mm and the return loss less than 10 dB and isolation less than 11 dB come at 17.5 mm as shown in Fi 3.

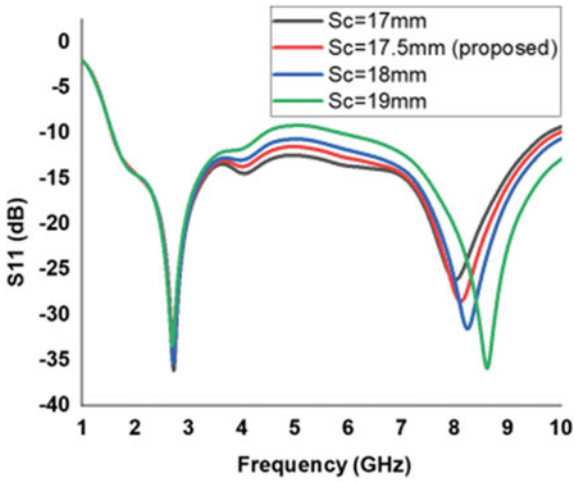
The radiation efficiency of the proposed antenna is shown in this Fig. 4a. The proposed antenna has an average radiation efficiency of 65%. The gain of the proposed antenna is shown in Fig. 4b. The proposed antenna has an average peak gain of 5.2dBi. Simulated gain for this antenna varies from 0.08 to 5.2 dBi as shown in Fig. 4b. Surface current is measured and plotted at 3.5 GHz as shown in Fig. 5.

ECC is a measure of isolation between the ports in a multiport MIMO antenna. It is desirable to have ECC as minimum as possible to minimize the isolation between the ports (Singh et al. 2021). ECC is calculated either using far-field pattern or using s-parameters of the system as given in Eq. 1 (Saxena et al. 2020). It is less than 0.02 throughout the band of operation.

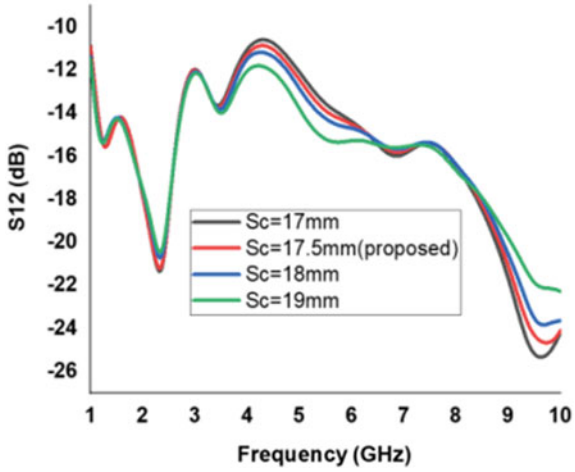
Another term is diversity gain which conveys same information about the isolation of the ports and it is related to ECC as given in Eq. 2 (Saxena et al. 2020). Diversity gain is 9.95 or above throughout the band of operation for this design (Fig. 6).

$$\mathrm{ECC} = \frac{\left|S_{11}^{*}S_{12} + S_{21}^{*}S_{22}\right|^{2}}{\left(1 - |S_{11}|^{2} - |S_{21}|^{2}\right)\left(1 - |S_{22}|^{2} - |S_{12}|^{2}\right)} \tag{1}$$

$$\mathrm{DG} = 10\sqrt{1 - \mathrm{ECC}^{2}}. \tag{2}$$



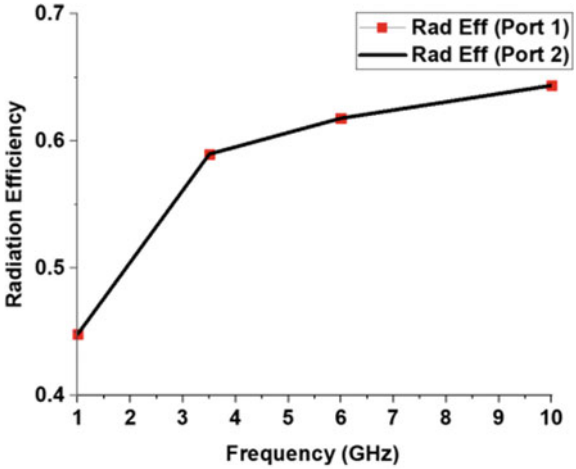
(a) Parametric S11 analysis



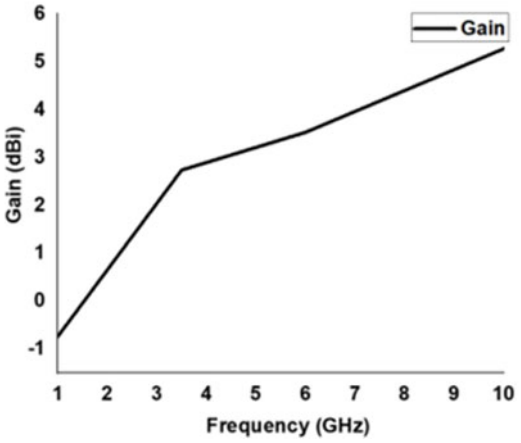
(b) Parametric S12 analysis

Fig. 3 Parametric analysis **a** S11 and, **b** S21

The proposed design has physical size of 58 × 58 mm^2 which is smaller as compared to (Sundarsingh et al. 2014; Moradikordalivand et al. 2014; Rais et al. 2013). The design has 1.58–10.08 GHz and 145.8% of fractional bandwidth which is better as compared to (Sundarsingh et al. 2014; Mao et al. 2020; Ononchimeg et al. 2010; Rais et al. 2013) as shown in (Table 1).



(a) Radiation efficiency vs. frequency



(b) Antenna gain vs. frequency

Fig. 4 Radiation efficiency and antenna gain with frequency

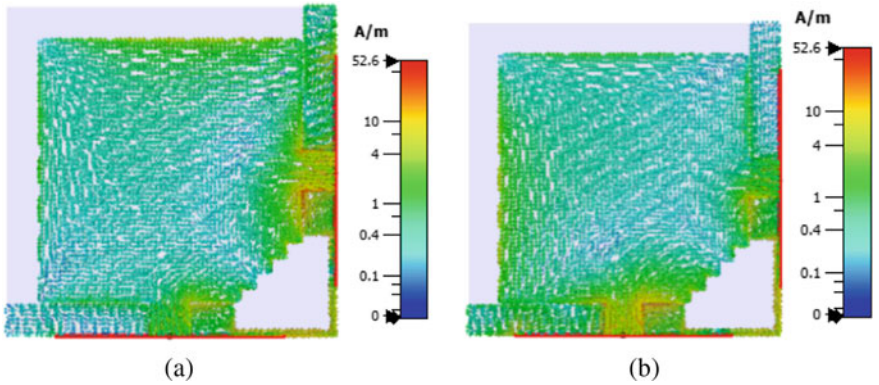
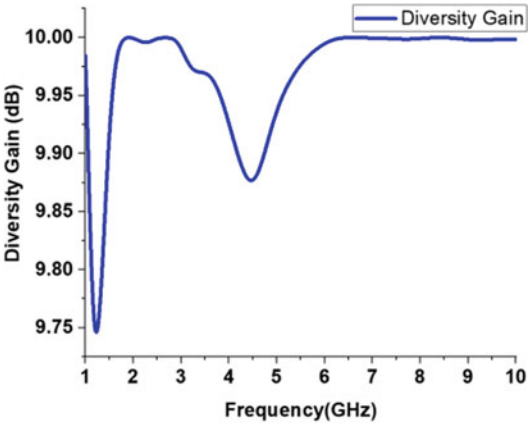
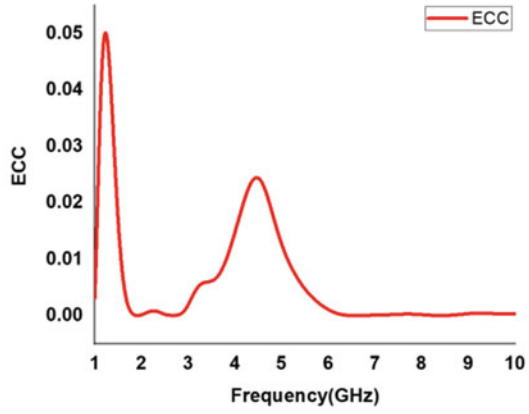


Fig. 5 Surface current of the proposed antenna when **a** Port-1 is excited and **b** Port-2 is excited

Fig. 6 Diversity gain and ECC



(a) Diversity gain versus frequency



(b) ECC versus frequency

Table 1 Comparison table

References	Material	Size (mm^2)	Bandwidth (GHz)	Isolation (dB)	Radiation efficiency (%)	Peak gain (dBi)
Sundarsingh et al. (2014)	Jeans	120 × 120	900 MHz,1800 MHz	NA	20	NA
Mao et al. (2020)	Evolon	NA	2.35–2.5	>15	90	7.5
Ononchimeg et al. (2010)	NA	45.0 × 40.0	14.5–15.2	>30	NA	6.2
Fallahzadeh et al. (2010)	Taconic RF35	30.2 × 38.8	2.6	>20	NA	4.5
Moradikordalivand et al. (2014)	FR-4	130 × 110	1–2.7	15	68–72	5.6
Upadhyay and Tripathi (2017)	Rogers RT/ Duroid 5880	96 × 96	25%	>11	NA	8
Hussein et al. (2019)	FR-4	70 × 70	3.2–3.3	−36	75	4.4
Rais et al. (2013)	Polyurethane foam	80 × 60	(2.22–2.48 GHz), (4.95–5.80 GHz)	NA	67–89	8.33
Proposed	Jeans	58 × 58	1.58 to 10.08	>15 at 1.7–2.7 and 6.4–10	65	5.2

4 Conclusion

A compact, dual-port, dual-polarized, ultra-wideband antenna for full-duplexing wearable application has been designed and simulated. The design consists of a stair cut patch antenna placed on jeans substrate backed by a partial ground with a square cut on the edge. The simulation result shows good reflection coefficients, isolation between the ports, and radiation characteristics throughout the band. Designed antenna is suitable for wearable and/or full-duplex applications such as health care, defense, and IoT. Isolation 15 dB has been achieved in the frequency range 1.7–2.7 GHz and 6.4–10 GHz. The efficiency of up to 65% is obtained in the required frequency band. Diversity gain of approximately 9.95 dB and envelope correlation coefficient below 0.02 have been obtained for diversity applications.

References

Choi DY, Shrestha S, Park JJ, Noh SK (2014) Design and performance of an efficient rectenna incorporating a fractal structure. Int J Commun Syst 27(4):661–679

Chung K, Yoon JH (2007) Integrated MIMO antenna with high isolation characteristic. Electron Lett 43(4):199–201

Fallahzadeh S, Bahrami H, Akbarzadeh A, Tayarani M (2010) High-isolation dual-frequency operation patch antenna using spiral defected microstrip structure. IEEE Antennas Wirel Propag Lett 9

Hussein AH, Abdullah HH, Attia MA, Abada AM (2019) S-band compact microstrip full-Duplex Tx/Rx patch antenna with high isolation. IEEE Antennas Wirel Propag Lett 18(10)

Jensen MA, Wallace JW (2004) A review of antennas and propagation for MIMO wireless communications. IEEE Trans Antennas Propag 52(11):2810–2824

Luo CM, Hong JS, Zhong LL (2015) Isolation enhancement of a very compact UWB-MIMO slot antenna with two defected ground structures. IEEE Antennas Wirel Propag Lett 14:1766–1769

Mao CX, Zhou Y, Wu Y, Soewardiman H, Werner DH, Jur JS (2020) Low-profile strip-loaded textile antenna with enhanced bandwidth and isolation for full-duplex wearable applications. IEEE Trans Antennas Propag 68(9)

Moradikordalivand A, Rahman TA, Leow CY, Ebrahimi S (2014) Dual-polarized MIMO antenna system for WiFi and LTE wireless access point applications. Int J Commun Syst

Ononchimeg S, Bang J-H, Ahn B-C (2010) A new dual-polarized gap-fed patch antenna. Progr Electromagn Res C 14:79–87

Rais NHM, Soh PJ, Malek MFA, Vandenbosch GAE (2013) Dual-band suspended-plate wearable textile antenna. IEEE Antennas Wirel Propag Lett 12:583–586

Sanjaria HR, Meratib AA, Varkiania SMH, Avakoli A (2013) A study on the effect of compressive strain on the resonance frequency of rectangular textile patch antenna: elastic and isotropic model. J Tex Inst 105(2):156–162

Saxena G, Jain P, Awasthi YK (2020) High diversity gain MIMO-antenna for UWB application with WLAN notch band characteristic including human interface devices. Wirel Pers Commun

Siddiqui JY, Antar YMM, Saha C (2019) Multifunctional ultrawideband antennas trends, techniques and applications. Taylor & Francis Group

Singh AK, Mahto SK, Sinha R (2021) Compact super-wideband MIMO antenna with improved isolation for wireless communications. Frequenz 75:407–417

Sundarsingh EF, Velan S, Kanagasabai M, Sarma AK, Raviteja C, Alsath MGN (2014) Polygon-shaped slotted dual-band antenna for wearable applications. IEEE Antennas Wirel Propag 13:611–614

Upadhyay G, Tripathi VS (2017) Pin-diode based switchable multiband dual feed microstrip patch antenna. Microw Opt Technol Lett 59(6).

Low-Power High-Speed Modified Three-Stage Comparator with Low Kickback Noise

T. Thamaraimanalan, S. Ramalingam, S. Dhanasekaran, and K. Baskaran

Abstract In this study, a three-level comparator architecture has been suggested to increase speed and decrease kickback noise. The additional amplified stage in the multiple comparators boosts voltage gain and speed in comparison to the traditional two-stage comparator. Furthermore, speed is increased by using nMOS input sets in both the regenerating and amplifies stages rather than just the regenerative stage as in the conventional two-stage arrangement. The three-stage comparator's kickback noise is greatly decreased by adding a pair of CMOS inputs at the amplification stage. This kickback noise is generated by NMOS kickback. Faster performance is achievable by introducing a second describing the steps while playback remains active. For better comparison, the suggested three-stage comparator and the standard two-stage comparator were both simulated by employing the identical 130-nm CMOS technology. According to the experimental findings, the redesigned comparator increases speed by 32% and decreases kickback noises by a factor of 10.

Keywords Comparator · Low power · High speed · Kickback noise

T. Thamaraimanalan (✉) · S. Ramalingam · S. Dhanasekaran
Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering, Coimbatore, India
e-mail: t.thamaraimanalan@gmail.com

S. Ramalingam
e-mail: ramece74@gmail.com

S. Dhanasekaran
e-mail: dhanselvaraj@gmail.com

K. Baskaran
Department of Electronics and Communication Engineering, Alagappa Chettiar Government College of Engineering and Technology, Karaikudi, India
e-mail: drbaskaran@gct.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_31

1 Introduction

The main goals of real-time application systems are speed, power, and area. Multiplication and addition are essential modules in DSP processors, CPUs, digital filters, modulators, demodulators, cryptosystems, and many other signal-processing modules (Harpe et al. 2013). There is a strong need to develop low-power and low-voltage circuit technology for such devices and critical components. Using the same power supply voltage for analog and digital circuits eliminates the need for multiple power supplies and lower overall system costs. It consistently incorporates the A/D interface into ICs, primarily for DSPs and controls digital modules. Multipliers are used in various applications, depending on the limitations of each. As a result, the power, area, and delay parameters of multipliers used in signal processing operations must be investigated (Bindra et al. 2018). Several multipliers are used to multiply two binary numbers: X carry-save adders, add-shift multipliers, ripple-carry parallel adders, pipeline multipliers, and serial-parallel multipliers.

Developing a new generation of A/D converters that are low-voltage device compatible is necessary. The cost of AD/DA converters can generally be decreased by improving the design of low-power comparators. This power reduction can be accomplished by switching to processes with smaller feature sizes. A/D converters for portable and mobile devices are among the applications that require low power consumption, high speed, low noise, low hysteresis, and low offset voltage. Such comparators' accuracy, speed, power consumption, and offset are all essential factors in achieving higher ADC performance (Razavi 2015; Babayan- Mashhadi and Lotfi 2014; Khorami and Sharifkhani 2018; Elzakker et al. 2010).

Supply voltages decrease as CMOS technology scales, and low supply voltages fundamentally limit the maximum input voltage swing and lower the SNR (Figueiredo and Vital 2006). Preamp-based comparators have a high offset voltage, which is their main disadvantage. Dynamic comparators commonly address this issue, which performs one comparison per clock cycle and requires less offset voltage (Miyahara et al. 2008; Zhuang et al. 2020a; Thamaraimanalan and Sampath 2019).

The preamplifier in the first stage implements a CMOS input pair, which helps to significantly decrease kickback noise. In the latch stage, an extra channel is included to boost playback speed and reduce input reference offset and noise.

2 Literature Survey

To increase speed and reduce kickback noise, a modified version and a three-level comparator are suggested. The capacity of the pMOS kickback to negate the nMOS return as a result results in a significant reduction of kickback noise. The speed can be raised even further by including more signal paths during the playback stage. The traditional two-level comparator and the suggested three-level comparator are simulated on the same 45 nm CMOS technology to facilitate comparison. High-speed,

high-resolution SAR ADCs are best suited for using these comparators. Finally, the measurement outcomes support the comparator's performance (Sonoli 2022).

An 8-bit, 5-stage, interleaved analog-to-digital converter is presented that uses differential pairs and source followers as the only open-loop analog processing components to produce a high conversion rate. ADCs do not use op amps; only source followers and differential pairs are used. As a result, there is little impact when gain, supply voltage, and speed are traded off. The prototype can use a 2.5 V power supply but is intended to work with 3.3 V systems (Wang, et al. 2000).

Various CMOS latching comparators' power consumption is compared to determine the power delay product of different comparators given a given V_{dd}. Simulation results show their effectiveness using dynamic latch comparators with inverters that use less power than other comparators. The latch stage and input of the comparator preamplifier both use pMOS transistors. Special local clock generators govern both stages. Latch activation is delayed during the evaluation phase to achieve good preamplifier gain while avoiding excessive power consumption (Moni and Lisha 2012). To demonstrate its effectiveness, a review for reducing recoil noise is made using HSPICE simulations of a comparator implemented in 0.18-m technology. Previously used recoil reduction techniques either did not fully solve the problem or significantly increase power consumption.

A 9-bit switching comparator SA-ADC with ultra-low power consumption is developed using 0.18 m CMOS technology. A comparator's fundamental structure and components are described. Basic component design techniques are covered, as well as sizing issues. The fundamentals of comparators and their role in ADCs and DACs are covered. Ultra-low power converters are regenerative step comparators to increase power efficiency and speed. This method was created for low-power, low-voltage, and high-speed applications. These simulations are based on CMOS technology available in 180, 90, 250, 130 nm, etc. Faster power transactions do not affect comparator circuits. Flash-based ADCs use high aspect ratio MOS transistors, which have a high-power requirement but are short and have very low latency (Valaee and Mohammad. 2003).

The three-stage comparator has been modified to improve speed and lower kickback noise. Further accelerating the speed is the addition of additional signal paths during the playback phase. The proposed three-level and improved three-level comparators are used on the same 45 nm CMOS process to simplify comparisons. This paper suggests improvements to the three-level comparator with high-speed and low recoil noise benefits (Dhanasekaran et al. 2020).

Kickback noise is a severe issue in traditional comparators, particularly ADCs. It can disrupt the input signal voltage and cause errors in the ADC output. Power reset, rail-to-rail output, and positive feedback are the benefits of reducing kickback noise. First, the small current source under the latch controls its regeneration speed. Another problem with the multiple stacked transistors is the requirement for a high-power supply voltage (Zhuang et al. 2021).

This paper proposes a modified three-stage comparator. Using a CMOS input pair significantly reduces kickback noise at the first-stage preamplifier. In the latch stage, a second path is added to accelerate regeneration while reducing input noise. The

proposed comparator in this work accelerates 25% faster than traditional two-stage comparators in 130 nm process. The suggested modified version accelerates by 32% while reducing kickback noise significantly.

3 Proposed Methodology

A small area, low-power delay line-based controller is developed to make the comparators resistant to process VDD temperature variation as the controller delay and comparator component delays vary in the same direction at different PVT corners. The transistor size is much smaller than the preamp input transistor size (about 7–10 times smaller). As a result, cross-connected circuits have minimal effects on area, power, and offset. When the preamp is turned off, the crossover circuit increases the differential voltage and speed, but its primary function is to reduce the latch's common-mode input voltage.

To combat the voltage swings on regeneration nodes in such circuits, our research presents a fully differential dynamics comparison with low kickback noise. Dynamic converters, especially pipelined ADCs, are essential circuits in ADCs because of their low-power consumption. Three other comparator topologies are simulated and contrasted with the suggested comparator. Over a broad input range, the proposed circuit exhibits low offset voltage errors and returns return noise values lower than those of conventional dynamic comparators. Figure 1 displays the three-stage comparator used in this research. The first phase of the three is connected to the other two. The addition of a second-stage preamplifier distinguishes this comparator from the Miyahara comparator. The latching stage can now operate faster using nMOS input pairs $M_{11}-M_{12}$ instead of pMOS input pairs. This extra preamp can also function as an inverter.

3.1 Three-Stage Comparator

The latched stage of Fig. 2(bmodified)'s version has different routes M29-32 than the original version does, and the first stages of the revised version change from those of the earlier design in the prior chapter (c). The additional initial two stages' kickback noise is lessened using the pMOS inputs pair M11–12. The latch nodes OUTN and OUTP receive additional signal lines to allow faster regeneration and lessen input distortion and referred offset.

The CLK signal rises to 1 during the amplification phase, while CLKB falls to 0. FP1 and FN1 are set to GND in Fig. 2b, whereas RP1 and RN1 rise to VDD (R represents rise) (F denotes fall). The transient activation of the extra pathways in Fig. 2c is caused by the increase of RP1 and RN1 earlier FP1 and FN1. As a result, the latching nodes OUTP and OUTN experience a differential current draw. By lowering distortion and offset the comparison input, the differential voltage at the

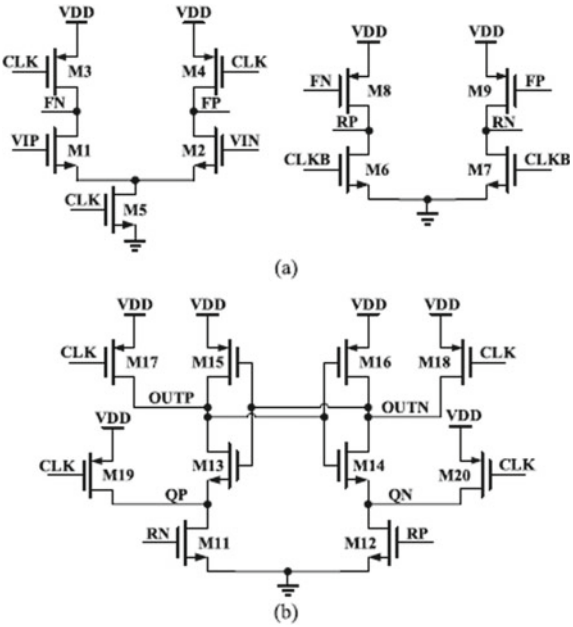


Fig. 1 Three-stage comparator **a** preamplifiers **b** latch stage

outputs' points (OUTN and OUTP) speeds up the restoration phase. To avoid static current, the routes in Fig. 2c are shut off after FP1 and FN1 approach GND. With less feedback offset and noise, the proposed three-stage comparator is faster than the traditional comparator. For high-speed, high-resolution SAR ADCs like the moment noise-shaping SAR ADC, this suggested adjustment is appropriate. The speed of the comparator, as already explained, restricts the ADC's velocity, and its sluggish noise limits its precision.

3.2 Design Consideration

We must minimize the comparator's integration time because we are using a high-speed, high-resolution ADC. Simultaneously, to reduce input-referred noise, the transconductance of each stage's input pairs (including each preamplifier input pair and the lock stage) should be as high as possible (Zhuang et al. 2020b).

Kickback

Using a positive feedback mechanism, the latching comparator adjusts the digital level during regeneration. Changes in voltage at the regenerative node coupled to the input interfere with it. This type of interference is known as kickback noise. Because the switches have non-zero resistance, the circuit amplifies, producing a small differential output voltage. A sampling switch can be placed in front of the

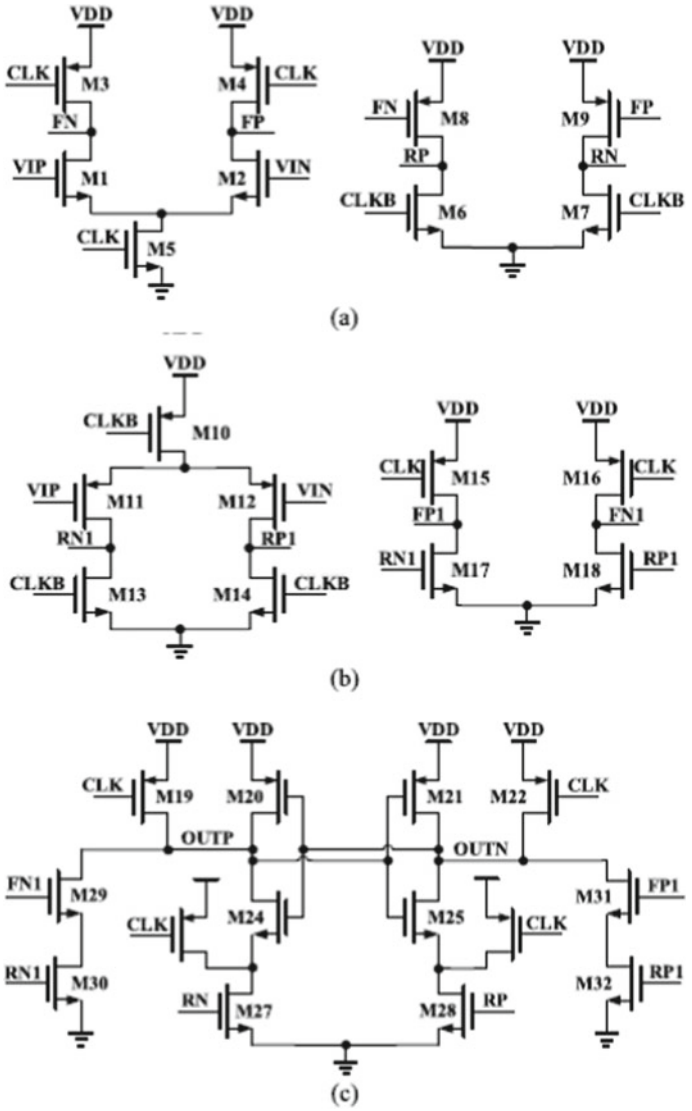


Fig. 2. **a** Preamplifiers with nMOS input pair **b** Preamplifiers with pMOS input pair **c** Latch stage

differential pair's input to minimize the noise kickback. It activates during the regeneration process. Reduce kickback noise by lowering the voltage swing at the input transistor's drain.

Sampling Switches
These switches open during the regeneration phase, disconnecting the input from the rest of the circuit. To minimize charge injection effects, the switch should be as small as possible in comparison to the total capacitance of the input.

Isolation Transistors
A pair of clocked NMOS and isolation transistors are typically positioned between the differential pair's drain and its regenerative output. The isolation transistor is disabled during regeneration to prevent charge infusion into the differential input. This method makes less noise from kickback.

Preamplifier
A preamplifier is the most common solution for reducing feedback effects before the comparator. In addition, the preamplifier amplifies the input difference while decreasing the input-referenced offset. Preamplifiers can boost system gain and bandwidth but consume more power.

4 Results and Discussion

Numerous simulation results have been run to assess the voltage offset error and kickback effect on the comparator input signal. To simulate the kickback effect, the gates of the input transistors (M_2 and M_4) of the four topologies are connected to V_{+IN} and V_{IN} through 50 resistors. Direct connections to the ideal voltage source exist between V_{+R} and V_{-R}. Figure 3 depicts the proposed method's schematic diagram.

Figure 4 depicts the proposed method's input frequency analysis. Figure 5 illustrates the proposed method's frequency as a time function. An entirely differential dynamic comparator with little kickback noise is presented in this article.

Compared to other comparator topologies, the circuit's kickback noise is reduced by at least one order of magnitude. The propagation delay is two orders of magnitude

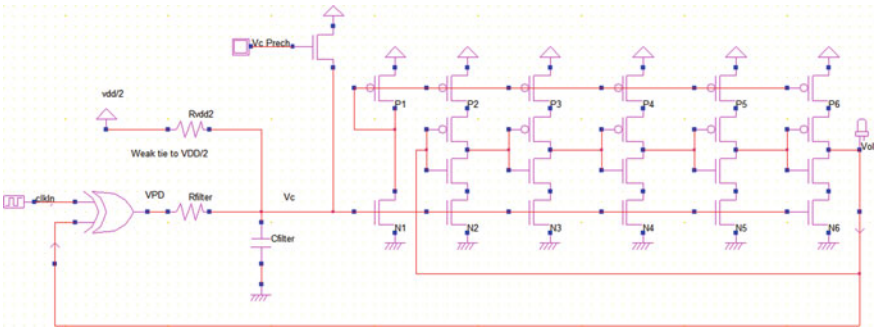


Fig. 3 Proposed three-stage comparator

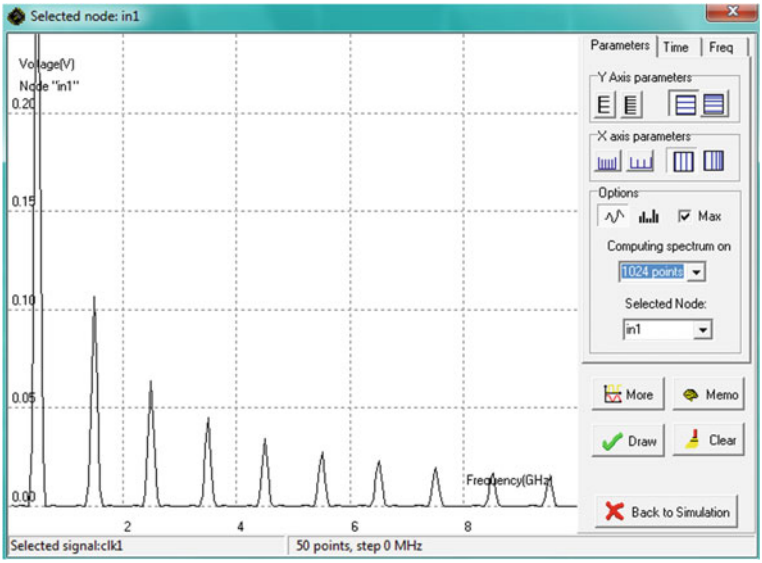


Fig. 4 Frequency analysis of the proposed comparator

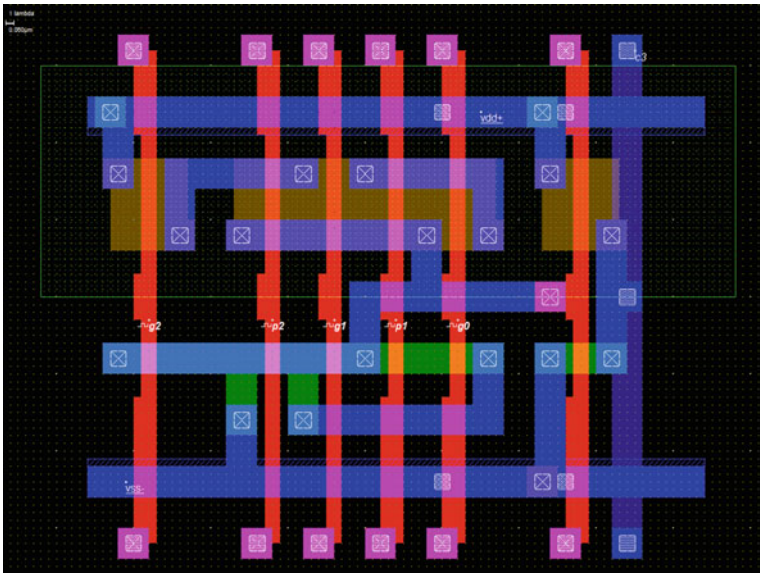


Fig. 5 Layout of the proposed comparator

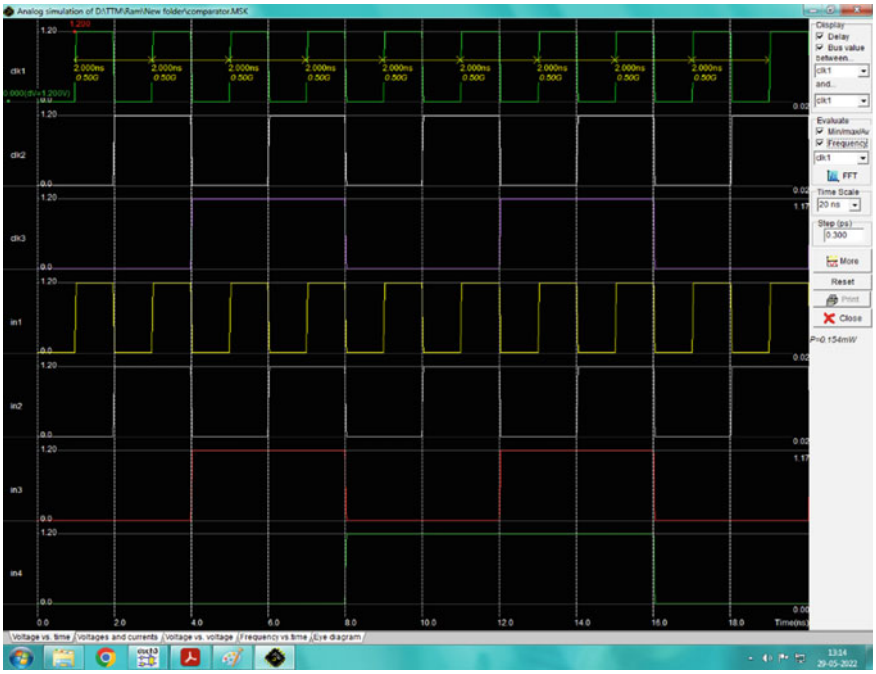


Fig. 6 Total power consumption of the proposed comparator

shorter than the clock cycle period. Figure 6 depicts the three-stage comparator layout diagram in the modified form that has been suggested.

Figure 6 show the voltage and current output and the voltage and time power analysis of the proposed method.

In Tables 1 and 2, a comparison is made between this study and current state-of-the-art research. The results indicate that the proposed comparator exhibits the shortest delay. Furthermore, the modified comparator produces the least amount of kickback noise.

Figure 7 depicts energy consumption as a function of differential input voltage. The three-level comparator consumes more power than the two-level comparator, as can be seen. This is due to increased complexity and additional circuitry. Although it consumes more power, a three-level comparator has significant advantages over a two-level comparator. This is the three-level comparator's advantage. The three-stage comparator's main advantages are increased speed and reduced kickback. It also does not include any input reference offset or noise. Table3 shows the delay comparison.

The input-referred noise is shown as a function of V_{cm}, and at $V_{cm} = 600$ mV, it is 0.44 mV. Three-stage comparators offer less intake reference noise than two-stage comparators; however, when V_{cm} reaches 600 mV, the input referenced noise increases due to the increased voltage gain.

Table 1 Comparison with current state-of-the-art works

Parameters	Miyahara's comparator (Miyahara et al. 2008)	Elzakker's comparator (Elzakker et al. 2010)	Three stage comparator (Thamaraimanalan and Sampath 2020)	Proposed comparator
Supply voltage	1.2 V			
Technology	130 nm			
Low kickback noise	No	No	No	Yes
Area (μm^2)	79	117	151	216
Input referred Offset (mV)	17	19	15	9
Energy per consumption (pJ)	0.09	0.08	0.17	0.29
Input referred noise (mv)	1	0.95	0.8	0.6
Delay (ps)	270	250	210	200

Table 2 Comparison of energy comparison (pJ)

Differential input voltage (mV)	Miyahara's comparator	Elzakker's comparator	Three-stage comparator	Proposed comparator
10^0	No	No	No	Yes
10^1	79	117	151	216
10^2	17	19	15	9

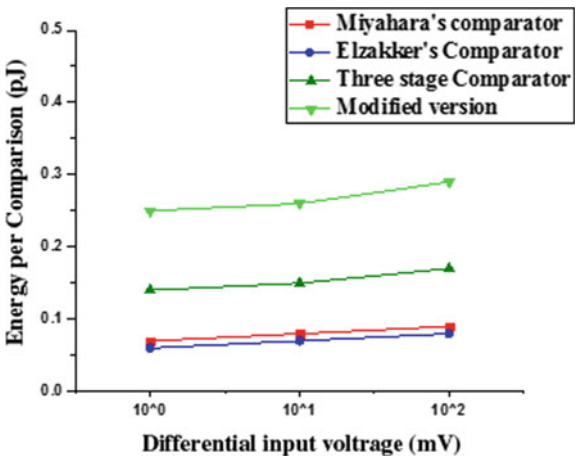


Fig. 7 Power consumption versus differential input voltage

Table 3 Delay comparison (ps)

Common-mode input voltage (mV)	Miyahara's comparator	Elzakker's comparator	Three-stage comparator	Proposed comparator
400	0.6	0.58	0.5	0.4
500	0.55	0.53	0.48	0.38
600	0.52	0.50	0.50	0.40
700	0.65	0.55	0.54	0.5
800	0.7	0.68	0.6	0.53
900	1	0.95	0.8	0.6

5 Conclusion

This research describes a three-level comparator and its modifications, which provide low kickback noise, high speed, noise, and low input-referenced offset. Eventually, the measurement outcomes attest to the efficiency of the comparators. A pMOS preamplifier, a pMOS latch, and a unique clock mode are employed in the presented comparator, along with a small cross-coupling circuit to tune the preamplifier's gain. To keep V_{cm} at the preamplifier's output low, sufficient preamplifier gain is achieved in clock mode using pMOS transistors at the latch circuit's input and a crossover circuit. This research's experimental and simulation results demonstrate that the proposed circuit consumes little power and can be an optimal solution for designing power-line ADC converters. Step noise has an effective resolution of 2.5 bits or less. Moreover, the circuit's low noise allows it to be utilized in other circuitry that calls for a comparator with a preparation stage, which lowers the circuit's power requirement.

References

Babayan-Mashhadi S, Lotfi R (2014) Analysis and design of a low voltage low-power double-tail comparator. IEEE Trans Very Large Scale Integr (VLSI) Syst 22(2):343–352

Bindra HS, Lokin CE, Schinkel D, Annema AJ, Nauta B (2018) A 1.2-V dynamic bias latch-type comparator in 65-nm CMOS with 0.4-Mv input noise. IEEE J Solid-State Circuits 53(7):1902–1912

Dhanasekaran S, Thamaraimanalan T, Anandkumar V, Manikandan A (2020) Analysis and design of fir filter using modified Carry Look Ahead Multiplier. Int J Sci Technol Res 9:1336–1339

Figueiredo PM, Vital JC (2006) Kickback noise reduction techniques for CMOS latched comparators. IEEE Trans Circuits Syst Exp Briefs II. 53(7):541–545

Harpe P, Cantatore E, van Roermund A (2013) A 2.2/2.7fJ/conversion step 10/12b 40kS/s SAR ADC with data-driven noise reduction. In: Proceedings IEEE international solid-state circuits conference digest of technical papers, pp 270–271

horami A, Sharifkhani M (2018) A low-power high-speed comparator for precise applications. IEEE Trans Very Large Scale Integr (VLSI) Syst 26(10):2038–2049

Miyahara M, Asada Y, Paik D, Matsuzawa A (2008) A low-noise self calibrating dynamic comparator for high-speed ADCs. In: IEEE Asian solid-state circuits conference, pp 269–272

lackuline Moni D, Lisha P (2012) Highspeed and low-power dynamic latch comparator. In: International conference on devices, circuits and systems

Razavi B (2015) The Strong ARM latch [a circuit for all seasons]. IEEE J Solid-State Circuits, Mag. 7(2):12–17

Sonoli S, Rajashekar V (2022) Design of three-stage comparator and its modified version using lector technique. Int J Res Trends Innovation IJRTI 7(7). ISSN: 2456–3315

Thamaraimanalan T, Sampath P (2019) A low power fuzzy logic based variable resolution ADC for wireless ECG monitoring systems. Cogn Syst Res 57:236–245

Thamaraimanalan T, Sampath P (2020) Leakage power reduction in deep submicron VLSI circuits using delay-based power gating. Natl Acad Sci Lett 43(3):229–232

Valaee A, Maymandi-Nejad M (2003) An ultra low-power low-voltage switched-comparator successive approximation analog to digital converter. IEICE Electronics Express IEEE J. Solid-State Circuits IEEE J. Solid-State Circuit Computer Design 6156:1098–1104. https://doi.org/10.1587/elex.6.1098

van Elzakker M et al (2010) A 10-bit charge-redistribution ADC consuming 1.9 μW at 1 MS/s. IEEE J Solid-State Circuits 45(5):1007–1015

Wang YT et al (2000) An 8-bit 150-MHz CMOS A/D converter. IEEE J Solid-State Circuit 35(3):308–317

Zhuang H, Tang H, Liu X (2020b) Voltage comparator with 60% faster speed by using charge pump. IEEE Trans Circuits Syst II Exp Briefs 67(12):2923–2927

Zhuang H, Liu J, Sun N (2020) A fully-dynamic time-interleaved noise shaping SAR ADC based on CIFF architecture. In: Proceedings on IEEE custom integrated circuits conference

Zhuang H, Cao W, Peng X, Tang H (2021) A three-stage comparator and its modified version with fast speed and low Kickback. IEEE Trans Very Large Scale Integration (VLSI) Syst 29(7):1485–1489. https://doi.org/10.1109/TVLSI.2021.3077624

Analysis of the Electronic and Optical Properties of MoS_2 and WSe_2 Using the First-Principle Density Functional Theory

Emarishi, Reshma Sinha, and Jasdeep Kaur

Abstract This paper describes the two-dimensional materials that are popular for their unique properties. The first is the introduction of transition metal dichalcogenides (TMDs) molybdenum disulfide (MoS_2) and tungsten selenide (WSe_2). Potential replacements for the traditional semiconductor materials used in the IC industry include MoS_2 and WSe_2 due to their unique properties including electronic and optoelectronic. Simulations were done by using Quantum Espresso for the electronic properties of MoS_2 and WSe_2.The potential bandgap-tuning of MoS_2 and WSe_2 has generated interest. The thickness of the material depends on the transition between the direct and indirect bandgap. To examine the features of the MoS_2 and WSe_2, density functional theory (DFT) is used for first-principle computations on electronic structures. The direct band gap is at K, and the indirect band gap of MoS_2 and WSe_2 is between the point of the first Brillouin zone and Γ and Γ-K. 'DFT material properties simulator' is used to investigate the optoelectronic properties for finding parameters such as absorption coefficient, extinction coefficient, and refractive index for the comparative analysis of mono- and bulk-layer MoS_2 by logarithmic graph scale. Our results show the analysis of the electronic properties of MoS_2 and WSe_2, where MoS_2 has better electronic properties than WSe_2 with SCF total energy value of MoS_2 (−993.3918 Ry) which is better than WSe_2 (−402.008Ry), and it will take lesser time for SCF to converge. Overall, calculating SCF and DOS are important steps in the first-principle DFT approach and provide valuable information about electronic properties. After that, we calculated the optoelectronic properties of its mono-layer and bulk-layer and their comparative analysis. The comparative analysis of bulk and mono-layer MoS_2 can be used to find visible light (light of wavelength below 500 nm) that can be used as a photodetector. Also,

Emarishi (✉) · J. Kaur
Department of Electronics and Communication Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi 110006, India
e-mail: emarishikataria2000@gmail.com

R. Sinha
Department of Instrumentation, Shaheed Rajguru College of Applied Sciences for Women, University of Delhi, Delhi 110096, India
e-mail: reshma.sinha16@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_32

it has been found that MoS_2 is most sensitive to light with a wavelength of 450 nm. MoS_2 and WSe_2 distinct bandgap characteristics can improve their applicability in energy devices like solar cells, FETs, and transistors, double gate FETs in pH sensors, optoelectronics, spintronics, and photoluminescence.

Keywords TMDs · FET · Optoelectronics · DFT · pH sensor · Spintronics

1 Introduction

The inherent properties of two-dimensional materials such as electronic, optoelectronic, and mechanical are popular for their unique characteristics (Liu et al. 2021). Most important aspect of these two-dimensional materials is the high mobility of charge carriers due to their 3-D quantum confinement (Ahmed and Yi 2017). This is the potential for applications in electronic devices such as transistors. Because of its extreme thinness and great mobility, two-dimensional layered material is promoted as a successor to present Si technology (Ahmed and Yi 2017; Ayari et al. 2007). TMDs are a good option for two-dimensional material that has a bandgap in the range of 1–2 eV. MoS_2 1.8 eV and WSe_2 1.7 eV, two transition metal dichalcogenides (TMDs), have a more desirable bandgap than graphene (Wang et al. 2012a). The devices of logic, such as MOSFET, MoS_2 and WSe_2 are one of the most promising materials because of their adjustable band gap, high on/off ratio, and incredibly low cost (Mir 2020). Among two-dimensional materials, TMDs currently exist in nature and are attracting increasing interest because of their special and varied characteristics. The symbolic representation of TMDs is $\mathbf{MX}_2$, M represents transition metal (like Zr, Ti, Mo, Ni, and V). It consumes 10^5 times less power than Si transistors. These considerations for 2D material growth and transfer lead to a discussion of what milestones need to be reached for integration with the current CMOS technology. Moreover, the dielectric environment can modulate electronic and optical properties. Monolayer TMDs have a direct bandgap in the electromagnetic spectrum's visible region. TMDs are therefore particularly well-suited for optoelectronics, digital electronics, and display devices (Ahmed and Yi 2017; Radisavljevic et al. 2011). In this paper, an analysis of the electronic structure of MoS_2 and WSe_2 using the first principle density functional theory (DFT) is done for the analysis of various parameters such as band structure, density of states, and self-consistent field. Our results show that on analyzing both these TMDs, we are getting better results for MoS_2 and because of that we calculated optoelectronic properties. Comparative analysis of mono-layer and bulk-layer MoS_2. Hence, this work gives an analysis that MoS_2 is better in comparison with WSe_2.

2 Computational Method

The Quantum Espresso software is used as a plane-wave pseudopotential technique to do DFT computation. To depict the exchange–correlation effects, the Perdew-Burke-Ernzerhof (PBE) functional was utilized as a generalized gradient approximation (GGA) (Projector augmented-wave method 1994). DFT first-principle computations were carried out using Quantum Espresso in the projector augmented-wave pseudopotential. According to the PBE form, exchange–correlation effects were described using the GGA (Perdew et al. 1997; Giannozzi et al. 2009). The parameters of electronic properties like band structure, density of states, and self-consistent field (SCF) were calculated and optical properties were also calculated using NanoHub: 'DFT Material Simulator' for parameters like absorption coefficient, refractive index, and extinction coefficient.

2.1 Molybdenum Disulphide (MoS_2)

The following computational information was utilized to analyze the crystal structure of MoS_2. The PAW-PBE platform for DFT, MoS_2, and space group P63/mmc(#194) was used for all calculation of lattice crystal: hex, crystal system: hex, and BL: HEX, unit cell parameters: a = b = 3.202 Å, c = 13.90 Å, c/a = 4.341, α = β = 90°, γ = 120°, unit cell volume: 116.7 $Å^3$, density: 4.553 g/cm^3, cut-off energy is 392 eV, and for Brillouin zone sampling, a mesh of 21 × 21 × 4 k-points was employed. In Fig. 1 the appropriate crystal structure of MoS_2 is shown.

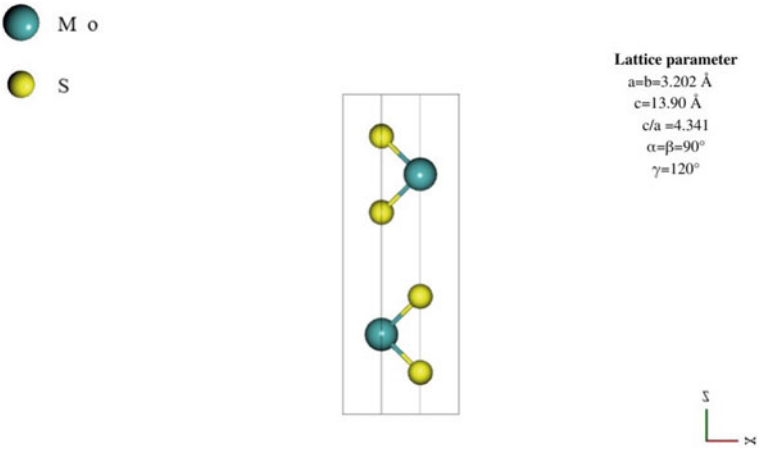


Fig. 1 Crystal structure of MoS_2

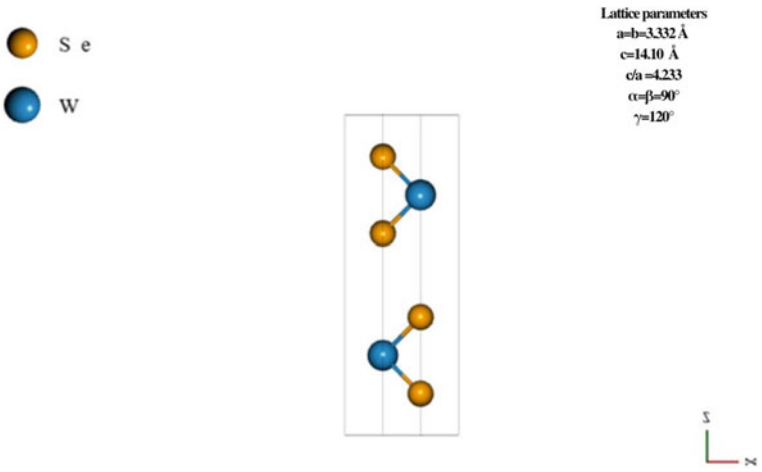


Fig. 2 Crystal structure of WSe_2

2.2 Tungsten Diselenide (WSe_2)

The following computational information was utilized to analyze the crystal structure of WSe_2. The PAW-PBE platform on the DFT, WSe_2, and space group P63/mmc(#194) was used for all calculations of lattice crystal: hex, crystal system: hex and BL: HEX unit cell parameters: a = b = 3.33 Å, c = 14.10 Å, $\alpha = \beta = 90°$, $\gamma = 120°$, unit cell volume: 135.6 $Å^3$, density:8.37 g/cm^3, cut-off energy is 312.3 eV, and for Brillouin zone sampling, a mesh of 21 × 21 × 4 k-points was employed. In Fig. 2 appropriate crystal structure of WSe_2 is shown.

3 Results and Discussions

Calculating the self-consistent field (SCF) and density of function (DOS) are two important calculations in the field of first principles (DFT) that provide valuable information about the electronic properties of a material. Here are some benefits of calculating SCF, DOS, and band structure:

3.1 Self-Consistent Field (SCF)

The SCF calculation is used to obtain the electronic ground state of a material by iteratively solving Kohn–Sham equations until the electron density converges to a self-consistent solution. This calculation provides information about the energy

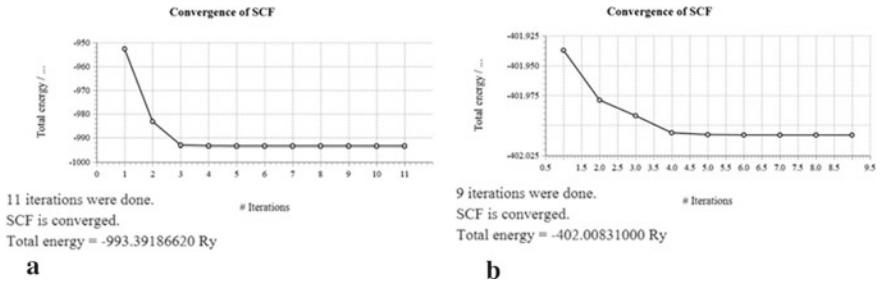


Fig. 3 SCF of **a** MoS_2, and **b** WSe_2

levels and distribution of electrons in the material, which can be used to calculate various properties such as electrical conductivity, optical properties, and thermoelectric properties. In this, SCF is carried out using 19 × 19 × 4 Brillouin zone sampling k-points. The result for SCF calculation was applied to the band structure extraction. On simulating the SCF of MoS_2 and WSe_2, we are getting the total energy of (−993.391 Ry) and (−402.008), respectively. A total of 11 iterations for MoS_2 and 9 iterations for WSe_2 were done for SCF to reach its ground state or total energy. On observing the convergence of these two materials, we analyze that MoS_2 will take lesser time in comparison with WSe_2 to reach its SCF convergence value as shown in Fig. 3a, b.

3.2 Band Structure

The band structure calculation is closely related to both SCF and DOS calculations. It provides information about the energy bands that electrons can occupy in a material, which is important for understanding the electronic and optical properties of the material. By analyzing band structure, it is possible to determine whether a material is a conductor, insulator, or semiconductor. MoS_2 and WSe_2 are indirect-gap semiconductors, according to band structure computation. With VB maxima and CB minima near the K-point mesh 19 × 19 × 4 in the Brillouin zone, MoS_2 is direct band gap material. WSe_2 and MoS_2 mono-layer structures were found to have direct bandgap materials with bandgap values of 1.6 eV and 1.8 eV. The k-point of the Brillouin zone, CB minima and VB maxima are observed. On simulating the bandgap of the two TMDs MoS_2 will show better bandgap results with a bandgap of 1.8 eV and bandgap-tuning in this is also good. The difference between CB minima and VB maxima is the energy bandgap. The band structure is shown in Fig. 4a, b

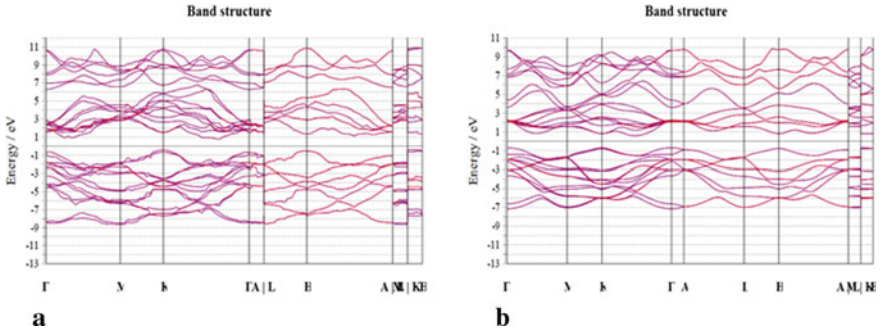


Fig. 4 Band structure of **a** MoS_2 and **b** WSe_2

3.3 Density of States (DOS)

The DOS calculation provides a detailed picture of the electronic structure of a material by showing the density of available electronic states at each energy level. This information can be used to calculate a range of physical properties such as electrical conductivity, optical properties, and thermoelectric properties. The total number of states that electrons can occupy at any given energy level is referred to as the DOS. DOS of MoS_2 and WSe_2 show the total spin up and spin down as shown in Fig. 5a, b. This compound is exploited for additional optoelectronic and spintronic applications.

By understanding the electronic structure of materials, scientists can design materials with specific properties for a wide range of applications, including energy generation, energy storage, and electronics. Overall, calculating SCF and DOS are important steps in the first-principle DFT approach and provide valuable information about the electronic properties of materials that can be used to predict a wide range of physical properties.

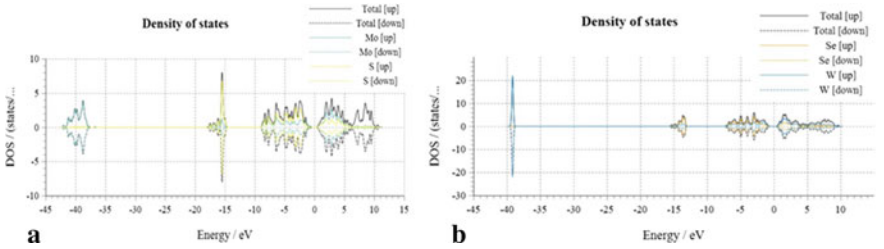


Fig. 5 Density of states **a** MoS_2 and **b** WSe_2

4 Optoelectronics Properties

4.1 Comparison of Mono-layer and Multi-layer MoS_2

Figures 6a and 7a show absorption coefficient of single- and multiple-layer MoS_2, respectively. The inverse relationship between wavelength and bandgap energy prevents the absorption of photons with energies lower or longer than the bandgap energy (Muhammad et al. 1957). When the photon energy of the incident light is less than the bandgap energy of the material, the material appears transparent to the incident light. The absorption coefficient determines how far a certain wavelength of light may go through a substance before being completely absorbed by it (Muhammad et al. 1957; Splendiani et al. 2010). Fundamental absorption is a phenomenon that results in a sharp rise of absorption coefficient for light showing a wavelength smaller than the bandgap. Figures 6a and 7a demonstrate that the absorption coefficients of monolayers and bulk layers are notably high for the visible spectrum (400–500 nm). These graphs demonstrate a rapid decline in the absorption coefficient under 500 nm of light. The absorption coefficient, which measures in scales of a nanometer, is what causes the fluctuating curve (Muhammad et al. 1957).

The extinction coefficient controls how readily a substance can be penetrated by the light of a specific wavelength. By the formula $k = \frac{a\lambda}{4\pi}$, where λ represents the wavelength and α represents the coefficient of absorption. The coefficient of extinction and the coefficient of absorption are connected. The extinction coefficient of single-layer and multi-layer MoS_2 are shown in Figs. 6b and 7b. At about 450 nm, the mono-layer extinction coefficient peaks (Wang et al. 2012b; M., Sanaullah and Masud H Chowdhury. 2015). This shows that mono-layer MoS_2 absorbs the majority of light at this wavelength. Moreover, the extinction coefficient drops significantly above 500 nm, making mono-layer MoS_2 transparent for wavelengths above 500 nm. When it comes to multi-layer MoS_2, the peak is also around 400 nm away (Muhammad et al. 1957). Multi-layer MoS_2 absorbs more effectively than that single-layer MoS_2 at the wavelength compared with mono-layer MoS_2, and the peak's magnitude is greater. The ratio of the speed of light in a vacuum to the speed of light in that substance is called a material's refractive index. The refractive index of a material at a particular wavelength can be calculated using the following formula: $n = \frac{c}{v}$, where c = speed in a vacuum, and v = material's phase velocity. Calculate the refractive index in logarithmic scale $\log 10(n)$ and then calculate the actual value using: $n = 10^{\log 10(n))}$ Figures 6c and 7c display the refractive index variation of mono-layer and multi-layer MoS_2 with respect to wavelength. The observation shows that the refractive index is between 1 and 2 below 250 nm in mono-layer MoS_2, showing its transparency to visible light. Hence, multi-layer MoS_2 always has a refractive index greater than 2 (with some exceptions at around 100 nm) (Radisavljevic et al. 2011). For both single and multi-layer MoS_2, almost 500 nm has the highest refractive index.

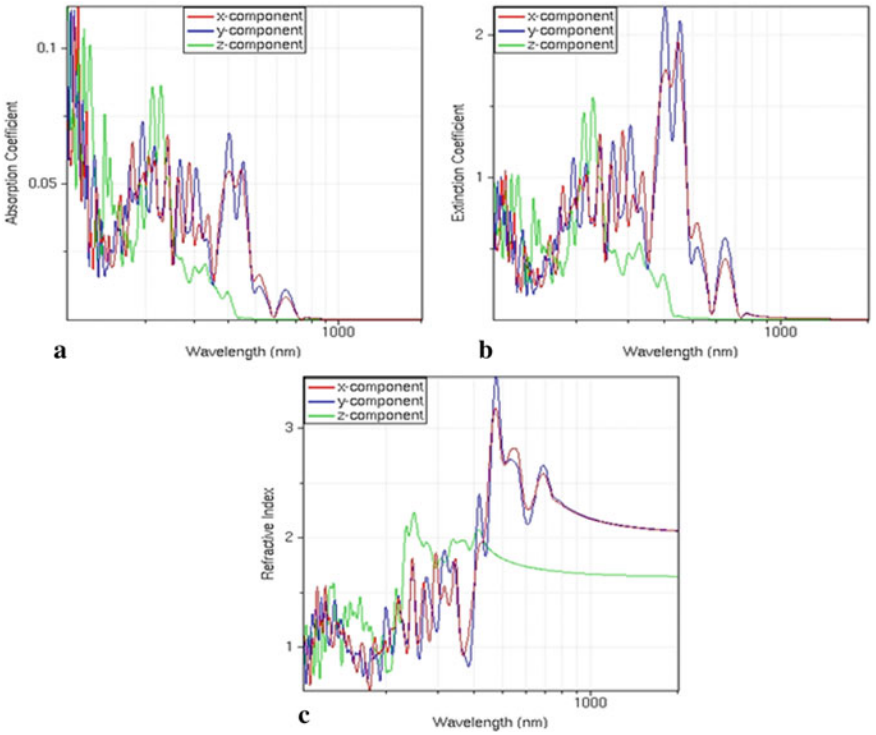


Fig. 6 Mono-layer representation of MoS_2 **a** Absorption coefficient, **b** Extinction coefficient, and **c** Refractive index

5 Conclusion

In this analysis, we have calculated the electronic properties using the first-principle DFT methods of MoS_2 and WSe_2 using Quantum Espresso, and we have observed that the results of MoS_2 are better in comparison with WSe_2. Since we are getting better electronic properties of MoS_2, after that we calculate the optoelectronics property of MoS_2 monolayer and bulk layer using the Nano Hub DFT material property simulator. Comparatively, investigate the optoelectronics properties like refractive index, extinction coefficient, and absorption coefficient of MoS_2 mono-layers and bulk-layers. The optoelectronics and electronic properties of mono-layer MoS_2 indicate its potential use in LASER and conventional field effect transistors applications. Therefore, we have concluded that MoS_2 will show better results among the two TMDs and that it may become one of the major important materials for new tunnel field effect transistor applications, and MoS_2 is among the potential materials for application in electronic devices such as solar cells, FETs, transistors, double gate FETs in pH sensors, optoelectronics, spintronics, and photoluminescence.

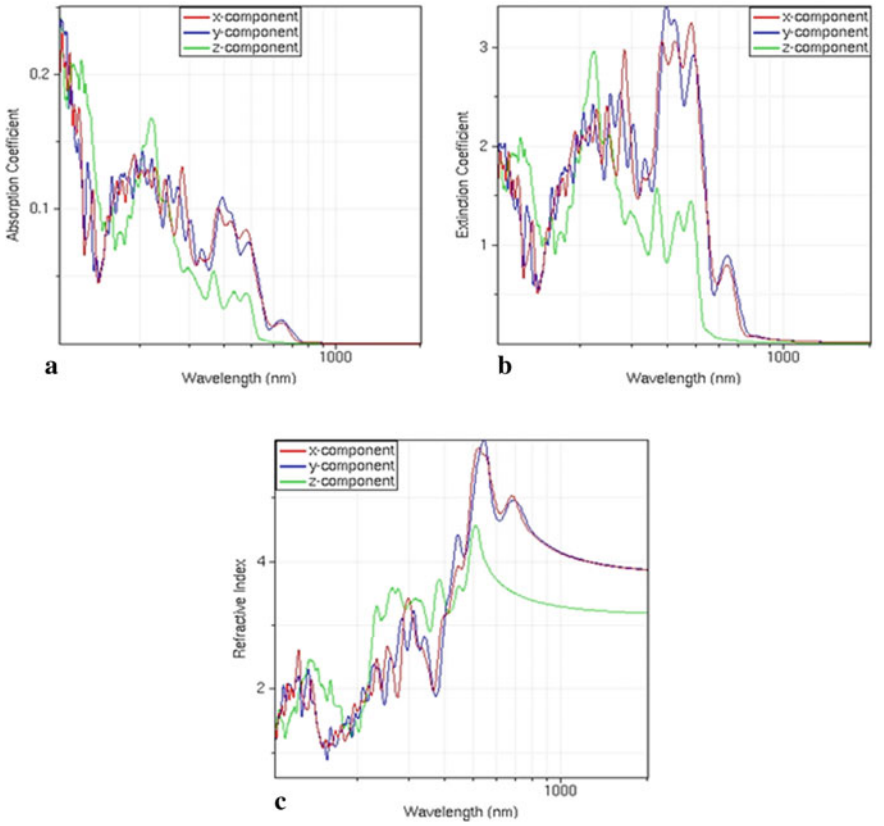


Fig. 7 Bulk-layer representation of MoS_2 **a** Absorption coefficient, **b** Extinction coefficient, and **c** Refractive index

References

Liu Y, Duan X, Shin HJ, Park S, Huang Y, Duan X (2021) Promises and prospects of two-dimensional transistors. Nature 591(7848):43–53

Ahmed S, Yi J (2017) Two-dimensional transition metal dichalcogenides and their charge carrier mobilities in field-effect transistors. Nanomicro Lett. 9(4):50

Ayari A et al (2007) Realization and electrical characterization of ultrathin crystals of layered transition-metal dichalcogenides. J Appl Phys 101:014505–014507

Wang QH, Kalantar-Zadeh K, Kis A, Coleman JN, Strano MS (2012) Electronics and optoelectronics of two-dimensional transition metal dichalcogenides. Nat Nanotechnol 7(11):699–712

Mir SH, Yadav VK, Singh JK (2020) Recent: advances in the carrier mobility of two-dimensional materials: a theoretical perspective. ACS Omega 5(24):14203–14211

Radisavljevic B, Radenovic A, Brivio J, Giacometti V, Kis A (2011) Single-layer MoS_2 transistors. Nat Nanotechnol 6(3):147–150

Blöchl PE (1994) Projector augmented-wave method. Phys Rev B 50:17953–21797

Perdew JP, Burke K, Ernzerhof M (1997) Generalized gradient approximation made simple. Phys Rev Lett 77(18):3865–3868

Giannozzi P, Baroni S, Bonini N, Calandra M, Car R, Cavazzoni C, Ceresoli D, Chiarotti GL, Cococcioni M, Dabo I, Dal Corso A, de Gironcoli S, Fabris S, Fratesi G, Gebauer R, Gerstmann U, Gougoussis C, Kokalj A, Lazzeri M, Martin-Samos L, Marzari N, Mauri F, Mazzarello R, Paolini S, Pasquarello A, Paulatto L, Sbraccia C, Scandolo S, Sclauzero G, Seitsonen AP, Smogunov A, Umari P, Wentzcovitch RM (2009) Quantum Espresso: a modular and open-source software project for quantum simulations of materials. J Phys Condens Matter 21(39):395502

Ullah MS, Yousuf AHB, Es-Sakhi AD et al (2018) Analysis of optical and electronic properties of MoS_2 for optoelectronics and FET applications. AIP Conf Proc, USA 1957:020001

Splendiani A et al (2010) Emerging photoluminescence in monolayer MoS2. Nano Lett 10:1271–1275

Wang Q, Kalantar-Zadeh K, Kis A et al (2012b) Electronics and optoelectronics of two-dimensional transition metal dichalcogenides. Nature Nanotech 7:699–712

Sanaullah M, Chowdhury MH (2015) Subthreshold swing characteristics of multilayer MoS_2 tunnel FET. In: IEEE 58th International midwest symposium on circuits and systems. Fort Collins, Colorado, USA, pp 1–4

Harmony Search-Based Fractional-Order Chebyshev Filter

Ashu Soni, Namrata Pandey, and Maneesha Gupta

Abstract This work focuses on the analysis of $(1 + \alpha)$ order Chebyshev low pass filter using the Harmony Search optimization technique. MATLAB simulations of fractional $(1 + \alpha)$ order low pass filters has been performed for α changes from 0.1 to 0.9 in the steps of 0.1. Further, the Tow-Thomas Biquad circuit is used to verify the optimized quantitative results of fractional-order Chebyshev low pass filter. MATLAB and LTSPICE simulations have been compared in order to construct the most effective $(1 + \alpha)$ fractional-order Chebyshev low pass filter.

Keywords Fractional-order filter · Chebyshev low pass filter · MATLAB · Optimization

1 Introduction

The numerous applications of fractional calculus have seen an ascent over the year. Fractional calculus is generally labeled as a simplification of integer calculus with the order of fractional operators for differential integral operators (Mahata et al. 2018; Psychalinos et al. 2016). These are the factors that limit the acknowledgment of fractional-order calculus in modern use. Also, fractional calculus in controller style has increased their workability and power. The "Caputo derivative" defines the fractional derivative of α order and is given by (Elwakil 2010; Carlson and Halijak 1964; Radwan et al. 2009)

$$ {}_{a}^{c}D_{t}^{\alpha} f(t) = \frac{1}{\tau(\alpha - n)} \int_{a}^{t} \frac{f^{(n)}(\tau) d\tau}{(t - \tau)^{\alpha+1-n}}. \quad (1) $$

A. Soni (✉)
Bharati Vidyapeeth College of Engineering, New Delhi, India
e-mail: soniashu.14@gmail.com

N. Pandey · M. Gupta
Netaji Subhas University of Technology, New Delhi, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_33

The Caputo described that the fractional-order derivative has been used over others as the initial condition is very similar to the differential equations of integer order. By using the Laplacian in (1) with initial condition of $a = 0$.

$$L\left\{{}_0^c D_t^\alpha f(t)\right\} = s^\alpha F(s) - \sum_{m=0}^{n-1} s^{\alpha-m-1} f^{(m)}(0), \tag{2}$$

where s^α is fractional Laplace operator, for zero initial condition, (2) can be written as

$$L\left\{{}_0^c D_t^\alpha f(t)\right\} = s^\alpha F(s). \tag{3}$$

Now it is general fractance device with impedances proportionate to s^α.

As per the literature survey, certain fractional-order analogue filters had already been built utilizing optimization techniques. Low pass Butterworth (Khanna and Upadhyay 2015), inverse Chebyshev, and Chebyshev filters in fractional domain have been optimized using the nonlinear least square optimization technique (Freeborn et al. 2013). Mahata et al. (2018) also proposed the GSA optimized fractional-order low pass Butterworth filter is to create the best fractional filter with precise magnitude response.

Harmony Search (HS) (Geem et al. 2001; Kumar et al. 2012) approach, on the other hand, has not yet used for the analysis of fractional-order Chebyshev low pass filters (FOCLPF). In the suggested work, HS is employed on FOCLPF. Our research aims to create a technique for discovering a solution that has a quick response and gives a more authentic solution to the issue while maintaining the shortest possible response time. The $(1 + \alpha)$ order low pass filters have MATLAB simulations with fractional steps ranging from 0.1 to 0.9. The flow of paper is as follows. The fitness function and meta-heuristic method are discussed in Sect. 2. Section 3 focuses on MATLAB simulation findings, whereas Sect. 4 focuses on LTSPICE implementation. Section 5 accomplishes the proposed work.

2 Proposed Work

The discussion of fitness function and the description of the flowchart of HS optimization method are studied in this section. Further, the HS optimization technique is employed to optimize the FOCLPF filter coefficients. Figure 1 shows the flow diagram of HS optimization.

The transfer function of low pass $(1 + \alpha)$ order filter is given by (Mahata et al. 2018; Psychalinos et al. 2016)

$$H_{LP}^{1+\alpha}(S) = \frac{a_0}{a_1 S^{1+\alpha} + a_2 S^\alpha + 1}, \tag{4}$$

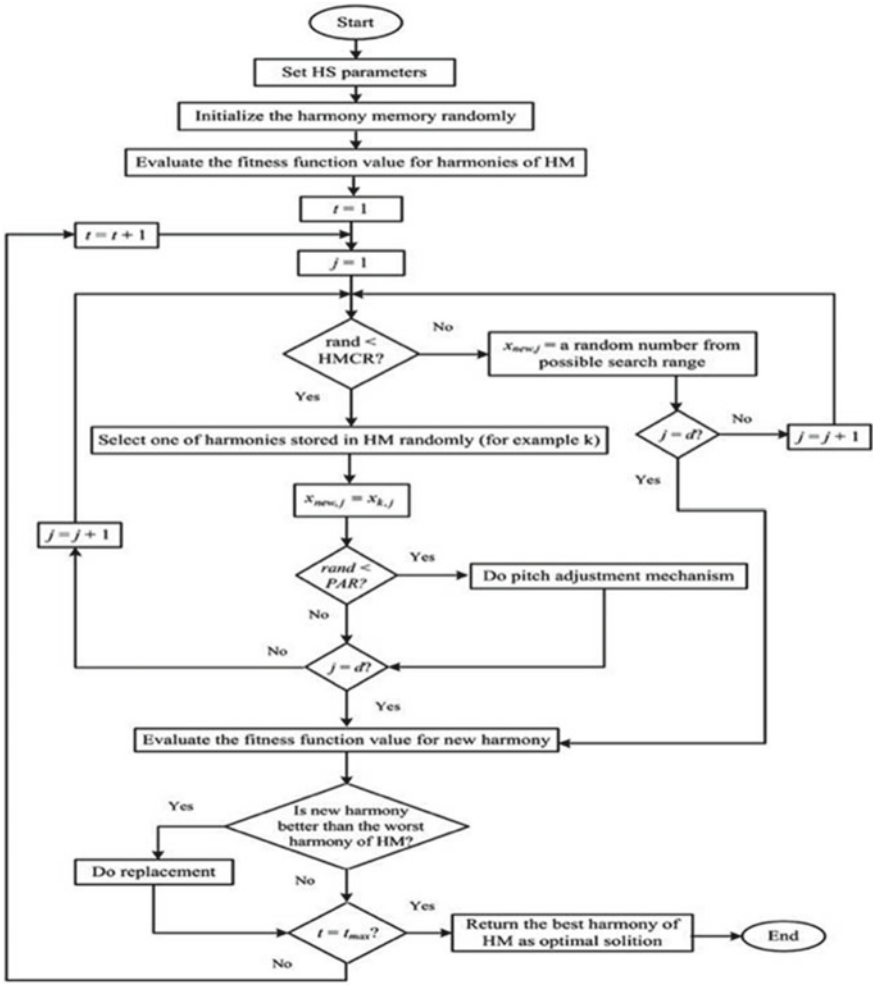


Fig. 1 Flowchart of HS optimization technique

where $a_1, a_2, a_3,$ and a_4 are filter coefficients to be optimized so that the fitness function value is reduced.

With a normalized frequency of 1 rad/s, a second-order transfer function of Chebyshev filter is

$$C_2(s) = i\frac{0.5012}{s^2 + 0.6449s + 0.7079}. \tag{5}$$

To reduce the mean square error (MSE) of the fitness function, HS optimization is used:

$$\min_x \| |H(x,\omega)| - |C_n(\omega)| \|_2^2 = \min_x \sum_{i=1}^{k} (|H(x,\omega)| - |C_n(\omega)|)^2$$
$$\text{such that } x > 0.1, \tag{6}$$

where $|H(x,\omega_i)|$ is the fractional-order filter, $|C_n(\omega_i)|$ is the function of integer order Chebyshev low pass filter, ω_i is the range of frequency between $10^{-5} < \omega$ (rad/s) < 1.5.

3 Numerical Analysis

The simulations of $(1 + \alpha)$ order Chebyshev low pass filter have been done on MATLAB 2018a. The HS optimization is applied to the function to reduce the MSE (3). The input variables are set in such a way that the optimization is completed in less than 5 s.

Table 1 lists the variables for HS optimization technique. The FOCLPF coefficients of filter utilizing the HS method are given in Table 2. Figure 2 depicts the responses of magnitude and phase of HS optimized FOCLPF.

Table 1 Input variables for HS optimization

Variable	Value
Iterations	1000
Harmony memory size (HMS)	25
Harmony memory consideration rate (HMCR)	0.9
Pitch adjustment rate (PAR)	0.1
Fret width (FW)	0.02*(Vmax−Vmin)
Vmax	10
Vmin	−10

Table 2 Filter coefficients of FOCLPF using HS optimization

Order	Filter's coefficients		
	a_0	a_1	a_2
1.2	0.7595	0.6005	0.05
1.5	0.7205	1.2218	0.1578
1.8	0.6605	1.1567	0.1853

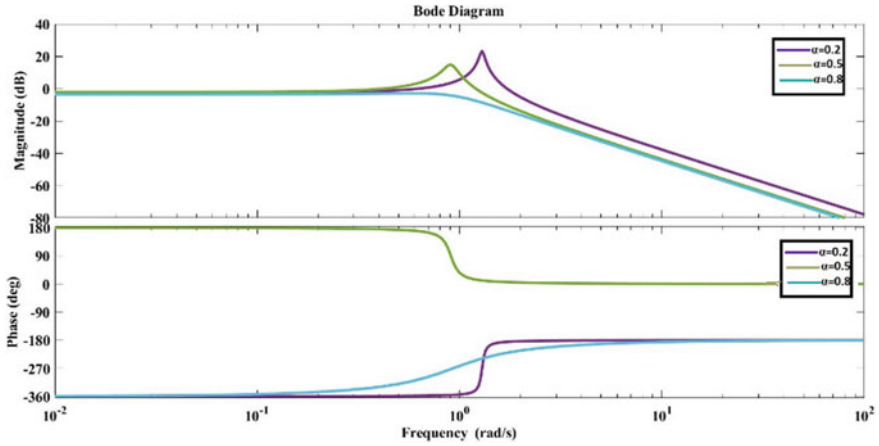


Fig. 2 Magnitude and phase characteristics of HS optimized $(1+\alpha)$ order filter

4 Circuit Implementation

The Tow-Thomas Biquad circuit, as shown in Fig. 3, can be used to implement the HS optimized FOCLPF, with $Z_{C2} = 1/s^{\alpha}c_2$ (C_2 is substituted by a fractional-order impedance capacitor) and $0 \leq \alpha \leq 1$ (Freeborn et al. 2013). The transfer function of the Tow-Thomas Biquad is provided by

$$\frac{V_{\text{out}}(s)}{V_{\text{in}}(s)} = \frac{R_3R_5/R_4R_6}{S^{1+\alpha}R_2R_3C_1C_2 + S^{\alpha}\left(\frac{R_2R_3C_2}{R_1}\right) + 1}. \tag{7}$$

The values of the required resistances for the HS optimized proposed filter are evaluated by comparing the coefficients of (4) and (7), with capacitance value $1F$ and R_4, R_5, R_6 (Ω). Further, the values of $R_1(\Omega)$, $R_2(\Omega)$, and $R_3(\Omega)$ and R_1, R_2, and R_3

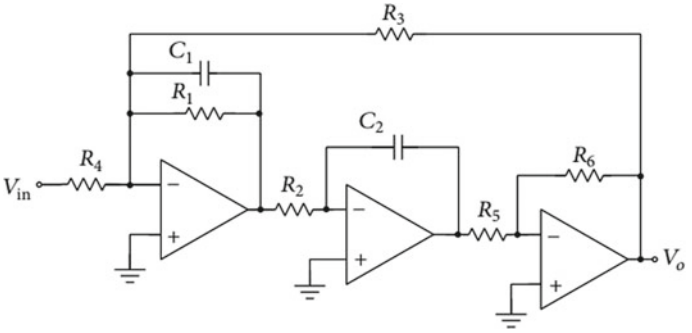


Fig. 3 Tow-Thomas Biquad topology

Table 3 Components values of $(1 + \alpha)$ order of FOCLPF using HS optimization technique with a scaling factor of 1000

Component	Value of FOCLPF		
	0.2	0.5	0.8
C_1 (F)	1	0.159μ	1
C_2 (F)	173.9μ	12.6μ	0.915μ
$R_1(\Omega)$	10.13 K	12.12 K	4.142 K
$R_2(\Omega)$	666.2	1.642 K	1.751 K
$R_3(\Omega)$	759.63	722.5	660.5
R_4, R_5, R_6 (Ω)		1000	

are scaled by the factor 1000. The capacitances and resistance values used in Fig. 3 are tabulated in Table 3.

Further, the LTSPICE simulations of Tow-Thomas Biquad using above value of components (as given in Table 3) is shown in Fig. 4. LTSPICE simulated magnitude characteristics agree well with the MATLAB simulated responses, with a reasonable cut-off frequency error (mentioned in Table 4). Therefore, Tow-Thomas circuit can be used to implement HS optimized FOCLPF responses using comparative fractional-order capacitor.

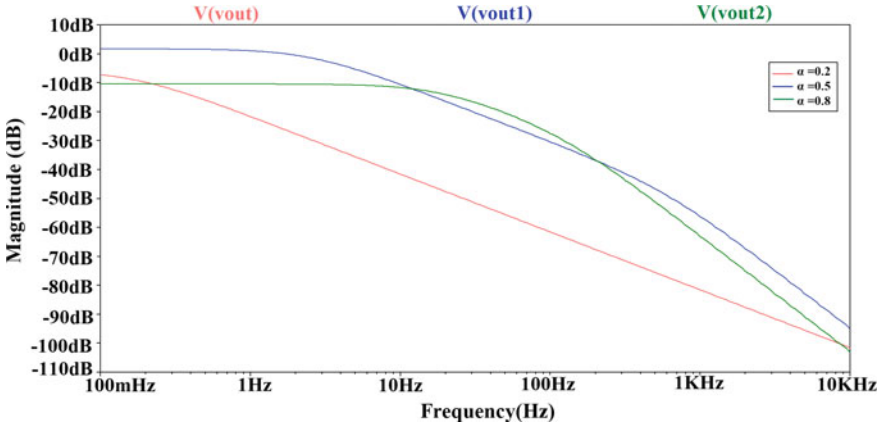


Fig. 4 Simulations result of the magnitude of FOCLPF of $(1 + \alpha)$ order filter using Tow-Thomas for the $\alpha = 0.2$, 0.5, and 0.8

Table 4 Comparison of cut-off frequency of $(1 + \alpha)$ order of FOCLPF using harmony search optimization technique of MATLAB and LTSPICE

Order	Cut-off frequency		
	MATLAB	LTSPICE	Absolute error
1.2	1.32	0.3602	0.0402
1.5	0.967	1.5328	0.5658
1.8	1.05	1.7524	0.7024

5 Conclusion

The proposed HS optimization strategy is used to determine the FOCLPF optimum coefficients. This optimized filter has achieved minimal cut-off frequency error and has given excellent frequency response. The results of the implementation reveal that the HS optimization results met the FOCLPF design requirements. Other approximation techniques and higher-order fractional filters can be used to extend this work.

References

Carlson GE, Halijak CA (1964) Approximation of fractional capacitors by a regular newton process. IEEE Trans Circuits Syst 11:210–213

Elwakil A (2010) Fractional-order circuits, and systems: an emerging interdisciplinary research area. IEEE Circuits Syst Mag 10:40–50

Freeborn TJ, Maundy B, Elwakil AS (2013) A fractional resonance-based RLβCα filters. Hindawi Publishing Corporation, pp 1–10

Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: harmony search. SIMULATION 76(2):60–68

Khanna T, Upadhyay DK (2015) Design and analysis of higher-order fractional step Butterworth filters. In: International conference on soft computing techniques and implementations (ICSCTI)

Kumar V, Chhabra JK, Kumar D (2012) Effect of harmony search parameters' variation in clustering. Proc Technol 6:256–274

Mahata S, Saha SK, Kar R, Mandal D (2018) Optimal design of fractional order lowpass butterworth filter with accurate magnitude response. Digital Signal Process 72:96–114

Psychalinos C, Tsirimokou G, Elwakil AS (2016) Switched-capacitor fractional-step Butterworth filter design. Circuits Systems Signal Process 35:1377–1393

Radwan AG, Soliman AM, Elwakil AS, Sedeek A (2009) On the stability of linear systems with fractional-order elements. Chaos Solitons Fractals 40(5):2317–2328

A Neural Network-Based Wavelet Thresholding Technique for De-Speckling of Ultrasound Images

Mayank kumar Singh, Indu Saini, and Neetu Sood

Abstract The mortality rate and life lost in suffering can be significantly improved if the cancer is detected in its earliest stage. For early detection, it is required to design a system that can be used frequently. Only ultrasound waves that are used for disease diagnosis are free from radiation. The ultrasound images suffer from speckle noise, which reduces their visual quality. Machine learning techniques have helped researchers to achieve significant results. Its application has been extended to almost every field of science. So, reducing the speckle noise in ultrasound images by using machine learning techniques is the aim of this study. In the proposed methodology, the image is transformed using logarithmic and wavelet transform. The histogram of the detailed sub-band was then used to train a neural network to generate modified coefficients. The network was trained in an unsupervised fashion. The study was tested on the breast and liver ultrasound images. The de-speckling of computer-generated images resulted in a PSNR and MSE of 45.6 and 58, respectively. The technique when applied to the ultrasound image dataset the reference less quality evaluator gave the value 31.6, which is comparatively better than the original image.

Keywords Ultrasound · Speckle noise · Wavelet thresholding · Artificial neural network

1 Introduction

Cancer has always been the biggest threat to humankind as it can possibly develop to any part of human body (Siegel et al. 2022). Cancer is the name given to the unauthorized and limitless growth of tissue, which sometimes penetrates surrounding organs (Dunn et al. 2004). There is no perceptual difference between a cancer patient and normal person. The cancer grows inside our body without exciting any visible

M. Singh (✉) · I. Saini · N. Sood
Department of Electronics and Communication, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab, India
e-mail: mayankks.ec.19@nitj.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_34

changes in the body. Most of people came to know about their cancer when it has spread and causing severe symptoms. Thankfully, there are successful treatment procedures for the early-stage cancer. The survival rate of a patient of higher stage cancer is very low. So, it is very crucial for survival of a cancer patient that the cancer is identified in its early stage.

The early detection is possible only through diagnosis. The invasive diagnosis technique is considered as gold standard, but it cannot be practiced frequently as it is painful. The non-invasive diagnostic imaging technique can be used for diagnosis without causing any pain. But unfortunately, most of the diagnosis techniques involve harmful radiation. Although they have low dose of radiation, still one cannot be allowed to its exposure without recommendation. The ultrasound (US)-based diagnosis is the only radiation free technique in practice. But the US has its one big limitation the speckles. The speckles are the multiplicative noise which are resulted from internal reflections. They reduce the quality of US which can potentially cloud the judgment of the pathologist. With an effective denoising technique, the quality of US image can be improved.

Spatial filters like Mean, Median, Weiner filter, and Weighted Mean filters were initially used for de-speckling. But they were not exhaustively designed to deal with the speckles. The filters reduce the speckle noise but at the cost of important details in the image. The averaging of images with images of different frames for de-speckling also resulted in information loss. After few years, a novel anisotropic diffusion method was introduced for de-speckling. The filter applies smoothing operation but in a guided way. The guided filter was able to reduce the noise while preserving most of the details. For spatial filters, the choice of filter dimension is not fixed, so there is always a tradeoff between efficient de-speckling and edge preservation. Another transform-based de-speckling technique is also famous among researchers. It is named Wavelet Shrinkage (WS) where the speckle is first converted into additive component followed by wavelet transform (Vidakovic 1998). The technique uses a threshold and a shrinkage function for modification of coefficients (Chang et al. 2000). The WS is independent from choice of filter dimension and also efficient in preserving the details. The optimal choice of threshold and shrinkage function can significantly reduce the speckles. Soft and hard thresholding has been used primarily as the shrinkage function (Kaur et al. 2003), but they produced irregular output. Exponential thresholding proposed in Andria et al. (2013) have results better than the conventional function. Later on studies in Randhawa et al. (2019); Bedi and Sunkaria 2022) generalized the thresholding functions, with better choice of parameter the efficiency can be improved. Another study in Elyasi et al. (2016) proposed a homogeneity modified Bayesian threshold. Instead of using conventional universal threshold, the technique used Bayesian threshold with an improvised methodology. All these techniques had a good de-speckling performance but with the intervention of machine learning techniques the performance can be improved further. Machine learning is known for its capability of improving the performance for any task in hand. It has been used for various image processing task like enhancement (Wang et al. 2020), segmentation (Chen et al. 2021), classification (Tan et al. 2013), etc. So, in this paper we are introducing a wavelet thresholding technique with artificial

neural network for coefficient modification. The network was trained with no target class or in unsupervised manner. The ANN was set to optimize its parameters for reducing the distortion level in the image. Referenceless quality metric together with structural similarity index (SSI) was used for training the network. The network was tested on public US dataset and compared with various practiced technologies.

The rest of the paper is arranged as follows. After an introduction to the imaging techniques and a brief description of extensive literature survey, Sect. 2 covers the methodology proposed for de-speckling in this literature. And the outcome of the proposed technology is provided in Sect. 3. And the section also contains the comparison of proposed technique with other best in class de-speckling techniques. The conclusion of this study is summarized in the last section.

2 Methodology

In Fig. 1, we have provided the process flow of the proposed methodology. The US is first transformed using logarithmic operator. As the speckle noise is in multiplicative in nature, after this transformation speckles are converted into additive noise. In Eq. (1), U_o is the original US image which contains the noise, U_i is the noise free image, and S_n is the speckle noise. The U_o', U_i', and S_n' in Eq. (2) are the log transformed version of original US image which contains the noise, noise free image, and speckle noise, respectively.

$$U_o = U_i \times S_n, \tag{1}$$

$$U_o' = U_i' + S_n'. \tag{2}$$

The corrupted image is next transformed using the Daubechies wavelet transform (Waqas et al. 2020). The image is divided into sub-bands consisting different components of the image. In Fig. 2, we have shown the images after logarithmic and wavelet transform. Apart from the multiresolution analysis, the benefit of using wavelet is that it also divides the image into sub-bands of different level of information. The approximation sub-band consists the majority of the important details of the image, while the rest of the sub-bands (detail sub-bands) consist comparatively less information of image. The diagonal (D) sub-band contains the least image information and most of the noise details.

For experimentation we have also involved some standard gray scale image (Meerwald 2023). The image is a natural image with no inherent speckle noise US image. We have intentionally added speckle noise in the image to visualize the changes after noise addition. The change in the sub-band histogram is seen in Fig. 3. The effect of speckles on coefficients of sub-bands can be seen in the graph. The noise has introduced many larger valued coefficients and has expanded the range of histogram. Since

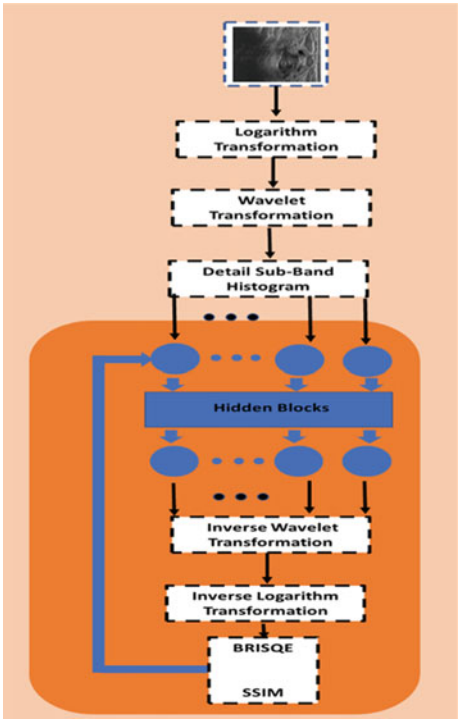


Fig. 1 Process flow of the proposed technique

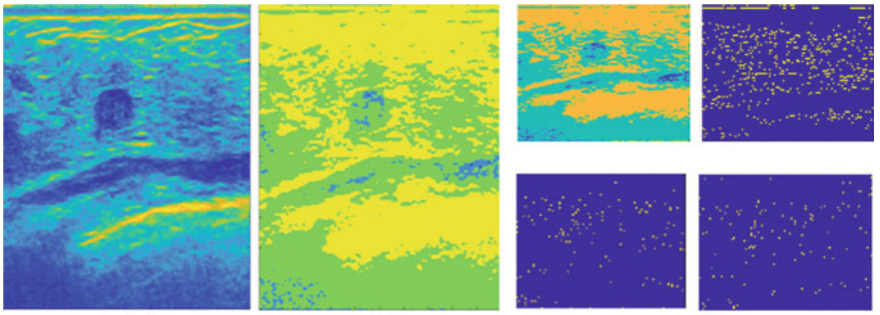


Fig. 2 **a** Original ultrasound image, **b** logarithm transformed version of the original, and **c** the sub-bands after wavelet transform of log transformed images

approximation sub-band has the major image details so, it will remain untouched throughout the process.

The motive of coefficient thresholding is to recover the distribution by assigning the coefficient a value after comparing with a calculated threshold. But instead of calculating the threshold and also avoiding the computation required to calculate

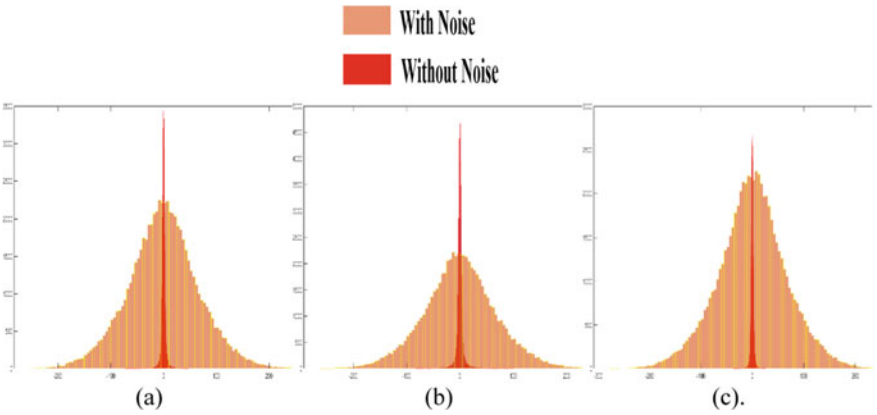


Fig. 3 Standard image in Meerwald (2023) was considered as a clean image and was intentionally corrupted by speckle noise. The noisy and clean image was decomposed into sub-bands using wavelets. From the sub-bands we extracted the coefficient distribution of **a** the H sub-band, **b** the V sub-band, and **c** the D sub-band, respectively, from both images

threshold, we trained a network using the histogram. ADAM optimizer was used to enhance the parameters of the network. For US image there is no noise free image that can be used as a target image, so supervised learning is not an option. To teach the network, an unsupervised learning a novel loss function was defined. The loss function has two parts each with its own agenda. The first part consists of Blind Reference less image quality evaluator (BRISQ). The property of the BRISQ to analyze distortion in an image without any reference was used as a noise factor. Lower the value the better the image quality, so it is taken as an additive factor. Hence, the network will evolve its parameter to reduce the BRISQ. But modification of the histogram may cause alteration in the fundamental properties of the image. Preserving the fundamental properties is utmost important as it may affect the outcome of diagnosis. Hence, we introduced the second component Limited Structural Similarity Index (LSSI). The LSSI limits the changes in the histogram to preserve the important properties of the image. Equation (3) is the loss function used to train the ANN.

$$L_{\emptyset} = \mathrm{BRISQ}(I') - \mathrm{LSSI}(I'), \tag{3}$$

$$\mathrm{LSSI}(I') = e^{|\mathrm{SSI}(I')-0.5|}, \tag{4}$$

$$\mathrm{SSI}(I') = \frac{(2\mu_I\mu_{I'} + C1)(2\sigma_{II'} + C2)}{(\mu_I^2 + \mu_{I'}^2 + C1)(\sigma_I^2 + \sigma_{I'}^2 + C2)}. \tag{5}$$

Here for input image I, I' is the image obtained after the ANN modified coefficients were used in inverse wavelet and logarithm transform, where μ_I, $\mu_{I'}$, σ_I, $\sigma_{I'}$, and $\sigma_{II'}$ are the local means, standard deviations, and cross-covariance, respectively.

3 Results and Discussion

The proposed denoising technique was applied over US image and a computer-generated image (CGI). To legitimize the performance of the technique, a CGI was used. The image was taken as a reference, and speckle noise was added to it. Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) were used to determine the efficiency of proposed technique in denoising and restoration. Table 1 consists the value of PSNR and MSE for the proposed technique. For comparison, the Exponential Shrinkage (Exp-SH) (Andria et al. 2013), Homogeneity modified Bayesian shrink (Homo-SH) (Elyasi et al. 2016), Optimized Bayesian Shrink (Pot-SH) (Bedi and Sunkaria 2022), and Wiener filter (Chen et al. 2006) were implemented. Table 1 provides the average PSNR obtained after denoising of the CGI, using the techniques.

The PSNR values in Table 1 represent the efficacy of the techniques in denoising if the CGI. The Weiner filter was observed to have the least PSNR, which means the filter has very less noise suppression. Among the other techniques the Opt-SH was able to achieve a good performance. But the proposed technique has even better results. The highest PSNR value implies that the proposed technique has significantly reduced the noise. The MSE in Table 2 quantifies the efficacy of technique in recovering the original image. There is a tendency of the denoising techniques to destroy or later the image information while removing the noise. It is obvious that a good denoising technique should have the least tendency to alter image information. This is what we can judge from the MSE values.

From the table it is clear that the proposed technique has the least tendency to destroy the image information. Both PSNR and MSE signify the capability of

Table 1 PSNR values of denoising techniques for CGI

Technique	PSNR
Exp-SH	31.69
Homo-SH	28.64
Weiner filter	25.61
Opts-SH	40.3
Proposed	45.6

Table 2 MSE values of denoising techniques for CGI

Technique	MSE
Exp-SH	297
Homo-SH	366
Weiner filter	132
Opt-SH	280
Proposed	58

proposed technique in removing speckles and preserving the image details simultaneously. The PSNR and MSE were calculated from the CGI, to evaluate the effect of our technique for US we have used BRISQ. The BRISQ determines the distortion in the image. Unlike the PSNR and MSE, it does not require reference image. The average BRISQ observed for the unprocessed original US was 43.1; but after processing with proposed technique, the value comes down to 31.7. The lower values imply less distortion in the image. Hence, the proposed technique was able to improve the quality of the US image.

4 Conclusion

In this paper, we have proposed a machine learning-based de-speckling technique for US. In the proposed methodology, the image is initially transformed to convert the multiplicative speckle noise into additive noise. Later, an ANN was trained to modify the coefficients of the transformed images. This ANN was trained using a novel loss function. The SSIM in the loss function makes sure that the transformation does not destroy the important features. While the BRISQE factor in the loss was introduced, that network learns to reduce the speckles. The ANN was set to train and learn to create a balance between the similarity and distortion in the image. The proposed technique was compared with other techniques in practice, and the proposed technique resulted in a better de-speckling The PSNR and MSE obtained were 45.6 and 58 respectively when observed over a synthetic image. For US image dataset and the BRISQE value was 31.7. The technique was able to remove the noise and most importantly preserve the important details of the image.

References

Andria G, Attivissimo F, Lanzolla AML, Savino M (2013) A suitable threshold for speckle reduction in ultrasound images. IEEE Trans Instrum Meas 62(8):2270–2279. https://doi.org/10.1109/TIM.2013.2255978

Bedi AK, Sunkaria RK (2022) Ultrasound speckle reduction using adaptive wavelet thresholding. Multidimens Syst Signal Process 33(2):275–300. https://doi.org/10.1007/s11045-021-00799-4

Chang SG, Yu B, Vetterli M (2000) Adaptive wavelet thresholding for image denoising and compression. IEEE Trans Image Process 9(9):1532–1546. https://doi.org/10.1109/83.862633

Chen J, Benesty J, Huang Y, Doclo S (2006) New insights into the noise reduction Wiener filter. IEEE Trans Audio Speech Lang Process 14(4):1218–1234. https://doi.org/10.1109/TSA.2005.860851

Chen H et al (2021) Segmentation of lymph nodes in ultrasound images using U-net convolutional neural networks and gabor-based anisotropic diffusion. J Med Biol Eng 41(6):942–952. https://doi.org/10.1007/s40846-021-00670-8

Dunn GP, Old LJ, Schreiber RD (2004) The immunobiology of cancer immunosurveillance and immunoediting. Immunity 21(2):137–148. https://doi.org/10.1016/j.immuni.2004.07.017

Elyasi I, Pourmina MA, Moin M-S (2016) Speckle reduction in breast cancer ultrasound images by using homogeneity modified bayes shrink. Measurement 91:55–65. https://doi.org/10.1016/j.measurement.2016.05.025

Kaur L, Gupta S, Chauhan RC (2003) Image denoising using wavelet thresholding. Third Conf Comput Vision Graph Image Process

Peter Meerwald, "No Title." https://www.cosy.sbg.ac.at/~pmeerw/Watermarking/lena.html. Accessed 28 Mar 2023

Randhawa SK, Sunkaria RK, Puthooran E (2019) Despeckling of ultrasound images using novel adaptive wavelet thresholding function. Multidimens. Syst. Signal Process. 30(3):1545–1561. https://doi.org/10.1007/s11045-018-0616-y

Siegel RL, Miller KD, Fuchs HE, Jemal A (2022) Cancer statistics, 2022. CA Cancer J Clin 72(1):7–33. https://doi.org/10.3322/caac.21708

Tan T, Platel B, Mus R, Tabár L, Mann RM, Karssemeijer N (2013) Computer-aided detection of cancer in automated 3-D breast ultrasound. IEEE Trans Med Imaging 32(9):1698–1706. https://doi.org/10.1109/TMI.2013.2263389

Vidakovic B (1998) Nonlinear wavelet shrinkage with Bayes rules and bayes factors. J Am Stat Assoc 93(441):173–179. https://doi.org/10.1080/01621459.1998.10474099

Wang W, Wu X, Yuan X, Gao Z (2020) An experiment-based review of low-light image enhancement methods. IEEE Access 8:87884–87917. https://doi.org/10.1109/ACCESS.2020.2992749

Waqas UA, Khan M, Batool SI (2020) A new watermarking scheme based on Daubechies wavelet and chaotic map for quick response code images. Multimed Tools Appl 79(9):6891–6914. https://doi.org/10.1007/s11042-019-08570-5

Artificial Neural Network (ANN)-Based Supervised Control of Single Wheel Robotic System (SWRS)

Ashwani Kharola, Ayush Krishali, Prateek Gurung, and Prince Kumar Jha

Abstract This study considers a supervised learning technique based on feedforward artificial neural network (ANN) for control of highly nonlinear and multivariable single-wheel robotic system (SWRS). The nonlinear governing equations of motion for the system were derived using Newton's second law and simulated in MATLAB/Simulink platform. The outcomes of PID control were considered for training of ANN controller. Excellent regression results of almost 0.99 were obtained after training. Finally, a comparison analysis has been done between PID and ANN controllers in terms of settling time, overshoot ranges and steady state error response.

Keywords SWRS · Nonlinearity · ANN · MATLAB · Simulink

1 Introduction

Wheeled mobile robots are very popular control devices widely used in various industrial applications (Kim and Singh 2022). These systems are capable of achieving better mobility and rotation in smaller spaces (Liu et al. 2022). The primary challenge while designing these systems is to attain an optimal position and tilt control (Khalaji and Jalalnezhad 2021). Researchers have been adopting various control techniques for control of these systems (Deng et al. 2010; Rubio et al. 2019; Abadi et al. 2020). For an instance, an artificial neural network (ANN)-based switching controller for control of two-wheeled self-balancing robot has been proposed in Unluturk and Aydogdu (2017). The controller balances the proposed robot on various granular surfaces. The results indicate more robust behaviour of proposed controller compared to classical ones.

A. Kharola (✉) · A. Krishali · P. Gurung · P. K. Jha
Graphic Era Deemed to Be University, Dehradun, Uttarakhand 248002, India
e-mail: ashwanidaa@gmail.com

P. K. Jha
e-mail: princzkumarjha@gmail.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_35

Further, a low cost self-balancing robot powered using stepper motors has been highlighted in Borja et al. (2020). The design was an improvement over a previous design developed using DC motor. The results indicate that the new model not only improved the performance but also reduced the overall cost of the complete system. Moreover, a neuro-fuzzy controller for stabilization of a two-wheeled autonomous robot has been proposed in Su et al. (2010). The study considered fuzzy inference as the main controller, while the neural network was kept as an auxiliary controller to observe the uncertainties arising in the system.

Furthermore, an approach based on combination of ANN and input–output linearization technique for trajectory tracking control of a two-degree of freedom self-balancing robot has been suggested in Gandarilla et al. (2022). Real-time experiments were conducted which highlighted better tracking accuracy and disturbance rejection via proposed approach. In this study, a feedforward ANN controller has been designed for control of SWRS. The responses of PID control were utilized for training of ANN controller. A mathematical model of SWRS has been initially developed and simulated in MATLAB/Simulink platform. The performance of ANN controller has been verified through responses obtained for settling time, overshoot and steady state error.

2 Mathematical Equations and Simulink of SWRS

The system comprises robot body having mass ($M = 1$ kg) and length ($L = 0.1$ m) attached on a single wheel of mass ($W = 0.25$ kg) and radius ($R = 0.05$ m) as shown in Fig. 1. The moment of inertia of the wheel (I) and pendulum (J) were 0.0123 kg m^2 and 0.1984 kg m^2, respectively. The aim is to stabilize the robot at desired angle (θ) and position (x), respectively. The system is acted upon by a control force (F) and gravity ($g = 9.81$ m/s^2).

The mathematical equations for SWRS were derived using Newton's second law (Huang 2010). These equations were used for developing a Simulink model of SWRS

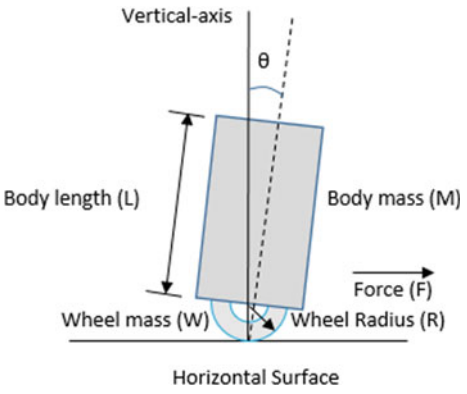


Fig. 1 SWRS moving on horizontal surface

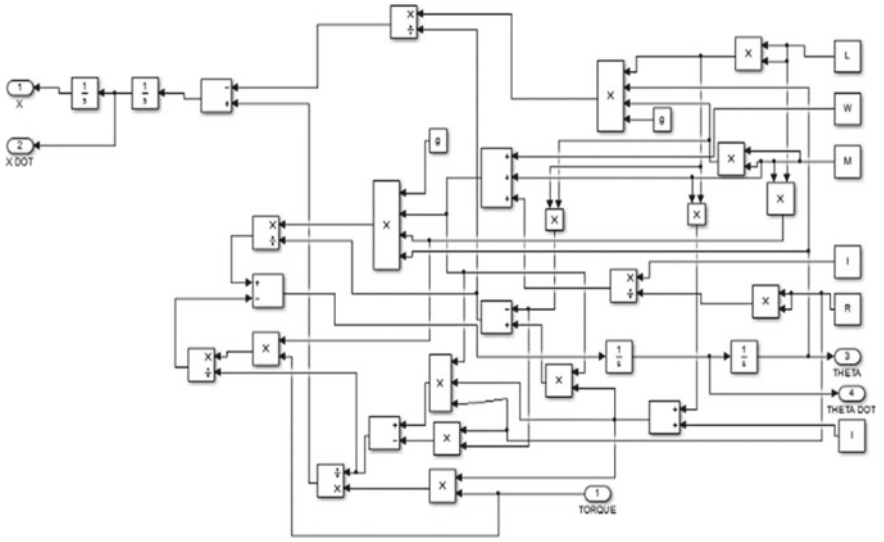


Fig. 2 Simulink model of SWRS

as shown in Fig. 2. The final expressions obtained for linear acceleration of the wheel ($\ddot{x}$) and angular acceleration of the chassis ($\ddot{\theta}$) are

$$\ddot{x} = \frac{-M^2L^2g\theta}{M_t\left(ML^2+I\right)-M^2L^2} + \frac{\left(M^2L^2+I\right)F}{M_t\left(ML^2+I\right)-M^2L^2}, \tag{1}$$

$$\ddot{\theta} = \frac{MgLM_s\theta}{M_t\left(ML^2+I\right)-M^2L^2} - \frac{MLF}{M_t\left(ML^2+I\right)-M^2L^2}, \tag{2}$$

where $M_t = M + W + \frac{J}{R^2}$.

3 ANN-Based Control of SWRS

ANNs are computing networks inspired by biological neurons of human brain (Kansal et al. 2023). These neurons have some weights which are adjusted during the learning process (Farias et al. 2020). ANNs learn by processing known inputs and evaluate an error as the difference between the processed and desired output (Dhiman et al. 2023). The best validation performance, regression results and error histogram obtained after training are shown with the help of Fig. 3.

The ANN controller proposed in this study has been designed using 20 neurons in the hidden layer and 1 neuron in the output layer and trained using Levenberg–Marquardt training algorithm. The best validation performance is 5.3803e-05

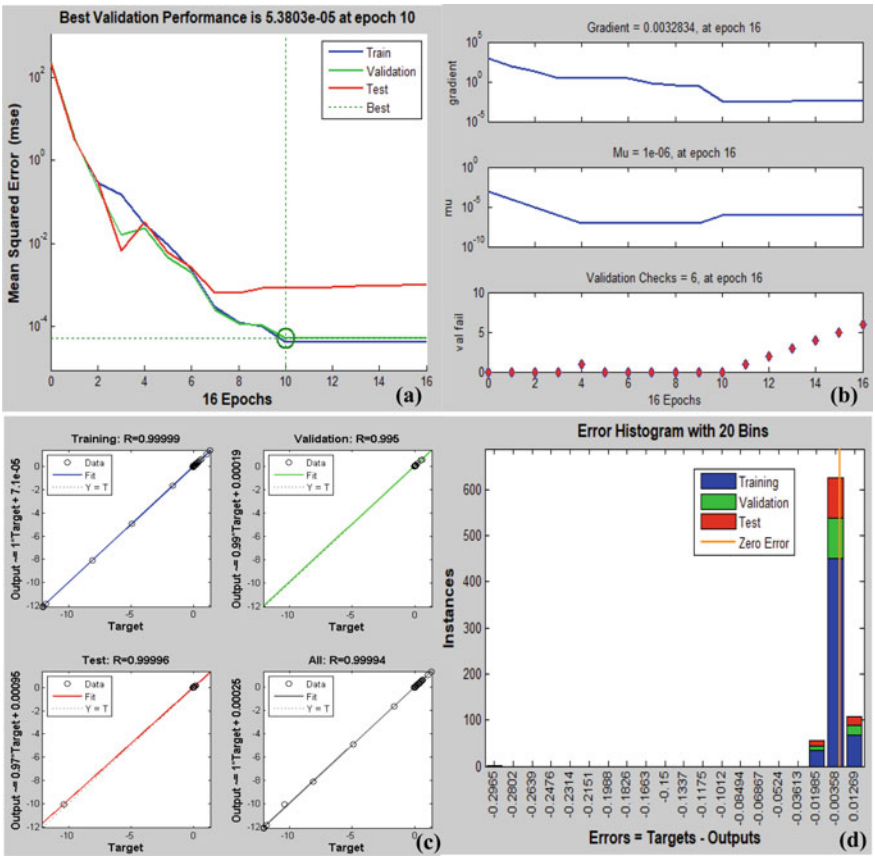


Fig. 3 a Best validation performance, **b** slopes for gradient, Mu and validation checks, **c** regression results, **d** error histogram obtained after training

obtained at 10 epochs as shown in Fig. 3a. The plots for gradient, *Mu* and validation checks are shown with the help of Fig. 3b. The value of gradient and *Mu* obtained after 16 epochs were 0.0032834 and 1e-06, respectively. The results indicate that excellent regression values in order of 0.99 has been obtained for all the category of samples as shown in Fig. 3c. The error histogram for 20 bins along x-axis is shown in Fig. 3d. It can be observed from the results that the zero error line which corresponds to zero error value falls under bin with value −0.00358.

4 Results and Comparison

The responses obtained after simulation ($t = 10$ s) for SWRS using PID and ANN controller are highlighted in Fig. 4.

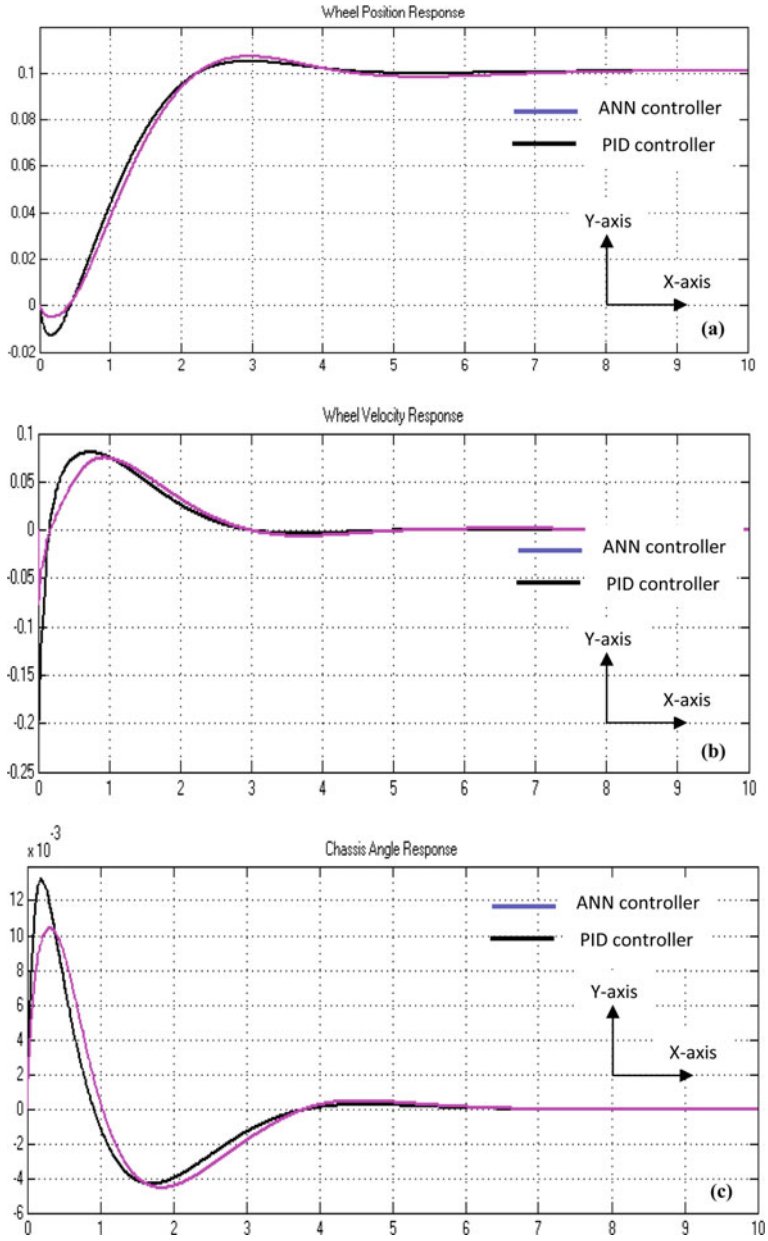


Fig. 4 Simulation responses for **a** wheel position, **b** wheel velocity, **c** chassis angle, **d** chassis angular velocity

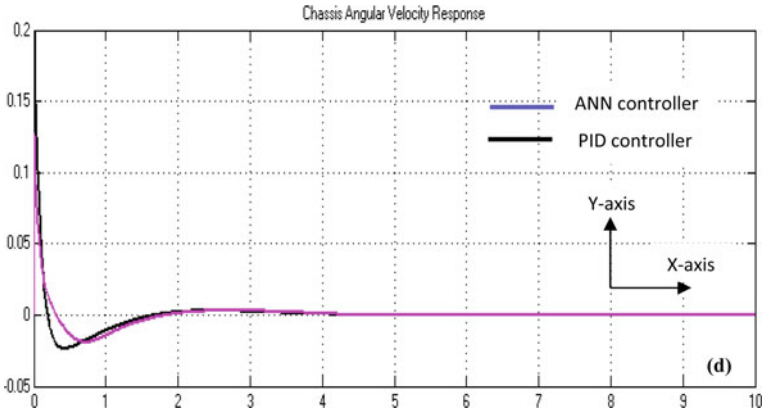


Fig. 4 (continued)

It can be clearly observed from the results indicated in Table 1 that both the PID and ANN controllers take almost similar settling time for stabilization of wheel position, wheel velocity, chassis angle and chassis angular velocity, respectively. It has also been observed that the overshoot responses have been significantly improved using ANN controller. Moreover, both the controllers showed excellent steady state error response with steady state error of almost zero.

Table 1 Comparison of results for different outputs

Controller	Settling time	Maximum overshoot	Steady state error
Wheel position			
PID	4.0 s	−0.01 m to 0.11 m	0
ANN	4.0 s	−0.005 m to 0.11 m	0
Wheel velocity			
PID	3.0 s	−0.2 m/s to 0.08 m/s	0
ANN	3.0 s	−0.05 m/s to 0.075 m/s	0
Chassis angle			
PID	4.0 s	-4.2×10^{-3} rad to 13.4×10^{-3} rad	0
ANN	4.0 s	-4.4×10^{-3} rad to 10.4×10^{-3} rad	0
Chassis angular velocity			
PID	2.0 s	−0.025 rad/s to 0.15 rad/s	0
ANN	2.0 s	−0.002 rad/s to 0.15 rad/s	0

5 Conclusion

The study successfully investigates application of ANN controller for supervised learning and control of highly nonlinear SWRS. The results of PID controller have been used for training of ANN controller with an excellent regression results of almost 0.99. The results indicate that both the ANN and PID controller take almost similar stabilization time for controlling different output attributes of SWRS, but the overshoot responses have been significantly reduced using ANN controller. The settling time taken by ANN controller for stabilization of wheel position, wheel velocity, chassis angle and chassis angular velocity were 4.0 s, 3.0 s, 4.0 s and 2.0 s, respectively. Further, the maximum overshoot responses obtained for wheel position were 0.005 m to 0.11 m, for wheel velocity were −0.05 m/s to 0.075 m/s, for chassis angle were -4.4×10^{-3} rad to 10.4×10^{-3} rad and for chassis angular velocity were −0.002 rad/s to 0.15 rad/s, respectively. The results also indicate that the ANN controller stabilizes SWRS with almost zero steady state error. Further, a working model of SWRS can be developed in future, and efforts can also be made to reduce the settling time of ANN controller by varying the count of neurons in the hidden layer.

References

Abadi A, Amraoui AE, Mekki H, Ramdani N (2020) Flatness-based active disturbance rejection control for a wheeled mobile robot subject to slips and external environmental disturbances. IFAC-PapersOnLine 53(2):9571–9576

Borja JA, Alvarado I, Pena DM (2020) Low cost two-wheels self-balancing robot for control education powered by stepper motors. IFAC-PapersOnLine 53(2):17518–17523

Deng M, Inoue A, Sekiguchi K, Jiang L (2010) Two-wheeled mobile robot motion control in dynamic environments. Robot Comput-Integr Manuf 26(3):268–272

Dhiman G, Kumar AV, Nirmalan R, Sujitha S, Srihari K, Yuvaraj N, Arulprakash P, Raja AR (2023) Multi-modal active learning with deep reinforcement learning for target feature extraction in multi-media image processing applications. Multimedia Tools Appl 82(4):5343–5367

Farias G, Garcia G, Montenegro G, Fabregas E, Dormido-Canto S, Dormido S (2020) Position control of a mobile robot using reinforcement learning. IFAC-PapersOnLine 53(2):17393–17398

Gandarilla I, Montoya-Chairez J, Santibanez V, Aguilar-Avelar C, Moreno-Valenzuela J (2022) Trajectory tracking control of a self-balancing robot via adaptive neural networks. Eng Sci Technol Int J 35:1–11

Huang CN (2010) The development of self-balancing controller for one-wheeled vehicles. Engineering 2:212–219

Kansal V, Jain U, Pant B, Kotiyal A (2023) Comparative analysis of convolutional neural network in object detection. Lecture Notes Netw Syst 520:87–95

Khalaji AK, Jalalnezhad M (2021) Robust forward\backward control of wheeled mobile robots. ISA Trans 115:32–45

Kim Y, Singh T (2022) Energy-time optimal control of wheeled mobile robots. J Franklin Inst 359(11):5354–5384

Liu S, Liu K, Zhong Z, Yi J, Aliev H (2022) A novel wheeled mobile robots control based on robust hybrid controller: mixed H2/H∞ and predictive algorithm approach. J King Saud Univ-Comput Inf Sci 34(10):9662–9676

Rubio F, Valero F, Llopis-Albert C (2019) A review of mobile robots: concepts, methods, theoretical framework, and applications. Int J Adv Rob Syst 16(2):1–22

Su KH, Chen YY, Su SF (2010) Design of neural-fuzzy-based controller for two autonomously driven wheeled robot. Neurocomputing 73(13):2478–2488

Unluturk A, Aydogdu O (2017) Adaptive control of two-wheeled mobile balance robot capable to adapt different surfaces using a novel artificial neural network–based real-time switching dynamic controller. Int J Adv Rob Syst 14(2):1–9

An Approach to the Classification of Ancient Indian Scripts Using the CNN Model

Muskan Agarwal, S. Indu, and N. Jayanthi

Abstract This research aims to investigate the performance of the convolutional neural network (CNN) for the classification of four rare ancient Indian scripts, namely, Devanagari, Nandinagari Newari, and Sharda. Script classification is an incredibly difficult task, especially in India where there are 14 Indian scripts used to write 22 official languages. Our work limits itself to the study of these four Indian ancient scripts to have a scope for developing an efficient classification model for script identification. In the context of our awareness of this work, earlier researchers have made significantly less effort to identify ancient Indian scripts: Devanagari, Nandinagari, Newari, and Sharda. Researchers have never implemented any deep learning techniques for the recognition of such ancient scripts. Hence our proposed approach is the first-time implementation of a CNN-based model for the multiclass classification of these ancient Indian scripts. For experimentation, we use the self-created dataset of IGNCA manuscripts, which consists of four rare Indian ancient scripts: Devanagari, Nandinagari, Newari, and Sharda. In this study, we accept the challenges of data collection, creation of our dataset, exploration of their distinguishable features, preprocessing, and classification of these Indian ancient scripts. The proposed convolutional neural network (CNN) model achieves classification accuracy of 99.03% and 98.71% on the training set and the testing set, respectively.

Keywords Ancient Indian scripts · Feature extraction · Classification · CNN · Script identification

M. Agarwal (✉) · S. Indu · N. Jayanthi
Delhi Technological University, New Delhi 110042, India
e-mail: muskanagarwal_2k22phdec01@dtu.ac.in

S. Indu
e-mail: s.indu@dtu.ac.in

N. Jayanthi
e-mail: njayanthi@dce.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_36

1 Introduction

India is a huge collection of historic Indian scripts. Ancient Indian scripts are conserved in temples, digital libraries, etc., but researchers do not have easy access to them. These ancient Indian scripts convey innate information about cultural, medical, religious, and historical values from the former generation to the later generation. These historical records are a rich foundation of exceptional evidence. Nevertheless, they endure ongoing deterioration throughout time as a result of temperature, moisture, insects, natural disasters, improper handling, and storage. These historical records should be preserved for future generations because of their importance. Several museums and digital libraries are attempting to digitize ancient Indian documents as a result of the technological advancement toward digitalization to preserve them for subsequent centuries and make them readily accessible sources of trustworthy knowledge. Digitization of these ancient Indian scripts is a challenging task because of several levels of degradation including noise, stains, fading, ink bleed, intensity variations, soft ink strokes, etc. (Shah and Badgujar 2013).

Motivated by this, we are implementing a method to classify these documents transcribed in ancient scripts. The current research work recognizes four ancient Indic scripts: Devanagari, Nandinagari, Newari, and Sharda, since the majority of historical document images are generally handwritten that forces numerous impressive challenges on Indian script recognition. First of all, every script is unique from other scripts in terms of its nature, structure, features, and writing style. Second, compared with printed fonts, handwritten scripts exhibit a wider range of writing styles from person to person. Third, handwriting styles vary slightly throughout time in handwritten script. Finally, noise and skewness present challenges in preprocessing of historical documents images.

We arranged this paper into succeeding subsections: Sect. 2 includes the related work. We discuss the distinguishable features of ancient Indian scripts in Sect. 3. Section 4 describes the script classification process. Section 5 explores the test results of our experiments. Section 6 completes our study with a conclusion.

2 Related Work

Hangarge and Dhandra (2010) have identified three Indian scripts, namely English, Devanagari, and Urdu using KNN classifier. He obtained an accuracy of 93% for the Devanagari script. Sarkar et al. (2010) have recognized two Indian scripts, namely Devanagari and Roman using MLP classifiers. He achieved a classification accuracy of 98.53%. Pati and Ramakrishnan (2008) have used SVM classifiers for recognizing 11 Indian scripts. He attained a classification accuracy of 89%. Table 1 gives additional related work of traditional feature extraction-based techniques of script classification and CNN-based script classification techniques. CNN-based deep learning

Table 1 Traditional and CNN-based classification techniques

SN	Authors	Script	Classifier	Accuracy
1	Rajput and Anita (2010)	Hindi, English, Telugu, Tamil, Malayalam, Kannada, Gujarati, and Punjabi	KNN	99.2% (Tamil and Malayalam)
2	Moussa et al. (2008)	Arabic and Latin	KNN and radial basic function (RBF)	98.72% (RBF)
3	Rashid et al. (2010)	Greek-Latin	CNN	98.40%
4	Rashid et al. (2010)	Greek-Latin, Arabic-Latin	CNN	98.40%
5	Naz et al. (2017)	Urdu	CNN + MD LSTM	98.12%
6	Vijayaraghavan et al. (2018)	Tamil	CNN	94.40%

model is only developed for the classification of non-Indian scripts and only a few Indic scripts as summarized in Table 1.

Conventional feature extraction techniques extract the appropriate features or characteristics from the document images which are given to the image recognition classifiers for classification. In contrast to this manual feature extraction, we are realizing the CNN model because of its natural capability to learn crucial features from image datasets, making it more suitable for document image classification. In the convolutional layer of CNN, many filters are used to perform convolution operations on document picture data to extract significant characteristics that are successively transmitted to the following layer. This automatic process of feature extraction keeps going on up to the last CNN layer. CNN model considerably enhanced the classification accuracy of ancient document images.

3 Problem Statement

The literature contains a lot of existing works related to language classification. The scripts of various languages differ greatly from one another. But, the scripts we're looking for (in Devanagari, Nandinagari, Newari, and Sharda) are written in the same language, and the majority of the characters are the same. Only a few characters differ from one another. Therefore it is a difficult task to categorize all scripts of the Sanskrit language. Here, we developed an efficient architecture once we create small image patches for recognizing the slight differences of all individual characters of the ancient scripts by appropriate image preprocessing techniques. These small image patches are further categorized using the CNN model of script classification.

4 Distinguishable Features of Ancient Indian Scripts

Although Indian ancient scripts are written in the same direction as English, from left to right, they differ from non-Indian scripts in numerous ways. The concept of upper-case and lower-case characters are absent in Indian scripts. These features make it challenging to identify ancient Indian scripts.

The identification of ancient Indian handwritten characters has not advanced to the same level of perfection as that of non-Indic scripts. We made an effort to recognize handwritten ancient Indian scripts like Devanagari, Tamil, Gurumukhi, and Telugu. A comprehensive identification system of ancient Indian script is not existing for commercialization (Patwardhan and Deshmukh 2015). We have obtained four ancient Indian scripts, Devanagari, Nandinagari, Newari, and Sharda, from the manuscript conservation center at Indira Gandhi National Centre for the Arts (IGNCA), New Delhi. The following is a description of some of the distinctive characteristics of these four rare ancient Indian scripts, as illustrated in Fig. 1.

4.1 Devanagari Script

Devanagari is a very popular ancient Indian script. Several languages including Bhojpuri, Nepali, Awadhi, Maithili, etc., are written in Devanagari script. The Devanagari script consists of 14 vowels and 33 consonants (Sonawane and Shelke 2018). Devanagari script contains a large number of character patterns or structures in form of compound characters and vowels with matras or modifiers. Even though some of them are nearly identical, there are small differences between them due to horizontal lines, loops, and dots. In contrast to English, which lacks strokes, nearly the whole Devanagari character set has important characteristics of curves, line segments (strokes), and holes. The Devanagari script is devoid of upper- and lower-case letters, unlike English. A long headline appears at the top of every Devanagari character set. Two or more characters form a specific word while joining the same headline. Because of the connected headline, character segmentation of the Devanagari script becomes more challenging (Mhaiskar 2014).

Scripts	Some examples of alphabets
Devanagari	क ka ख kha ग ga घ gha ङ ṅa च ca छ cha ज ja झ jha ञ ña
Sharda	𑆑 𑆒 𑆓 𑆔 𑆕 𑆖 𑆗 𑆘 𑆙 𑆚
Nandinagiri	𑦑 𑦒 𑦓 𑦔 𑦕 𑦖 𑦗 𑦘 𑦙 𑦚
Newari	𑐎 𑐏 𑐐 𑐑 𑐒 𑐓 𑐔 𑐕 𑐖 jha 𑐗 ña

Fig. 1 Style variation of a few characters of Devanagari, Nandinagari, Newar, and Sharda script

4.2 Sharda Script

Sanskrit, Kashmiri, and other languages are written in the Sharda script in northern South Asia. Sharda letters, however, have a very different appearance overall from Devanagari. The Sharda script comprises 14 vowels and 34 consonants. The Sharda and Devanagari share a similar structure. In comparison with the Devanagari script, all of the characters in the Sharda script have stiff, thick strokes. In contrast to the Devanagari script, the Sharda script does not include a headline at the top of the characters. Virama and numbers are the most distinctive characteristics of the Sharda script, despite the fact that they share structural similarities with the Devanagari alphabet. Sharda virama immediately follows the rightmost consonant letter, joining to the headline in place of directly below it as in Devanagari script. Sharda numerals are written using the Devanagari script's decimal system, but the notation system is incredibly distinctive because zero is denoted by a dot and one is represented by a circle.

4.3 Nandinagari Script

The Sanskrit language is written using the Nandinagari script. In Karnataka, Kannada is also written in Nandinagari. The Nandinagari script comprises 12 vowels and 35 consonants. Devanagari script and Nandinagari script are analogous in terms of their glyphic representation, character repository, and structural layout. Characters written in Nandinagari, however, seem different from those written in Devanagari. Because headline linkages only apply to a single character and not to nearby characters, adjacent characters are separated. The shapes of the vowels and some of the consonants in the Nandinagari script differ from those in the Devanagari script.

4.4 Newar Script

Newar is written using the Newar script. People in the states of Bihar and Sikkim spoke Newar. Newar was adopted as the state's official language by the Sikkim government. The languages like Nepali, Hindi, Sanskrit, Bengali, and Maithili are written in Newar script. The structure of the Newar script is analogous to that of Devanagari. The Newar script contains 14 vowels and 33 consonants.

5 Script Classification Process

Script classification is regarded as one of the preliminary phases in language processing. There are three steps in the script classification process—image collection, image preprocessing, and image classification. Figure 2 depicts the fundamental flow diagram for the script classification process.

5.1 Image Collection

At present, we have digitized small numbers of standard datasets for a few ancient Indian scripts in comparison with non-Indian scripts in the past couple of decades. Lack of benchmarked datasets for all ancient Indian scripts is a significant problem. There is a vital requirement to create huge datasets for ancient Indian scripts identical to other non-Indian scripts. In this research, we have made an attempt to develop datasets for a few of the ancient Indian scripts. Our research work begins with the collection of four ancient Indian scripts, namely Devanagari, Nandinagari, Newari, and Sharda. We obtained the scanned document images of these four Indian ancient scripts from the reference library of the manuscript conservation center Indira Gandhi National Centre for the Arts (IGNCA), New Delhi, India. We obtained a total of 429 scanned document images, out of which 27 were Devanagari, 86 were Nandinagari, 86 were Newari, and 230 were Sharda. Figure 3 illustrates the sample image of each Indian ancient script from our datasets.

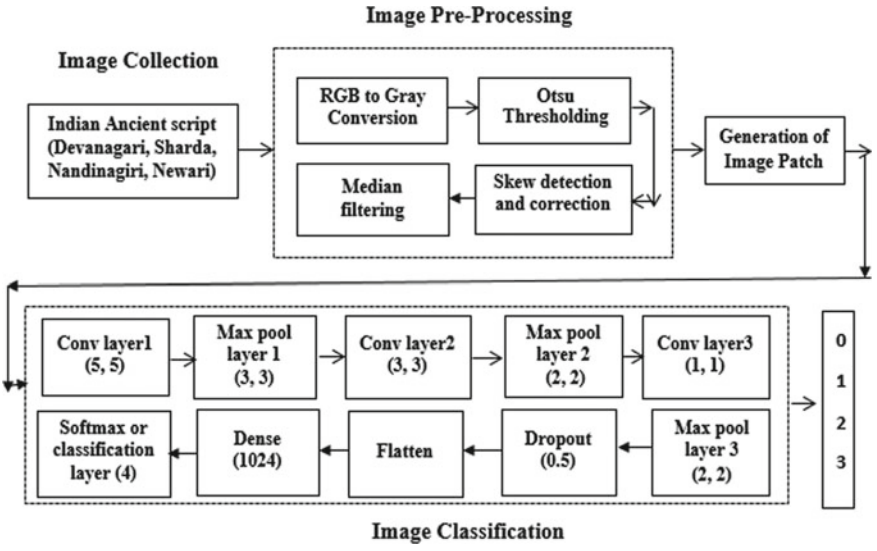


Fig. 2 Architecture of script classification process

Fig. 3 Images dataset of **a** Devanagari, **b** Sharda, **c** Nandinagari, **d** Newari

5.2 Image Preprocessing

Image preprocessing includes numerous modifications in the document scanned images, like aspect ratio, grayscaling, binarization, filtration, and skew correction. We initially converted the document image to a gray value image. Grayscaling is used to accomplish a number of benefits. Since 2D array representation is easier when compared with 3D operations, advanced image processing techniques are simpler in gray value images. Grayscale images require less memory than color images. Later, the grayscale image is converted into a binary image using Otsu's global thresholding technique. We preferred the Otsu technique because of the uniform background of dataset images. As a result, we determine that the threshold value for Devanagari, Sharda, Nandinagari, and Newari document images is 130, 160, 145, and 140, respectively. Scanned document images have a partial tilt of text from an x-coordinate axis because of improper alignment of the document image on the scanner, which we know as skew.

Then, we rotated the skewed image so that the text was aligned parallel to the x-coordinate axis, a method known as a skew correction. Because of the salt and pepper noise in the document image, we choose median filtering of kernel size (5*5). Filtered images are divided into image patches to increase the amount of training data for

the script classification model. Our data preprocessing algorithm generates 2711 image patches from 429 document images. We created image patches from scanned document images to accomplish two benefits: concentrate on specific characters and produce additional training datasets for the deep learning model.

5.3 Image Classification

We input image patches directly to our proposed CNN-based script classification model. We successfully classified the four Indian scripts based on their structural characteristics. CNN model can be considered as both a feature extractor and an image classifier. CNN classified all four categories of Indian scripts by automatically extracting pertinent features. The architecture of the CNN model is shown in Fig. 2. We supply image patches to the convolutional layer, which carries out the convolutional operation and produces feature maps. After that, the max pooling layer applies the max pooling operation on feature maps to minimize their spatial dimension. Convolutional and maximum pooling layers are together referred to as feature extractors. The low-level CNN layers extract features or structures like lines, edges, corners, etc., while high-level CNN layers extract features such as text, shapes, object contours, etc.

To avoid overfitting, these feature maps are also sent to the dropout layer. The CNN model then flattened the resulting feature maps to vectors and fed them into a fully connected layer (Manocha and Tewari 2021). The softmax activation function, which specifies the probability of the appropriate class at the softmax layer, maps all connections in the fully connected layer to the softmax layer.

The CNN model learning takes place because of the loss function and optimizer. The loss function tells the CNN model to reduce the errors between predicted and actual classes. The optimizer backpropagates this error through convolutional layers and instructs the CNN model of updating weight adaptively to minimize that error. We achieve this using Adam optimizers and the cross-entropy loss function, which improves the CNN model's accuracy.

6 Result and Discussion

We have implemented the script classification using the CNN model. We evaluated our developed method to script classification on four different classes. These include Devanagari, Nandinagari, Newari, and Sharda. We developed the CNN deep learning model using Tensor flow and Keras library. To train the CNN model, we divided the image dataset into two sections: training (80%) and testing (20%). We found classification accuracy of 99.03% and 98.71% on the training and the testing dataset, respectively. Performance metrics are used to evaluate the performance of our developed algorithm. Table 2 depicts performance metrics for each class of the

CNN model. Tables 3 gives the overall performance metrics for the proposed CNN model.

We plotted training and validation accuracy and loss curves as illustrated in Fig. 4. As given in Table 2, the classification accuracy achieved on our own datasets of ancient Indian scripts is higher than that of the current model. The training time of various classifiers is a serious issue that needs to be addressed. Consequently, it takes less time to train the CNN model on GPU than CPU, taking roughly 2.85 min instead. The results demonstrate that the CNN deep learning model outperformed all other traditional feature extraction techniques.

Table 2 Performance metrics for CNN model (%)

Model	Performance metrics	Devanagari	Nandinagari	Newari	Sharda
CNN	Precision	98	99	97	100
	Recall	100	97	99	99
	F1 score	99	98	98	100

Table 3 Overall performance metrics for CNN model (%)

Performance metrics	Proposed model (CNN)
Training accuracy	99.03
Testing accuracy	98.71
Macro precision	98.59
Macro recall	98.92
F1 score	98.75

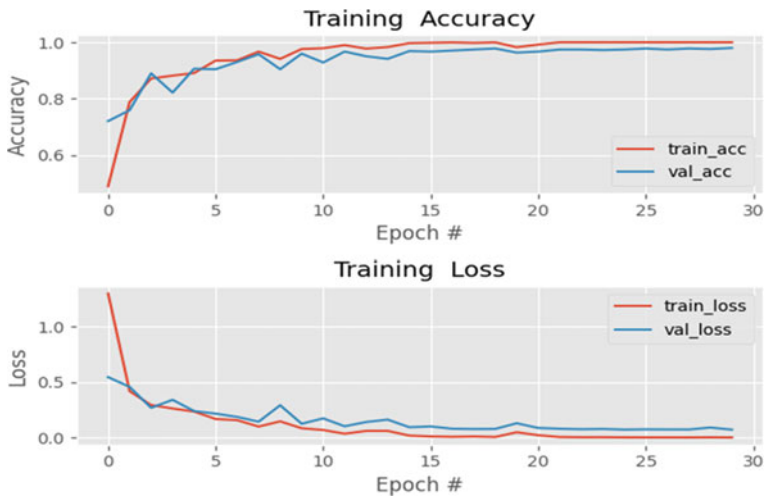


Fig. 4 Training and validation accuracy and loss curve for CNN model

7 Conclusion

This paper is mainly dedicated to the development of the script classification model. A brief overview of Indian ancient script is presented. We also cover a study of four ancient Indian scripts and their unique characteristics. Recognition of document images in ancient Indian scripts, including Devanagari, Nandinagari, Newari, and Sharda was one of the crucial challenges because of complicated patterns and the non-availability of standard datasets. We described a deep learning method for recognizing the ancient Indian script included in a document image. An image dataset of 426 documents from four ancient Indian scripts was used to implement this approach. We categorized the documents by the distinct writing styles, often termed structural characteristics. We developed the convolutional neural network (CNN) for classifying handwritten document images. Overall classification accuracy for the four ancient Indian scripts was 99.03% and 98.71% on the training and testing dataset, respectively.

References

Hangarge M, Dhandra BV (2010) Offline handwritten script identification in document images. Int J Comput Appl 4(6):6–10

Manocha SK, Tewari P (2021) Comparative study of deep learning models for Devanagari OCR. In: 2021 international conference on smart generation computing, communication and networking (SMART GENCON). IEEE

Mhaiskar R (2014) Devanagari: a historical overview and a study of writing style in Aga Khan palace photo gallery. Bull Deccan College Res Inst 74:197–208

Moussa SB et al (2008) Fractal-based system for Arabic/Latin, printed/handwritten script identification. In: 2008 19th international conference on pattern recognition. IEEE

Naz S, Umara AI, Ahmadc R, Siddiqid I, Saad S, Ahmede B (2017) Urdu Nastaliq recognition using convolutional-recursive deep learning. Neurocomputing 243:80–87

Pati PB, Ramakrishnan AG (2008) Word level multi-script identification. Pattern Recognit Lett 29(9):1218–1229

Patwardhan SS, Deshmukh RR (2015) A review on offline handwritten recognition of Devnagari script. Int J Comput Appl 117(20)

Rajput GG, Anita HB (2010) Handwritten script recognition using DCT and wavelet features at block level. IJCA, Spec Issue RTIPPR (3):158–163

Rashid S, Shafait F, Breuel T (2010) Connected component level multiscript identification from ancient document images. In: Proceedings of the 9th IAPR workshop on document analysis system, pp 1–4

Rashid SF, Shafait F, Breuel TM (26–29 Sept 2010) Discriminative learning for script recognition. In: IEEE 17th international conference on image processing, pp 2145–2148

Sarkar R et al. (2010) Word level script identification from Bangla and Devanagari handwritten texts mixed with roman script. arXiv preprint arXiv:1002.4007

Shah KR, Badgujar DD (2013) Devnagari handwritten character recognition (DHCR) for ancient documents: a review. In: 2013 IEEE conference on information and communication technologies. IEEE

Sonawane PK, Shelke S (2018) Handwritten Devanagari character classification using deep learning. In: 2018 international conference on information, communication, engineering and technology (ICICET). IEEE

Vijayaraghavan P, Sra M (2018) Handwritten Tamil recognition using a convolutional neural network. In: 2018 international conference on information, communication, engineering and technology (ICICET), pp 1–4

Addressing Vehicle Safety and Platooning Using Low-Cost Object Detection Algorithms

Prathmesh Sharma, Priti Gangwar, Ritik Gupta, and Poornima Mittal

Abstract Object detection is a computer vision-based technology that is very pertinent to the study of automation. You only look once (YOLO) is an object detection algorithm used to detect objects belonging to certain classes in images and videos. YOLO provides an increased speed and accuracy of object detection over conventional solutions. This research paper compares two versions of YOLO: YOLOv5 and YOLOv3, in terms of precision and recall, by training the models from the ground up from a dataset containing vehicle images. These results have been integrated with a single-frame lane detection pipeline and were evaluated over two different architectures: an i5 11th gen processor and Nvidia GTX 1050 graphics card. The best possible solution based was able to achieve an average speed of 12 frames per second for the entire pipeline.

Keywords YOLO: you only look once · Precision · Recall · Mean average precision · Object detection · Vehicle detection · Lane detection

1 Introduction

Object detection characteristics, like accuracy and speed are required to make self-driving cars. For object detection, object localization is required. For this, a bounding box needs to be created. Object localization is finding what and where an object exists in an image. For object localization, a bounding box is defined and a CNN is used (Felzenszwalb et al. 2010). R-CNNs resize all potential bounding boxes to a fixed size of 224 by 224 to run through convolutional bounding boxes, however this is a slow, two-step process. You only look once (YOLO) detects, localizes, and classifies objects in one go. The YOLO algorithm is based on regression. YOLO has trained

P. Sharma · P. Gangwar (✉) · R. Gupta · P. Mittal
Delhi Technological University, New Delhi, India
e-mail: pritigangwar77@gmail.com

P. Mittal
e-mail: poornimamittal@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_37

on PASCAL Visual Object Class (VOC) dataset (Everingham et al. 2015). Multiple bounding boxes and probabilities of the class (Mittal et al. 2020a) are identified by a single neural network. Hence, it is faster than previous algorithms (Redmon et al. 2016).

YOLO has double the mean average precision (mAP) than previous algorithms. Background errors in YOLO are less than in fast R-CNN (Bharti and Mittal 2021). The idea behind YOLO is to divide the image into S $\times$ S grid-cell, where each grid-cell will output a prediction with a corresponding bounding box. A grid-cell can only detect one object. If more than one object exists in a grid-cell, the grid-cell should be finer (Girshick 2015). In this paper, a literature survey of YOLOv5 and YOLOv3 was done. Single-frame lane detection (Mittal and Jha 2021) was performed using purely computer vision-based techniques. YOLOv5 and YOLOv3 models of differing complexity were trained on a vehicle images dataset. The vehicle detection models were condensed with lane detection into a pipeline, and then evaluated by applying different weights to metrics to get the best-fit model for the task (Mittal et al. 2020b).

2 Comparison Between YOLOv3 and YOLOv5

YOLO is unable to detect small objects in a cluster. YOLOv3 is an update over the original YOLO algorithm and is faster than many other algorithms. The major difference is in their architectures. The original YOLO generated an output of a single grid of 7 $\times$ 7. It was observed that this feature was good for the large bounding boxes, but it created problems for the smaller ones. The original YOLO generated an output of a single bounding box per cell, but in YOLOv3, the output is three per cell. Considering the three scales of prediction, it leads to a massive amount of bounding box predictions. Hence anchor boxes are used. Multilevel classification is done on each bounding box to predict the object class. YOLOv3 performs better than most algorithms and is comparable to Retina-Net (Girshick et al. 2014).

YOLOv5 includes a lot of data augmentation. YOLOv5 achieves high accuracy, precision values of up to 50 mAP with datasets (Choudhary et al. 2021) along with the high operation speed. Input is an image link, made up of three components: image enhancement, image size adjustment, and automated adapting anchor frames (Jaint et al. 2019). YOLOv5 enhances data through scaling, cropping, arranging, and stitching the input (Ahlawat et al. 2021) to work on smaller objects. During the dataset training, a uniform image size is achieved for all the input images before the image is passed to the model for detection (Chen et al. 2022; Rawat and Mittal 2021; Redmon and Farhadi 1804; Zhang et al. 2022).

3 Lane Detection Using Computer Vision

The first step within the pipeline was creating a computer vision-based lane detection model (Chougule et al. 2018). The preliminary step was to Gaussian blur the image as the input is a high-definition image, and the Canny edge detection algorithm may detect false positive edges if the image is not blurred. This was followed by extracting the edges of the blurred image (Bottazzi et al. 2014). This was done using the multistage Canny edge detection. The initial step involves noise reduction using a Gaussian filter of 5×5 dimension. Further steps measure the intensity gradient of the filtered image. A Sobel filter was used, along the x and y axes (denoted by Sx and Sy). Then, the direction and edge gradient of every pixel are given by:

$$\text{Angle} = \tan^{-1}(\text{Sy}/\text{Sx}) \tag{1}$$

$$\text{Edge Gradient} = \sqrt{\left(\text{Sx}^2 + \text{Sy}^2\right)} \tag{2}$$

Next step was to perform non-maximal suppression, a full scan of the image was done to check and remove pixels from the edge. For this, every pixel was checked for being a local maximum in its neighborhood; hysteresis thresholding was performed, selecting a min and a max value and any edge with intensity greater than max was selected as a 'sure edge.' Any edge with intensity below the minimum was discarded.

Those that lied in between were classified based on their connectivity to sure/discarded edges. Next, unimportant area was removed from the image. These values depend on the height of the camera mounted on a vehicle, and an ideal value was reached by observing the dataset. The chosen values were stored in the program, and the others were discarded. Next, the Hough transform was applied. Frequently, the problem faced in images is that the road lanes extracted have broken lines. This is solved by the Hough transform, which extends the extracted feature to an appropriate set of lines. Now, within this graph, there are either completely dark pixels or completely white ones. Hence, intensity graphs plots in the x and y axes were done to find endpoints of the trapezium that need to be found to see whether a car is coming into the lane or not. This was done using NumPy arrays.

Figure 1 shows that in the y-axis, the line begins between the pixel range 400 to 500 and ends between the pixel range 500 and 600. Upon further computation across both axes the ends of the rectangle were computed to be: (473, 653, 298, 843). One image takes 0.015s to be processed in this manner, making processing speed close to 66 frames per second.

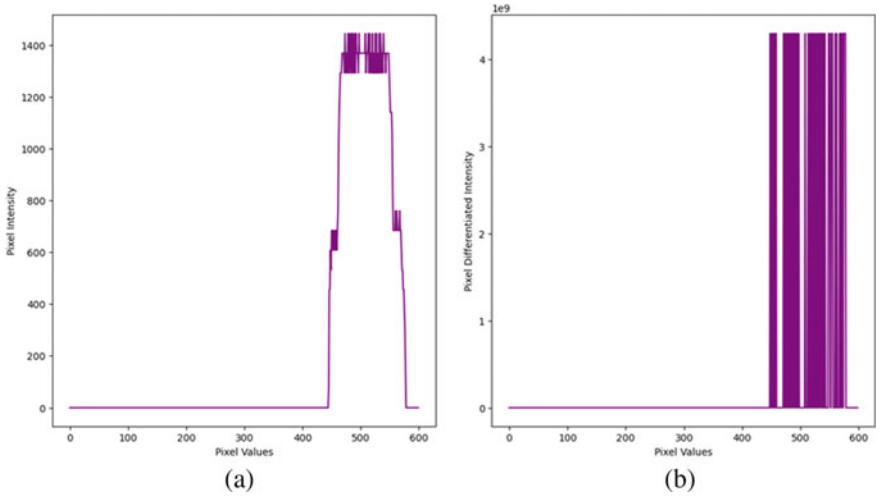


Fig. 1 **a** Pixel intensity graph **b** Differentiated intensity graph

4 Vehicle Detection Dataset Description

The size of the dataset used was 29.1 GB, and it contained ≡70,500 images for training and ≡18,500 images for validation. Dataset specifications were given in a yaml-type file. This contains the image location in the memory and the number of classes. The classes map on to an observation. Every line gives the class array and pixel location for the bounding box. These values were normalized, so that all of them lie in between 0 and 1, as when the images are input into the YOLO network, they all get resized to a 416 × 416 size.

5 Result and Analysis

In this paper, different models of YOLOv5 and YOLOv3 were trained and three evaluations metrics were chosen. The first was model speed, for real-time object detection to avoid damage due to delayed results. This was achieved by noting the time taken per frame using the 'timeit' function in Python. The second metric is mean average precision. Mean average precision is the average of precisions over all classes. These values are calculated over a particular I.O.U value (0.5 in this case). Recall is the third metric, this is one of the most important metrics as YOLO chooses best possible recall (BPR) as the metric to be optimized. In Fig. 2, the mAP value increases for a few epochs, after which it stagnates, and may decrease and hence training is stopped after that epoch. The maximum value of I.O.U is reached soonest by YOLOv5 XL. YOLOv5 Medium has a significantly larger mAP than Small and

Nano. After a few epochs, YOLOv5 Small overtakes Nano in terms of mAP. Figure 3 represents recall vs epochs. Here the recall converges for YOLOv5 Small, YOLOv5 Tiny, and Yolov5 Medium. YOLOv5 XL does significantly better, in significantly lesser epochs.

Model speed is architecture variant, and must be calculated for different architectures as given in Table 1. In the computational pipeline, two models were run in series, one for road detection (on YOLOv5 Nano and Small only) and the other for vehicle detection (encompassing all five YOLOv5 models and two YOLOv3 models). Bracket values are when CPU is pushed into overdrive and can only be sustained temporarily.

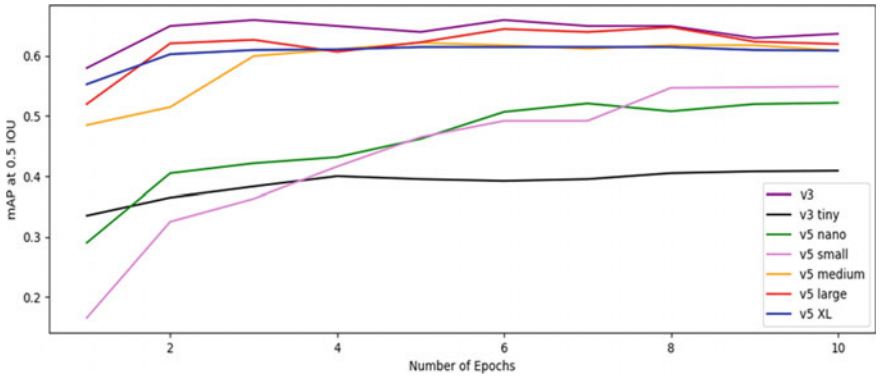


Fig. 2 Graphical comparison of mAP for each model per epoch at IOU of 0.5

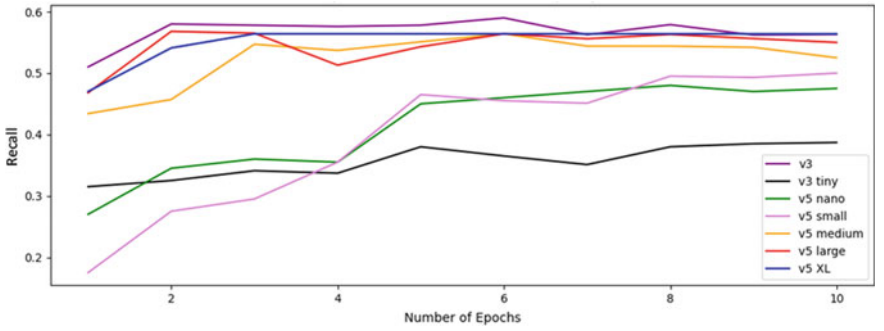


Fig. 3 Graphical comparison of recall for each model per epoch

Table 1 Pipeline evaluation on i5 11th gen processor and GTX 1050 graphics card

	i5 1135G7, 8 GB RAM, 2.4 GHz, 4 Cores, 8 Threads	Nvidia GTX 1050 graphic card, 4 GB RAM, 640 CUDA
V3 small	4.2	20 (22)
V3	1.2	19 (19)
V5 nano	4.3	16 (20)
V5 small	2.9	14 (15)
V5 medium	2.0	12 (12)
V5 large	1.4	9 (9)
V5 XL	0.6	6 (6)

6 Conclusion

In this paper, a pipeline for vehicle safety was created. This pipeline takes images as inputs in a sequential manner and processes them in order to detect road lanes and vehicles. The pipeline involves road lane detection, purely based on computer vision, and vehicle detection based on YOLOv5 and YOLOv3. The models were evaluated based on precision, recall, and model speed. The performance of the entire pipeline was evaluated on two architectures: i5 11th-gen CPU and Nvidia GTX 1050 graphic card. Based on the analysis, YOLOv5 Medium for vehicle detection was concluded to be the best fit for the task, as it had a mAP value of approximately 0.56, along with a speed of 12 frames per second on a GTX 1050 architecture.

References

Ahlawat S, Siddharth Rawat B, Mittal P (16–18 Dec 2021) A comparative performance analysis of varied 10T SRAM cell topologies at 32 nm technology node. In: International conference on modeling, simulation and optimization 2021, organized by NIT Silchar

Bharti R, Mittal P (20–21 Aug 2021) Frequency analysis of ring oscillator at different technology nodes. In: International conference on simulation, automation & smart manufacturing (SASM 2021), organized by GLA University, Mathura

Bottazzi VS, Borges PVK, Stantic B, Jo J (2014) Adaptive regions of interest based on HSV histograms for lane marks detection. In: Kim JH, Matson E, Myung H, Xu P, Karray F (eds) Robot intelligence technology and applications 2. Advances in intelligent systems and computing, vol 274. Springer, Cham

Chen Z, Cao L, Wang Q (2022) YOLOv5-based vehicle detection method for high-resolution UAV images. Mobile Inf Syst 2022:11, Article ID 1828848

Choudhary P, Jain A, Agrawal A, Mittal P (07–09 May 2021) Comparative analysis of two hardware based square root computational algorithms. International conference on paradigms of communication, computing and data sciences (PCCDS 2021) organized by national institute of technology, Kurukshetra, Haryana

Chougule S, Koznek N, Ismail A, Adam G, Narayan V, Schulze M (8–14 Sept 2018) Reliable multilane detection and classification by utilizing CNN as a regression network. Munich, Germany

Everingham M, Eslami SMA, Gool LV, Williams CKI, Winn J, Zisserman A (2015) The pascal visual object classes challenge: a retrospective. Int J Comput vis 111(1):98–136

Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645

Girshick RB (2015) Fast R-CNN. CoRR, abs/1504.08083

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE conference on computer vision and pattern recognition (CVPR). IEEE pp 580–587

Jaint B, Singh V, Singh SK, Mittal P, Indu S, Pandey N (28–30 March 2019) A novel approach for detection of malicious nodes in WSN using linear AR prediction and clustered weighted trust evaluation. In: IEEE international conference on signal processing, VLSI and communication (ICSPVC-2019). Delhi Technological University, New Delhi, India

Mittal P, Jha S (20–21 Aug 2021) Traffic congestion control system based on microcontroller. In: International conference on simulation, automation & smart manufacturing (SASM 2021), organized by GLA University, Mathura

Mittal P, Gupta P, Shri S, Goel T (2020a) Development of neural network approach towards MPPT optimization. Int J Adv Sci Technol 29(7):1436–1444. (ISSN: 2005-4238)

Mittal P, Gupta P, Shri S, Goel T (Apr 2020b) Development of neural network approach towards MPPT optimization. Int J Adv Sci Technol 29(7):1436–1444 (ISSN: 2005-4238)

Rawat B, Mittal P (July 2021) A 32 nm single ended single port 7T SRAM for low power utilization. Semicond Sci Technol 36(9):095006–095022. (ISSN: 1361-6641), (IOP Sciences, I. F. 2.048)

Redmon J, Farhadi A. YOLOv3: an incremental improvement, in arXiv:1804.02767

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, NV, USA pp 779–788. https://doi.org/10.1109/CVPR.2016.91

Zhang Y, Guo Z, Wu J, Tian Y, Tang H, Guo X (2022) Real-time vehicle detection based on improved YOLO v5. Sustainability 14(19):12274

Facial Appearance Discerning Using Convolutional Neural Networks

Kushal Kashyap and Vijay Kumar

Abstract Humans use their facial expressions to convey their emotions, which make them a potent tool in communication. Detecting facial expressions is one of the difficult and effective social communication jobs since non-verbal communication depends heavily on facial expressions. Facial expression recognition (FER), a topic of ongoing research in the robotics area, has recently been the subject of multiple experiments. In the research uses, CNNs show how to classify FER utilizing static photos, without the need for any feature extraction or preprocessing work. The article gives examples of preprocessing methods that can be used to increase future accuracy in this field. Apparent features of the face are extracted via feature extraction. We also talk about the methodology, the CNN design, difficulties with max-pooling, and how dropout helped us get higher performance. On the FE dataset, we were able to complete a multiclass classification with a training accuracy of 82.3%.

Keywords Facial expression (FE) · IoT in healthcare · Convolutional neural networks (CNN)

1 Introduction

Recognition of facial emotions is a crucial component of human emotion detection, which is widely employed in disciplines such as machine vision, pattern recognition, picture understanding, and human–computer interaction (Xia et al. 2017). Seven fundamental emotional face expressions are generally acknowledged by researchers. They are: happy, surprise, disgust, fear, neutral, anger, and sadness. These phrases

K. Kashyap (✉) · V. Kumar
Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India
e-mail: kushal.kashyap@chitkara.edu.in

V. Kumar
e-mail: vijay.jodan@chitkara.edu.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_38

are used consistently across all racial and cultural groups. When detecting emotional expressions, there are two common methods.

The initial strategy consists of two steps: (i) recognizing small variations in what are known as action units (AUs) on the face; (ii) matching combinations of these AUs that correspond to seven standard facial emotional expressions. The second method classifies the entire facial expression into one of the seven prototypical emotions (Agarwal et al. 2012; Lilhore et al. 2021). Automated FE is still seen as a difficult issue, particularly when it comes to practical applications, despite the fact that approaches based on convolutional neural networks (CNNs) have shown promised accuracy in a wide range of applications (Fard et al. 2021).

Emotions on the face: Traditional techniques identify the features of face organs like the eyes and mouth by their geometric relationships or by features extracted from variations in their shapes. When we compare two images of a human face, we can determine which image clearly depicts the facial emotions. Therefore, we believe it is crucial to create a measurement method for the intensity of facial expressions as a step in extending the facial expression recognition process. Handling the variety in appearance caused by variations in stance, expression, and lighting conditions is one of the fundamental issues still present in face recognition (Donato et al. 1999; Lien et al. 2000).

In the field of artificial intelligence, facial expression recognition (FE2020) is a hot topic for research. It has applications in a wide range of fields, including e-learning, medical, marketing, law enforcement, security, and monitoring. A number of fields, including data analytics and automatic facial expression recognition can be useful for human–computer interactions, social gaming, and psychological research. Some facial expressions have been shown to have a common meaning. Happy, surprise, disgust, fear, neutral, anger, and sadness are seven facial expressions that appear to be universal across all cultures, according to research by Ekman and Friesen, who developed the Facial Action Coding System (FACS) in 1978 (Singh and Nasoz 2020).

In the related work section, we will provide more extensive insights into the background and prior research concerning the utilization of the FE2020 facial expression dataset by CNNs. In order to classify photos of human faces into distinct emotion categories using CNNs, the goal of this work is to create a novel architecture from scratch (Kaushal et al. 2022). It also demonstrates feature extraction and preprocessing methods that can be used to increase accuracy on the FE2020 dataset.

2 Related Work

In the nineteenth century, Guillaume-Benjamin-Amand Duchenne de Boulogne, a French neurologist, had an interest in physiognomy and sought to comprehend how the muscles of the face produce facial emotions since he felt that these were directly related to a person's soul. In order to achieve this, he employed electric probes to cause muscle contractions. He then used newly invented camera technology to take

images of his patients' faces, capturing the contorted looks he was able to produce. He published his findings in the book "The Mechanism of Human Physiognomy," along with images of the induced facial emotions, in 1862.

CNNs are used to identify facial expressions, and the dropout and dataset extension strategies are used to address the issues of inadequate training data and overfitting. Using the LBP and SVM algorithms, C. Shan et al. classified face movements with an accuracy of 95.10%. The accuracy of the depth convolution neural network employed by Andre Teixeira Lopes et al. to classify face expression was 97.81%. Low-level network models, however, perform poorly when using the convolutional neural network technique, and deep network models frequently experience issues like data scarcity, overfitting, really have times, and so on (Duchenne de Boulogne 1862; Lopes et al. 2015).

On the FER2013 in 2016, Pramerdorfer and Kampel used CNNs to obtain accuracy of 75.2%. Additionally, they set up the engineering for up to 300 epochs and used stochastic angle drop to reduce the cross-entropy loss while maintaining a momentum value of 0.9. Various boundaries, including learning rate with 0.1, cluster size with 128, and weight decay with 0.0001 (Pramerdorfer and Kampel 1612).

Zhang et al. (2015) employed a Siamese network to describe a method for deciphering social interaction patterns from photographs and attained a test precision of 75.1% on the challenging Kaggle look dataset. The authors increased the training data by using numerous datasets with various names; they also devised a feature extraction approach to cope with feature extraction.

Kim et al. (2016) proposed a group of CNNs and showed that it is beneficial to employ both registered and unregistered versions of given face photos during testing and training. On the FER2013 dataset, the authors attained a test accuracy of 73.73%. Additionally, they carried out illumination normalization and interface for a traditional 2-D alignment, both of which are openly accessible for landmark detectors. Based on the outcomes of facial landmark detection, they executed registration selectively in order to avoid the registration error.

3 Dataset

According to the Facial Expression (FE2020) Dataset present at the largest dataset, which is publicly accessible on the Kaggle's Facial expression dataset, FE2020 dataset includes 31,058 face cropping, including photos used for training and testing, with 23,879, 7,179, respectively as shown in Fig. 1.

Seven various emotions are represented by photos, along with their respective labels, in this graph (0 = anger, 1 = disgust, 2 = fear, 3 = happy, 4 = neutral, 5 = sad and 6 = surprise). All photos are grayscale and 48 × 48 pixels in size. Ian Good fellow determined this dataset's accuracy to be about 82.6%.

Fig. 1 FE dataset photos with labels

4 Methdology

We will explore our CNNs' architecture and procedures in this section in order to improve the dataset's accuracy.

5 Preprocessing

According to the feature extraction procedure, preprocessing can be used to enhance the performance of the (FE2020) dataset framework. Different techniques are included in picture preprocessing, such as face identification and placement, brightness, occlusion, pose, and data augmentation correction.

The brightness and contrast of the image change depending on the object's brightness and illumination. These variations increase the complexity of feature sets and detection techniques (Lilhore et al. 2021) The architecture for face detection is depicted in general in Fig. 2.

In order to identify faces in images, the input block saves the image that was collected. The preprocessing block that eliminates undesirable noise and normalizes the image uses the face area as its input. The trainer module receives the output, processes it, trains the image, determines whether it belongs to the face class or not, and then provides information about face recognition (Saudagare and Chaudhari 2012).

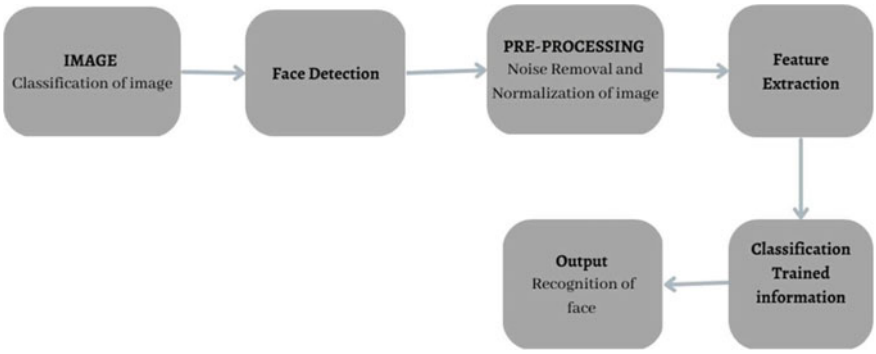


Fig. 2 Architecture for face detection is represented

6 Feature Extraction

In the following module, which extracts facial features, appropriately preprocessed data is provided as input. The important information in an image is chosen by feature extraction, making it simple to identify the emotions. The characteristics retrieved by CNN and HOG (Nandi et al. 2020) are integrated in this study to perform a crossover technique for face component extraction, which includes facial landmarks (Bodini 2019). HOG depicts the appearance and shape of local items by navigating edge bearings throughout a scene. HOG is change-invariant since it operates on local cells. It is easier to identify these HOG highlights because they vary for each expression. This is the rationale behind their decision to use HOG as the component selector in this framework.

Using this framework, a convolutional neural network is implemented (CNN) (Pramerdorfer and Kampel 1612) using the additional feature extracted in the previous module along with the feature extracted using CNN's capabilities for grouping emotions, four convolutional layers and four pooling layers.

7 CNN Architecture

There are various computer vision applications that frequently use CNNs, and the FE2020 dataset is just one of them. A number of FE dataset studies have been published in the early twenty-first century. (Fasel 2002; Gupta et al. 2020) have shown that both changes in scale and changes in face placement perform well with CNNs. Additionally, they were discovered to perform better than multilayer perceptron (MLP) while analyzing novel changes in face posture.

Researchers used CNN to help them address issues with translation, rotation, subject independence, and scale invariance in facial emotion recognition (Matsugu et al. 2003).

As seen in Fig. 3, finding an optimum architecture or network structure is a very challenging task. Our model was trained using the following traits:

- Three convolution layers are using "RELU" as an activation function.
- Four max-pooling layers out of which the one, two, three, are using pool size (2, 2) and stride (1, 1), and four layers are using (2, 2) max-pooling. Every convolutional layer is followed by max-pooling layer.
- In total, six dropout values for one to five are (0.4) and for six are (0.3).
- One flattened layer, three dense layers, two of which have the activation functions "RELU" and other is "SOFTMAX," respectively.
- There are 4.2 million total parameters and trainable parameters.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 46, 46, 128)       1280
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 23, 23, 128)       0
_________________________________________________________________
dropout (Dropout)            (None, 23, 23, 128)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 21, 21, 256)       295168
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 256)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 10, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 512)         1180160
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 512)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 512)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 2, 2, 512)         2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 1, 512)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 1, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 512)               262656
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_5 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 1799
=================================================================
Total params: 4,232,199
Trainable params: 4,232,199
Non-trainable params: 0
```

Fig. 3 CNN architecture illustration

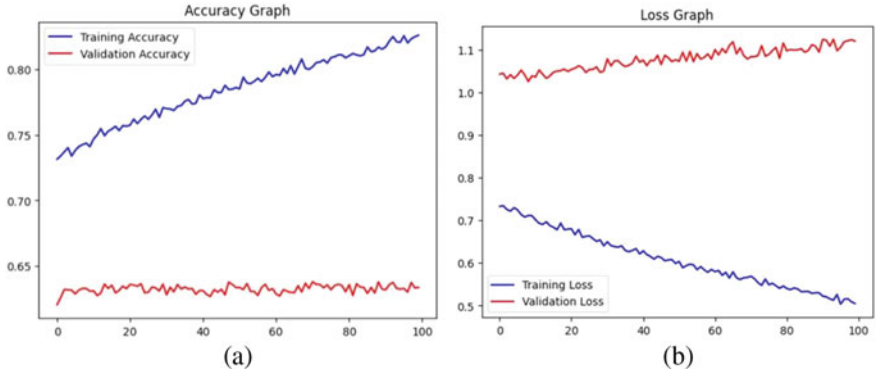


Fig. 4 **a** Accuracy **b** Loss

8 Result

We compared our findings with training and validation sets for the FE dataset. Our total training accuracy is 82.6%, training accuracy graph and our validation's total accuracy is 63.4%, as shown in Fig. 4a.

On the dataset that was provided, we utilized 100 epochs to train our model more effectively, but we can potentially use fewer epochs if necessary. Figure 4b shows the total training loss and validation loss above 100 epochs loss function will contain the difference between the output we get and the desired output.

Suppose the label of the input image is 0 and the model outputs 0.25, then the error will be (loss = output − true value) mean squared error function (loss function).

9 Conclusion

In this experiment, we offered a fresh approach to facial expression dataset and face detection, utilizing CNN and deep learning methods. In particular, we naturally correlate them. We ran a broad range of experiments on the well-known FE dataset to see how it performed. Even though our testing accuracy is 82.3% and validation accuracy is 63.4%, we think that less layers and filters might help the network perform even better. The accuracy and effectiveness of the suggested method for other devices will be further improved.

References

Agarwal S, Chatterjee M, Mukherjee DP (2012) Recognizing facial expressions using a novel shape motion descriptor. In: Proceedings of the eighth Indian conference on computer vision, graphics and image processing, pp 1–8

Bodini M (2019) A review of facial landmark extraction in 2D images and videos using deep learning. Big Data Cogn Comput 3(1):14

Donato G, Bartlett MS, Hager JC, Ekman P, Sejnowski TJ (1999) Classifying facial actions. IEEE Trans Pattern Anal Mach Intell 21(10):974–989

Duchenne de Boulogne G-B (1862) Mécanisme de la physionomie humaine, ou analyse électro-physiologique de l'expression des passions, applicable à la pratique des arts plastiques. Vve J. Renouard, Paris

Fard AP, Abdollahi H, Mahoor M (2021) ASMNet: a light weight deep neural network for face alignment and pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1521–1530

Fasel B (2002) Head-pose invariant facial expression recognition using convolutional neural networks. In: Proceedings fourth IEEE international conference on multimodal interfaces. IEEE, pp 529–534

Gupta A, Arunachalam S, Balakrishnan R (2020) Deep self-attention network for facial emotion recognition. Procedia Comput Sci 171:1527–1534

Kaushal C, Islam MK, Althubiti SA, Alenezi F, Mansour RF (2022) A framework for interactive medical image segmentation using optimized swarm intelligence with convolutional neural networks. Comput Intell Neurosci 2022

Kim B-K, Dong S-Y, Roh J, Kim G, Lee S-Y (2016) Fusing aligned and non-aligned face information for automatic affect recognition in the wild: a deep learning approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 48–57

Lien JJ-J, Kanade T, Cohn JF, Li C-C (2000) Detection, tracking, and classification of action units in facial expression. Robot Auton Syst 31(3):131–146

Lilhore UK et al (2021) Impact of deep learning and machine learning in industry 4.0: impact of deep learning. In: Cyber-physical, IoT, and autonomous systems in industry 4.0. CRC Press, p 179197

Lopes AT, De Aguiar E, Oliveira-Santos T (2015) A facial expression recognition system using convolutional networks. In 2015 28th SIBGRAPI conference on graphics, patterns and images. IEEE, pp 273–280

Matsugu M, Mori K, Mitari Y, Kaneda Y (2003) Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Netw 16(5–6):555–559

Nandi A, Dutta P, Nasir M (2020) Automatic facial expression recognition using histogram oriented gradients (HoG) of shape information matrix. In: Intelligent computing and communication: proceedings of 3rd ICICC 2019, Bangalore 3. Springer, pp 343–351

Pramerdorfer C, Kampel M (2016) Facial expression recognition using convolutional neural networks: state of the art. arXiv preprint arXiv:1612.02903

Saudagare PV, Chaudhari D (2012) Facial expression recognition using neural network–an overview. Int J Soft Comput Eng 2(1):224–227

Singh S, Nasoz F (2020) Facial expression recognition with convolutional neural networks. In: 2020 10th annual computing and communication workshop and conference (CCWC). IEEE, pp 0324–0328

Xia X-L, Xu C, Nan B (2017) Facial expression recognition based on tensorflow platform. In: ITM web of conferences, vol 12. EDP Sciences, p 01005

Zhang Z, Luo P, Loy C-C, Tang X (2015) Learning social relation traits from face images. In: Proceedings of the IEEE international conference on computer vision, pp 3631–3639

Deep Learning-Based Multi-state Colorectal Cancer Histological Image Classification

Hardeep Kaur, Anil Kumar, and Varinder Kaur Attri

Abstract Colorectal carcinoma is one of the most prevalent cancers of intestinal tract. Globally, it is the second biggest reason in cancer-associated deaths. A swift diagnosis is required to enhance the lifespan of colon cancer patients. Nevertheless, disease categorization becomes challenging since histopathological images include a variety of cells and features. This research provides a deep learning-based model for colorectal cancer multi-state classification. Expert-level reliability in medical image categorization has lately been proven using deep learning methods. A deep learning model, Dense Net 121, is employed to classify the colorectal cancer in eight tissue types. Dense Net 121 is being trained and assessed at several epochs to measure the learning rate. The learning rate, confusion map, and accuracy rate are used to assess the effectiveness of the pretrained model. Dense Net 121 obtained 97% classification accuracy with an average CPU time of 31 s. This model will be able to detect and categorize tissue types from similar histology imaging datasets in the future.

Keywords Colorectal cancer · Histological images · Multi-state classification · Deep learning approach

H. Kaur
Guru Nanak Dev University, Amritsar, Punjab, India
e-mail: hardeepcet.rsh@gndu.ac.in

A. Kumar (✉) · V. K. Attri (✉)
Guru Nanak Dev University Region Campus, Jalandhar, Punjab, India
e-mail: anil.dcse@gndu.ac.in

V. K. Attri
e-mail: varinder2002@yahoo.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_39

1 Introduction

Variations in genes cause aberrant tissue proliferation in the rectum and colon parts of the digestive tract, leading to the growth of malignant polyps and the onset of colorectal cancer. Colorectal cancer is a deadly disease with a high death and incidence rate in advanced nations. It is considered the third typical type of cancer, leading to nearly 10% of overall cases across the globe (Jang et al. 2020). Clinicians and researchers confront a difficulty when attempting to identify colorectal cancer.

Immunotherapy, chemotherapy, radiation therapy, and biopsy are some of the treatments used to cure colon cancer. Its treatment will be easier to manage if the progress of abnormal tissue development is recognized early (Devvi et al. 2020). It is often detected by microscopic observation of colon biopsy images. Therefore, this approach is unreliable as well as time-intensive, resulting in considerable observer heterogeneity. Several initiatives have been undertaken to offer suitable methods for the automatic diagnosis of colorectal cancer. The significance of computer-assisted procedures to help with clinical diagnosis has increased since the annual increase in specimens (Keum and Giovannucci 2019).

Computer-aided diagnostic (CAD) systems are the computer-based system that helps the healthcare professionals to understand images. Such systems process data input, including histopathological images and carry out different operations such as detection, classification, and segmentation. Clinical image classification approaches are becoming more frequently utilized as computer technology has grown in recent years (Xu et al. 2020; Escorcia-Gutierrez et al. 2022). In this study, a computer-assisted diagnosis system is built using a deep learning model for automated multiclass classification of colorectal cancer from healthy controls.

In earlier studies, multiple deep learning (DL) methods for the CAD diagnosis and categorization of clinical images have been investigated. Anurodh Kumar et al. (2023) suggested a lighter and less complicated CNN framework to automate the multi-state categorization of colorectal cancer. They used histological data, namely NCT-CRC-HE-100 K, colorectal histology, and an integrated dataset. When compared with four deep learning pretrained models, the developed model consumes less computing time to train. The proposed model attains 99.21% accuracy on the combined dataset. Chen et al. (2022) developed a semi-supervised colorectal histological image classification system that is based on multi-channel attention processes and interactive learning. It precisely distinguishes attention areas across spatial and channel parameters by combining different attention processes.

Lakpa and Byung (2021) provide a retroactive summary of basic machine learning and image processing-based computer-aided diagnostic (CAD) system. Additionally they present a complete analysis of use case scenarios with multiple kinds of deep learning methods for cancer diagnosis. To make use of multiple pretrained models and to evaluate their performance, Kather et al. (2021) investigated whether the existing pretrained CNN models could extract the prognosticators directly from H&E-stained tissue slides. According to Rizalputri et al. (2019) the classification of the colorectal histology images was carried out using a variety of techniques, including K-nearest

neighbor, logistic regression, random forest, and convolutional neural network. The classification model that outperformed the other models, such as K-nearest neighbor, logistic regression, and random forest in terms of accuracy has been selected. The convolutional neural network had an accuracy of 82.2%.

Using endo-microscopic pictures, Rasti et al. (2019) offered three methods for classifying three stages of tumors based on health, mice-colon walls, and inflammation. With the support vector machine approach, fully automatic machine learning techniques, such as traditional texture-based classification, deep learning, and transfer learning were introduced. They contrasted different training methods. They achieved a 99.93% accuracy rate on the Image Net ILSVRC. Using a GBM approach that relies on supervised methods, Na et al. (2019) demonstrated and tested a completely automated classification approach on a large number of pictures. According to the experimental findings, the first strategy achieved 98.49% correctness. On the other hand, the second strategy performed best overall, with an accuracy rate of 99.93% (Table 1).

The rest of the article is presented below: Sect. 2 describes the dataset and methodology employed for multi-state classification, while Sect. 3 illustrates and summarizes the outcomes. Subsequently, Sect. 4 concludes the study.

2 Methodology

This section demonstrates the materials utilized and the procedures employed in the investigation. This begins with the dataset adopted in this research, then moves on to data augmentation approaches, and finally to the deep learning model employed in this research. Figure 1 illustrates the work flow of methodology for multi-state colorectal cancer classification.

2.1 Dataset

The Kather texture 2016 image is a set of samples from eight cell types of colorectal lesions associated with humans. It has 5000 H&E-stained histological images with a pixel size of 150 × 150. Images were captured using an Aperio Scan Scope at a magnification of 20x. The Kather Texture 2016 is an open library of colorectal cancer histology images. The University Medical Center Mannheim Heidelberg University, Mannheim and Germany gathered this information. Each image represents one of eight distinct tissues, namely stroma, complex, tumor, empty lympho, debris, mucosa, and adipose. Table 2 illustrates the description of each tissue type. On the other hand, the Fig. 2 depicts the variation observed in similar tissue types as well as the similarities across samples of various types; such unique features support the classification's complexity. Kaggle.com offers a colorectal histopathology image collection.

Table 1 List of deep learning-based research to classify the colorectal cancer (Lakpa and Byung 2021)

Author name/ reference	Aim	Image classification approach	Image type	Imaging dataset	Findings (%)
Devvi et al. (2020)	Classification as benign and malignant, and diagnosis of colorectal cancer using deep learning models	ResNet-50 and ResNet-18 employed for cancer classification	Colon gland images	Warwick-QU	Accuracy = 80 Specificity = 83 Sensitivity = 87
Yao et al. (2021)	To automate the classification and segmentation of colorectal data using a deep learning	To overcome the issue of labeling imbalances and to classify the problematic imaging samples, the STVGG architecture was implemented	Endoscopy images	3061 colorectal images	Accuracy = 96
Mallela et al. (2022)	Multi-grade categorization of colorectal cancer using deep learning	VGG-16, ResNet-50, and suggested model adaptive ResNet-152 employed for classification	Histology images	Kather texture image (2016)	Accuracy of VGG-16 = 96.14 ResNet50 = 97.08 Adaptive ResNet152 = 98.38
Fahami et al. (2021)	Classification and reorganization of genes in colon lesion	Decision tree and K-nearest neighbor used for colon cancer classification	Histology images	HTSeq-FPKM-UQ	Accuracy:97.49 ± 2.92 Percision:100.0 ± 90.00 Recall: 95.00 5.83
Talukder et al. (2022)	Diagnosis of colon and lung cancers	RF, SVM, LR, MLP, XGB, and LGB used for classification	Histology images	LC25000	Accuracy of colon: 99.05 Lung:100 Colon and lung: 99.30

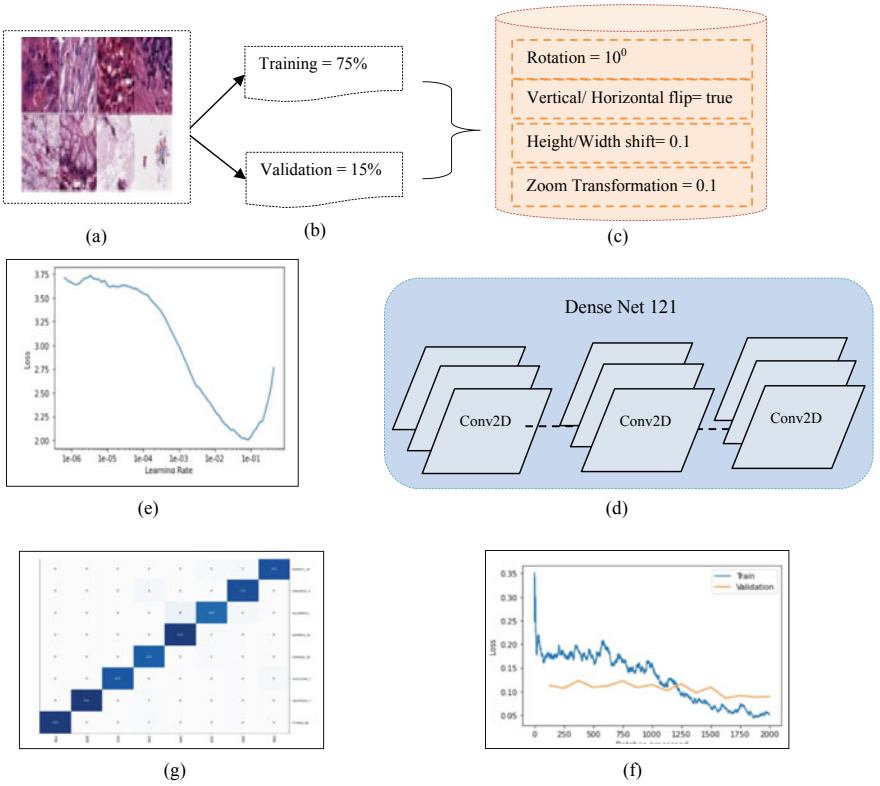


Fig. 1 Represents step-by-step workflow of methodology: **a** Histology images with eight tissue classes, **b** Imaging data distribution, **c** Data augmentation, **d** Pretrained model, **e** Learning rate versus loss rate curve, **f** Curve for batch processed against loss for training and validation, and **g** Confusion map for eight tissue types

Table 2 Description of eight classes of histology dataset

Class no	Class name	Description
1	STROMA	A stroma is a functional component of a cell
2	COMPLEX	Complex cells are ones that have multiple cells that work together to achieve a certain activity
3	TUMOR	Cell with the growth of aberrant tissues
4	EMPTY	Background of histopathological images
5	DEBRIS	Debris also known as hematoxylin and eosin stain tissues
6	MUCOSA	Mucous membrane
7	LYMPHO	A component of immune system
8	ADIPOSE	Body fat

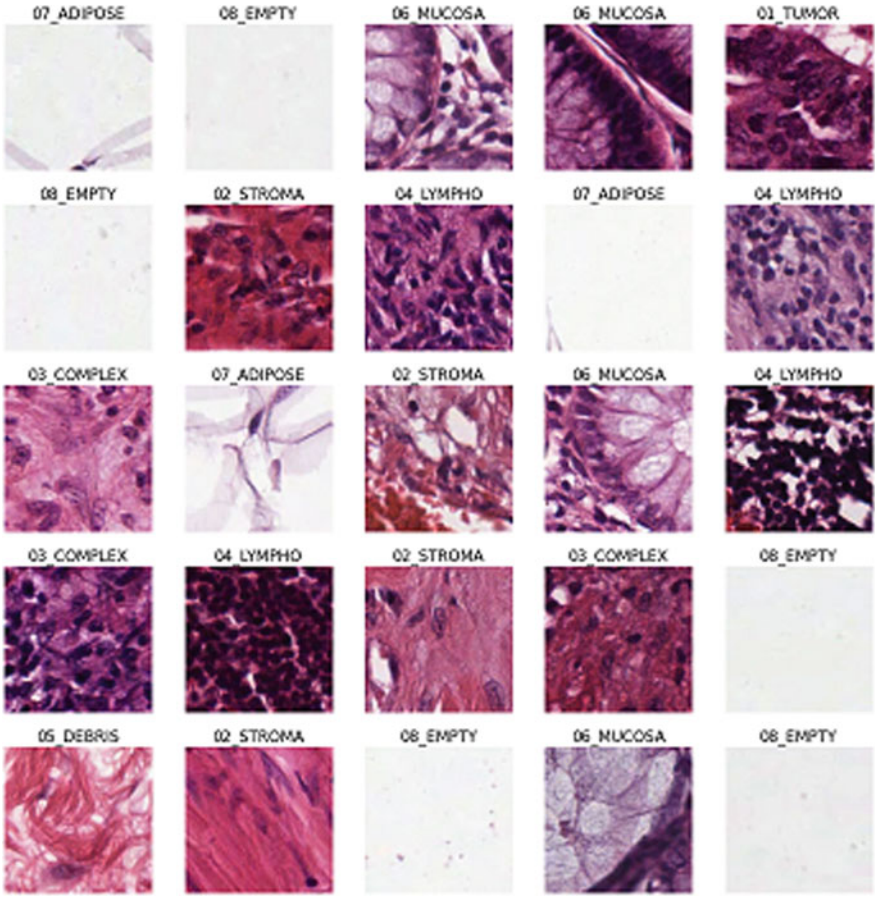


Fig. 2 Pictorial representation of independent sample histology images with eight classes

2.2 Dataset

The Kather texture 2016 image is a set of samples from 8 cell types of colorectal lesions associated with humans. It has 5000 H&E-stained histological images with a pixel size of 150 × 150. Images were captured using an Aperio Scan Scope at a magnification of 20×. The Kather Texture 2016 is an open library of colorectal cancer histology images. The University Medical Center Mannheim Heidelberg University, Mannheim, and Germany gathered this information. Each image represents one of eight distinct tissues namely stroma, complex, tumor, empty lympho, debris, mucosa, and adipose. Table 2 illustrates the description of each tissue type. On the other hand, the Fig. 2 depicts the variation observed in similar tissue types as well as the similarities across samples of various types; such unique features support the

Table 3 Tissue type distribution of colorectal cancer images

Tissue type	Training set	Validation set	Test set	Total images
Stroma	437	94	94	625
Debris	437	94	94	625
Adipose	437	94	94	625
Mucosa	437	94	94	625
Empty	437	94	94	625
Tumor	437	94	94	625
Lympho	437	94	94	625
Complex	437	94	94	625

classification's complexity. Kaggle.com offers a colorectal histopathology image collection.

2.3 Dataset Separation

The authors generated a new database that contains a total of 5,000 histological images of colorectal cancer with eight classes of tissues (Kather et al. 2016). All eight classes are evenly distributed and each tissue set contains 625 images, i.e., 75 $\times$ 75 μm. Following Table 3 presents the equal distribution of all eight classes. Each tissue contains 437 images for training, 94 images for validation, and 94 images for testing, respectively. This clinical imaging data is separated into three sets: training, testing, and validation. The training set contains 75% of images, validation set contains 15%, and validation set contains 15%.

2.4 Data Augmentation

Images can be preprocessed in a variety of ways. The preparation methods are well-suited to the image's present state. Because of the small number of data points in the dataset adopted in this research, learning a deep neural network (DNN) with such a small dataset may result in network over fitting. As a consequence, data augmentation methods are employed to enhance the training images. They regulate visual contrast, rotation, de-noising, brightness, and flipping. Although a variety of transformations, including picture orientation, lighting, size fluctuations, posture, and backdrop, should be decreased. In this article, zoom, rotation, height shift, width shift, vertical flip, and horizontal flipping transformation have been performed to preprocess the images. To rotate the image to a specific angle, the rotation transformation is applied. Hence, 10 degrees are assigned to rotate the picture. Flip has been

used to horizontally and vertically flip the picture. Horizontal and vertical flip is set to 'true' value. Images are moved dynamically to the left or right by adjusting the width shift range. The width shift factor got fixed to 0.1. Despite the fact that the images can be rotated vertically, the height shift range has been adjusted to 0.1. The zoom conversion is then performed using the zoom range parameter with a range of 0.1

2.5 Dense Network

To attain cutting-edge image classification performance, pretrained algorithms are increasingly being deployed. It is especially effective when the available computer power and data are both restricted. Dense network is an abbreviation for densely connected convolutional topology. Huang and colleagues designed dense network (Huang et al. 2017). Because the extracted features do not need to be relearned, it needs fewer factors than conventional networks. Dense Net preserves the integrity of the other feature maps while feeding a constrained feature map to the topology's overall data. Advanced information and gradient fluxes, which make it simpler to train throughout the network, are one benefit of the dense network. In this article, classification is performed using a pretrained deep learning model, i.e., Dense Net 121. The goal of a dense network is to maximize the sharing of data across network levels. It comprises 121 layers, three transition layers, and four dense blocks in total. These 121 layers are made up of 117 convolutional layers, three transition layers, and one classification layer. The key component of such a structure is dense block. Every layer in this model is intimately linked to all other layers in a feed-forward style. If there are 'm' layers in a conventional convolutional architecture, there will also be 'm' connections, whereas in dense networks, this number is m (m + 1)/2. This makes it easier to train more intricate network configurations.

3 Results

A deep learning algorithm is employed in this experiment on histology imaging dataset, namely Kather texture 2016. Kather texture 2016 contains eight tumor classes. These classes are dispersed uniformly. The deep learning model is programmed in Python language using PyTorch library. Google Colab is used to train and test the model. It is an open source available Python-based environment that runs directly in the cloud. The effectiveness of the pretrained model in terms of several outcome metrics is explored below. The basic key parameters, as well as the confusion map, represents the amount of accurate forecasts for each class based on the images being generated to determine output. The three separate parameters used to assess the model's productivity are as follows: learning rate, loss, and confusion matrix. Accuracy has been used to assess progress.

The learning is continued in two stages. The first cycle Fig. 3 illustrates the learning rate versus loss curve. Although it depicts the plotting curve for eight epochs in the second cycle with the variable value adjusted to 5e-3, when the variable value is le-6, the curve is shown at 0.14 losses. A curve in Fig. 4 depicts the batches processed against loss for the training and validation sets and confusion map for eight classes. The confusion map has been utilized to demonstrate the effectiveness of categorization for eight classes.

In a confusion matrix, the Y-axis represents the actual classes of the images, while the X-axis represents the anticipated classes. This enables comparison of the results with the actual classifications of the pictures.

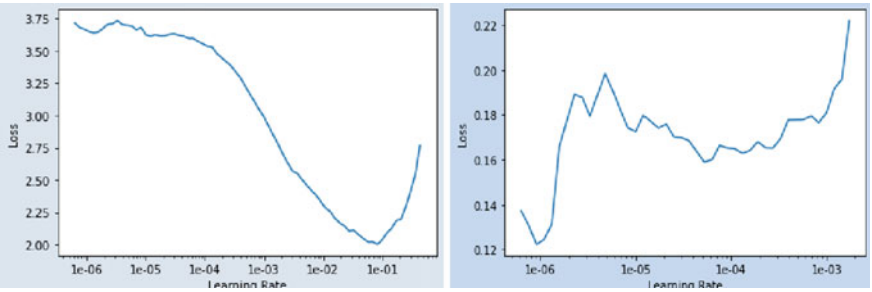


Fig. 3 Learning rate curve against loss for training and validation set

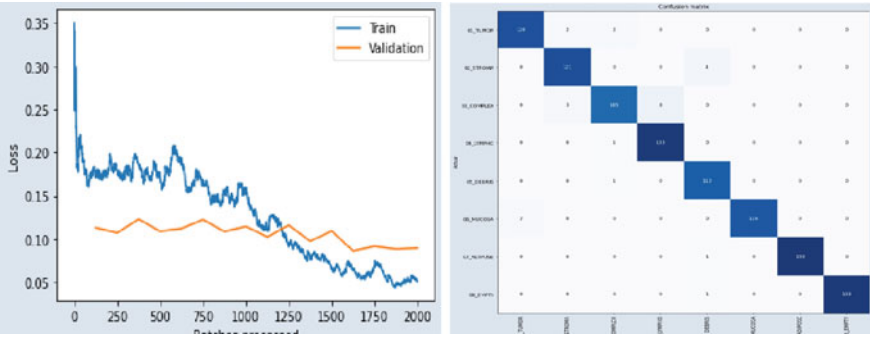


Fig. 4 Loss versus batches processed curves for training and validation set and confusion map for eight classes

4 Conclusion

Worldwide, one of the leading causes of deaths associated with lesions is colorectal carcinoma. Deep transfer learning's most recent developments have made it possible to automate the study of histological imaging information as a resource for computer-aided clinical imaging classification, segmentation, and detection. A deep learning approach called Dense Net 121 is utilized in this article to classify the malignancy into eight classes: stroma complicated, tumor, empty, lympho, debris, mucosa, and adipose. For the implementation, a publicly accessible image dataset, Kather texture 2016, is employed. This dataset contains 5,000 colorectal cancer histology images. The model's productivity is evaluated using three distinct parameters: learning rate, loss, and confusion matrix. The accuracy rate is used to measure the performance of the model. The findings are shown in Figs. 3 and 4 as loss rate curves, learning rate curves, and batch processed curves for training and validation sets, and a confusion matrix of eight tissue types for cancer classification. The performance of the model achieved 97% classification accuracy.

References

Chen H, Chen L, Xiaoyan L, Mamunur RM, Weiming H, Yixin L, Wanli L, Changhao S, Hongzan S, Xinyu H, Marcin G (2022) IL-MCAM: an interactive learning and multi-channel attention mechanism-based weakly supervised colorectal histopathology image classification approach. Comput Biol Med 143:0010–4825

Devvi S, Radifa HP, Alhadi B, Pinkie A (2020) Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. In: 5th international conference on computer science and computational intelligence 2020. Procedia Computer Science, Science direct pp 423–431

Escorcia-Gutierrez J, Mansour RF, Beleno K, Jimenez-Cabas J, Perez M, Madera N, Velasquez K (2022) Automated deep learning empowered breast cancer diagnosis using biomedical mammogram images. Comput, Mater Continua 71(3):4221–35

Fahami MA, Roshanzamir M, Izadi NH, Keyvani V, Alizadehsani R (2021) Detection of effective genes in colon cancer: a machine learning approach. Inform Med

Huang G, Liu Z, Van DML, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708

Jang HJ, Lee A, Kang J, Song IH, Lee SH (2020) Prediction of clinically actionable genetic alterations from colorectal cancer histopathology images using deep learning. World J Gastroenterol 26(40):6207–6223

Kather NJ, Weis AC, Bianconi F, Melchers MS, Schad LR, Gaiser T, Marx A, Zollner GF (2016) Multi-class texture analysis in colorectal cancer histology. Sci Rep

Kather JN, Krisam J, Charoentong P, Luedde T, Herpel E, Weis CA, Gaiser T, Marx A, Valous NA, Ferber D (2019) Predicting survival from colorectal cancer histology slides using deep learning: a retrospective multicenter study. PLoS Med 16(1)

Keum N, Giovannucci E (2019) Global burden of colorectal cancer: emerging trends, risk factors and prevention strategies. Nat Rev Gastroenterol Hepatol 16(12):713–732

Kumar A, Vishwakarma A, Bajaj V (2023) CRCCN-net: automated framework for classification of colorectal tissue using histopathological images. Biomed Sign Process Control 79(2):1746–8094

Lakpa DT, Byung WK (2021) Deep learning approaches to colorectal cancer diagnosis: a review. Appl Sci 11(22)

Mallela S, Naga R, Battula SR (2022) Colorectal multi-class image classification using deep learning models. Bull Electr Eng Inf 11(1):195–200

Na KS (2019) Prediction of future cognitive impairment among the community elderly: a machine-learning based approach. Sci Rep

Rasti P, Wolf C, Dorez H, Sablong R, Moussata D, Samiei S, Rousseau D (2019) Machine learning-based classification of the health state of mice colon in cancer study from confocal laser endomicroscopy. Sci Rep

Rizalputri LN, Pranata T, Tanjung NS, Auliya HM, Harimurti S, Anshori I (2019) Colorectal histology CSV multiclassification accuracy comparison using various machine learning models. In: 2019 international conference on electrical engineering and informatics. IEEE

Talukder MA, Islam MM, Uddin MA, Akhter A, Hasan KF, Moni MA (2022) Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning. Expert Syst Appl

Xu L, Walker B, Liang PI, Tong Y, Xu C, Su YC, Karsan A (2020) Colorectal cancer detection based on deep learning. J Pathol Inform 11:1–5

Yao Y, Gou S, Tian R, Zhang X, He S (2021) Automated classification and segmentation in colorectal images based on self placed transfer network. Bio Med Res Int

Transcription of American Sign Language (ASL) Using Convolutional Neural Networks (CNNs)

Nishu Chaudhary, Ridham Sheel, Ranjan Kumar, and Poornima Mittal

Abstract Sign language is most natural and the fastest way to gather knowledge and information, and it is an expressive and efficient way for the people with hearing and speech impairment to convey their thoughts and feelings. Sign language to text conversion technology allows these individuals to more easily communicate with others by converting their hand gestures into text. This technology can bridge the communication gap and improve integration and accessibility for people who rely and primarily utilize sign language as their prior means of expression. This translation is done using image processing, computer vision and machine learning technique. This technology can process video input of American hand signs and convert them into text in more than one language, providing real-time captioning and subtitle improving communication and accessibility for people with HSD and with our method we get 95.7% accuracy for 26 letters. The technology not only helps people with hearing and speech impairments, but also helps in teaching sign language to those who are not familiar with it.

Keywords Machine learning · Computer vision · Image processing · Sign language · Sign language recognition · Hearing and speech impairment · Convolution neural network

N. Chaudhary · R. Sheel · R. Kumar (✉) · P. Mittal
Delhi Technological University, Delhi, India
e-mail: ranjankumar_2k19ec143@dtu.ac.in

N. Chaudhary
e-mail: nishuchaudhary_2k19ec118@dtu.ac.in

R. Sheel
e-mail: ridhamsheel_2k19ec145@dtu.ac.in

P. Mittal
e-mail: poornimamittal@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_40

1 Introduction

Sign language is a visual language that uses physical features of the human body such as hand gestures, facial expressions and body language for communication. It is used by people who are completely deaf or hard-of-hearing as the main method of communication. There are different sign languages worldwide, such as American Sign Language (ASL) (Lopez 2017; Padmanabhan and Sornalatha 2014) and British Sign Language (BSL). Sign language recognition, processing and translation technology are needed to overcome the communication gap between such people and us and facilitate better understanding between non-signers and the signers, especially in education, healthcare and consumer services (Negi et al. 2020). This technology helps people with hearing impairment or hearing loss to have equal access to information and services.

American Sign Language (ASL) is an intricate and ever-evolving mode of communication which has four parts handshapes, movements, facial expression and non-manual makers. The number of signs in ASL is difficult to pin down because it is a rich and vibrant language that is constantly evolving. However, it's estimated that ASL has thousands of signs each with its own unique meaning (Shirbhate et al. 2020). In this paper, sign language to text conversion is a process of converting handshapes sign gestures and movements into written text. We have used 27 signs, 26 letters of English alphabets and one for space as shown in Fig. 1.

Convolutional neural networks (CNNs) and sequence-to-sequence (Seq2Seq) algorithms are two popular deep learning techniques used in natural language processing (NLP) and computer vision (CV) tasks (Kumar 2018). CNNs are commonly used in image and video analysis for feature extraction and classification (Rawat and Mittal 2021). In contrast, Seq2Seq models are designed to handle sequential data such as text, speech, or time-series data (Yuan et al. 2021). However, these two techniques are not mutually exclusive, and researchers have explored combining them to tackle complex problems. For instance, in machine translation, a CNN can be used as an encoder to extract features from the input text, and a Seq2Seq model can be used as a decoder to generate the translated text (Monika 2021). This combination has shown promising results in improving the accuracy and fluency of machine translation systems. One of the significant applications of CNNs is object recognition, where CNNs can identify objects in images by learning from a large dataset

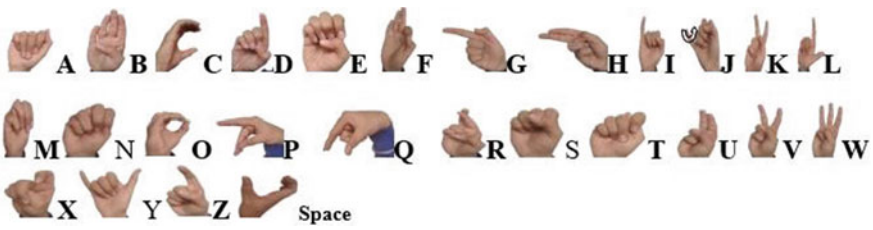


Fig. 1 American sign language (ASL)

of labeled images (Mittal and Jha 2021). Additionally, CNNs can be used for image classification, where they can classify images into various categories based on the features they extract from the input image. Another important application of CNNs is in image segmentation, where CNNs can partition an image into various regions or objects based on their visual characteristics. Moreover, CNNs have been applied in image restoration, where they can remove noise, blur, or other distortions from an image to improve its quality. In summary, CNNs have a wide range of applications in image processing, including object recognition, classification, segmentation and restoration (Bharti and Mittal 2021).

2 Literature Survey

Recent research in the field of sign language comes up with the wide range of topics such as development of sign language, the cognitive process involved in using sign language, the education and teaching of sign language and the use of sign language in various settings, such as the classroom or the workplace (Fok et al. 2015). Some researchers also study the use of sign language in specific population, such as deaf individuals or individuals with communication disorders (Kumar 2018). Additionally, researchers also study the technology related to sign language such as sign language recognition (Padmanabhan and Sornalatha 2014), sign language translation and sign language generation. Some notable examples include.

In 2015, researchers developed a machine learning system based on hidden Markov model algorithm that can recognize American Sign Language (ASL) signs with an accuracy of 85%. The system was trained on a dataset of over 10,000 ASL signs (Mittal et al. 2022). Later in 2016, researchers developed a deep learning base system using recurrent neural network (RNN) architecture that can translate from ASL to spoken English with an accuracy of around 90%. The system was trained on a dataset of around 10,000 sentences (Dakhore et al. 2021). For more accuracy, in 2017, researchers developed a sign language translation system using a sequence-to-sequence neural network architecture, which can be translated from ASL to written English and vice versa with the accuracy of 96%. The system was trained on the dataset of around 150,000 sentences. This model is quite impressive but all above proposed models are unable to process the colored (both front and back side of hand) photo. For that in 2019, researchers proposed an original approach for automatic sign language recognition using deep neural network with the combination of both depth and color information (Kumar and Mittal 2020). In 2018, a researcher developed a system which used linear discriminant analysis (LDA) algorithm for gesture recognition and then converted it to text as well as voice (Rahman et al. 2019). In 2020, researchers proposed a system that could work in a real-time environment using convolutional neural network and long short-term memory (CNN-LSTM) with an accuracy of around 98% (Lee et al. 2021). This model dataset is small that makes it

inefficient for recognizing hand gestures for different condition, different light intensity and in different background. In 2021, researchers proposed a system to identify hand gestures using the Otsu thresholding algorithm (Lean et al. 2019).

3 Proposed System

This paper is proposing a sign language to text conversion application that uses pretrained deep learning models to recognize and predict text from hand gestures captured by a webcam. Shown below is the Fig. 2, the flowchart depicting our proposed system. The application loads four pretrained models from disk and uses them to predict the text of the sign language input. The models are fed image data of American Sign Language (ASL) alphabet for the purpose of training.

The application capture video input from the user's webcam is then passed through the loaded deep learning models to predict the text. The predicted text is then spell checked to ensure accuracy and give the suggestions for words. The application

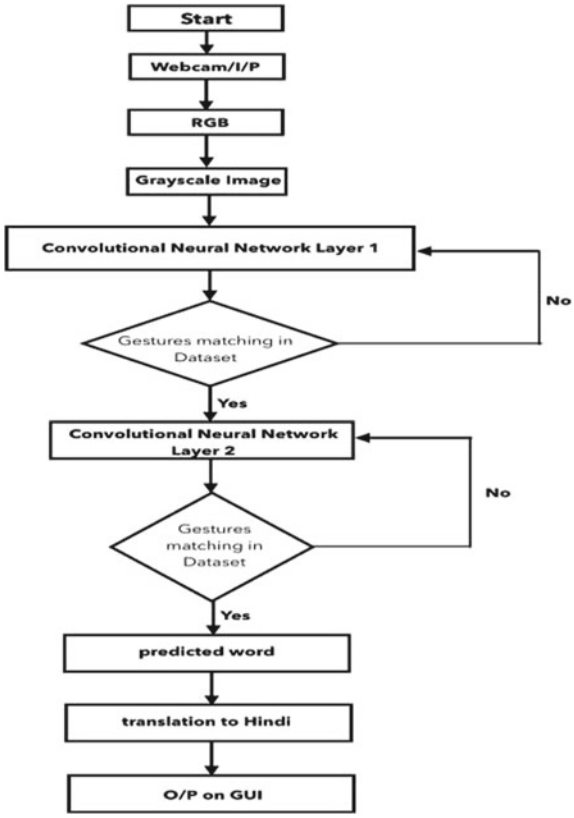


Fig. 2 Flowchart depicting the working of the project

creates a graphical user interface (GUI) for the application. The GUI allows the user to see the webcam input as well as the predicted text in real time in English as well as in Hindi. The user can also see the current symbol and the current word in the GUI in both languages. The application targets to make communication with people who use sign language easier by providing an easy-to-use application that can convert and translate sign language to text in English and in Hindi in real time.

4 Implementation

The first and foremost step for building a hand gesture recognition system is to gather a comprehensive dataset of images capturing sign language gestures for different characters (Fig. 3). The collected dataset must be diverse and inclusive of images captured under varying lighting conditions and from different angles to ensure the model can accurately recognize sign language gestures from various perspectives.

Data Preprocessing: After the collection of the dataset, the next step is to preprocess the images in the dataset. This step involves resizing the images to a standardized size, normalizing the pixel values and converting the images to grayscale. These preprocessing steps are essential in ensuring the images are in a format suitable for the model to process and learn from. *Model Training*: The third step is to train a deep learning model using the preprocessed dataset. Convolutional neural networks (CNNs) are commonly used in image recognition tasks, and for this sign language recognition system, a CNN will be trained using popular deep learning libraries such as Tensorflow, Keras, or Pytorch.

Model Testing: After the training of the DL model is complete, it's essential to check how it performs. For this purpose, a separate test dataset is used to determine its

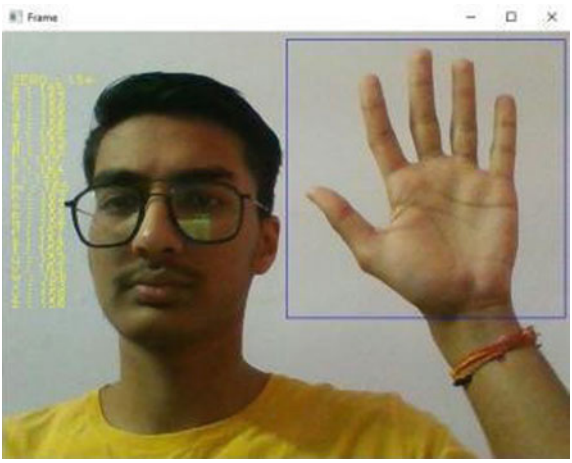


Fig. 3 Data collection

accuracy. This step is crucial in determining if the model has learned to accurately recognize sign language gestures, or if additional training is necessary.

GUI Implementation: The next step is to implement a graphical user interface (GUI) using a suitable library such as Tkinter. The GUI will be designed to display the webcam input, the predicted text, the current symbol and the current word. This interface will allow the user to interact with the sign language recognition system and see the predicted results in real time.

Spell Checking: To further improve the accuracy of the system, spell checking will be implemented using a library such as Hunspell. This will provide the user with auto-suggestions for predicted words, ensuring that the system's output is as accurate as possible.

Translation: In a subsequent step, the predicted word will be translated into Hindi using the *Google Translation API*. This feature will allow users to understand the predicted sign language in their preferred language, making the system more accessible.

Integration and Testing: The final step is to integrate all the above steps and thoroughly test the application to ensure that it is functioning properly. This step will ensure that the sign language recognition system is reliable and ready for use by individuals who rely on sign language for the purpose of communication.

5 System Working

This is nine-layer deep learning model designed to classify images. The architecture of the model is a sequential model that starts with an input layer that takes in grayscale images of size 128 $\times$ 128. The input image is then passed through two sets of convolutional and max pooling layers, which are used to extract features from the input image.

The first two layers of the model are Conv2D layers, with 32 filters of size 3 $\times$ 3 each. These layers perform the convolution operation on the input image, applying a set of filters (also known as kernels) to extract features. The 'padding' parameter is set to 'same', which means that the output size of the feature map will be the same as the input size. The 'strides' parameter is set to (1,1) which means that the filter will slide 1 pixel at a time, and the 'activation' parameter is set to 'relu' which applies rectified linear unit activation function after the convolution operation.

Following the Conv2D layers, are the maxpooling2D layers which perform down-sampling operation on the input feature maps. The 'pool_size' parameter is set to (2,2) which means that the maximum value of a 2 $\times$ 2 window is taken and the rest of the pixels are discarded. This helps to reduce the spatial dimensions of the input feature maps and also helps to make the model more robust to small translations of the input image.

After the feature extraction process, a flatten layer is used to flatten the input feature maps into a 1D array, so that it can be fed to the dense layers. The next two layers are dense layers also called fully connected layers; it applies a dense connection to all the neurons in the previous layers. The number of neurons in each of these layers is 128, and the activation function is 'relu'.

Finally, the model has a dropout layer, which is used to prevent overfitting by randomly dropping some of the neurons during training, this helps to reduce the dependency on any one feature in the input data.

6 Result

Utilizing the power of Python libraries, mainly TensorFlow, Keras and OpenCV, and with the application of convolutional neural networks, we have developed a software which can convert American Sign Language symbols of the alphabets to text. By showing a sequence of hand signs, deaf and mute people can generate a word and with the help of this application, others can understand what they are trying to convey.

Figure 4 shows the final working system which is able to identify the 26 alphabetical hand signs of the American Sign Language (ASL). Figure 5 is the confusion matrix obtained when only one layer CNN was used. It can be observed that there were some incorrect outputs, which were removed after the application of the second layer of CNN as shown in Fig. 6.

This application can also convert the formed words and sentences into Hindi and to other languages as well by just making a minor change in the code. This feature has been added considering the lingual diversities across the world. It helps to bridge the lingual differences not only between the disabled and us, but also between people coming from different backgrounds.

The algorithm used in our work is a type of sequence-to-sequence algorithm (Seq2Seq). As mentioned in the literature survey, previous works that have been done on this problem have used algorithms like HMM, LDA and Otsu thresholding. Although these algorithms do a perfectly fine job for image, data and speech processing, Seq2Seq has its own advantages. One of the main advantages of seq2seq models is their ability to handle variable length sequences, which is critical in NLP tasks such as language translation and speech recognition, where input and output sequences can have varying lengths. In contrast, algorithms like HMM and LDA may struggle to handle such sequences. Another advantage of seq2seq models is their ability to model long-term dependencies between sequences, which is important in tasks such as language translation and summarization, where the meaning of a word or phrase can depend on the context of previous words or phrases. On the other hand, algorithms like Otsu thresholding and LDA may not be effective in capturing such long-term dependencies.

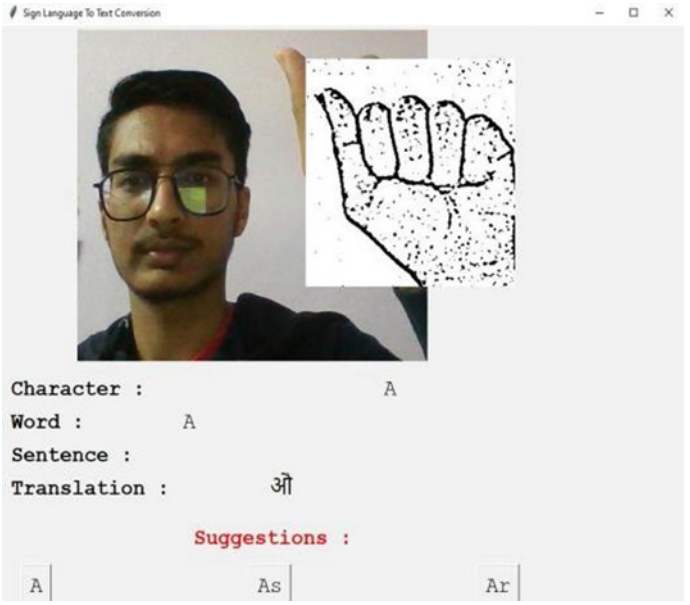
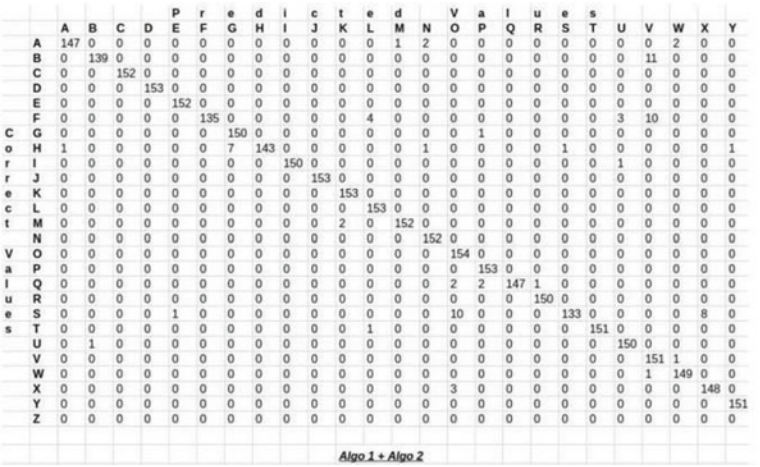


Fig. 4 Final working system

		Predicted Values																							
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Correct Values	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	2	0	0	
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	
	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	
	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	
	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	
	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	
	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	
	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	
	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	
	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0	
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1

Fig. 5 Confusion matrix when only one layer was used



Correct \ Valuues	A	B	C	D	PE	Fr	Ge	Hd	Ii	Jc	Kt	Le	Md	N	OV	Pa	Ql	Ru	Se	Ts	U	V	W	X	Y
A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
H	1	0	0	0	0	0	7	143	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
S	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 6 Confusion matrix when two layers were used

7 Conclusion

A cutting-edge technology has been developed to help the deaf and mute people communicate with us. Utilizing the power of convolutional neural networks (CNNs), this system accurately recognizes and categorizes various signs used in sign language, thereby providing an accurate translation to text. This technology is intended to eliminate communication barrier between those who are deaf and mute and the society as a whole. This system is designed to effectively identify and interpret hand gestures, extract key features and consolidate data from various sources. It is a powerful tool that is able to recognize and interpret signs, identify key features and combine relevant information from various sources. Furthermore, it is an effective solution for developing expert knowledge and detecting edges in sign language. With this technology, communication between the deaf and mute community and society becomes seamless and effortless. This innovative solution is an important step toward providing equal opportunities and breaking down barriers for this community.

8 Future Scope

There are several potential areas for future development for the sign language recognition. Such a model's performance can be improved by training it on a larger and more diverse dataset and by fine-tuning the model's architecture and hyperparameters. The model can be trained to recognize a wider range of signs, including those used in different sign languages or dialects, or to recognize signs made by different people with different styles. The system can be designed to provide feedback to the

user, such as indicating when a sign is not recognized, or providing suggestions for alternate signs that could be used instead. The system can be improved by developing a more robust system that can handle complex hand gestures, signs and finger spelling, variations in lighting and background, and able to function under different orientation or angle. System can have all four elements of ASL recognition to behave in full-fledged assistance.

References

Bharti R, Mittal P. Frequency analysis of ring oscillator at different technology nodes. International conference on simulation, automation & smart manufacturing (SASM 2021)

Dakhore H, Landge M, Patil S, Patil T, Zyate S, Moon A, Lade R (2021) Sign language recognition using machine learning. Int J All Res Educ Sci Methods (IJARESM) 9(6)

Fok K-Y, Ganganath N, Cheng C-T, Tse CK (2015) A real-time ASL recognition system using leap motion sensors. International conference on cyber-enabled distributed computing and knowledge discovery. Xi'an, China, pp 411–414. https://doi.org/10.1109/CyberC.2015.81

Kumar M (2018) Conversion of sign language into text. Int J Appl Eng Res 13(9):7154–7161. ISSN 0973-4562

Kumar N, Mittal P (2020) Performance analysis of FinFET based 2:1 multiplexers for low power application. 6th students' conference on engineering and systems (SCES-2020), organized by MNNIT, Allahabad, India, July 10–12, 2020 (IEEE, Scopus Indexed). https://doi.org/10.1109/SCES50439.2020.9236770

Lean KS, Tolentino RO, Serfa J, August C, Thioac MA, Pamahoy B, Forteza JR, Xavier JO (2019) Static sign language recognition using deep learning. Int J Mach Learn Comput 9(6)

Lee CKM, Ng KKH, Chen C-H, Lau HCW, Chung SY, Tsoi T (2021) American sign language recognition and training method with recurrent neural network, vol 167, p 114403. ISSN 0957-4174

Lopez VI (2017) Hand gesture recognition for sign language transcription. Boise State University, Research and Economic Development

Mittal P, Jha S. Traffic congestion control system based on microcontroller. International conference on simulation, automation & smart manufacturing (SASM 2021)

Mittal P, Rawat B, Kumar N (2022) Tetra-variate scrutiny of diverse multiplexer techniques for designing a barrel shifter for low power digital circuits. Microprocess Microsyst J 90:104491. (ISSN: 0141-9331), (Elsevier, I. F. 3.503). https://doi.org/10.1016/j.micpro.2022.104491

Monika MP (2021) A novel modified current comparator based on extremely low voltage high compliance current mirror. Int J Inf Technol (BJIT) 14:323–331. (ISSN: 2511-2112), (Springer, ESCI). https://doi.org/10.1007/s41870-021-00823-7

Negi S, Mittal P, Kumar B (2020) Numerical modeling and parameters extraction of novel triple hole block layer based organic light emitting diode for display. J Soc Inf Display 22(12):956–964. (ISSN: 1938-3657), (Wiley, I. F. 2.017). https://doi.org/10.1002/jsid.952

Padmanabhan V, Sornalatha M (2014) Hand gesture recognition and voice conversion system for dumb people. Int J Sci Eng Res 5(5)

Rahman MM, Islam MS, Rahman MH, Sassi R, Rivolta MW, Aktaruzzaman M (2019) A new benchmark on American sign language recognition using convolutional neural network. International conference on sustainable technologies for industry 4.0 (STI). Dhaka, Bangladesh pp 1–6. https://doi.org/10.1109/STI47673.2019.9067974

Rawat B, Mittal P (2021) A 32 nm single ended single port 7T SRAM for low power utilization. Semicond Sci Technol 36(9):095006–095022. (ISSN: 1361-6641) (IOP Sciences, I. F. 2.048). https://iopscience.iop.org/article/10.1088/1361-6641/ac07c8

Shirbhate SR, Shinde DV, Metkari AS, Borkar UP, Khandge AM (2020) Sign language recognition using machine learning algorithm. Int Res J Eng Technol (IRJET) 7(3)
Yuan G, Liu X, Yan Q, Qiao S, Wang Z, Yuan L (2021) Hand gesture recognition using deep feature fusion network based on wearable sensors. IEEE Sens J 21(1):539–547. https://doi.org/10.1109/JSEN.2020.3014276

Analysis of Greedy, Semi-greedy, and Random Scheduling Heuristics with DVFS for Heterogeneous Fog Computing Platform

Savina Bansal, Rakesh K. Bansal, and Nikita Sehgal

Abstract Internet of things (IoT)-based real-time applications, demanding energy-aware and low latency responses, are leading to a wider interest of researchers toward fog computing platforms to complement cloud computing by provisioning support for real-time computations near the end devices. However, the development of an optimized full-fledged fog platform, to meet latency, energy, and other quality of service (QoS) requirements of users and applications, is quite challenging. For real-time deadline-constrained tasks, energy efficiency and timeliness are two crucial parameters. Judicious scheduling of tasks can play a vital role in the effective utilization of such resource-constrained heterogeneous computing platforms. Greedy, semi-greedy, and random scheduling approaches are suggested by researchers to address energy efficiency and complexity-related issues. In this work, we employ dynamic voltage and frequency scaling (DVFS) technique on these three heuristics to ascertain its impact on reducing network energy consumption further without violating task deadline constraints. Simulations done on wide variety of task sets, on heterogeneous fog nodes, indicate that active energy consumption can be reduced with DVFS, with random heuristic exploiting the maximum benefit of DVFS followed by semi-greedy and greedy approach.

Keywords Scheduling · Real-time tasks · Deadline laxity · Fog computing

S. Bansal (✉) · R. K. Bansal · N. Sehgal
Department of ECE, Giani Zail Singh Campus College of Engineering and Technology, MRSPTU, Bathinda, Punjab 151001, India
e-mail: savina.bansal@gmail.com; savinabansal@mrsptu.ac.in

R. K. Bansal
e-mail: rkbansal@mrsptu.ac.in

N. Sehgal
e-mail: nikita.ece@mrsptu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_41

1 Introduction

In the present age of digitization, IoT is coming up as one of the fastest emerging technologies for interconnecting virtually anything that can be digitized, thereby generating huge amount of data and increasing computational demands for cloud computing platforms. To ease out the load and to handle latency and other QoS requirement of IoT tasks, fog or edge computing is being advocated to complement the cloud resources (Bonomi et al. 2012; Fog computing and the internet of things: Extend the cloud to where the things are, cisco white paper 2015; Matrouk and Alatoun 2021; Rao et al. 2015; Fizza et al. 2022; Oma et al. 2018). As these edge devices are normally resource-constrained, therefore, energy-efficient task scheduling in fog computing plays a major role and also poses a great challenge before researchers. Task scheduling can play a crucial role in efficient resource utilization, especially for resource-constrained fog computing platforms (Aslanpour et al. 2020; Naha et al. 2018; Belmahdi et al. 2021;Oma et al. 2018; Dabiri et al. 2022; Lee et al. 2021). Scheduling broadly refers to allocation and assignment of tasks on to nodes in the computing network to meet scheduling objectives such as minimizing makespan, deadline violations, energy consumption, or maximizing user satisfaction, fault tolerance, and reliability with optimized resource consumption (such as computational time, network bandwidth, or computing cost) (Jamil et al. 2020; Aslanpour et al. 2020; Kaur et al. 2015a, b; Lee et al. 2021; Balmahdi et al. 2021). Most of the available works in real-time task scheduling are being reassessed for their effectiveness in such a scenario (Stankovik et al. 2012; Dabiri et al. 2022; Bansal et al. 2020). Surveys of fog scheduling algorithms (Stankovik et al. 2012; Kaur et al. 2021; Matrouck et al. 2021; Naha et al. 2018; Belmahdi et al. 2021) reflect that the major focus of researchers is toward judicious assignment of tasks on to fog nodes to meet user's QoS requirements, with energy consumption and makespan projected as the most important metrics for fog environment. For real-time tasks, however, minimization of deadline violation is more important than makespan, so, finish time of a task can be delayed up to the maximum allowable response time (deadline) of the task without compromising user's satisfaction. Accordingly, deadline laxity that refers to the amount by which the finish time of a task can be delayed without violating deadline condition becomes an important parameter, as it can be exploited further for energy management or reduction by using dynamic power management (DPM) or dynamic voltage and frequency scaling (DVFS) schemes (Kaur et al. 2015a, b; Kaur et al. 2021; Xu et al. 2020; Rizvandi et al. 2011). DVFS amounts to reduction in supply voltage and clock frequency, in tandem, for lower workloads. It results in slowing down a system without affecting its performance and is vastly advocated by researchers for energy-efficient computing. More succinctly, DVFS trades off speed and energy by exploiting the switching activity-related component of dynamic energy consumption in computing machines, which is given by the interplay among supply voltage (V), clock frequency (f), physical capacitance (C), and an activity factor α as given below

$$Pdyn \sim \alpha CV2f \tag{1}$$

Lowering of supply voltage decreases drive current available for charging and discharging transistor circuit capacitances, resulting in increased gate delays seeking lowering of clock frequency for flawless circuit working. This feature of scaling supply voltage and frequency in tandem is termed as dynamic voltage and frequency scaling (DVFS). However, reduction in clock frequency may result in slowing down computational speed which in turn shall elongate task completion time and may not benefit toward overall system energy consumption. But, since the relation (Eq. 1) is quadratic in voltage so there may be net saving in dynamic power consumption with reduction in voltage as supported by many a works available in the literature (Kaur et al. 2015a, 2016, 2017; Dabiri et al. 2022; Bansal et al. 2020; Rizvandi et al. 2011). Here, in this work, a recently proposed deadline and energy-aware priority-aware semi-greedy (PSG) algorithm (PSG) (Azizi et al. 2022) that has been shown to significantly outperform many other popular scheduling algorithms, is modified and assessed for further energy reductions. DVFS scheme is used for exploiting the deadline laxities available in the algorithm for the purpose. The algorithm is also compared with its greedy and random scheduling counterparts with the incorporation of DVFS parameters. It is shown that significant amount of deadline laxities remains available that can be exploited for slowing down the node speed without violating task deadline by reducing operating voltage that may reduce the total computational energy on the heterogeneous fog platform.

2 Present Work

The basic fog and task model are similar to the one used in Sect. 3.2.1 of Azizi et al. 2022, wherein a priority-based semi-greedy scheduling algorithm meant for scheduling independent real-time IoT tasks on fog computing platform is proposed by Azizi et al. Its working is broadly overviewed here for the sake of understanding the work in a better manner. Real-time tasks are first arranged in an increasing order of their respective deadlines. Next, prospects of scheduling the highest priority task subjected to fulfillment of task deadline on each fog node are explored. In each case, record is kept for its impact on the overall computation energy consumption of the system. If deadline is not satisfied on any fog node, then the task is scheduled on a node with minimum response time and deadline violation penalty is recorded. If task deadline gets satisfied on more than one node, then there exist three possibilities (i) random, (ii) greedy, and (iii) semi-greedy. In case of random scheduling, task can be scheduled on any of the potential fog node at random, whereas in the greedy approach task shall get scheduled on a fog node offering the lowest overall computational energy consumption. In the semi-greedy approach Azizi et al. 2022, prospective fog nodes are first arranged in increasing order of their energy consumptions as recorded earlier, and then a fixed percentage of minimum indexed nodes (based on

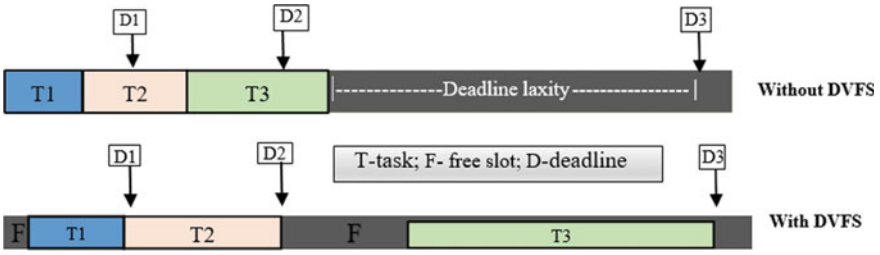


Fig. 1 An example of a task schedule on a fog node with and without DVFS

an alpha parameter value (taken as 0.4)) is taken up for consideration and task can get scheduled at random on any of these fog nodes.

The algorithm can behave as greedy (alpha = 0), semi-greedy (0 < alpha < 1), or random (alpha = 1) algorithm based on the alpha parameter value. After the resulting schedule is obtained, deadline laxity available under the above three algorithmic approaches is exploited for further reduction in energy consumption without sacrificing any of the task deadline constraints. For the purpose, we look for the possibility of applying voltage and frequency scaling on each fog node in maximum of two steps first by employing 50% voltage reduction that reduces the energy consumption by fourfold, but doubles the execution time of a task on the node and if it is not viable then 25% voltage reduction is tested. For checking viability, availability of suitable time slot (deadline laxity) to accommodate extended execution times needs to be checked. Furthermore, deadline of the task should not be violated with slowing down of machines with voltage scaling. If these conditions are met then the task is scheduled to be executed at the said voltage on the said node. For applying DVFS, we start from the last scheduled task on the fog node. The outline of the pseudocode is shown in Figs. 1 and 2.

3 Simulation Setup

Simulation of the fog environment is somewhat similar to the one used in the base work and is comprised of heterogeneous fog nodes (FNs) interconnected with random mesh topology. Fog nodes are varied from 10 to 100 with step size as 10, and tasks are varied from 100 to 1000 with step size as 100. Fog nodes are taken up as heterogeneous nodes with varying computing power uniformly distributed from 2000 to 4000 MIPS. Active power consumption of nodes is also randomly distributed in the range (80–200) W, while idle state power consumption is taken as 60–70% of the active power consumption. Propagation delay between FNs is considered to be (1–3) millisecond (ms), and communication links bandwidth is kept as 1000 Mbps. The task deadline of a task is randomly distributed to (2–10) times, its average computation cost of fog network, to give a level playing field to all nodes in the network. Task sizes

Input:	Task list: L1 and Fog network F, α
Output:	Scheduled list of tasks in L1 on F with DVFS using any of the three scheduling schemes (based on α)
/----------Scheduling without DVFS /	
Step 1:	Sort L1 in EDF order
Step 2:	For each task Ti in L1
Step 3:	Make list Li of viable fog nodes satisfying Ti deadline
Step 4:	count= number of nodes in Li
Step 5:	if count=0, task is rejected.
Step 6:	If (α= 0) % random approach
Step 7:	Schedule Ti on any randomly selected node in Li
Step 8:	If (α≠ 0)
Step 9:	For each fog node Fj in Li
Step 10:	Record active energy consumption of the whole network, assuming Ti gets scheduled on Fj
Step 11:	Sort Li in ascending order of network energy consumption
Step 12:	if (α=1) % greedy approach
Step 13:	Schedule Ti on the first node in Li
Step 14:	if (α=0.4) % semi-greedy approach
Step 15:	Schedule Ti on any of the first (α * count) nodes in Li in a random manner
Step16:	Calculate total network active energy consumption without DVFS
/----------Applying DVFS on the non-DVFS schedule ----------/	
Step17:	For each fog node Fj in F
Step18:	Explore possibilities of applying DVFS (with 50% /else 25% voltage reduction) by exploiting available deadline laxities starting from the last scheduled task
Step19:	Update total network active energy consumption with DVFS
/----------end----------/	

Fig. 2 Pseudocode for DVFS enabled random, greedy, and semi-greedy scheduling approaches

are mixed bag of small and bigger size tasks randomly varying from (100–10,000) million instructions (MI). The input and output file sizes are randomly selected in the range (100–10,000) and (1–1000) kilobytes, respectively. Type of greedy, semi-greedy, or random scheduling scenario is selected by keeping the alpha parameter $\alpha = 0$, 0.4, and 1, respectively. All simulations are coded in C/C++ programming language. The experiments were carried out on a desktop running on Intel(R)Core (TM)i7-6700 CPU@3.40 GHz, 3.41 GHz, 8 GB of RAM, and Windows 10 operating system. Each experiment is repeated five times with different seed values for random number generation, and average value is reported in the results reported. The source code of the base work (Rao et al. 2015) was modified to incorporate the change.

4 Performance Metrics

Makespan parameter that refers to the maximum finish time of a task in a given job set is deliberately not taken up here, as used in the base work. Our rationale for the same is that the present job set comprises of independent tasks with their own independent deadlines, and hence, overall makespan might not be of much concern to a particular task user. However, in case of jobs comprising of precedence-constrained tasks, makespan parameter becomes more relevant and also the main scheduling objective for minimization. Further, while calculating overall energy consumption of the fog network only active computational energy consumptions is being used after modification in the base work. Further, it is assumed that free slots left after invoking DVFS may be utilized for fitting in batch tasks that have no specific time deadlines.

*4.1 Energy Consumption (*EC*)*

Energy consumption of the system refers to total active energy consumed by all the m fog nodes in the system. Energy consumption of a fog node (f_j) is amount of energy consumed when it is actually computing a task (t_{ij}) scheduled on it and is given by the product of its active power consumption and task execution time.

Energy consumption of node

$$f_j = \mathrm{EC}_j = \sum_{i=0}^{n} P_{\max}^{j} * \mathrm{et}(t_{ij}). \tag{2}$$

Total energy consumption of the network

$$\mathrm{EC} = \sum_{j=0}^{m} \mathrm{EC}_j. \tag{3}$$

4.2 Deadline Laxity

Deadline laxity of a task is the difference between task deadline and its response time (that equals the task's finish time plus communication delay to pass on the results to end device). In real-time task scheduling, the earliest completion of a task is not always desirable as the response can be delayed up to the task deadline without affecting user's satisfaction. Total deadline laxity is calculated by summing up all deadline laxities for the tasks scheduled on different fog nodes.

5 Experimental Results

Figure 3 shows deadline laxity with and without DVFS in all the three fog scheduling approaches. As expected, deadline laxity reduces after employing DVFS, as the tasks employing DVFS consume lesser energy at the cost of increase in task execution time by exploiting these. Deadline laxities without DVFS are minimum in the greedy approach and maximum in random scheme. This is expected as greedy scheduling tends to accumulate tasks onto limited number of most energy-efficient nodes in a greedy manner, leaving lesser scope for deadline laxities; whereas semi-greedy adopts somewhat lesser and random approach does not consider energy consumption a criterion for scheduling and hence spreads the tasks onto larger number of nodes, resulting in higher deadline laxities as observed.

As reflected in Fig. 3d, on an average, without DVFS, in comparison with the greedy approach, 151% and 448% higher deadline laxities were observed in semi-greedy and random approach, respectively. As expected, with DVFS, random approach benefitted the most and margin of deadline laxity reduced to 241% over its greedy counterpart, whereas for semi-greedy it was around 113%only. On an average over all the tasks, the actual number of tasks that underwent voltage and frequency scaling in greedy, semi-greedy, and random scheduling approaches were about

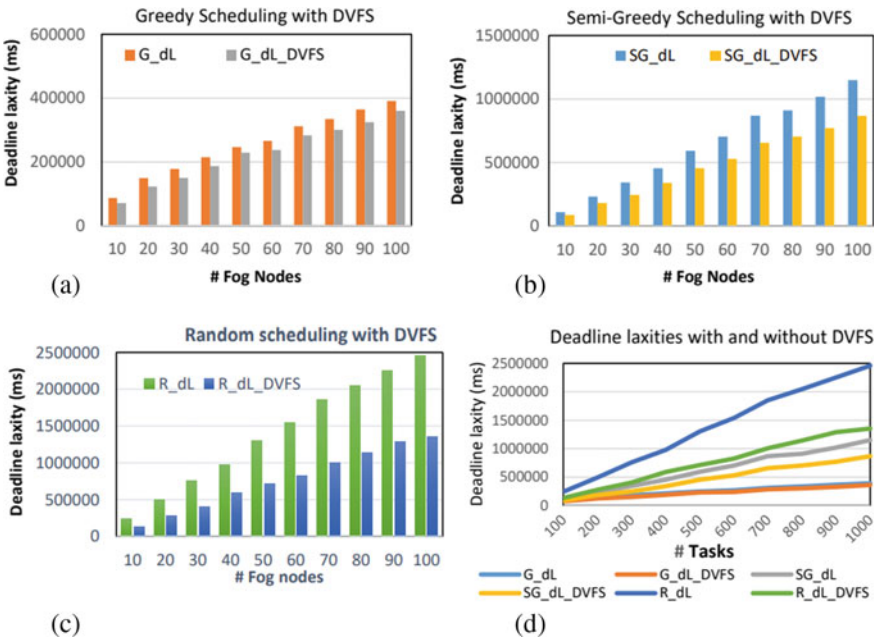


Fig. 3 Deadline laxity for **a)** greedy, **b)** semi-greedy, **c)** random, and **d)** overall scheduling schemes with and without DVFS

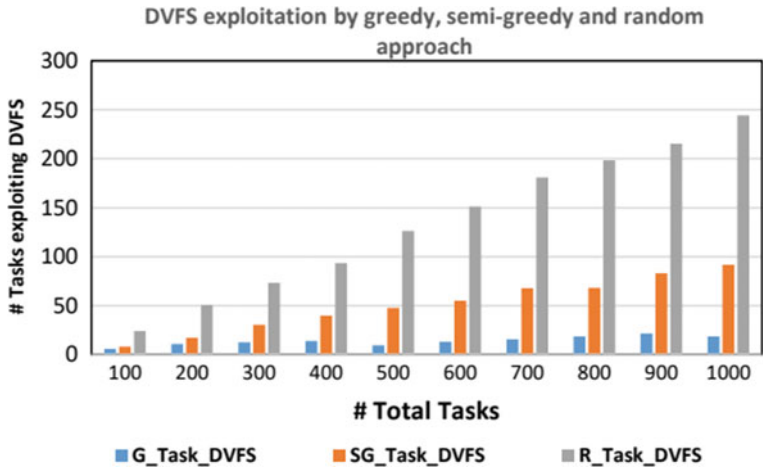


Fig. 4 Comparison of DVFS exploitation by greedy, semi-greedy, and random scheduling approach

3.2%, 9.2%, and 24.7%, respectively (Fig. 4). As expected, random approach makes maximum utilization of DVFS due to higher availability of deadline laxities.

Figure 5 shows the total computational (active) energy consumed with and without DVFS by using the three scheduling schemes on the tested experimental setup. It can be seen that DVFS results in good amount of energy savings in all the three scheduling approaches. Benefits achieved through DVFS are maximum in random approach owing to maximum deadline laxity available. On an average, energy consumption improvements by DVFS over their non-DVFS counterpart, as observed for greedy, semi-greedy, and random approaches are 2.7%, 9.5%, and 22.5%, respectively.

On an average, the energy consumption without DVFS was 4.9% and 26.8% higher than greedy scheduling approach in semi-greedy and random scheduling approaches, respectively. However, interestingly, after employing DVFS, this margin came down to −2.4 and 1% for semi-greedy and random schemes respectively with respect to the greedy approach, thereby making the semi-greedy approach to be marginally better than the other two after the application of DVFS. This is due to the fact that greedy approach tends to confine the tasks on to the most energy-efficient fog node leaving lesser scope for deadline laxities and hence DVFS exploitations. Whereas, semi-greedy and random approaches tend to utilize the available fog nodes in a more balanced manner leaving a good margin for deadline laxities as observed above and hence benefitting the most from DVFS exploitation in the process. After applying DVFS, as observed, the difference of heuristics diminished considerably making all these more or less at par to one other.

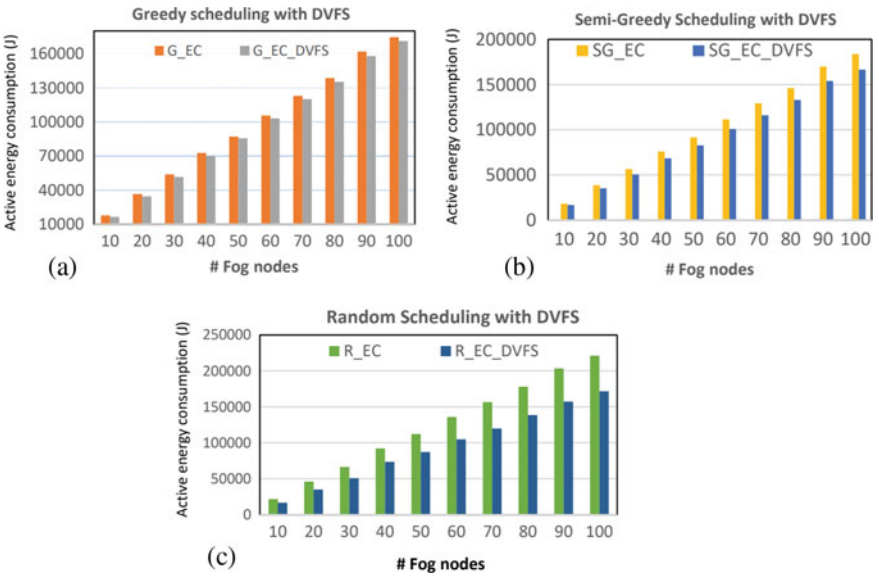


Fig. 5 Active energy comparison for **a)** greedy, **b)** semi-greedy, and **c)** random scheduling approaches with and without DVFS

6 Conclusion

Fog computing is a well-advocated computing scenario supplementing and complementing cloud computing, especially for IoT tasks that are latency-sensitive and energy-constrained. Effective task scheduling can play a major role in judicious exploitation of energy-constrained fog nodes for timely completion of real-time tasks. In this work, the greedy, semi-greedy, and random scheduling heuristics are analyzed for scheduling real-time deadline-constrained tasks on heterogeneous fog computing platform. The deadline laxity available in these schemes is first analyzed for the tested experimental setup, and it is observed that a huge amount of laxity remains available, especially in random scheduling approach. Next, the possibility of improvement in active energy consumption using DVFS scheme is explored by exploiting these deadline laxities by modifying the base work without violating task deadlines. On the simulated test setup, it was observed that DVFS can result in good energy savings with maximum savings observed in random scheduling approach followed by semi-greedy and greedy approach. Further, after applying DVFS, the difference among different scheduling heuristics reduced considerably making all of them more or less at par with each other.

References

Aslanpour MS, Gill SS, Toosi AN (2020) Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet of Things 12:100273. https://doi.org/10.1016/j.iot.2020.100273

Azizi S, Shojafar M, Abawajy J, Buyya R (2022) Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: a semi-greedy approach. J Netw Comp Appl 201:103333

Bansal S, Bansal RK, Arora K (2020) Energy-cognizant scheduling for preference-oriented fixed-priority real-time tasks. J Syst Architect 108:101743

Bansal S, Bansal RK, Arora K (2021) Energy efficient backup overloading schemes for fault tolerant scheduling of real-time tasks. J Syst Architect 113:101901

Belmahdi R, Mechta D, Harous S (2021) A survey on various methods and algorithms of scheduling in fog computing. Ingénierie des Systèmes d'Information 26(2):211–224

Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, pp 13. https://doi.org/10.1145/2342509.2342513 (2012).

Dabiri S, Azizi S, Abdollahpouri A (2022) Optimizing deadline violation time and energy consumption of IoT jobs in fog–cloud computing. Neural Comp Appl 34:21157–21173

Fizza K, Auluck N, Azim A (2022) Improving the schedulability of real-time tasks using fog computing. IEEE Trans Serv Comp 15(1):372–385

Fog computing and the internet of things: Extend the cloud to where the things are, cisco white paper (2015) http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

Iorga M, Feldman L, Barton R, Martin MJ, Goren N, Mahmoudi C (2018) NIST: fog computing conceptual model, recommendations of the National Institute of Standards and Technology. NIST Special Publication 500-325. https://doi.org/10.6028/NIST.SP.500-325

Jamil B, Shojafar M, Ahmed I, Ullah A, Munir K, Ijaz H (2020) A job scheduling algorithm for delay and performance optimization in fog computing. Conc Comp: Pract Exp 32(7):e5581

Kaur N, Bansal S, Bansal RK (2015a) Task scheduling & energy conservation techniques for multiprocessor computing systems. Int J Netw Syst 2(2):5–8

Kaur N, Bansal S, Bansal RK (2015b) Towards energy efficient scheduling with DVFS for precedence constrained tasks on heterogeneous cluster system. In: 2nd IEEE International conference on Recent Advances in Engineering & Computational Sciences, pp 1–6

Kaur N, Bansal S, Bansal RK (2016) Energy efficient duplication-based scheduling for precedence constrained tasks on heterogeneous computing cluster. Multiagent Grid Syst, IOS Press 12(3):239–252

Kaur N, Bansal S, Bansal RK (2017) Duplication-controlled static energy-efficient scheduling on multiprocessor computing system. Concur Comp: Practice Exp 29(12):e4124

Kaur N, Bansal S, Bansal RK (2021) Survey on energy efficient scheduling techniques on cloud computing. Multiagent Grid Syst 17(4):351–366

Lee S, Lee SK, Lee S-S (2021) Deadline-aware task scheduling for IoT applications in collaborative edge computing. IEEE Wireless Comm Lett 10(10):2175–2179

Matrouk K, Alatoun K (2021) Scheduling algorithms in fog computing: a survey. Inter J Netw Distributed Comp 9(1):59–74. https://doi.org/10.2991/ijndc.k.210111.001

Naha RK, Garg S, Georgakopoulos D, Jayaraman PP, Gao LX, Xiang Y, Ranjan R (2018) Fog computing: Survey of trends, architectures, requirements, and research directions. IEEE Access 6:47980–48009. https://doi.org/10.1109/ACCESS.2018.2866491

Oma R, Nakamura S, Duolikun D, Enokido T, Takizawa M (2018) An energy-efficient model for fog computing in the Internet of Things (IoT). Internet of Things 1–2:14–26

Rao TVN, Khan MA, Maschendra M, Kumar MK (2015) A paradigm shift from cloud to Fog computing. Inter J Comp Sci Eng Tech 5(11):385–389

Rizvandi NB, Taheri J, Zomaya AY (2011) Some observations on optimal frequency selection in DVFS-based energy consumption minimization. J Parallel Distrib Comp 71(8):1154–1164. https://doi.org/10.1016/j.jpdc.2011.01.004

Stankovic JA, Spuri M, Ramamritham K, Buttazzo GC (2012) Deadline scheduling for real-time systems: EDF and related algorithms, vol 460. Springer Science & Business Media
Xu J, Sun X, Zhang R, Liang H, Duan Q (2020) Fog-cloud task scheduling of energy consumption optimisation with deadline consideration. Inter J Internet Manufact Serv 7(4):375–392

The Effect of Ground Temperature on the Operating Temperature of Solar Photovoltaic Module Using FEM Analysis

Prakash Pratik and Kunal Ghosh

Abstract To understand the performance losses and deterioration brought on by thermal conditions, the operational temperature of photovoltaic modules must be measured. The environmental factors like solar irradiation, ambient temperature, wind speed and direction, module design and orientation, and ground temperature have an impact on the determination of the operating temperature of the module. In this work, a three-dimensional thermal model of p-Si solar photovoltaic module has been developed in Hypermesh software, and the effect of ground temperature on the operating temperature of p-Si photovoltaic module has been simulated on the ANSYS Fluent Software. The simulation results show that there is linear impact of the ground temperature on the module temperature. Additionally, it has been observed that the effect of the wind speed on the operating temperature is nonlinear and it is inversely related.

Keywords Three-dimensional thermal modelling · FEM analysis · CFD analysis · Photovoltaic module

1 Introduction

A PV module generates electricity directly from the solar rays. India is blessed with 330 days–350 days in a year bright sunshine that helps in solar power generation. In solar power generation process, a huge part of the solar radiation gets converted into heat. It raises the operating temperature of the module. The electrical and thermal performance of the polycrystalline Si-PV module is influenced by its working temperature, and for 1 °C rise in solar module temperature, the output power gets reduced by 0.4% of rated output over standard testing condition. Additionally, with the rise

P. Pratik (✉) · K. Ghosh
Indian Institute of Technology, Mandi, Himachal Pradesh 175001, India
e-mail: ppratik.iitmandi@gmail.com

NIELIT Ropar, Ropar, Punjab 140001, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_42

in module's temperature, thermal stress also increases within the module and the performance of the module's operation is got compromised. The module temperature under working conditions is influenced by a wide range of variables, many of which are strongly interdependent. To precisely forecast a module's temperature under operational conditions, a thermal model is therefore necessary. To ascertain the temperature gradient of a PV module, thermal models based on analytical and finite element methods (FEM) are typically used. Faiman (2008), Mattei et al. (2006), Skoplaki and Palyvos (2009), King et al. (2004) have proposed the development of analytical relationship conditional thermal models by the solving the thermal equations and energy balance equations. However, each of these models is designed for a specific site location, module configuration, which results in varying heat transfer coefficients.

The mentioned drawbacks of analytical models are addressed using finite element method (FEM). Previously developed thermal have discussed the effect of environmental parameters on the temperature distribution of PV module, and some of the famous thermal models are as follows: Lee and Tay have examined the thermal profile of each constituent layers of the module using the numerical simulation method. Siddiqui et al. (2012) calculated the influence of solar insolation level and surrounding temperature on the performance of PV modules. Zhou et al. (2015) have done a detailed work in the mentioned area. A polycrystalline-based 3D thermal model has been developed to analyze the influence of varying climatic variables on the thermal variation of PV module and optimize the performance using finite element simulation.

However, there are certain limitations associated with the previously developed models. The structural model does not take aluminium frames into account. Furthermore, the work does not model the impact of ground temperature. Additionally, the optical model used to calculate absorbance in each layers needs improvement.

In this work, a three-dimensional thermal model of p-Si photovoltaic module is developed to determine the effect of ground temperature on the module's operating temperature. The structure simulated incorporates the aluminium frame on the edges, along with the presence of essential layers, namely are glass, EVA, solar cells, and backsheet. In order to perfectly predict the operating temperature of the module, the calculation of solar radiation absorbed in each layers is based on a ray tracing software PVlighthouse. Finally, the developed model is used to analyze and simulate the thermal profile of a PV module under open-circuit condition for varying ground temperatures.

The continuing of the paper is categorized as follows. The comprehensive simulation setup for the developed model is kept at open-circuit condition with suitable thermal assumptions as described in Sect. 2. In Sect. 3, the aerial description of temperature of backsheet layer of PV module under a defined environmental condition has been analyzed. It provides an understanding of the spatial distribution of the temperature. Also, the influence of varying ground temperature on the operating temperature of the PV module is discussed with the help of a graph, while keeping other environmental parameters constant. Section 4 concludes the paper.

2 Simulation Setup and Thermal Hypothesis

For the characterization of the temperature distribution of the layers, ANSYS Fluent software simulates a three-dimensional thermal model based on FEM. This simulation involves two steps. The finite element programme Hypermesh is used to design a three-dimensional structure of poly-silicon solar PV and meshed it into finite elements in the first step. The meshed structure is exported into ANSYS Fluent in the successive level, where its steady-state thermal profile is simulated under open-circuit condition.

Figure 1 shows a representative structure of the PV module. As shown in Fig. 1, the developed structure is having five essential layers: glass, front EVA, silicon solar cell, rear EVA, and backsheet as the last layer. Due to their low thermal masses, the busbars, solder connector, junction box, and 2-mm gap between solar cells are disregarded in order to simplify the computational complexity of the model. An aluminium frame encircles these five layers of the module on all sides.

Due to open-circuit condition, the heat losses from thermal processes equalize the overall energy flux received as a result of incident solar insolation. The materials' optical properties, like absorption, transmittivity, reflectivity, and emissivity, used in the simulation are listed in Table 1 and are assumed to be isotropic and independent of temperature. The absorption in the constituent layers is calculated using ray tracing software called PVLight house, which takes the values of the solar radiation's intensity, spectrum, and angle of incidence to predict the absorption in each of the module's layer. For the calculation, we consider normal incidence and AM 1.5 G for solar gamut. It is assumes that convection and radiation are the means by which the PV module interacts thermally with its surroundings. The transfer of the radiative heat for the glass layer occurs with respect to the sky, which is assumed to be at air temperature. For backsheet layer, radiation is calculated with reference to the ground temperature, which is varying from 15 to 45 °C in this work, to study the effect of ground temperature.

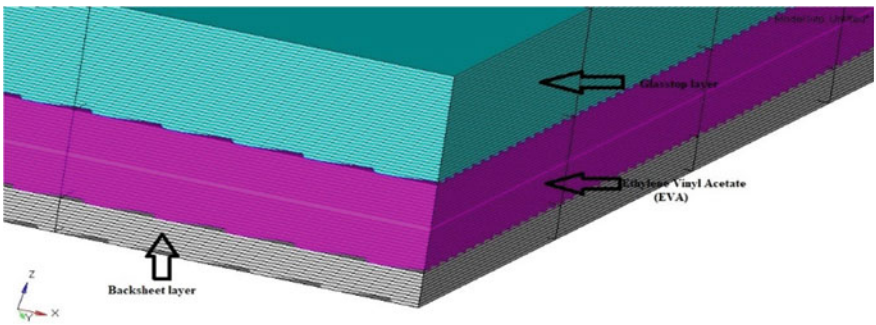


Fig. 1 Geometry of the meshed three-dimensional thermal model of standard p-Si PV module. The different layers of the module from top to bottom are glass, EVA, and backsheet, respectively, as shown in the figure

Table 1 Material properties of different layers. Also, the amount solar insolation absorbed in each layer is given in the last column. In the calculation for absorption of solar radiation, it has been assumed that 1000 W/m^2 radiation is incident

Different parts of the module	Thickness (in mm)	Thermal conductivity (W/mK)	Specific heat capacity (J/KgK)	Density (Kg/m^3)	Refractive index	Emissivity	Solar energy absorbed (in W/m^2)
Glass	3.2	0.8	800	2500	1.486	0.85	38
EVA	2.4	0.34	1400	950	1.5	–	7.5
Silicon	0.18	130	700	2329	3.44	–	834
Backsheet	1.0	0.1583	1760	1720	1.33	0.9	13
Aluminium	2.0	202.4	871	2719	3.08	0.71	800

3 Results and Discussion

We will analyze and discuss the simulation results in this section. We begin the discussion with by analyzing the aerial distribution of temperature of the backsheet layer of the developed module. The temperature differential between the constituent layers of the PV module is very small, less than 1 °C, and thereby, the thermal profile of the backsheet layer can be considered representative of other layers. The simulation is performed for a climatic condition corresponding to a solar insolation of 1000 W/m^2, ambient temperature of 298 K, wind speed of 1 m/s, and ground temperature at 308 K (in Fig. 2a) and 318 K (in Fig. 2b). The specific condition is closest to the standard testing condition. Figures 2a, b show the thermal profile. At the edges, there exists a variation because of the inclusion of aluminium frames. The temperatures at the edges are lower than at the centre, signifying a cooling effect of the aluminium frame. The specific condition is closest to the standard testing condition.

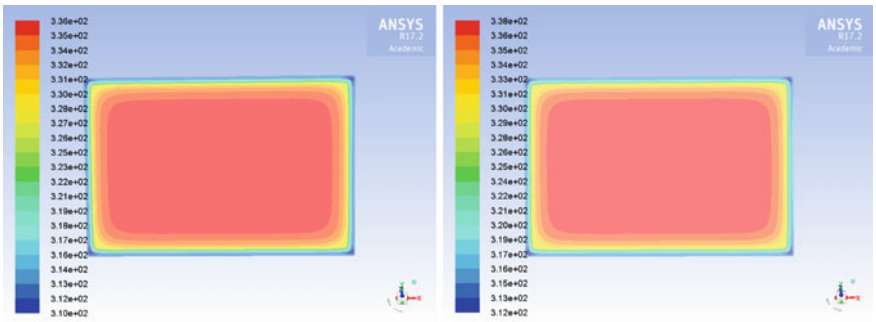


Fig. 2 Temperature distribution of backsheet layer when ground temperature is at 308 K on the left (**a**) and ground temperature is at 318 K on the right (**b**)

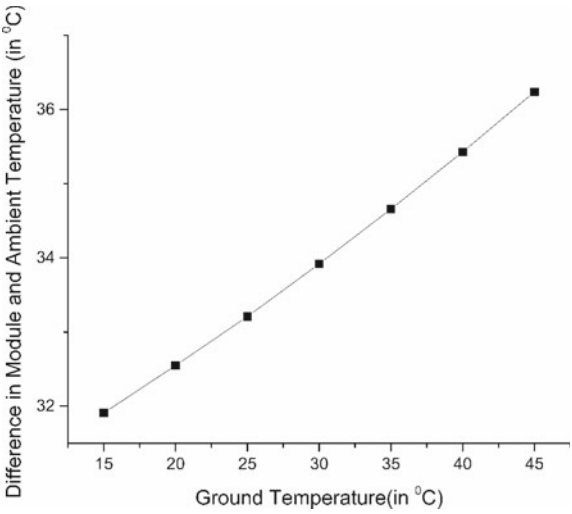


Fig. 3 Influence of ground temperature on the thermal distribution of PV module. X-axis represents the varying ground temperature from the 15 to 45 °C. While y-axis plotted the temperature differential between module and surroundings. From the plot, it is observed that with the increase in ground temperature, module temperature increases and lower ground temperature will keep the module temperature at a lower level

An experiment is performed to recognize the effect of varying ground temperatures. In a PV module, the backsheet layer (rear surface) faces the ground and the radiative heat transfer depends on the differential between T_{mod} and the ground temperature (T_{ground}). We have kept the U_{POA}, T_{a}, and WS to be 1000 W/m^2, 25 °C, and 1 m/s, respectively. Figure 3 shows the effect of deviation in ground temperature from 25°C on ΔT, and hence, on T_{mod}. In Fig. 3, the x-axis represents the ground temperature in degree celsius, while the y-axis represents ΔT. From the analysis, we can easily observe that the effect of ground temperature is significant and lower value of ground temperature helps in keeping operating temperature of the module at a lower value.

4 Conclusion

A three-dimensional thermal model of a poly-Si solar photovoltaic module is developed in this work. The analysis of thermal profile shows that the aerial variance in temperature is low, with the edges being at lower temperature because of the cooling influence of the aluminium frames. Additionally, we simulated the effect of ground temperature and found that a lower ground temperature, with other conditions remaining the same, assists to keep T_{mod} at a lower temperature.

References

ANSYS Inc. (2016) Fluent is a product of ANSYS, Inc., Release 17.2

Armstrong S, Hurley WG (2010) A thermal model for photovoltaic panels under varying atmospheric conditions. Appl Therm Eng 30:1488–1495. https://doi.org/10.1016/j.applthermaleng.2010.03.012

Bevilacqua P, Perrella S, Bruno R, Arcuri N (2021) An accurate thermal model for the PV electric generation prediction: long-term validation in different climatic conditions. Renew Energy 163:1092–1112. https://doi.org/10.1016/j.renene.2020.07.115

Charfi W, Chaabane M, Mhiri H, Bournot P (2018) Performance evaluation of a solar photovoltaic system. Energy Rep 4:400–406

Dhaundiyal A, Atsu D (2020) The effect of wind on the temperature distribution of photovoltaic modules. Sol Energy

Dhaundiyal A, Atsu D (2021) Energy assessment of photovoltaic modules. Sol Energy 218:337–345. https://doi.org/10.1016/j.solener.2021.02.055

Faiman D (2008) Assessing the outdoor operating temperature of photovoltaic modules. Prog Photovolt Res Appl 16:307–315. https://doi.org/10.1002/pip.813

Gu W, Li S, Liu X, Chen Z, Zhang X, Ma T (2021) Experimental investigation of the bifacial photovoltaic module under real conditions. Renew Energy 173:1111–1122. https://doi.org/10.1016/j.renene.2020.12.024

Hegazy AA (2000) Comparative study of the performances of four photovoltaic/thermal solar air collectors. Energy Convers Manag 41:861–881. https://doi.org/10.1016/S0196-8904(99)00136-3

Khelifa A, Touafek K, Moussab HB et al (2015) Analysis of a hybrid solar collector photovoltaic thermal (PVT). Energy Procedia 74:835–843. https://doi.org/10.1016/j.egypro.2015.07.819

King DL, Boyson WE, Kratochvil JA (2004) Photovoltaic array performance model, SANDIA report SAND2004-3535. Sandia Rep No 2004–3535(8):1–19

Lee Y, Tay AAO (2012) Finite element thermal analysis of a solar photovoltaic module. Energy Procedia 15:413–420. https://doi.org/10.1016/j.egypro.2012.02.050

Mattei M, Notton G, Cristofari C, Muselli M, Poggi P (2006) Calculation of the polycrystalline PV module temperature using a simple method of energy balance. Renew Energy 31:553–567. https://doi.org/10.1016/j.renene.2005.03.010

Photon J (2017) Experimental and computational fluid dynamics analysis of a photovoltaic/thermal system with active cooling using aluminum fins. J Photon Energy 7:1. https://doi.org/10.1117/1.JPE.7.045503

Rouholamini A, Pourgharibshahi H, Fadaeinedjad R, Abdolzadeh M (2016) Temperature of a photovoltaic module under the influence of different environmental conditions—experimental investigation. Int J Ambient Energy 37:266–272. https://doi.org/10.1080/01430750.2014.952842

Shaheed R, Mohammadian A, Kheirkhah Gildeh H (2019) A comparison of standard k–ε and realizable k–ε turbulence models in curved and confluent channels. Environ Fluid Mech 19:543–568. https://doi.org/10.1007/s10652-018-9637-1

Siddiqui MU, Arif AFM (2012) Effect of changing atmospheric and operating conditions on the thermal stresses in PV modules. In: ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, ESDA 2012. pp 729–739. https://doi.org/10.1115/ESDA2012-82740

Siddiqui MU, Arif AFM, Kelley L, Dubowsky S (2012) Three-dimensional thermal modeling of a photovoltaic module under varying conditions. Sol Energy 86:2620–2631. https://doi.org/10.1016/j.solener.2012.05.034

Skoplaki E, Palyvos JA (2009) On the temperature dependence of photovoltaic module electrical performance: a review of efficiency/power correlations. Sol Energy 83:614–624. https://doi.org/10.1016/j.solener.2008.10.008

Sun X, Silverman TJ, Zhou Z et al (2017) An optics-based approach to thermal management of photovoltaics: selective-spectral and radiative cooling. arXiv 7:566–574

Tina GM, Abate R (2008) Experimental verification of thermal behaviour of photovoltaic modules. In: Proceedings of the Mediterranean Electrotechnical Conference—MELECON, pp 579–584. https://doi.org/10.1109/MELCON.2008.4618497

Toledo C, López-Vicente R, Abad J, Urbina A (2020) Thermal performance of PV modules as building elements: analysis under real operating conditions of different technologies. Energy Build 223:110087. https://doi.org/10.1016/j.enbuild.2020.110087

Zhou J, Yi Q, Wang Y, Ye Z (2015) Temperature distribution of photovoltaic module based on finite element simulation. Sol Energy 111:97–103. https://doi.org/10.1016/j.solener.2014.10.040Silicon

A Descriptive Analysis of Different Dual-Port and Single-Port 11T SRAM Cells for Low-Voltage Operations

Yogita Chopra and Poornima Mittal

Abstract Here, this paper is providing a descriptive comparison of three different SRAM cells in terms of mode of accessing, stability, power consumption, and accessing delay. Three different 11T cells are compared in this paper; two are dual-ended, while one is a single-ended memory cell. After all the analyses, it is found that amongst the three cells, the single-ended cell is found to be better in performance in every aspect. All the three cells are evaluated on an operating voltage of 0.4 V with the technology node 32 nm. The obtained values of SNM for hold of the three cells are 0.241 V, 0.26 V, and 0.19 V for $11T_1$ SRAM cell, $11T_2$ SRAM cell, and $11T_3$ SRAM cell and for read operation are 0.239 V, 0.256 V, and 0.187 V, respectively. The power consumed by these cells is 327.9 nW for $11T_1$ SRAM cell, 104.45 nW for $11T_2$ SRAM cell, and 82.36 nW for $11T_3$ SRAM cell. And, the accessing time for read/write of all three SRAM cells is minimum for $11T_3$ SRAM cell which is 332 ps and 192 ps, respectively.

Keywords Low power · 11T SRAM · Static noise margin · Leakage current

1 Introduction

The load of processing using processors present on System on Chip has increased due to the huge popularity of Internet of Things. The devices with SoC integrated with artificial intelligence and machine learning make the devices more desirable and reliable. The external peripheral circuitry for any Static Random Access Memory (SRAM) bit cell constitutes of bit line, sense amplifiers, and decoders mainly. SRAM bit cell comprises various processors and each processor is relying on the internally

Y. Chopra (✉) · P. Mittal (✉)
Department of Electronics and Communication Engineering, Delhi Technological University, New Delhi 110042, India
e-mail: yogitachopra_2k20phdec04@dtu.ac.in

P. Mittal
e-mail: poornimamittal@dtu.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_43

present cache memory; these are processors that are required for the faster execution applications (Rawat and Mittal 2021a). Memory is already known to be one of the most crucial parts of any electronic system. However, a class of memory called the cache memory is even more crucial amongst the type of memories since it is the one working closely in synchronization with the central processing unit. It is a faster and smaller memory and is also exploited to store data often used by CPU. Its usage is inevitable as it is being used in System on Chip (SoC), IC fabrications, and also into personal computers. The main composition of cache memories is SRAM cells as cache memories are just a wide arrangement of various numbers of SRAM cells arranged in a major size array (Rawat and Mittal 2021b, 2022). Thus, to construct or form a reliable cache memory, it is a necessary consideration that SRAM cells must have these features, namely: higher stability for data retainment, lower dynamic and static consumption of power, and significantly lesser delays during both read and write (Islam and Hasan 2012). The new advancement in the industry can be towards nanometer technology feature size as it focuses on sleek designs with larger battery life that can accommodate larger needs. Also, by keeping with Moore's law in nanotechnology, the feature size of the nano electronic devices which are highly integrated is being marginally deduced (Krishna and Duraiswamy 2021). To keep up with the requirement for larger battery life and to minimize the consumption of power by the bit cell has become a major designing concern (Abbasian and Gholipour 2022). The major component to measure the total power consumed for an SRAM cell is calculated in terms of static power consumed by the cell as cell commonly executes in the hold mode (Lorenzo and Pailly 2020; Chiu et al. 2013). In this paper, Sect. 1 is the introduction that is basically providing the overview of the work being explained in later sections, Sect. 2 comprises the schematic comparison of three different SRAM cells, Sect. 3 demonstrates the observations and analysis carried out, and lastly, Sect. 4 is the final conclusion of this paper.

2 Schematic Representation

There is a huge variety of SRAM cells that are present in the literature, but in this article, we are focussing on the 11T SRAM cells (Lorenzo and Pailly 2020). In this work, three different 11T SRAM cells have been designed with CMOS technology at 32 nm technology node at a supply voltage of 0.4 V. The major problems occur when it is required to design SRAM cell for very low operating voltage for higher number of transistors which is VDD scaling. In this section, we have analyzed three SRAM cells, namely $11T_1$ SRAM cell (Chiu et al. 2013) which is a dual-ended cell and has different bit lines for reading and writing operations; similarly, $11T_2$ SRAM cell (Islam and Hasan 2012) is also a dual-ended cell, but here accessing is quite simpler as it has dedicated access transistors that are explained in Table 2 in Sect. 3 and lastly $11T_3$ SRAM cell (Lorenzo and Pailly 2020) which is single-ended SRAM cell, i.e. it has only one bit line for reading and writing operations.

In $11T_1$ SRAM cell (Chiu et al. 2013), accessing or basic operations is done via transistors M7, M8, M9, M10, and M11, and transistors MP1, MP2, MP3, MP4, M5, and M6 form the cross-coupled inverter pairs for the cell that forms the basic circuit of the bit cell. And, the cell has two bit lines BLB and BL to form the arrangement (Fig. 1).

In $11T_2$ SRAM cell (Islam and Hasan 2012), transistors M5, M6 M7, M8, and M9 are the access transistors for basic operations, while transistors M1, M2, M3, M4, M10, and M11 form the cross-coupled inverter pairs of the cell. Cell is having dual-line architecture, which increases the overall versatility of the cell in terms of the feasibility and stability of the cell during cell operations.

In $11T_3$ SRAM cell (Lorenzo and Pailly 2020), has a single bit line, and transistors M2, M3, M4, M9, and MP7 are the access transistors for basic operations, while transistors MP6, MP8, M1, MP5, M10, and M11 form the cross-coupled inverter

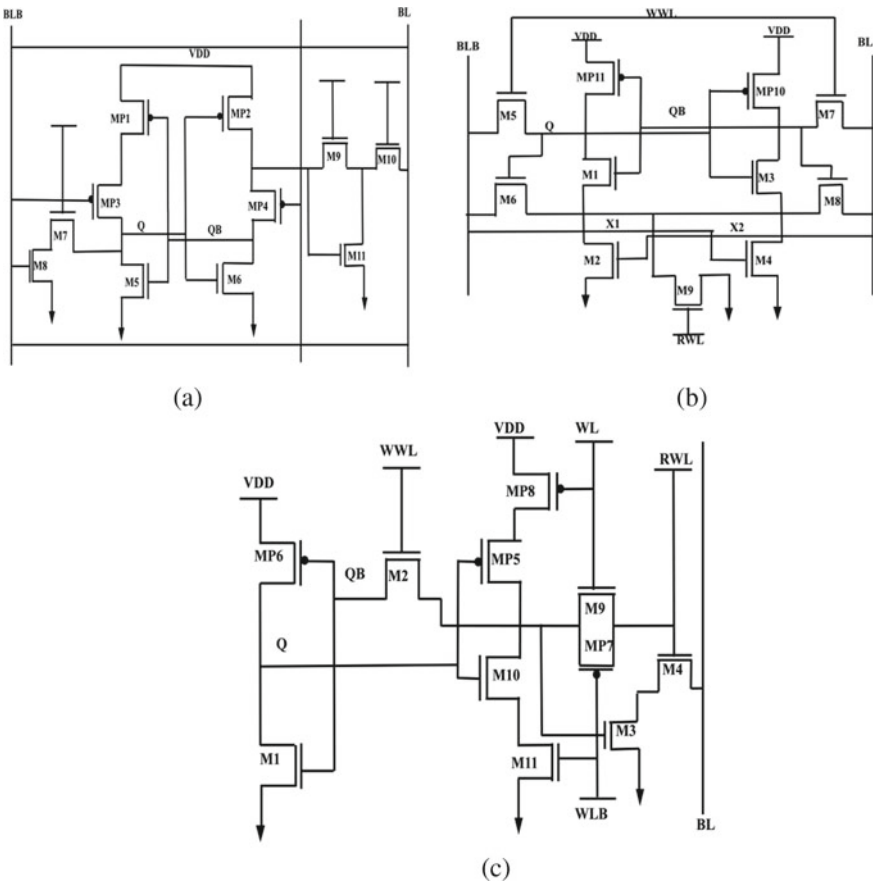


Fig. 1 Schematic diagram of **a** $11T_1$ SRAM cell **b** $11T_2$ SRAM cell **c** $11T_3$ SRAM cell

pairs of the cell. Further, the simulations and observation are explained in Sect. 3 of this paper in detail.

3 Simulation Results

LT-SPICE software has been used to simulate and analyze all the 11T SRAM cells at the 32 nm technology node that are described in Sect. 2 of this brief. All the SRAM cells are evaluated at an operating voltage of 0.4 V. The observed results are further explained in this section. All the transistors are series connected; it makes the resistance to increase; hence in order to reduce the equivalent resistance, the size of the other transistors needs to be decreased (Eslami et al. 2020). All the other transistors used here are having the minimum size possible, which makes the overall occupied area by the circuit much lesser (Jose and Balan 2016; Rawat and Mittal 2021c; Cho et al. 2020).

Simulation is performed under the stable temperature and constant power supply. The W/L ratios for the cross-coupled inverters and access transistors of different 11T cells that are being comparatively analyzed are shown in Tables 1, 2, and 3;

Here, these depict the *W/L* ratios required to design any bit cell corresponding to the transistors.

Table 1 Width and length values of $11T_1$ SRAM cell (in nm)

	Access transistors					MP1 and MP4	MP2 and MP3	M5 and M6
	M11	M9	M7	M10	M8			
Width (nm)	32	32	32	32	32	96	80	32
Length (nm)	32	32	32	32	32	32	32	32

Table 2 Width and length values of $11T_2$ SRAM cell (in nm)

	Access transistors					MP11	MP10 MP4 and M3	M1 and M2
	M5	M6	M7	M8	M9			
Width (nm)	48	48	32	32	32	32	32	32
Length (nm)	32	32	32	32	32	32	32	32

Table 3 Width and length values of $11T_3$ SRAM cell (in nm)

	Access transistors					MP8 and MP6	MP5	M1 and M11	M10
	M2	M3	M4	M7	M9				
Width (nm)	64	64	64	64	64	80	96	96	96
Length (nm)	32	32	32	32	32	32	32	32	32

3.1 Stability Analysis

The main aspect for SRAM cell is evaluated in terms of data stability during read, write, and hold operation which is quantified by Static Noise Margin (SNM) and is calculated and observed for the three different modes of operation (Rawat and Mittal 2021c; Cho et al. 2020; Mishra et al. 2019; Kumar and Ubhi 2019). SNM is the maximum amount of noise that can be provided to the storage nodes before that state stored is flipped and this applies to all three operating modes. SNM is demonstrated with the help of butterfly curves which are showcased in Fig. 2.

The observed values of the SNM for hold operation for the three analyzed SRAM cells are found to be 0.241 V for $11T_1$ SRAM cell (Chiu et al. 2013), 0.26 V for $11T_2$ SRAM cell (Islam and Hasan 2012), and 0.19 V for $11T_3$ SRAM cell (Eslami

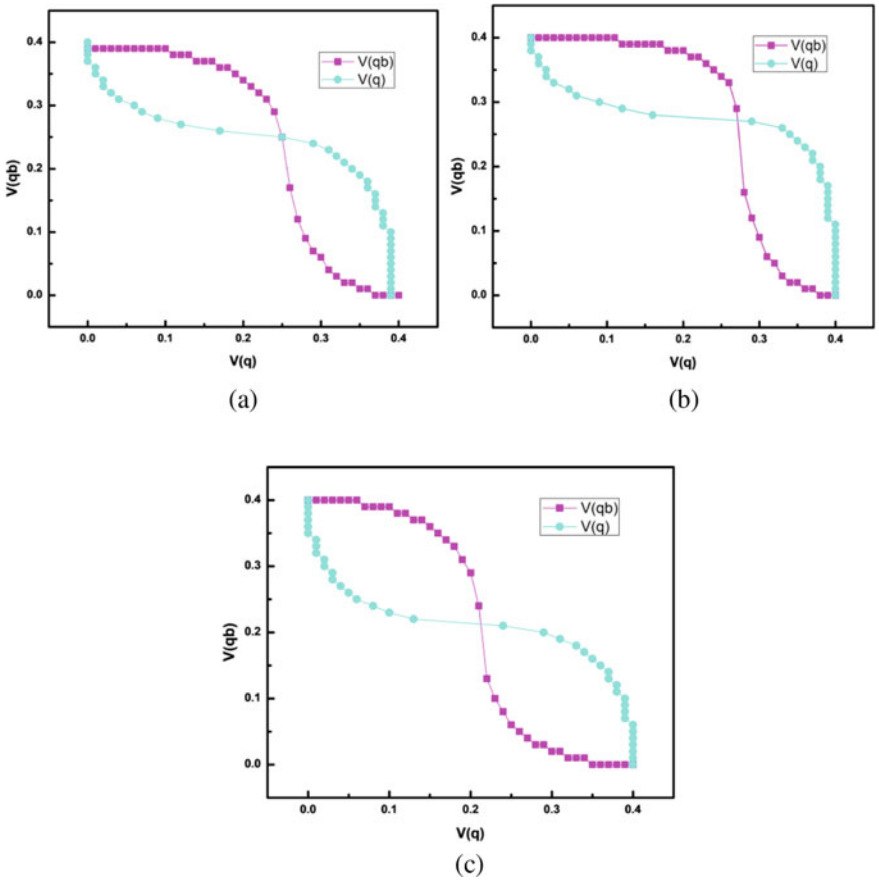


Fig. 2 HSNM and RSNM for **a** $11T_1$ SRAM cell **b** $11T_2$ SRAM cell **c** $11T_3$ SRAM cell

Table 4 SNM values for different SRAM cells

SRAM bit cells	WSNM (V)	RSNM (V)	HSNM (V)
$11T_1$ SRAM cell0.32	0.239	0.241	
$11T_2$ SRAM cell	0.34	0.256	0.26
$11T_3$ SRAM cell	0.29	0.187	0.19

et al. 2020) and the SNM values for all the three cells for read operation are 0.239 V, 0.256 V, and 0.187 V, respectively.

The obtained values of the SNM for hold and read operation and almost similar for the three cells while the value for the write margin for $11T_1$ SRAM cell is 0.32 V, $11T_2$ SRAM cell is 0.34 V and $11T_3$ SRAM cell is 0.29 V. Amongst these three cells, it is clear that the single-line 11T SRAM cell is efficient in performance as compared to the other two cells. This stability comparison clearly shows that single-port cell has better immunity towards noise (Mishra et al. 2019), i.e. noise tolerance capability of the cell in comparison with the other cell models (Kumar and Ubhi 2019). Specifically, if we talk about the accessing techniques of all three cell models, it is clearly observed that despite being single bit line cell, $11T_3$ SRAM cell is showing better stability during cell operations as compared to dual bit cells under same arrangement and supply (Table 4).

3.2 Power Consumption

The analysis confers the power consumption for different variants of the 11T SRAM cell. The analysis associated depicts that the power consumed is maximum in $11T_1$ SRAM cell, whereas it is minimum in case of $11T_3$ SRAM cell. The power consumed by $11T_1$ SRAM cell (Chiu et al. 2013) is 327.9 nW which is approximately three times as compared to the $11T_2$ SRAM cell (Islam and Hasan 2012) which is 104.45 nW and $11T_3$ SRAM (Lorenzo and Pailly 2020) cell 82.36 nW (Fig. 3).

The technology node used in the simulation is 32 nm, so the corresponding power consumption is also very low (Surana and Mekie 2019). Comparison of power consumed between the three cells is illustrated graphically in Fig. 3a.

3.3 Read and Write Access Times

Access time is the period required to read/write a particular bit into/from the cell. The total time required to read the data '1'/'0' from the cell is known as read access time and time required to write any specific data '1'/'0' from the cell is known as write access time (Divya et al. 2022; Mittal et al. 2022; Mittal and Kumar 2020). The different cell models have different read and write access delays; after this comparison, it is found that $11T_3$ SRAM cell has reduced time delay or access time

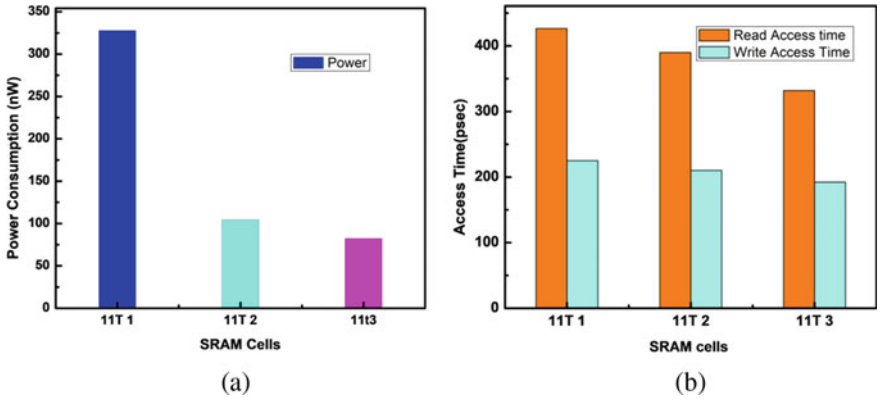


Fig. 3 **a** Power consumption in different 11T SRAM cells, and **b** read/write access time comparison in different 11T SRAM cells

delay which means that speed at which data can be read or written into the cell is faster than other two models. The read and write access time comparison for different variants of SRAM cell is demonstrated by Fig. 3b. The analysis associated depicts that read access time is maximum in $11T_1$ SRAM cell (Chiu et al. 2013) which is 426.5 ps and write access time is 224 ps. The read access time and write access time are minimum for $11T_3$ SRAM cell (Lorenzo and Pailly 2020) which are 332 ps and 192 ps, respectively. As per these observed results, we can conclude that amongst the three 11T SRAM cells, $11T_3$ cell (Lorenzo and Pailly 2020) depicts best performance in terms of SNM, read/write access time, and power consumption.

4 Conclusion

In this paper, three different 11T SRAM cells are compared on the basis of their accessing modes precisely in which two are dual-ended cells, while one is a single-ended memory cell. These cells are compared on parameters such; SNM, power consumption, and accessing time. After all the analyses, it is found that amongst the three cells, the single-ended cell is found to be better in performance in every aspect. All the three cells are evaluated on an operating voltage of 0.4 V with the technology node 32 nm. The obtained values of SNM for hold operations are 0.241 V, 0.26 V, and 0.19 V and read operations of the three cells are 0.239 V, 0.256 V, and 0.187 V for $11T_1$ SRAM cell, $11T_2$ SRAM cell, and $11T_3$ SRAM cell, respectively. Total power consumed by these cells is 327.9 nW for $11T_1$ SRAM cell, 104.45 nW for $11T_2$ SRAM cell, and 82.36 nW for $11T_3$ SRAM cell. And the accessing time for read/write of all three SRAM cells is minimum for $11T_3$ SRAM cell which is 332 ps and 192 ps, respectively.

References

Abbasian E, Gholipour M (2022) Robust transmission gate-based 10T subthreshold SRAM for internet-of-things applications. Semicond Sci Tech

Chiu W, Hu Y, Tu M, Zhao J, Jou S, Chuang C (2013) 40 nm 0.32 V 3.5 MHz 11T single-ended bit-interleaving subthreshold SRAM with data-aware write-assist. In: Symposium on Low Power Electronics and Design, 978-1-4799-1235-3/13

Cho K, Park J, Oh TW, Jung OK (2020) One sided Schmitt-Trigger based 9T SRAM cell for near threshold operation. IEEE Trans Circuits Syst I Regul Pap 67(5):1551–1561

Divya D, Mittal P, Rawat B, Kumar B (2022) Design and performance analysis of high-performance low power voltage mode sense amplifier for static RAM. Inter J Adv Elect Elect Eng 19(2):145–154. https://doi.org/10.15598/aeee.v19i2.3821

Eslami N, Ebrahimi B, Shakouri E, Najaf D (2020) A single-ended low leakage and low voltage 10T SRAM cell with high yield. Analog Integrat Circuits Sig Proces 105:263–274

Islam A, Hasan M (2012) Variability aware low leakage reliable SRAM cell design technique. Microelectron Reliab 52(6):1247–1252

Jose AA, Balan NC (2016) Static noise margin analysis of 6T SRAM cell. Artif Intell Evol Comput Eng Syst, 249–258

Krishna R, Duraiswamy P (2021) Low leakage 10T SRAM cell with improved data stability in deep sub-micron technologies. Analog Integrat Circuits Sig Process 109(1):153–163

Kumar M, Ubhi JS (2019) Design and analysis of CNTFET based 10T SRAM for high performance at nanoscale. Inter J Circuit Theory Appl 47(11):1775–1785

Lorenzo R, Pailly R (2020) Single bit-line 11T SRAM cell for low power and improved stability. IET Comp Digital Tech 14 (3):114–121. https://doi.org/10.1049/iet-cdt.2019.0234 www.ietdl.org

Mishra N, Mittal P, Kumar B (2019) Analytical modeling for static and dynamic response of organic pseudo all-p inverter circuits. J Comput Electron 18:1490–1500

Mittal P, Kumar N (2020) Comparative analysis of 90 nm MOSFET and 18 nm FinFET based different multiplexers for low power digital circuits. Inter J Adv Sci Tech 29(8):4089–4096

Mittal P, Rawat B, Kumar N (2022) Tetra-variate scrutiny of diverse multiplexer techniques for designing a barrel shifter for low power digital circuits. Microproces Microsyst J 90:104491

Rawat B, Mittal P (2021a) Single bit line accessed high-performance ultra-low voltage operating 7T static random access memory cell with improved read stability. Int J Circuit Theory Appl 49(5):1435–1449

Rawat B, Mittal P (2021b) A 32 nm single-ended single-port 7T static random access memory for low power utilization. Semicond Sci Technol 17:36095006. https://doi.org/10.1088/1361-6641/ac07c8

Rawat B, Mittal P (2021c) Analysis of varied architectural configuration for 7T SRAM bit cell. In: International Conference on Recent trends in Communication and Electronics (ICCE-2020)

Rawat B, Mittal P (2022) A reliable and temperature variation tolerant 7T SRAM cell with single bitline configuration for low voltage application. Circuits Syst Sig Process 41(5):2779–2801

Surana N, Mekie J (2019) Energy efficient single-ended 6-T SRAM for multimedia applications. IEEE Trans Circuits Syst II Express Briefs 66(6):1023–1027

ASIC Design of High-Performance MIPS Processor Using Aprisa

Naga Surya Manikanta Talla and Prakash Kodali

Abstract In this present era of electronics, processors play a critical role. Everyone needs processors with low cost, high operating speed, and low power consumption. There are a wide range of unfavorable risks in processors as technology advances to sub-micrometer levels. These risks (hazards) could result in timing, area, and power disruptions that depart from the desired proportions. As technology shrinks, new architectures and better EDA tools are needed to implement them. Our paper primarily focuses on finding solutions to some of these problems with a tool named Aprisa. Here we propose a high-performance 32-bit pipelined MIPS processor which is designed and implemented using the tools from Siemens EDA, namely Oasys (synthesis tool) and Aprisa (place and route tool). The proposed architecture can also detect data hazards caused by pipelining. The entire ASIC flow is done using TSMC 45 nm libraries, and a frequency of 3.6 GHz is achieved with a power consumption of 18.4mWatts and an instance area of 9091 sq.m. Also, chip utilization of 85% is achieved with good routability. Also, the same work is done in the 16 nm lower node and both results are compared. Compared with existing work, the area got improved by 65% and the frequency got almost doubled.

Keywords MIPS · Aprisa · EDA · TSMC · Oasys · Routability

1 Introduction

The heart of the computer is its processor. There are a large number of processors on the market. All those microprocessors are categorized into two types based on speed and power. One is a Reduced Instruction Set of Computers (RISC) and another is a Complex Instruction of Computers (CISC). CISC architecture is used to minimize

N. S. M. Talla · P. Kodali (✉)
Department of ECE, NIT Warangal, Warangal, Telangana, India
e-mail: kprakash@nitw.ac.in

N. S. M. Talla
e-mail: tnec21107@student.nitw.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_44

the instruction count. The advantage of CISC is each instruction can do a wide variety of operations but the drawbacks are complex system size and pipelining. Pipelining is not supported in CISC (even if it supports it, it is very difficult to implement). RISC allows fixed instruction length which supports pipelining, thereby improving the processor's performance (Topiwala and Saraswathi 2014). Smartphones, tablets, computers, and many other sophisticated gadgets all use RISC architecture. MIPS is designed based on RISC architecture. MIPS is used in routers and other such small computing devices (Venkatesan et al. 2019). Pipelining MIPS enhances throughput better but data hazard detection unit and forwarding are necessary to prevent hazards (Prasad and Prakash 2021). PPA (performance, power, and area) are crucial for every VLSI designer, and hence much research is conducted regularly to enhance them. The dominant force while aiming for higher frequency is dynamic power. The two techniques that are frequently used to reduce dynamic power are clock gating and multi-vt (Bharadwaja et al. 2015).

$$\text{Dynamic Power} = \alpha f C V_{\text{DD}}^2, \tag{1}$$

where α = switching activity factor, f = frequency, C = capacitance, and V_{DD} = supply voltage (Schimd 2012; Bharadwaja et al. 2015).

In this work, clock gating and multi-vt have been used to reduce power. EDA tools from Siemens EDA, namely Oasys and Aprisa, are used to get optimized frequency and area. This paper is divided as follows. The next section talks about the literature survey. Section 3 focuses on implementation methodology. Section 4 speaks about the results.

2 Literature Survey

This chapter presents a quick evaluation of modern literature on the design of MIPS architecture. Numerous works have previously been developed and presented with different frequencies, power, and area. Power is an important performance parameter, so its reduction is achieved by bypassing pipelining stages that cause unnecessary switching activity (Topiwala and Saraswathi 2014). For boosting processor performance, pipelining is incorporated and also a data forwarding unit is proposed which resolves the data hazards (Prasad and Prakash 2021). RTL to GDSII implementation of RISC is done and achieved a power of 0.53W, an area of 17,067.7584 sq.m in TSMC 180 nm by using Cadence Innovus (Aradhya et al. 2021).

Stalling and optimized forwarding units can be used to get high-performed pipelined processors (Dewangan et al. 2021). The processor won't go to the high impedance or unknown state, which in turn results in performance enhancement (Ashok and Ravi 2017). Clock gating is the process of preventing the clock signal from accessing the processor's various components. By preventing flip-flops or registers from changing their values, this signal from the clock lowers the leakage power (Joseph et al. 2009).

The author in Islam et al. (2006) suggested an energy-saving design that uses both architectural and circuit principles. In branch-specific applications, the advanced branch technique is also utilized to lessen stalls, which further minimizes energy waste.

3 Work Proposed

For the MIPS design, the entire ASIC flow beginning from RTL to GDSII is done. The following Fig. 1 (Aradhya et al. 2021) depicts the general flow of any ASIC. The work uses ModelSim and Oasys tools (from Siemens EDA) for the front-end design procedure.

3.1 Front-End Design

Architecture Design: The architecture (Prasad and Prakash 2021) of five stages of pipelined MIPS with data hazard detection and forwarding unit is shown in Fig. 2.

RTL Design: A 5-stage 32-bit pipelined MIPS processor is designed using Verilog HDL and the designed RTL is simulated with ModelSim HDL simulator.

The code should be written in such a way that it does not infer latches during synthesis. Following are the Verilog modules developed PC, ALU32bit,

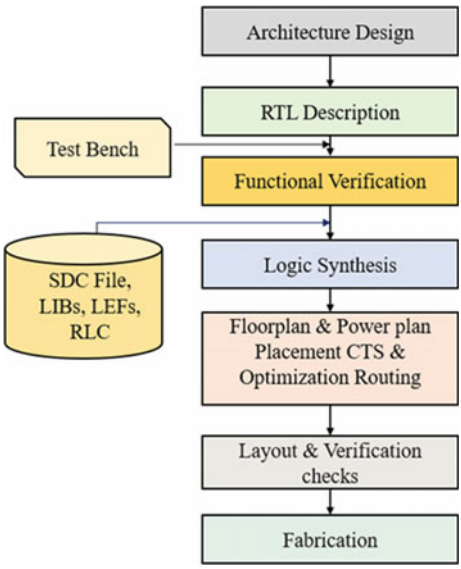


Fig. 1 ASIC flow of digital design

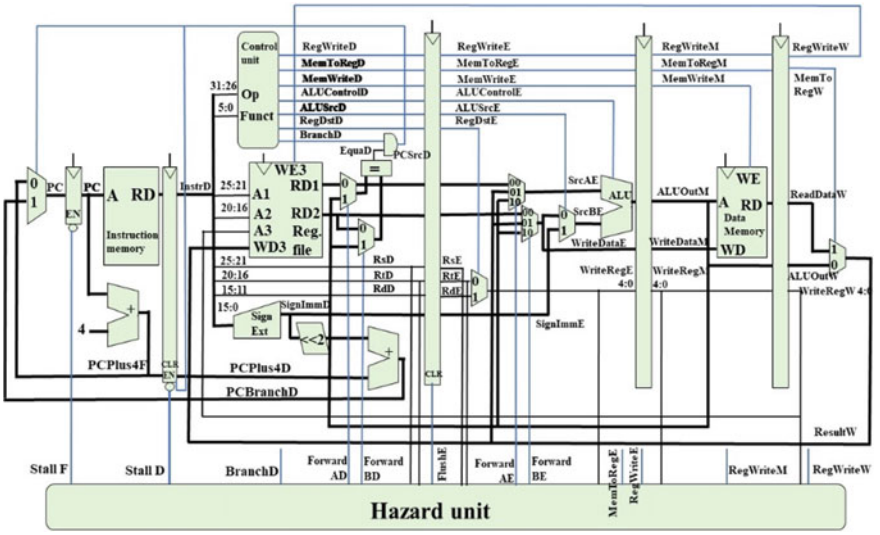


Fig. 2 MIPS architecture with hazard detection

ALUController, RegisterFile, Mux, PCAdder, BranchAdder, InstructionMemory, DataMemory, Controller, IF ID Reg, ID Ex Reg, Ex Mem Reg, Mem Wb Reg, Hazard Detection Unit, Data Forwarding, and SignExtend Unit.

RTL Simulation: Designed MIPS is thoroughly verified with a suitable test bench for arithmetic and logical instructions. Figure 3 shows the simulation results of a few instructions.

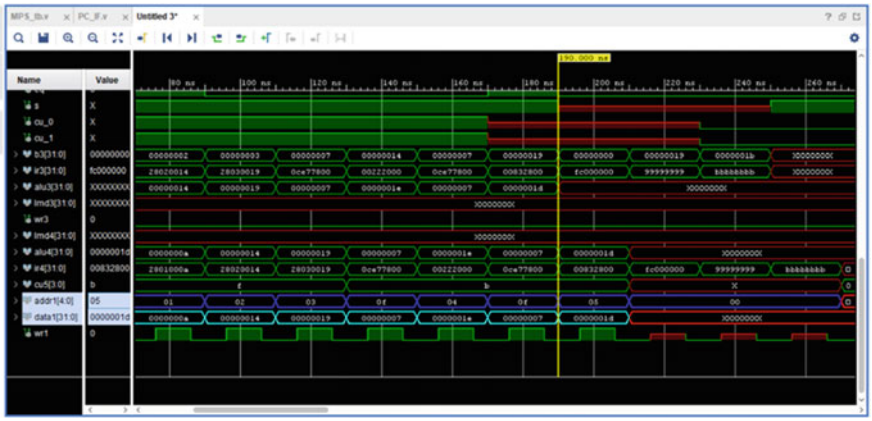


Fig. 3 Simulation results

Synthesis: This process generates a design netlist from logically verified synthesizable Verilog HDL codes by utilizing standard-cell libraries for the 45 nm CMOS technology provided by the TSMC semiconductor foundry. After validating the design's functionality, it is ready for synthesis if the code is synthesizable. In this work, the Oasys tool from Siemens EDA is used for synthesis. In any EDA tool, synthesis happens in three major steps: translation, optimization, and mapping. After technology mapping, Oasys provides a clock gating feature to reduce dynamic power, so in this work, it is enabled and got better power than existing works. Multi-bit optimization is also enabled to reduce area and power. SDC File, Libraries (.libs), LEF, and RLC File are the inputs for the synthesizer.

3.2 Back-End Design

The gate-level netlist's physical implementation is referred to as back-end design. The Aprisa tool is utilized in this work for the place and route steps. It is a full-featured back-end design tool from Siemens EDA.

Broadly, the following steps are performed:

Import Design: The gate-level netlist obtained from the logic synthesis tool, .lef file, .lib file, tech file, rlc file, and DEF from Oasys which are necessary for physical implementation are imported here. Two scenarios are considered, namely func_ss_ 0p85 and func_ss_0p95.

Floorplan: The floor plan is a critical step in physical design because it determines the quality of design implementation. A bad floorplan can create issues like congestion, timing violations, and blowing up the area. Aspect ratio, core-to-boundary distance, and core utilization must all be defined for the floor plan. The chip's form is determined by the aspect ratio. The height/width of the core is used to define the aspect ratio. For this MIPS project, the aspect ratio of 1 is used, making the core seem square. Core utilization of 85% is chosen. Flipped and abutted row array is made based on DEF from Oasys. Here, M-10 has been selected as the maximum metal layer that can be employed in the design.

Power plan: Power strips and rails are created around the core area which carries VCC and GND. Here, the widths of the metal layers for these rings can also be specified but these are mostly specified by foundry people based on technology. For horizontal and vertical directions, metals with odd and even numbers are employed, in this work, metal-9 is used for the horizontal direction while metal-10 is used for the vertical direction. Power stripes are routed in higher metal layers as they are wider and decrease resistance. Thereby, the IR drop is low. Power rails are routed in metal-1. Metal-10's vertical stripes are 1.6 um wide, whereas Metal-9's horizontal stripes are 0.8 um wide.

Placement: The objective of placement is to place all unplaced cells in legal positions to achieve the best possible timing and the least congestion. Placement determines to interconnect length and hence interconnect delays. Interconnect delays can consume as much as 75% of the clock cycle. So, good placement can substantially improve the performance of a circuit. Inputs to the place stage are Netlist from Oasys, floor plan and power plan, logical and physical libraries, and constraints. The Aprisa reads in the Tech LEF file and places the standard cells in the rows of the floorplan as per the rule's information contained in the LEF. The standard cells are placed according to the netlist. Following the placement of standard cells, mainly timing violations and congestion overflows are checked. As the clock tree is not built yet, there is no skew and hold time checking. Any EDA tool will perform the major optimization steps like buffer insertion and deletion, vt-swapping, netlist restructuring, and resizing gates while placing cells.

Clock Tree Synthesis: Actually in the place stage, clocks are not yet built. Hence, tool assumes an ideal clock. So, clock nets have infinite drive capability, zero rise/fall transition time, latency, and skew at the place stage. The actual clock tree is built here. The objective is to do Buffer/inverter insertion to distribute the clock evenly to all sequential elements by minimizing skew. The tool uses a skew group approach to balance the skew while generating the clock tree. A skew group is a type of clock network designed to reduce skew between a set of pins. Typically, SDC constraints are used to define the various clocks and the clock pins they drive. For CTS to form a tree, high-drive strength buffers, and inverters are supplied as input. NDR, or non-default routing, is used to decrease crosstalk. Make sure the latency and skew are acceptable during CTS, failing to do so will affect timing during subsequent phases. Furthermore, 45ps of uncertainty are added to account for clock jitter.

Post-CTS Optimization: Although very identical optimization strategies to placement are used here, the propagating clock is real rather than ideal. Hold timing is taken into account for optimization in this case, unlike placement optimization.

Routing: The routing stage establishes the precise routes for connecting standard cells, macros, and I/O pins. Routing aims to complete wiring with acceptable timing, a minimum number of DRCs, and wires that are both short and have few vias. Three main processes make up the routing process: global routing, track assignment, and detail routing. Global cells, or gcells, are the units used to categorize the area that needs to be routed. It chooses tile-to-tile pathways for the nets while also attempting to minimize the length without creating any physical connections. When assigning tracks, the tool uses the global routed layout to determine which tracks should be used for each net. It doesn't adhere to the physical DRC guidelines. The router employs the plan created during the global routing and track assignment phases to lay metals to connections. Complete time-driven and DRC-aware routing are implemented here. Factors affecting SI like crosstalk (delay and noise), IR drop, antenna effect, and electromigration are also considered during routing for the more realistic case.

4 Results and Analysis

Results mainly contain information regarding frequency, area, and power at the synthesis stage and route stage. Table 1 shows a detailed analysis of the area and power report generated from the synthesis tool. As data memory consists of 32-bit locations each with a 32-bit wide, it stood as the highest dominating block in terms of power and area. The processor block which consists of the register file, program counter (PC), instruction memory, and all multiplexers is the second dominating block in terms of power consumption and area occupied. Multi-bit optimization has enabled during synthesis flow so got better power and area. Generally, multi-bit optimization reduces switching power significantly.

From synthesis to route, power is optimized better which has been shown in Table 2. Table 2 also shows the other metrics that are achieved at the route stage which is the final design step in the ASIC flow. The proposed design has achieved a frequency of 3.6 GHz and an area of 9081 sq. um. Especially leakage power has been reduced in p & r flow by using the multi-vt approach and clock gating which makes total powerless. The utilization metric says how efficiently we are using the core area. At the route stage, 85% utilization is attained with 0 violations, demonstrating high routability.

Table 3 shows the comparison of this proposed work at 45 nm with the existing work which is also done at 45 nm. The area was drastically reduced (almost reduced by 65%) and the frequency nearly doubled. Two of three metrics, i.e., frequency, and area, are improved. Coming to power, it is acceptable because, with increasing

Table 1 Synthesis results

Instance name	Area (sq.m)	Total power (mW)
Processor	2800	20.01
IF stage	20	0.02
ID stage	36.18	0.108
Ex stage	336.76	14.32
Mem_Wb stage	144.7	0.407
Hazard detection	118.24	2.10
Forwarding unit	0.53	0.09
Data memory	4741.18	1.08
ALU	578.28	1.37
Control unit	163.86	0.0094
Total	8939.73	39.5

Table 2 Route stage results

Frequency	Area	Leakage	Switching	Internal	Total Power
3.6 GHz	9091 sq.m	0.03 mW	5.21 mW	13.17 mW	18.41 mW

frequency, power increases linearly (evident from Eq. 1). Also in the existing work, a 16-bit processor is only implemented, whereas in the proposed design 32 bit with a data hazard detection unit and forwarding unit is incorporated, so obviously circuit logic itself is more so a little bit of power increase will be there. To make sure that the proposed work and the tools are better in lower technology also, the same work is carried out in lower technology 16 nm. For the smaller node 16 nm, this implementation has produced much better results across the board, including performance and area. Lower technology node needs to give high frequency and low area. But it is at the cost of sacrificing power. At lower nodes, the leakage power dominates. The final layout which is generated after all the design steps and checks is shown in Fig. 4. Overall, the performance of the EDA tools Oasys and Aprisa was good in terms of outperforming previous work.

Table 3 Performance comparison

Parameter	Venkatesan et al. (2019)	Proposed work	Proposed work
Technology (nm)	45	45	16
Frequency	2.08	3.6	12.5
Area (GHz)	26,475	9091	785
Total power (mW)	2.6	18.4	31.02
Processor width (bit)	16	32	32
Capable of hazard handling	no	yes	yes

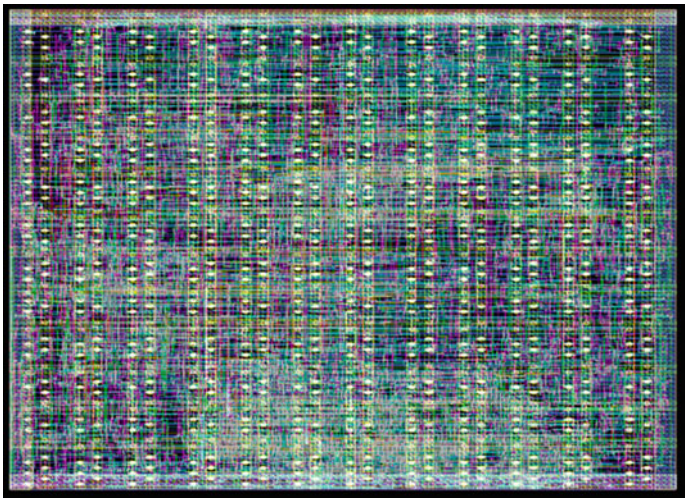


Fig. 4 Layout of the designed processor

5 Conclusion

Using Oasys and Aprisa EDA tools, a 32-bit MIPS, which is capable of handling data hazards is successfully implemented from RTL to GDSII with better results compared to existing works. In the future, the same design can be done in lower technologies to seek still better performance, power, and area (PPA). Built-in self-test (BIST), a testing system that assures the processor's proper functionality, can be added to this work to improve further.

References

Aradhya HVR, Kanase G et al (2021) RTL to GDSII of Harvard structure RISC processor. In: 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, pp 1–4. https://doi.org/10.1109/CONECCT52877.2021.9622735

Ashok A, Ravi V (2017) ASIC design of MIPS based RISC processor for high performance. In: 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), Chennai, India, pp 263–269. https://doi.org/10.1109/ICNETS2.2017.8067945

Bharadwaja KR, Teja MN, Neelima K et al (2015) Advanced low power RISC processor design using MIPS instruction set. In: 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, pp 1252–1258. https://doi.org/10.1109/ECS.2015.7124785

Dewangan GK, Prasad G, Mandi BC (2021) Design and implementation of 32 bit MIPS based RISC processor. In: 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, pp 998–1002. https://doi.org/10.1109/SPIN52536.2021.9566007

Islam S, Chattopadhyay D, Das MK, Neelima V, Sarkar R (2006) Design of high speed pipelined execution unit of 32-bit RISC processor. In: India Conference Annual IEEE, pp 1–5

Joseph N, Sabarinath S, Sankarapandiammala K (2009) FPGA based implementation of high performance architectural level low power 32-bit RISC core. In: International Conference on Advances in Recent Technologies in Communication and Computing Annual IEEE, pp 53–57

Prasad KK, Prakash V (2021) Designing and implementation of 32-bit 5 stage pipelined MIPS based RISC processor capable of resolving data hazards. In: 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, pp 1–6. https://doi.org/10.1109/ICMNWC52512.2021.9688435

Schimd L (2012) Processor implementation in VHDL. University of Ulster, pp 1–20

Topiwala MN, Saraswathi N (2014) Implementation of a 32-bit MIPS based RISC processor using Cadence. In: 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, pp 979–983. https://doi.org/10.1109/ICACCCT.2014.7019240

Venkatesan C, Sulthana MT, Sumithra MG, Suriya M (2019) Design of a 16-bit Harvard structure RISC processor in cadence 45 nm technology. In: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, pp 173–178. https://doi.org/10.1109/ICACCS.2019.8728479

A Proficient Multi-level Data Analytic Suite for Ascertaining Preliminary Gestational Hazards Associated with Its Influences

G. Bhavani and C. Jeyalakshmi

Abstract The basis of sustaining positive health outcomes begins from womb in the current era is safeguarding pregnancy from problems including maternal mortality and morbidity. A pregnancy with a hypertensive issue and diabetes is regarded as having a very high risk since it may result in long-term diseases such cardiovascular disease, obesity due to fetal growth restriction, anemia, and thyroid problems. Low birth weight, early delivery, and other serious neonatal disorders can affect babies who go untreated throughout pregnancy. Advances in Intelligence system and health data analytics allow for the early diagnosis of pregnancy concerns, reducing risks. This proposed effort includes the prediction of early gestational problems like diabetes and hypertension as well as the analysis of the risk variables that may be present. In our suggested work, multi-level data analysis is employed to better understand the health of pregnant women throughout the first trimester. In this data analysis suite, summarizing of data with deep insights is done at data preparation using exploratory data analysis (EDA) to look at the core pattern of data with its properties. The KNN algorithm has been used to preprocess data. Pregnancy risks including gestational hypertension and diabetes are identified before 13 weeks of pregnancy machine learning techniques. In order to comprehend analytical data with graphical representations, visual analytical approaches have been utilized. As a consequence, 97.18% accuracy is achieved.

Keywords Preeclampsia · Machine learning classifiers · Ensemble learning · Gestational diabetes · Gestational hazard

G. Bhavani (✉)
Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India
e-mail: bhavanisomaselvakumar@gmail.com

C. Jeyalakshmi
Department of Electronics and Communication Engineering, K. Ramakrishnan College of Engineering, Tiruchirappalli, Tamil Nadu, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_45

1 Introduction

In recent years, one of the most promising fields of study has been the analysis of healthcare data. Now, healthcare data sets can incorporate genetic and sensor data in addition to electronically recorded clinical data. This amount of high-e collected data cannot be managed manually, and the accuracy that results is poor.

Machine learning provides a wide range of statistical approaches (Beernink et al. 2022) and algorithms for the analysis of healthcare data. Healthcare data analysis using machine learning provides useful insights for forecasting and decision-making. Physicians will benefit from this practice in a number of ways, including better patient care and treatment, increased patient knowledge of their health state, and improved access to vital information.

The dangers of pregnancy can be decreased with the help of supportive and caring gestational care. Two important factors that might significantly increase your risk of pregnancy difficulties are age and overall health. Pregnancy risk has a substantial impact on both mother health and fetus development. Untreated pregnancy issues are seldom the primary cause of chronic disorders. Early diagnosis of pregnancy issues and their strategic linkage with risk factors are essential for a successful pregnancy. The following are the consequences of difficulties during pregnancy:

Instantaneous impact:

1. Preterm birth
2. Pre- or post-eclampsia
3. A small birth weight
4. Metabolic inconsistencies
5. Congenital conditions
6. A limit on fetal development
7. Mortality from stillbirth
8. Birth via C-section.

Long-term impact: an iron deficit.

Long-term effects of diabetes and cardiovascular disease (Sharma et al. 2021). Pregnancy risk can be low medium and high. Pregnancy-related complications on both maternal and fetal might occasionally have a significant influence on gestational risks. Risk factors include previous pregnancy status of mother including renal disease, diabetes, and hypertension, as well as behavioral and biological indicators.

2 Methodologies

The work module is depicted in detail in Fig. 1. Three modules—Data collection and pattern analyzing, Training of Model, and Evaluation—make up the total study. To comprehend the data, data exploration was done, and prediction analysis using model training. Models must be assessed if analysis findings are to be more accurate.

2.1 Data Summary

Data Source

Both publicly available repositories and publications that have been submitted contain all of the data and related metadata.

The data set used in this proposed work consists of the prenatal clinical visit status of 400 pregnant women. Which includes 16 risk factors as column values such as age, height, weight, BMI, sysbp, diabp, hb, pcv, tsh, platelet, creatinine, plgf:sflt, SEng, cysC, pp_13, glycerides. Among all variables of the dataset 16 are identified as numerical data, other two are categorical data. Out of 400 pregnant women 105 are recognized as risky mothers since they have both complications of HTN and diabetes.

2.1.1 Data Cleaning and Transformation

The preparation and cleaning of data includes handling NAN and missing values. There are other methods employed, such as dropping, filling, imputation, and interpolation. Since many characteristics exhibit continuous and high levels of similarity,

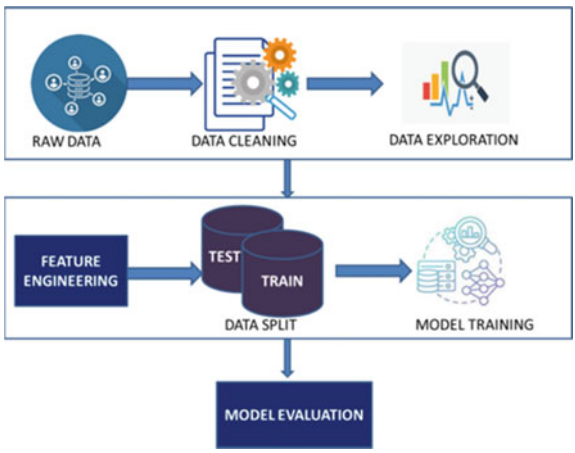


Fig. 1 Work flow diagram

missing values have been imputed using K-nearest neighbor (KNN) imputation. Euclidean distance measurements are used to establish similarity. The parameter neighbor may be supplied with a range of numerical values. A parameter neighbor of 2 has been given to complete the gaps in this work with the most similar values.

It is feasible to clean data using a number of data transformation techniques, including aggregation, normalization, and standardization, selection of feature, discrepancy, and conceptual grading construction. In this suggested effort, standardization aims at reducing all features to a same uniform scale in the range of values. Z-score normalization: This technique scales the data to have a mean of 0 and a standard deviation of 1. It is also known as standardization.

The formula for *Z*-score normalization is:

$$\text{Normalized} = (x - \text{mean}(x))/\text{standard deviation}(x) \quad (1)$$

2.1.2 Data Examination

EDA is a technique for accurately comprehending different facets of data (Philip et al. 2022). Data should be tidy, without repetition, missing values, NANs, or null values while being explored. The main purposes of EDA are to recognize crucial values and eliminate noise. The development of highly accurate models requires EDA (Philip et al. 2022). EDA may also be used to show how different variables are related to one another (Philip et al. 2022). Understanding the data, cleansing the data, and figuring out how variables relate to one another are all phases in the EDA process (Kaya and Kuncan 2022). Visual representations like probability distributions and histograms may be made using the EDA approach. Used for dimensionality reduction as well as cluster building. Three different forms of analysis can be employed in EDA.

Univariate Analysis:

A single variable in a data collection can be analyzed and described using a statistical technique called univariate analysis. It entails looking at the variable's distribution and its statistical properties.

Bivariate Analysis:

Usually this statistical analysis is performed to discover the relationship between two variables most probability between target and input variables to know the dependency (Philip et al. 2022).

Analysis of Correlation:

Correlation analysis examines connections between quantitative or categorical variables (Morimune et al. 2022; Kaya and Kuncan 2022; Philip et al. 2022). It evaluates how well the variables are relates to one another. The study of correlation involves examining the connections between various variables. Future predictions can be

made with the use of correlation analysis. The below mentioned Corr computation is used to calculate the correlation of target and input risk factors:

$$\text{Corr}_{xy} = \frac{\sum(\mathbf{Sx} - \overline{\boldsymbol{x}})(\mathbf{Sy} - \overline{\boldsymbol{y}})}{\sqrt{\sum(\mathbf{Sx} - \overline{\boldsymbol{x}})^2 \left(\boldsymbol{y}_j - \overline{\boldsymbol{y}}\right)^2}} \tag{2}$$

where

Corr_{xy}—the correlation coefficient.

Sx—*X*'s Sample value.

$\overline{\boldsymbol{x}}$—Average of *x*.

Sy—*Y*'s sample value.

$\overline{\boldsymbol{y}}$—Average of *y*.

The positive, negative, and neutral categories for correlations are possible as a result.

While performing analysis with clinical or health care data, positive and negative correlation outcomes can be taken into consideration because a decrease in a risk factor may result in an complications in consequences. Table 1 shows 1Relation between target values (output features—diabetes and hypertension) and risk factors.

Multidimensional Analysis

A statistical technique called multivariate analysis is used to examine multifactors of same data set toward the target HTN and diabetes.

Table 1 Relation between target value and risk factors

HTN		Diabetes	
Age	−0.044316	Age	0.093013
Height	0.035534	Height	−0.054274
Weight	−0.046537	Weight	0.003723
BMI	−0.038557	BMI	0.082728
sysbp	−0.048266	sysbp	0.060266
diabp	0.035519	diabp	0.071795
hb	0.011615	hb	0.029939
pcv	0.022120	pcv	0.017576
tsh	0.050729	tsh	−0.043162
Platelet	0.014776	Platelet	0.013614
Creatinine	−0.007294	Creatinine	0.002937
SEng	0.005352	SEng	0.037157
cysC	−0.071971	cysC	0.043899
pp_13	−0.017201	pp_13	0.081698
Glycerides	−0.008021	Glycerides	−0.000540

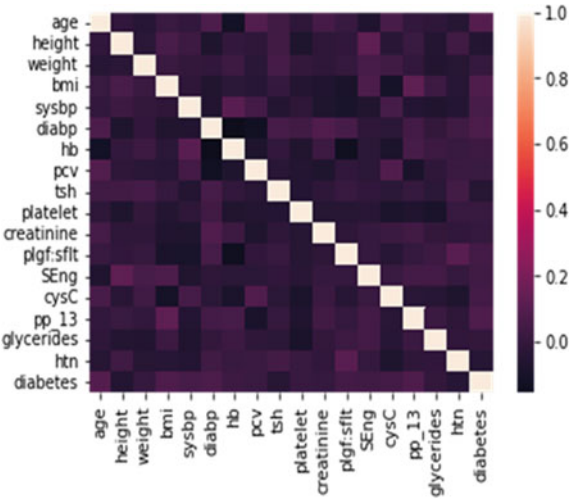


Fig. 2 The Spearman's rank correlation coefficient

Risk Warmth Map

Heat maps are an effective tool for understanding and analyzing vast amounts of data (Philip et al. 2022). To examine the link between variables, it uses a representation based on color coding. Figure 2 displays a heat map with correlation values of pregnant data set.

Dendrograms

Dentrograms can be used to identify clusters by comparing the characteristics of two items (Philip et al. 2022). In real-world data, it is challenging to identify enough clusters to create a systematic model. As a consequence, to analyze item similarity by calculating their distance, dendrogram tree is utilized with has hierarchical cluster kind of structure. Every single data point is considered as an independent cluster in this method. The data pair with minimum distance is then created for each cluster. Up until there is just one cluster left, this process is repeated. Cluster hierarchies are visualized using dendrograms.

The distance measure among two data points is calculated as

$$\text{dis} = \sqrt{\left((x2-x1)^2 + (y2-y1)^2\right)} \tag{3}$$

where dis—Distance of clusters (with high similarity).

(x, y)—cluster's coordinates.

To build a cluster, a matrix concept with distance values has been made for each data point. The distance matrix should be updated with the clusters and the shortest distance needed to obtain a single cluster.

3 Model Training

In the suggested work, the training and test data sets can be divided from the full data set in the ratio of 80:20. The model is subjected to a variety of machine learning classifiers to obtain the best outcome as high accurate prediction. They are listed below.

3.1 SVM

The acronym SVM stands for support vector machine. Both regression and classification problems benefit from the usage of this technique in supervised machine learning algorithms. Two classes are typically separated as output using SVM. The construction of an adequate hyper-plane allows SVM classification to be done. A hyper-plane is just a line that divides two data points by passing across them. Positive and negative points can be found on each side of the hyper-plane. A single data set can have many hyper-planes generated for it.

Two parallel lines to the hyper-plane are the boundary, also known as the margin or decision boundary. The closest positive and negative data points are paralleled by a line. The marginal distance is the separation between the hyper-plane and the margin. SVM's main objective is to increase the precision of model creation. By calculating the shortest marginal distance, the objective was accomplished. Linearly and non-linearly separable data are two kinds of data that may be separated. Real-world data is utilized in this proposed study to apply kernel function analysis to the training of SVM-based models. By enhancing dimensions and accuracy, the kernel function is employed (Launbo et al. (2022)). Kernel functions are mostly employed to convert lower-dimensional data to higher-dimensional data. The following kernel parameters are used in this study, and greater dimensions are found using other mathematical formulae. Radial Basis Function (RBF) Sigmoid Kernel Polynomial Kernel Linear Gaussian.

3.2 Logistic Regression

This machine learning approach works well with values of labeled data that are known. Three categories of logistic regression methods exist:

1. Binary—When there are only two potential results, such as yes or no, this sort of regression is utilized.
2. Multinomial—When there are numerous outcomes.
3. Ordinal—This model was employed to forecast outcomes that were ordered.

The equation below is used to determine the logistic function, which is used for classification (Morimune et al. 2022).

$$\text{prob} = \frac{1}{1 + \left(e^{-(\beta 0 + \beta 1 x)}\right)} \tag{4}$$

Which problem is probability result for the sigmoid function with independent variable x and slope 1 and intercept 0.

3.3 Bayesian Classifier

A classification model based on probabilistic machine learning is known as a Naive Bayes classifier. The core of the classifier is the Bayes theorem. Depending on the issue, several Bayes classifier approaches are employed.

- Multinomial when the model's output is discrete, Naive Bayes is used.
- Naive Bernoulli Bayes: For binary vectors, the binomial model is helpful.
- Gaussian Naive Bayes: This technique assumes a normal distribution for the feature data.

The following is the Bayes theorem equation:

$$P(y|X) = \frac{\mathrm{P}(X|y)\mathrm{P}(y)}{\mathrm{P}(X)} \tag{5}$$

X: feature range probably from 1 to n.
Y: target.
Function P () to determine probability.

3.4 Ensemble Boosting Classifier

After training the model with various classifier algorithms and evaluating the accuracy value, the best training algorithm was identified and used in this proposed work. A streamlined recurrent gradient boosting library called XGBoost was created with the goals of being extremely effective, adaptable, and compact. It uses the gradient boosting framework to carry out AI calculations. With the help of XGBoost, you may quickly and accurately handle a range of data science problems using an equal tree boosting method also known as GBDT or GBM. XGBoost is a flexible and incredibly accurate implementation of gradient assistance that exceeds the limits of computing power for boosted tree algorithms. It was created primarily to accelerate the execution of AI models and computational speed. Trees are underlying equal in XGBoost as opposed to successively in GBDT. It uses a level-by-level approach, comparing gradient values and using these fractional sums to evaluate the character of parts at each potential split in the preparation set.

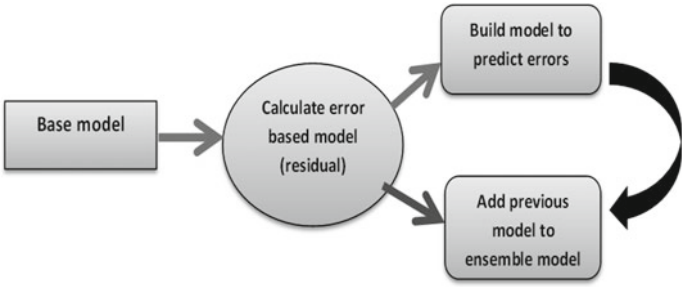


Fig. 3 XGBOOST implementation steps

XGBoost builds new models and combines them to form an ensemble model. It first constructs a basic model and computes errors for each finding in the data set. Then create a new model to forecast those errors (residuals). Then, to the new ensemble model, add predictions from the previous outcome. XGBOOST workflow steps are shown in Fig. 3.

It is a combination of multiple classification models that is mostly beneficial for multiclass classification. The basic model's construction is depicted below.

Calculate residuals by subtracting the dependent variable from the data frame's average of dependent variables.

$$R = x - \mu \tag{6}$$

where

R = residuals x = independent variable μ = mean of input features.

Then, calculate the similarity weight (SW) using the equation below.

$$\text{SW} = \sum\left(R^2\right) / \sum(p(1-p)\Big) + \lambda \tag{7}$$

Similarity weight has been calculated for left, right child, and root of constructed decision tree. Then gain () is calculated using

$$\text{Gain}\ () = \text{SW (left\_child)} + \text{SW (right\_child)} - \text{SW (root)} \tag{8}$$

The calculated gain value is compared against each column to see highest impact on prediction. In post pruning lower gain values has been pruned or deleted. If the cover value is less then gain then it can be pruned.

The cover value is computed as

$$\text{Cover\_Value} = \left(\sum(p(1-p)\right) \tag{9}$$

The following calculation was used to construct the basic model as initial step of boosting algorithm.

$$B\_V = (\text{Prop}/1 - \text{Prob}) \tag{10}$$

Sigmoid value has been calculated as

$$\text{Value}\ (V) = B\_V + (\text{learning\_Rate} + G) \tag{11}$$

Sigmoid function is applied to find t new prob value as follows:

$$\sigma\ (t) = 1/\ (1 + \exp\ (-V)) \tag{12}$$

Repeat above calculations till get minimum residual value, finally calculate

$$\sigma \left\{ \sum (\sigma\ (\text{ti}) \right\} \tag{13}$$

where $i \leq 0 \geq$ n: denotes iteration of each tree. In the above calculation most appropriate value for β is zero.

F score is calculated (shown in Fig. 4) using Gain, Cover Value, Similarity Weight. Gradient boosting decision tree construction will aid in determining the importance score of features, which defines how the feature will be useful in decision tree construction.

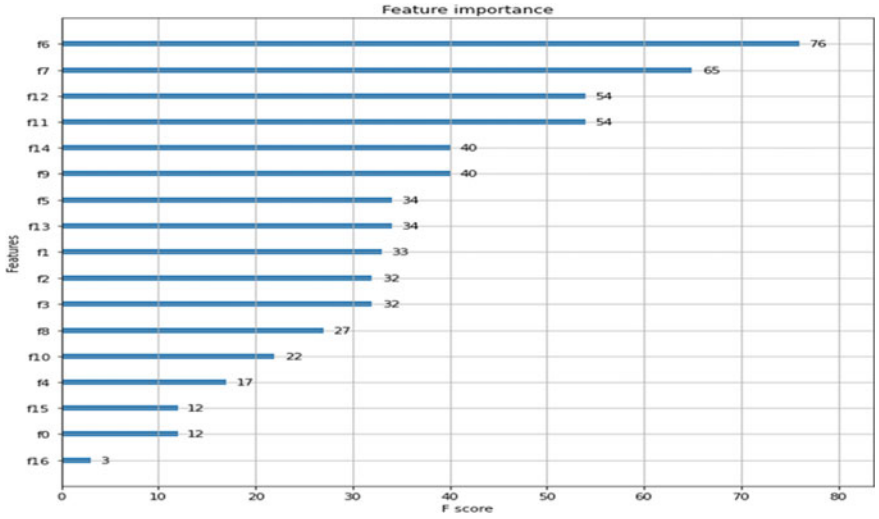


Fig. 4 Feature score for HTN

Table 2 Classification accuracy

Classifier	Outcome accuracy value (%)
SVM (kernel parameter:Linear)	58.475
SVM (kernel parameter:'sigmoid')	75.948
SVM (kernel parameter:'rbf')	86.125
SVM (kernel parameter:'poly')	95.16
Logistic classifier	57.55
Naïve Bayes classifier	58.735
Boosting algorithm	97.1082

4 Model Evaluation

SVM, logistic and Bayes classifiers, and ensemble classifier (XGBoost) were just a few of the classifiers used to train the model. The XBoost approach will be more accurate than other methods since real-world data is not linear. The results of implementing SVM with various kernel settings are displayed in Table 2.

5 Conclusion

To improve the efficiency of gestational risk likelihood at starting state, data pattern has been examined using EDA which can be done using python profile reader package, which aids in choosing the appropriate machine learning classifier method for model training. Comparing XGBOOST gradient classifiers to other classifier methods, they have a high accuracy of 97.18%. The information can be utilized to understand the connection between risk factors of blood sample collected during less than 13 weeks gestational period, the research proposal highlights the elements that significantly has impact on both risk such as gestational HTN and diabetes during the first trimester. Visual data analytics approaches are used to create a graph with the corresponding frequency that shows the distribution of those strongly related variables or components. One sign of pregnancy difficulties that is closely connected to BMI (shown in Fig. 5) is that it is above 24.9 for 38% of identified high-risk moms, as shown in the distribution map. In future this work can be expanded with multi test results of pregnant women like blood urine MRI and ultrasound for better prediction of risk as early as possible to ensure better treatment.

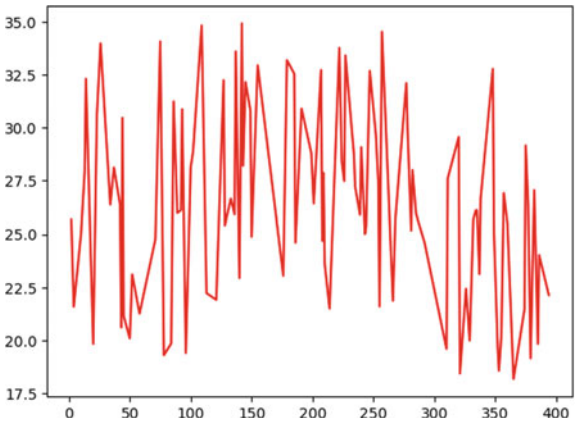


Fig. 5 BMI distribution of high risk mother

References

Azmi J, Arif M, Nafis T, Alam A, Tanweer S, Wang G (2022) A systematic review on machine learning approaches for cardiovascular disease prediction using medical big data. Med Eng Phys 105:103825

Beernink RHJ, Zwertbroek EF, Schuitemaker JHN, Cremers TIFH, Scherjon SA (2022) First trimester serum biomarker discovery study for early onset, preterm onset and preeclampsia at term. Placenta 128:39–48. ISSN 0143-4004

Fiaidhi J (2020) Envisioning insight-driven learning based on thick data analytics with focus on healthcare. IEEE Access 8:114998–115004. https://doi.org/10.1109/ACCESS.2020.2995763

Kaya Y, Kuncan F (2022) A hybrid model for classification of medical data set based on factor analysis and extreme learning machine: FA + ELM. Biomed Signal Process Cont 78:104023. ISSN 1746-8094

Launbo N, Davidsen E, Granich-Armenta A, Bygbjerg IC, Sánchez M, Ramirez-Silva I, Avila-Jimenez L, Christensen DL, Rivera-Dommarco JA, Cantoral A, Nielsen KK, Grunnet LG (2022) The overlooked paradox of the coexistence of overweight/obesity and anemia during pregnancy. Nutrition 99–100:111650. ISSN 0899-9007

Morimune A, Kimura F, Moritani S, Tsuji S, Katusra D, Hoshiyama T, Nakamura A, Kitazawa J, Hanada T, Amano T, Kushima R, Murakami T (2022) The association between chronic deciduitis and preeclampsia. J Reprod Immun 150:103474. ISSN 0165-0378

Philip NY, Razaak M, Chang J, O'Kane SMM, Pierscionek BK (2022) A data analytics suite for exploratory predictive, and visual analysis of type 2 diabetes. IEEE Access 10:13460–13471. https://doi.org/10.1109/ACCESS.2022.3146884

Sharma S, Skog J, Timpka S, Ignell C (2021) Preeclampsia and high blood pressure in early pregnancy as risk factors of severe maternal cardiovascular disease during 50-years of follow-up. Pregnancy Hyperten 26:79–85. ISSN 2210-7789

Su Y, Tian X, Gao R, Guo W, Chen C, Chen C, Jia D, Li H, Lv X (2022) Colon cancer diagnosis and staging classification based on machine learning and bioinformatics analysis. Comp Biol Med 145:105409. ISSN 0010-4825

Yamashita T, Wakata Y, Nakaguma H, Nohara Y, Hato S, Kawamura S, Muraoka S, Sugita M, Okada M, Nakashima N, Soejima H (2022) Machine learning for classification of postoperative patient status using standardized medical data. Comp Methods Prog Biomed 214:106583. ISSN 0169-2607

CO-OFDM System with Self-Organizing Map (SOM) for Improving the Spectral Efficiency by the Compensation of Phase Noise

Gurpreet Kaur and Gurmeet Kaur

Abstract For optical signal transmission to increase the spectral efficiency in coherent systems, the phase noise compensator (PNC) without any pilot and with extreme learning machine (ELM) was proposed earlier. Moreover, in place of fast Fourier transform (FFT), it is preferred to use discrete wavelet transforms (DWPTs), since there is no need of cyclic prefix (CP) for removing the dispersion in ELM-based PNC. In this article, a self-organizing map (SOM)-based phase noise compensator is proposed for 16-QAM modulation format. It has been seen from the simulation results that there is improvement in performance of coherent systems with the proposed technique in comparison with ELM technique. Moreover, this proposed technique is improving the spectral efficiency comparatively.

Keywords Coherent optical orthogonal frequency division multiplexing (CO-OFDM) · Self-organizing map (SOM) · Phase noise (PN)

1 Introduction

In recent days, 5G communication and other applications require very fast speed of processing of information which results in requirement of optical communication systems with high amount of spectral efficiency. In order to fulfill this requirement, CO-OFDM was made in all types of optical passive networks and systems (Armstrong 2009; Shieh and Djordjevic 2010). From the literature of transmitting the longer symbol duration data, it has been observed that laser phase noise is one of the major reason for degrading the system enactment (Wu and Bar-Ness 2004). Therefore, for

G. Kaur
Indian Institute of Information Technology Una, Una, Himachal Pradesh 177209, India
e-mail: gurpreetkaur@iiitu.ac.in

G. Kaur (✉)
Punjabi University Patiala, Patiala, Punjab 147002, India
e-mail: gurmeetkece@pbi.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_46

compensating the effects of phase noise, some techniques (pilot-aided (PA) and RF-pilot (RFP)) were proposed (Pasandi and Plant 2010; Randel et al. 2010; Yi et al. 2007; Zhu et al. 2015). The main issue with these proposed techniques is requirement of extra bandwidth for pilot subcarriers.

By keeping this problem into consideration, some non-pilot techniques were introduced (Ha and Chung 2013; Mousa-Pasandi and Plant 2010; Nguyen et al. 2017; Venkatasubramani et al. 2019). However, it has been observed that these techniques are more computationally expensive. Some more techniques from literature were used for reducing the phase noise impact on symbols of long duration (Balogun et al. 2018; Chen et al. 2020).

In this paper, self-organizing map (SOM)-based system has been suggested. In this work, no pilot subcarriers statics knowledge is used for mitigating the effects of phase noise. With this technique there is an improvement of 23% in spectral efficiency when compared with the traditional pilot subcarriers-based technique. The work of this paper is divided into five sections, i.e., introduction, network and mathematical modeling, spectral efficiency calculation, simulation results, and the conclusion.

2 Network and Mathematical Modeling

Self-organizing maps are networks with unsupervised type of competitive learning approach for updating the weights of network.

The main difference in SOM-based network and other artificial neural networks is that the stochastic gradient decent method is not used for learning activity of network and this network is not having activation function. A SOM (Haykin 1994; Jain et al. 2000) is basically made up of set of neurons with lattice structure in two-dimensions as shown in Fig. 1. There are two layers in this network, first layer is used as input layer and number of neurons in input layer is equal to the number of inputs. Second layer is in lattice structure for mapping the features.

The main algorithm used for this work is presented in Fig. 2. The formula used for updating the weights is presented below:

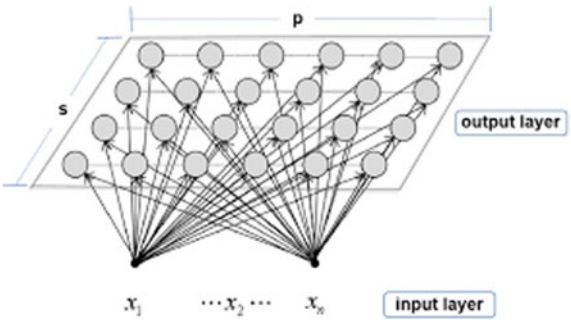
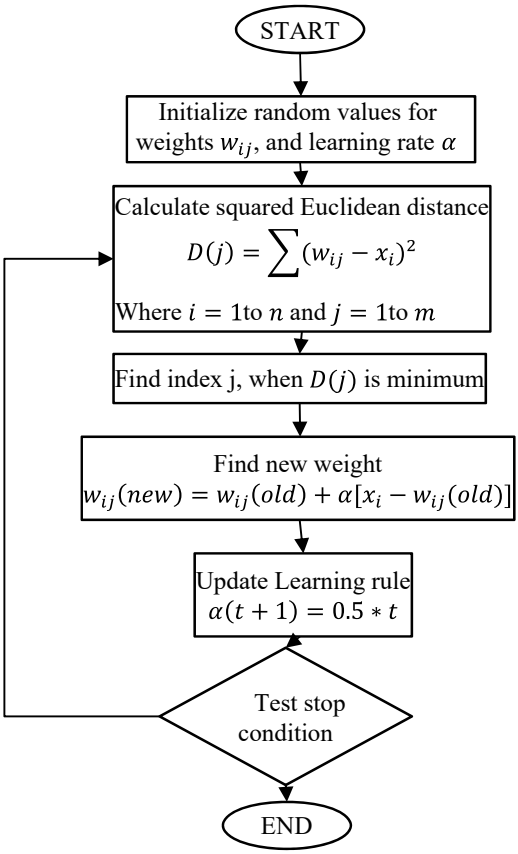


Fig. 1 Self-organizing map (SOM) network

Fig. 2 Flowchart



$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \text{alpha}(t) * \left(x_{ik} - w_{ij}(\text{old})\right) \tag{1}$$

where $w_{ij}(\text{new})$ is the updated weight, $w_{ij}(\text{old})$ is the old weight, x_{ik} is the input, and alpha is the learning rate.

3 Spectral Efficiency Calculation

The formula used for the calculation of spectral efficiency with this proposed work is:

$$SE\left(\frac{\frac{\text{bits}}{\text{sec}}}{\text{Hz}}\right) = \frac{N_{\text{filled}}}{N_{SC}} \times \frac{32G}{\text{BW.Occupied}} \times b \tag{2}$$

and BW occupied is calculated as:

Table 1 Comparison of improvement in spectral efficiency (SE) in different techniques

Sr. No.	Technique	Percentage improvement in SE
1	Pilot-aided technique (existing)	18
2	ELM-based technique (existing)	21
3	SOM-based technique (proposed)	23

$$\text{BW occupied} = \frac{2}{T_S} + \frac{N_{SC} - 1}{t_s} \tag{3}$$

where N_{SC} is number of subcarriers, N_{filled} subcarriers loaded with PRBS-15 data, T_S is period of CO-OFDM symbol with observation period t_s. The comparison of improvement in spectral efficiency w.r.t without compensator calculated using above formulae is presented in Table 1 for total number of 128 subcarriers. From this Table 1, it is clearly seen that the improvement in SE with proposed technique is more than the existing techniques. It has been observed the percentage improvement of calculated spectral efficiency is 23.

4 Simulation Results and Discussions

For receiving the results, various performance parameters, such as EVM, BER, and Q-Factor are calculated for the proposed technique and existing ELM-based technique. The formulae used for their calculation are as follows (Fatadin 2016; Schmogrow et al. 2012; Shafk et al. 2006):

$$EVM_{rms} = \sqrt{\frac{\frac{1}{N}\sum_{k=1}^{N}|y_k - x_k|^2}{S_{\max}^2}} \tag{4}$$

$$BER = \frac{\left(1 - M^{-\frac{1}{2}}\right)}{\frac{1}{2}\log_2 M}\text{erfc}\left[\sqrt{\frac{\frac{3}{2}}{(M-1)EVM_{rms}^2}}\right] \tag{5}$$

$$Q = 20\log_{10}\left(\sqrt{2}erfc^{-1}(2BER)\right) \tag{6}$$

Here, idyllic and obtained constellation points are represented as x_k and y_k, respectively, maximum number of points is represented as $S_{\max}$, and M is the modulation format number.

The simulation results are presented in Figs. 3, 4, and 5 and the comparison is done in existing technique (ELM—extreme learning machine-based technique) and self-organizing map (SOM) with results obtained without any phase noise compensator. In Fig. 3, the results are shown for power (dBm) versus BER with proposed technique in comparison with ELM technique at 200 kHz laser linewidth. From graph, it is clear

that the optimum value of launched power is 4 dBm, since at 4 dBm launched power the value of BER is minimum. The improvement in performance at 4dBm launched power is 0.93 with proposed technique when compared with 0.9 with ELM technique.

In Fig. 4, length (Km) versus BER with proposed technique in comparison with ELM technique at 200 kHz laser linewidth is presented. It has been observed from this plot that with the increase in length, there is degradation in systems performance. Moreover, it has been noticed in this plot that the proposed technique is more effective

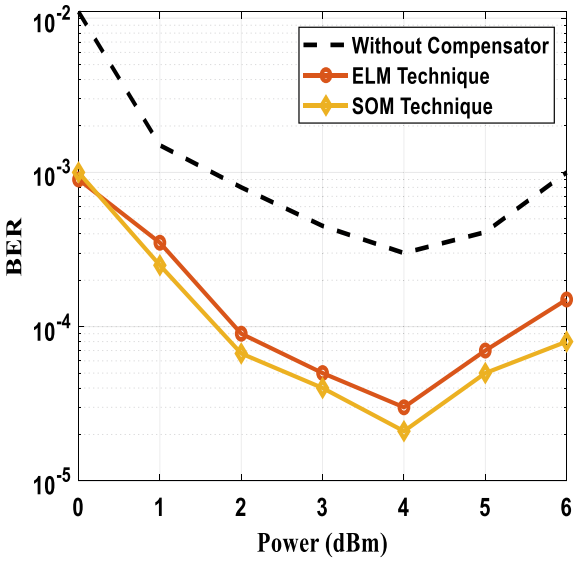


Fig. 3 Power (dBm) versus BER

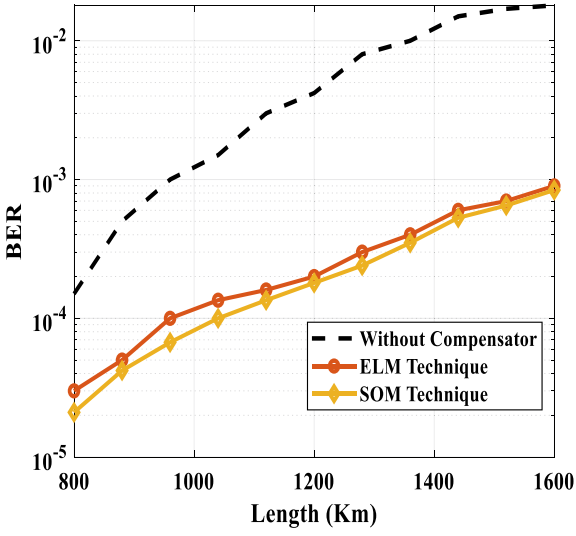


Fig. 4 Length (Km) versus BER

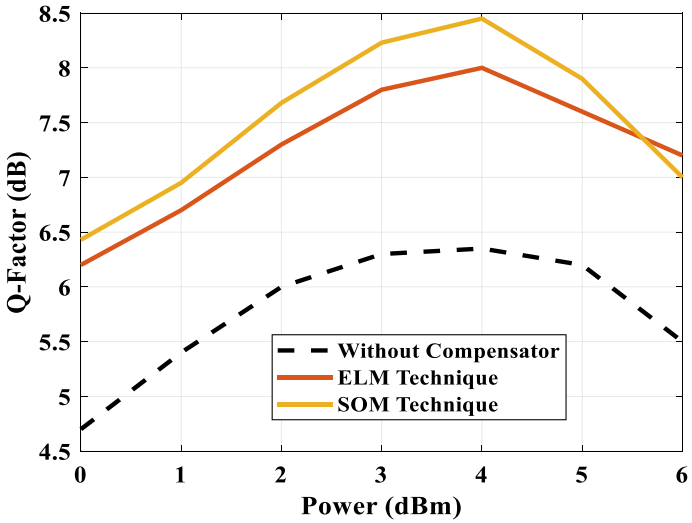


Fig. 5 Power (dBm) versus Q-factor (dB)

at lower transmission lengths when compared with ELM technique. Furthermore, one more conclusion can be withdrawn from this plot that at higher transmission lengths the proposed technique is giving same performance as ELM technique.

In Fig. 5, power (dBm) versus Q-factor (dB) with proposed technique in comparison with ELM technique at 200 kHz laser linewidth is shown. It has been noticed that the percentage improvement in Q-factor at optimum launched power (4 dBm) is 33 with proposed technique when compared with 26 with ELM technique.

5 Conclusion

In this work, a self-organizing map (SOM)-based phase noise compensator (PNC) for 16-QAM modulation format has been proposed in coherent systems. It has been observed in the simulation results that the Q-factor (for performance measurement) and spectral efficiency (for effective utilization of bandwidth) are improved by a noteworthy amount in comparison with the previous techniques of phase noise compensation. In addition to these improvements, there is an improvement in complexity of the system, since DWPT technique is used in this work in place of FFT. For getting more robust outcome, SOM weight update formula is used for the compensation of phase noise. Furthermore, the training speed of this algorithm is more with better generalization performance.

References

Armstrong J (2009) OFDM for optical communications. J Lightwave Tech 27(3):189–204

Balogun MB, Oyerinde OO, Takawira F (2018) Adaptive subspace method for phase noise estimation in CO-OFDM systems. IEEE Access 6:56368–56377

Chen Y, Huang Y, Han Y, Fu J, Li K, Li Y, Yu J (2020) A novel phase noise suppression scheme utilizing Gaussian wavelet basis expansion for PDM CO-OFDM Superchannel. IEEE Photonics J 12(2):1–9

Fatadin I (2016) Estimation of BER from error vector magnitude for optical coherent systems. Photonics 3:21

Ha Y, Chung W (2013) Non-data-aided phase noise suppression scheme for CO-OFDM systems. IEEE Photonics Technol Lett 25(17):1703–1706

Haykin S (1994) Neural networks: a comprehensive foundation. Prentice-Hall, Upper Saddle River, NJ, USA

Jain AK, Duin RPW, Mao JC (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Mousa-Pasandi ME, Plant DV (2010) Zero-overhead phase noise compensation via decision-directed phase equalizer for coherent optical OFDM. Opt Express 18(20):20651–20660

Nguyen TT, Le ST, Wuilpart M, Yakusheva T, Megret P (2017) Simplified extended Kalman filter phase noise estimation for CO-OFDM transmissions. Opt Express 25(22):27247–27261

Pasandi MEM, Plant DV (March 2010) Improvement of phase noise compensation for coherent optical OFDM via data-aided phase equalizer. In: 2010 conference on optical fiber communication (OFC/NFOEC), collocated national fiber optic engineers conference, pp 1–3

Randel S, Adhikari S, Jansen SL (2010) Analysis of RF-pilot-based phase noise compensation for coherent optical OFDM systems. IEEE Photon Technol Lett 22(17):1288–1290

Schmogrow R, Nebendahl B, Winter M, Josten A, Hillerkuss D, Koenig S, Meyer J, Dreschmann M, Huebner M, Koos C (2012) Error vector magnitude as a performance measure for advanced modulation formats. IEEE Photonics Technol Lett 24:61–63

Shafk RA, Rahman S, Islam R (2006) On the extended relationships among EVM, BER and SNR as Z. In: ICECE'06 international conference on electrical and computer engineering, pp 408–411

Shieh W, Djordjevic I (2010) OFDM for optical communications. Academic Press, California

Venkatasubramani LN, Vijay A, Venkitesh D, Koilpillai RD (2019) Pilot-free common phase error estimation for CO-OFDM with improved spectral efficiency. IEEE Photon J

Wu S, Bar-Ness Y (2004) OFDM systems in the presence of phase noise: consequences and solutions. IEEE Trans Commun 52(11):1988–1996

Yi X, Shieh W, Tang Y (June 2007) Phase estimation for coherent optical OFDM transmission. In: COIN-ACOFT 2007—joint international conference on the optical internet and the 32nd Australian conference on optical fibre technology, pp 1–3

Zhu J, Omomukuyo O, Venkatesan R, Li C, Dobre OA (2015) RF-pilot phase noise compensation for long haul coherent optical OFDM systems. In: 2015 IEEE 14th Canadian workshop on information theory (CWIT)

Intelligent Car Damage Detection System

Ved Mistry, Chirag Jagad, Umang Jhunjhunwala, and Prachi Tawde

Abstract This research develops an automated car damage detection system using computer vision and deep learning algorithms to detect and classify damage on cars. Traditional methods rely on human inspectors, which are time-consuming and costly. The proposed system uses convolutional neural networks and transfer learning on VGG-16 for efficient and accurate detection. The developed pipeline breaks down the process into distinct stages for maximized efficiency and accuracy. This allows for more efficient and informed decision-making by insurance companies, maintenance providers, or individuals. The system provides a more efficient, accurate, and cost-effective way to assess car damages, enhancing road safety with more objective and consistent assessments.

Keywords Object detection · Computer vision · Transfer learning · VGG-16 · Car damage

1 Introduction

Car accidents can cause significant physical and emotional harm to those involved, as well as lead to costly damages to the vehicles involved. India has one of the highest rates of road accidents in the world. According to the Ministry of Road Transport and Highways of India, over 400,000 road accidents were reported in 2021 (Ministry of Road Transport Highways, Government of India 2023). Traditional methods of assessing car damage rely on human inspectors, which can be time-consuming, prone to errors, and costly. Therefore, there has been a growing interest in developing automated systems for car damage detection, which provides a more efficient, accurate, and cost-effective way to assess car damages. These systems rely

V. Mistry (✉) · C. Jagad · U. Jhunjhunwala · P. Tawde
Dwarkadas J. Sanghvi College of Engineering, Mumbai University, Mumbai, Maharashtra, India
e-mail: vedmistry619@gmail.com

P. Tawde
e-mail: prachi.tawde@djsce.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_47

on computer vision and machine learning algorithms to analyze images and videos of cars and identify the locations and types of damages.

The main objective of this project is to develop a car damage detection system that can automatically detect and classify different types of damages on cars. The system will be trained on a large dataset of labelled images and videos of damaged cars using deep learning algorithms for feature extraction and classification. The performance of the system will be evaluated on a test dataset, and the results will be compared with the state-of-the-art methods in car damage detection. An accurate and efficient car damage detection system can help to improve the accuracy and efficiency of car damage assessment, reduce the costs associated with manual inspection, and enhance road safety by providing a more objective and consistent assessment of car damages.

2 Related Works

One proposed method compared Google LeNet and ResNet CNN architectures in identifying and locating various types of damages in automobiles. Both architectures are pretrained on Imagenet and fine-tuned on a 3000-picture dataset with four damage categories: undamaged, scratch, mild, and severe. The same architectures are used to locate damage and classify it into nine classes (Moreira 2017). A study used the classification and regression tree (CART) method to develop an expert system for identifying car damage. Experts provided information on damage symptoms, which was used in the input stage. The learning stage determined potential branches, while the classification stage used decision trees to diagnose damage based on user input of symptoms (Putri et al. 2014). In another method, the authors developed a pipeline for detecting automobile damage to streamline insurance claims. They manually collected and tagged photographs of diverse damage using a web crawler and employed pretrained models to prevent overfitting. They described a method called class activation mapping (CAM) that enables CNN models to learn object localization. The authors created a pipeline using YOLOv3 and a CNN model trained on the damage dataset to detect and classify damaged regions (Dwivedi et al. 2021). Unlike previous approaches, the car damage assessment system in Zhang et al. (2020) interacts with consumers using videos rather than images, making the entire process as easy as possible. They use computer vision techniques such as object and video detection and segmentation and numerous frames taken from videos to achieve excellent damage recognition accuracy. VGG-16 is a deep convolutional neural network created by researchers at the University of Oxford in 2014. It consists of 16 layers and was trained on the ImageNet dataset with over 1.2 million images from 1000 different classes. VGG-16 uses small 3×3 convolutional filters repeatedly to extract features at different scales. VGG-16 achieved state-of-the-art performance on the ImageNet dataset, with 92.0% accuracy on the validation set (Karen and Zisserman 1409; Matthew and Fergusn 2014).

Traditional machine learning and deep learning algorithms have up until now been created to operate independently. These algorithms have been designed to carry out

particular functions. The models must be completely rebuilt whenever the feature-space distribution changes. Breaking out from the isolated learning paradigm, transfer learning is the idea of using the information acquired for one activity to address related problems.

3 Methodology

We trained a pipeline using convolutional neural networks and transfer learning on VGG-16 (Karen and Zisserman 1409) with Keras for automated damage detection and estimation. The use of transfer learning allows for the model to leverage the pretrained weights of VGG-16, which was trained on the ImageNet dataset (Deng et al. 2009) having a large number of images, to help accelerate the training of our model. The pipeline uses a series of convolutional and pooling layers to extract features from the image, followed by fully connected layers to generate the required results. Once the model is trained, it can take an input image and locate the areas of damage on a vehicle and predict the severity of the damage. The output of the pipeline includes the estimated cost of repairing the damage, as well as the location and severity of the damage.

The pipeline for automated damage detection and estimation of repair costs is designed to maximize efficiency and accuracy by breaking the process down into several distinct stages, or "gates". The gates are designed to serve a unique purpose and work together to produce a final report that details the location and severity of the damage, as well as an estimated repair cost. Figure 1 shows the system architecture of the entire pipeline.

Gate 1 of the pipeline is a simple binary classifier trained using a pretrained VGG-16 network that detects whether the input image is of a car or not. This is an important first step because if the input image is not of a car, there is no need to waste processing time by sending it through the other gates. If a car is detected in the input image, the image is then sent to Gate 2.

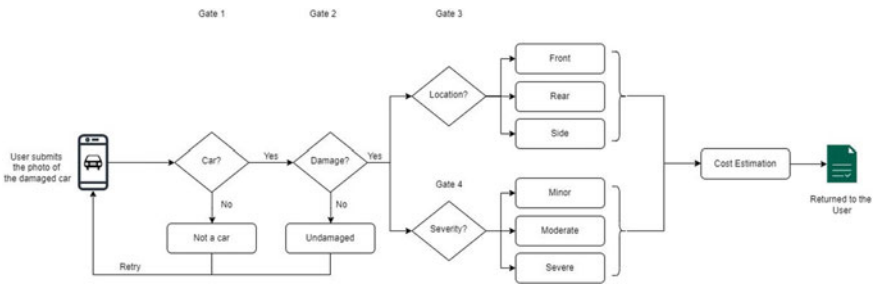


Fig. 1 System architecture

Gate 2 is another simple binary classifier that checks whether any damage has been incurred to the car in the input image. If the car is undamaged, the pipeline terminates and does not proceed any further. If damage is detected, the image is then sent to Gate 3.

Gate 3 is designed to detect the location of the damage in the input image. This is accomplished using a multi-class classifier that categorizes the location into three classes—front, rear, and side. Once the location of the damage has been detected, the image is then sent to Gate 4.

Gate 4 is designed to detect the severity of the damage in the input image. This is accomplished using another multi-class classifier that categorizes the severity of the damage into three classes—minor, moderate, and severe. Once the severity of the damage has been detected, the predictions of Gates 3 and 4 are gathered and fed into a repair cost estimation model.

The repair cost estimation model takes into account the location and severity of the damage to accurately predict the estimated cost of repairing the damage. The final output of the pipeline is a generated report that consists of all the details predicted by the various models of the pipeline, including the location and severity of the damage, and the estimated cost of repairing the damage.

4 Implementation

The implementation of this project involves the use of several technologies, including Python, JavaScript, Flask, HTML, CSS, OpenCV, TensorFlow, Keras, and Scikit-learn. Python is used for its powerful image processing libraries like OpenCV, enabling us to analyze images. TensorFlow and Keras are used for building and training deep learning models for image recognition, classification, and segmentation. Scikit-learn is used for building and training machine learning models for data analysis and prediction. JavaScript is used for frontend web development, enabling dynamic interactions with the user interface. Flask is used as the web framework for the backend development, allowing for the creation of RESTful APIs and web services. HTML and CSS are used for creating the structure and styling of web pages. Implementing the entire system on Intel(R) Core(TM) i5-1035G1 CPU and NVIDIA GeForce MX250 GPU ensures that the system has the necessary hardware resources. Moreover, the system can also leverage the power of NVIDIA's CUDA technology, which allows for the efficient use of the GPU's computing power.

5 Dataset

We have a dataset of 4600 images divided into three sections, each further divided into training and validation. Part A has 920 images each of damaged and undamaged cars in training, and 230 in validation. Part B has 419 front view, 288 rear view, and

272 images of damaged and undamaged cars in training, and 73 front view, 50 rear view, and 48 images of damaged and undamaged cars in validation. Part C has 278 minor damage, 315 moderate damage, and 386 severe damage images in training, and 48 minor, 55 moderate, and 68 severe damage images in validation.

6 Results

On the original data for stage 1, which is to determine whether the car is damaged or not, we can see that when fine-tuned, the VGG-16 performed significantly better than the original VGG-16 model. Though the accuracy of VGG-16 is not the highest as displayed in Table 1, we have used its simplicity and ease of use. By fine-tuning the pretrained VGG-16 model on a smaller dataset, we can achieve high accuracy. The accuracy of this model is 93.02% which increased to 96.43% after fine-tuning. We obtained a precision of 91% and a recall of 90%. For stage 2 which is to determine the damage localization again fine-tuned by the same process made a model with an accuracy of 71.93% which increased to 78.95% after fine-tuning, we obtained a precision of 76% and recall of 76%. For stage 3 the model had an accuracy of 66.67% which increased to 71.35% after fine-tuning, we obtained a precision of 71% and recall of 69%. The stage 1 binary classification fine-tuner model uses stochastic gradient descent, RMSProp, L2 kernel regularizer, and sigmoid activation function. The training was done in 50 epochs of which the best model was selected. The stage 2 and 3 categorical classifications fine-tuner model uses an L2 kernel regularizer, and softmax activation function. The training was done in 50 epochs of which the best model was selected.

Table 1 Top-performing pretrained models on the ImageNet dataset

Model	Accuracy (in %)
ResNet50 (He et al. 1512)	93.0
DenseNet-121 (Huang et al. 1608)	92.3
InceptionV2 (Szegedy et al. 1409)	92.2
VGG-19 (Karen and Zisserman 1409)	92.2
VGG-16 (Karen and Zisserman 1409)	92.0
ShuffleNet (Zhang et al. 1707)	89.80
AlexNet (Krizhevsky et al. 2017)	84.9

6.1 Training Results

To ensure the greatest possible flow between the network's layers, the VGG-16 model was selected. Additionally, the dense connectivity pattern gives us the ability to adjust the parameters to meet our needs and produce the best results.

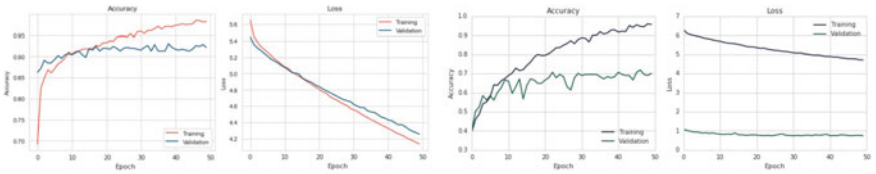
The accuracy of the overall model was close to 70%.

Gate 0: Pre-trained VGG-16 accuracy: 92.7%.
Gate 1: Accuracy: 93.04%, validation loss: 4.4627.
Gate 2: Accuracy: 71.93%, validation loss: 0.7495.
Gate 3: Accuracy: 66.67%, validation loss: 0.7424.

6.2 Fine-Tuning Results

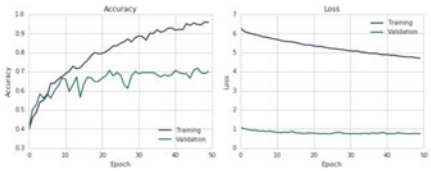
Fine-tuning in machine learning involves taking a pretrained model and training it further on a specific task, often with a smaller dataset, to improve its performance on that task. We build a classifier model to put on top of the convolutional model. The dense layers used ReLU (with kernel regularization) and softmax as activation functions in the classifier model (Figs. 2 and 3, Table 2).

Gate 1 (SGD with learning rate = 0.0001, sigmoid, with l2 = 0.001): Accuracy: 96.43%, validation loss: 4.1755.
Gate 2: Accuracy: 78.95%, validation loss: 0.6357.
Gate 3: Accuracy: 71.35%, validation loss: 0.7054.



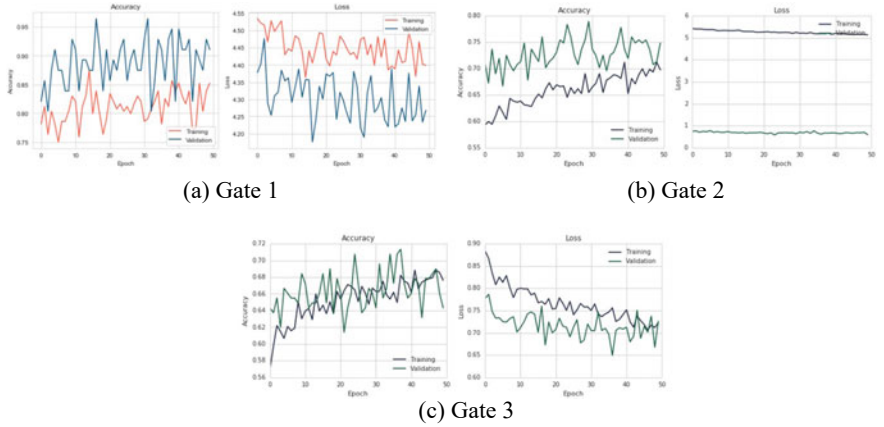
(a) Gate 1

(b) Gate 2



(c) Gate 3

Fig. 2 Training results



(a) Gate 1

(b) Gate 2

(c) Gate 3

Fig. 3 Fine-tuning results

Table 2 Comparison between untuned and fine-tuned model

Gate	Accuracy before fine-tuning (in %)	Accuracy after fine-tuning (in %)
Gate 1	93.04	96.43
Gate 2	71.93	78.95
Gate 3	66.67	71.35

7 Conclusion

In conclusion, the development of an automated car damage detection system is a promising area of research that has the potential to revolutionize the way car damages are assessed. The system can provide a more efficient, accurate, and cost-effective way to assess car damages, which can lead to better decision-making and reduced costs for insurance companies, car repair shops, and individuals looking to purchase a used car. Moreover, the system can help to enhance road safety by providing a more objective and consistent assessment of car damages. The results demonstrated that the developed system was able to accurately detect and classify different types of damage on cars, achieving high accuracy scores. Future research in this area can focus on developing more sophisticated deep learning models that can handle more complex damage scenarios and improve the system's generalizability to unseen data.

References

Deng J, Dong W, Socher R, Li L, Li K, Fei L (2009) ImageNet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, pp. 248–255

Dwivedi M, Malik H, Omkar S, Monis E, Khanna B, Samal S, Tiwari A, Rathi A (2021) Deep learning-based car damage classification and detection. In: Advances in artificial intelligence and data engineering. Springer, pp 207–221

He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. In: arXiv preprint, arXiv:1512.03385v1

Huang G, Liu Z, Maaten L, Weinberger KQ (2018) Densely connected convolutional networks. In: arXiv preprint, arXiv:1608.06993

Karen S, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: arXiv preprint, arXiv:1409.1556

Krizhevsky A, Ilya S, Geoffrey H (2017) Imagenet classification with deep convolutional neural networks. In: Communications of the ACM 60, no 6, pp 84–90

Matthew Z, Fergusn R (2014) Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part I 13. Springer International Publishing, pp 818–833

Ministry of Road Transport Highways, Government of India. Accessed 25 March, 2023. https://morth.nic.in/sites/default/files/RA2021Compressed.pd

Moreira JPLM (2017) Identificação de danos em véıculos sinistrados através de imagens. In: U.PORTO. https://repositorio-aberto.up.pt/handle/10216/107814

Putri PK, Putra IKG, Mandenni NIMI (2014) An expert system to detect cardamage by using cart method. J Theor Appl Inf Technol

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. In: arXiv preprint, arXiv:1409.4842

Zhang X, Zhou X, Lin M, Sun J (2017) Going deeper with convolutions. In: arXivpreprint, arXiv:1707.01083v2

Zhang W, Cheng Y, Guo X, Guo Q (2020) Automatic car damage assessment system: reading and understanding videos as professional insurance inspectors. AAAI Conf Artif Intell 34(09):13646–13647. AAAI

An Efficient Malicious URL Detection Approach Using Machine Learning Techniques

Swati Chaudhari, Archana Thakur, and Alpana Rajan

Abstract IT security is a major concern for everyone. Malicious Uniform Resource Locators (URLs) are major source of cyber-attacks. Blacklisting of malicious IP addresses is a primary technique used for safeguarding mission critical IT systems. It is repetitive, time-consuming, computationally intensive, and incapable of detecting new URLs. Heuristic classification is an improvement to blacklisting approach. Cyber-attacks are increasing at a rate of 20–25% per year, with newly generated URLs serving as a primary attack surface. Techniques currently used are ineffective because they cannot classify newly generated URLs. This issue is handled by maintaining and updating a comprehensive database of millions of URLs with the new URLs at regular interval. In spite of having greater accuracy, it requires human involvement to update and maintain URL databases, which is a big concern. To overcome these limitations, machine learning (ML) is used for detecting malicious URLs efficiently. The feature set composed of lexical, binary, address bar, domain, and HTML-JavaScript-based features is used with the logistic regression, decision tree, and random forest algorithms. These models are compared for accuracy, recall, and precision over malicious and benign URL datasets. Methodology to update URL datasets using security logs is discussed to improve efficiency of ML model. Innovatively, a parallel programming technique to improve training time of the ML models is also presented. The presented malicious URL detection approach is efficient, and accuracy is approximately 99%. It can be used to build a malicious URL block list and prevent accesses to or from malicious URLs to safeguard clients against potential cyber-attacks.

Keywords Malicious URL · Machine learning · Parallel programming · Cyber-attacks

S. Chaudhari (✉) · A. Rajan
Department of Atomic Energy, Raja Ramanna Centre for Advanced Technology, Government of India, Indore, Madhya Pradesh, India
e-mail: swati@rrcat.gov.in

A. Thakur
School of Computer Science and Information Technology, Devi Ahilya Vishwavidyalaya, Indore, Madhya Pradesh, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_48

1 Introduction

As our usage of the Internet increases, the amount of malware attacks happening each day are becoming more and more prevalent. Majority of digital resource holders feels their current security practices are not adequate. Data breaches are increasingly common, affecting individuals and organizations. The current pandemic has increased the challenges manifolds. Besides various measures, IT security is still a very big concern to many especially in scientific organizations which are of national importance. Malicious URLs are a major source of cyber-attacks. Potential cyber-attacks are launched and distributed by attackers using malicious domains, URLs through spam, text messages, drive by downloads, pop-ups, etc. Internet users are targeted to visit malicious URLs in the form of web pages to spread malware, virus, worms, etc. Objective of attacker is gaining confidential information such as credentials, security PIN or to plant malicious software in target system. So, it is necessary to adapt the system which should detect the malicious URL and prevent from the attack. There are conventional methods for detection of malicious URLs. Blacklisting of malicious IP address is a primary technique commonly used for safeguarding mission critical IT systems. The technique is repetitive, time-consuming, computationally intensive, and incapable to detect new URLs. Heuristic classification is an improvement to the blacklisting. There is a growth of attacks by 20–25% annually. Threats are on continuous rise, which majorly uses newly generated URLs as a surface of attacks. These techniques are very ineffective as they are not able to classify newly generated URLs. One way to handle this issue is to maintain a comprehensive database of millions of URLs through web-based companies which are updated on regular intervals with the new URLs. In spite of having greater accuracy, a big concern is that, it requires human involvement to update and maintain URL database. Conventional methods like Blacklisting & Heuristic classification for detecting malicious URL are ineffective and inefficient as they requires manual intervention for updating new URLs and work for newly generated URLs. A selflearning system based on ML is proposed in the paper to detect malicious URLs accurately. Feature extraction is important part in machine learning approach for data transformation and enhancing model accuracy.

A URL specifies web document's location address and resources of web on the Internet. It has two parts, the first is the protocol identifier and second one is the address of IP or name of domain indicating where actually the resource is sited. The URL is considered as a base entity for performing feature engineering in our proposed approach for detecting malicious URL. The set of features prepared using the lexical like length, count, binary features such as use of IPs and shortening of URLs. Address bar features like redirection, http/https in the domain name are added in the feature list to for improving the accuracy of detection model. Domain-based features like web traffic, dns time and HTML, JavaScript-based features are also tested. Logistic regression, decision tree and random forest algorithms are used for building ML model.

This paper is organized is like: Sect. 2 discusses related work of the field. The proposed methodology is presented in Sect. 3. Brief discussion about the results is presented in Sect. 4. The concluding remarks and the future scope of the presented approach are provided in Sect. 5.

2 Related Work

Different approaches have been tried to handle the problem of malicious URL detection. It can be categorized into: (i) blacklisting or heuristics, (ii) malicious URL detection tools, (iii) machine learning approach.

2.1 Blacklisting or Heuristic Approach

Database of malicious URLs is maintained in blacklisting approach (Sheng et al. 2009; Sinha et al. 2008). While detecting malicious URL, a query is made to database for its existence in it. If URL is existing in the database, then it is taken as a malicious else assumed to be benign. Maintaining exhaustive list of all possible malicious URLs is a big challenge as new URLs gets generated daily. It makes difficult to detect new threats (Sheng et al. 2009). It becomes critical when attackers create new URLs using some algorithms and get escaped from blacklist. Blacklisting is still used by many anti-virus systems because it is simple and efficient (Sinha et al. 2008). Blacklisting approach is improvised as a heuristic approach in which blacklist of signatures of different attacks is created (Seifert et al. 2008). These signatures cover identified attack types. Signatures of different attacks are scanned by Intrusion Detection Systems for probing suspicious behaviors and raises alarms in case such behaviors are detected. This approach covers behaviors of threats and hence can cover larger set of URLs. It has the capability to find threats in new URLs and hence better than blacklisting. However, signatures can be created for limited number of common threats and cannot generalize to zero day or new attacks. It is easy to use obfuscation techniques to avoid them. A customized version of heuristic methods is also used in which analysis of execution dynamics of the webpage is carried out (Kim et al. 2011). Signatures of malicious behavior which generally covers uncommon process forking, recurring redirection, etc., are looked. The URLs are required to be visited in this approach which can launch attack, and hence, it requires controlled environment like sandbox. My concern is if sandbox environment is not available then checking malicious behaviour is risky. In case of malicious page, the attacks may not get initiated immediately but can get planted for future and may get unnoticed.

2.2 Malicious URL Detection Tools

Commercial malicious URL detection tools are available in market. URL Void is a program; it uses several engines and blacklist domains databases to check (Domain Reputation 2022). Many browsers are supported by URL Void program is an advantage. The process of detecting malicious URL which greatly depends upon the predefined signature sets is a big drawback of the program. Another tool is Unmask Parasites (2022), which tests URLs by retrieving and analyzing, iframes, Hypertext Markup Language—HTML contents, outward links, and JavaScripts. It can detect iframe quickly and correctly is the plus point of this tool. Though this tool is of use only when the client suspects anything unusual is happening on the sites which are being accessed. Comodo Site Inspector (2022) is used to detect malware and security laps. This tool assists users to check URLs or allows web administrators to use a sandbox-controlled environment to execute the downloaded particular sites to perform regular checks. There are few more URL checking tools, such as Browser Defender (2022), Google Safe Browsing Diagnostic (2022), Online Link Scan (2023), Sucuri (2023), UnShorten.it (2023), and VirusTotal (2022).

Signature-based URL detection approach is broadly used in all current malicious URL detection techniques and tools. Hence, the effectiveness of these tools is restricted. Flaws of current detection methods are, it is very difficult to maintain an up-to-date blacklist. URLs are modified using obfuscation like mystifying the host by an IP, confusing the host through another domain, obscuring using large host names for the host, and misspelling. Modifying URLs by obfuscating makes it difficult to identify as malicious and adding it to blacklist.

2.3 ML Approach

A ML algorithm helps computer to perform task without human interactions. Basically, we train a machine with inputs, and after enough training, datasets are provided; it can predict results on its own. This approach is used for malicious URL detection.

Conditional Kolmogorov Complexity calculation of the set of benign and malicious URLs data to detect malicious URL is discussed in Pao et al. (2012). Machine learning algorithm-based detection of malicious URLs using their behaviors and attributes is presented in Xuan et al. (2020). In the paper lexical, host-based and correlated feature groups have been used for training ML models. Training performance for Support Vector Machine (SVM) and random forest algorithms have been also given in detail. The analysis and study of malicious URLs and proposed methodologies for discovering the malicious URLs of the web applications is presented in Prasad and Rao (2021). Capturing the image of the web page by visiting the URL and generation of perceptual hash for the URL page methodology have been given in the paper (Prasad and Rao 2021). SVM algorithm with the set of features prepared out of the 18 features, like count of tokens, largest path, average path token, largest

token, etc., for the URL detection has also been suggested. The suggested technique which can be improved with more features and is anticipated to offer an enhanced result is also mentioned in the paper (Naveen et al. 2019). Based on empirical study carried by Patgiri Ripon et al., the split ratio 80:20 is observed as more accurate split and average accuracy of random forests is more than SVMs (Patgiri et al. 2019).

Considering the limitations of above-mentioned approaches, supervised learning algorithms of ML with additional features and parallel programming techniques for feature computations to detect malicious URL are adopted for our work and discussed in the paper. We are going to feed inputs along with labels saying whether URLs are benign or malicious. Based on the selected features, the machine is going to make predictions whether URL is malicious or benign.

3 Methodology

Malicious URL detection which involves prediction of URL as benign or malicious is a binary classification problem. Explicitly, a data set is given with N URLs $\{(u1, y1), ..., (uN, yN)\}$, where URLs of the training dataset is represented as un for $n = 1, ..., N$, and corresponding label is represented as $yn \in \{1, 0\}$ where $yn = 1$ classify a malicious URL and $yn = 0$ classify a benign URL.

- Feature Representation: Extracting the suitable feature representation: $un \rightarrow xn$ where $xn \in$ Rd is a d-dimensional URL feature vector.
- ML: Function f: Rd $\rightarrow R$ is a prediction function which does binary classification for URL instance x using suitable feature representation.

Figure 1 shows block diagram of methodology used for malicious URL detection using ML.

Major components of our approach for malicious URL detection using ML are described as follows.

3.1 Training and Testing Dataset

The dataset for implementation of our malicious URL detection model comprises: URLs gathered from Kaggle datasets website (2023), data source from Malicious_n_Non-Malicious URL (2023). While accessing Internet by multiple clients, www access log gets generated. Logs are examined for bulk unauthenticated accesses made by clients. The clients trying to reach such URLs can be categorized as malicious, and rests of the URLs are safe to visit. Novelty of the proposed model is that the dataset is updated by adding some of such URLs on regular basis being as more labeled data for training can be beneficial for accuracy improvement of the model. Currently, the dataset has more than 514,736 labeled URL data. It contains 78% URLs which are benign and safe to access; on the other hand, left over 22% URLs

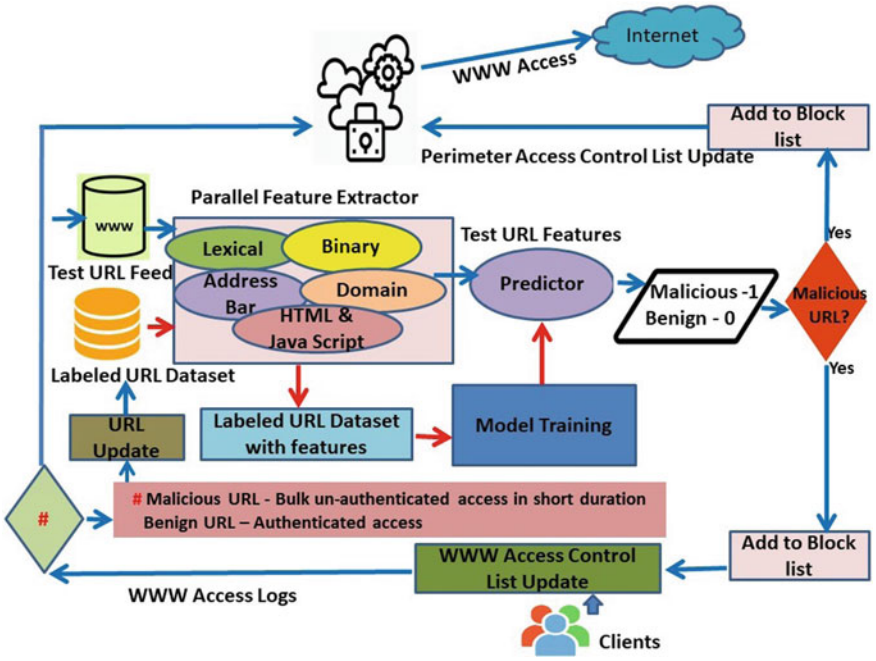


Fig. 1 Block diagram of methodology used for malicious URL detection using ML

are malicious. CSV file is used to store the set of data. Each URL in a dataset has label 1 for 'malicious' and 0 for 'benign'.

3.2 Prediction Dataset

This dataset is built using unique URLs visited by clients from WWW access log.

3.3 Feature Engineering

Binary classifier, using lexical, binary, address bar, domain-based and HTML-JavaScript-based features, has been built to classify the URLs as malicious or benign. Total 27 features which are based on different categories are extracted from URL dataset for training and prediction model. Lexical features like lengths of URL, host-name, path, first directory, and top level domain have been computed. Count of '−', '@', '?', '%', '.', ' = ', 'http', 'www', digits, letters, number of directories are added to feature list. Binary features like use of IP address in URL or not, use of shortening URL or not is also made part of feature list. Presence of unexpected IP Addresses in

URL could be indicators for malicious activities (Chiba et al. 2012). URL shortening not only reduces the URL length but also obfuscate the actual URL behind a shortened link. Attackers take advantage of this obfuscation to mislead the net users to land to malicious page by posting malicious links [22, 23]. Normally, the '//' in URL path is placed after the protocol which navigates the user to another website after redirection. If the '//' is present anywhere instead of after the protocol then it is more likely that, it is a malicious URL. 'http/https' in a URL represents its protocol but sometime attacker may place 'HTTPS' token in domain portion of URL to phish the users. Hence, domain portion is checked for 'http/https'. WHOIS database (2023) is source of information related to URLs. If DNS record is not getting verified or no record is found for the hostname of URLs from WHOIS, then it is recognized as malicious website. Visitors count of a website and number of pages visited by them determines its popularity. The popularity parameter is used to compute web traffic feature. Alexa database (2023) does not recognize malicious websites as they remain in existence for short period of time. By observing dataset, it is seen that in worst case situation, legitimate websites are ranked among top 100,000. In case if the domain has not received any traffic or it is not found in the database of Alexa then URL is classified as malicious. WHOIS database provides information of the feature 'Age of Domain' also. Malicious websites domain keeps short domain age. In our implementation, a minimum age of 12 months is used to determine the genuine domain. Age is difference between creation and expiration time. 'End Period of Domain' feature is extracted from WHOIS database by finding the difference between expiration time and current time. The end period considered for the legitimate domain is 6 months or more for the implementation work. Many HTML and JavaScript-based features are also considered. The 'iframe' tag can be used to hide it or make it without frame borders by attackers. Instead, attackers are using 'frameBorder' attribute which enables the browser to generate and show a visual description. If the iframe is empty or response is not found, then the URL is noted as malicious. Illegitimate URL is displayed in status bar using JavaScripts in case of malicious URL. Web page source code is checked, especially the 'onMouseOver' event to understand its effect on the status bar. If an empty response is received on 'onMouseOver', then the URL is tagged as malicious. Attackers restrict the users to view and download web page source code by disabling right-click function with the use of JavaScript. The source code of the web page is checked for the use of event 'event.button==2' to verify disabled or not disabled right click. If the response is empty, then the URL is tagged as malicious. Mostly, legitimate websites redirect only once unlike as malicious websites get redirected quite a few times. Website forwarding feature is used to distinguish between malicious and benign URL.

3.4 Parallel Feature Extractor

Both the safe and malicious URLs data stated above is passed through feature extraction module. Domain-based and HTML and Java-based features are queried over

Internet and computed after retrieving the information in real time. Latency time for receiving response for each URL is considerable compared to calculating other features locally. So, generating these features for larger no. of URL dataset is very high. Parallelization paradigm has been applied for reducing time required for generating such features for training and test data. Parallel libraries of Python have been used for the development. The URL dataset is divided across the multiple processing pools. Feature extraction is computed in parallel across the pools. This improved the time complexity of the model by factor of number of cores of the machine on which the code is executed. It is a great advantage in the scenarios where dataset is big and is getting updated frequently which can be used for training the model after feature extraction for improved model accuracy.

3.5 Machine Learning Model Selection for Predictor

Various machine learning algorithms are studied for our development work (Bhattacharyya and Bhattacharya 2021). They are widely used for detecting malicious URLs (Sahoo et al. 2019). The prediction model is trained using three different machine learning algorithms—logistic regression, decision tree, and random forest. The prediction function in these algorithms $f: \mathrm{R}^{\mathrm{d}} \rightarrow R$ is mathematically expressed as an optimization problem which try to achieve maximum possible detection accuracy.

Logistic Regression. The logistic regression classifier simply uses the sigmoid function to generate the probability for the class 1, i.e., p1. Sigmoid function value ranges between [0, 1]. Hence, probability for class 0 is $\mathrm{p0} = 1 - \mathrm{p1}$ (Sahoo et al. 2019).

Decision Tree. For a classification problem, we have bunch of features of URL and a corresponding target variable, i.e., label, where the target variable takes on values which are dependent on the value of the features. The algorithm uses features in the root node to generate appropriate condition which evaluates to '1' or '0' and have the targets along with the target values in the leaf nodes (Sahoo et al. 2019).

Random Forest. The random forest classifier is an ensemble of many decision trees. It builds decision trees on different samples and takes their majority vote for classification (Sahoo et al. 2019).

3.6 Building Malicious URL Block List

The URLs classified as malicious are added to block list of perimeter level and client level WWW access control list for preventing future referencing to them to reduce cyber-attacks surface.

4 Results and Performance Analysis

Dataset with lexical, binary, address bar features is separated into two subsets. Training of the predictor model is done using around 80% of the dataset, and about 20% of the dataset for testing model yields highest accuracies as indicated in graph shown in Fig. 2. The unique WWW URLs visited by clients can be parsed from access logs and given as a test URLs to the model for prediction. As a result, malicious URL block list can be built so as to prevent the further accesses to such domains. This way the proposed model can improve the organizational level security posture. Decision tree classifier's accuracy is 98.16% with TP = 302,592, FP = 2234, FN = 2551, TN = 104,409. Random forest classifier's accuracy is 99.32% with TP = 303,904, FP = 922, FN = 1848, TN = 105,114. To avoid over fitting hyperparameters of the algorithm are tuned. Logistic regression classifier's accuracy is 94.47% with TP = 295,675, FP = 9151, FN = 13,592, TN = 93,368. Table 1 enlists the performance parameters of different tested ML models. The implementation work is carried out on following experimental setup:

Software. Python version 3.9.7, Centos 7.

Hardware. CPU-Intel(R) x86_64@4.8GHz, RAM-2 GB.

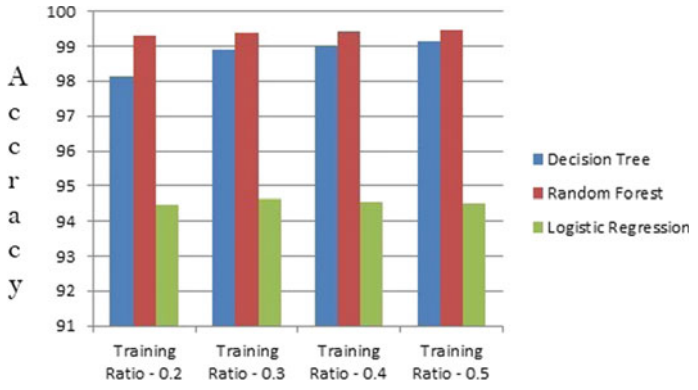


Fig. 2 Accuracies of models with training/testing ratios

Table 1 Test results based on confusion matrix

Sr. No.	ML algorithm (approx. 514736 URLs dataset)	Model accuracy (%)	Recall (%)	Precision (%)	Mean absolute error	Root mean squared error	R squared metric
1	Decision tree	98.16	99.16	99.26	0.0116	0.1077	0.9395
2	Random forest	99.32	99.39	99.69	0.0067	0.0819	0.9650
3	Logistic regression	94.47	95.60	96.99	0.0552	0.2350	0.7127

Table 2 Feature extraction time

Sr. No.	No. of URLs in dataset (with lexical, binary, address bar)	Training time (seconds) CPU cores = 1	Training time (seconds) CPU cores = 4
1	514,736	166	46
2	337,632	130	36
3	225,088	48	20
4	150,059	40	12
5	1000 (with lexical, binary, address bar and domain and HTM/JavaScript features)	1500	486

Table 2 gives feature extraction time compared for single and multiple cores execution. It is seen that with the parallel feature extractor, computation time is reduced approx. by 3.5 times which is good achievement for repetitive model training and testing with updated datasets. In case of domain and HTM/JavaScript feature extraction, parallel feature extractor is beneficial as it takes huge amount of time for feature.

5 Conclusions and Future Work

In this paper, an efficient approach of detection of malicious URL using machine learning technique is presented. Total 28 features have been tested for model training. New features like domain-based feature, HTML and JavaScript-based feature are also experimented for malicious URL detection using ML methods. Novelty of the approach is parallel implementation of feature extraction, which improved processing time efficiency of the model approximately by number of cores of the executing machine. This helped in improving feature extraction time while training and prediction with updated and big datasets. We have trained the model with lexical, binary, address bar-based features for final results. The presented approach is resulted in malicious URL detection ML model which is processing time efficient with high prediction accuracy. Random forest algorithm with tuning of hyperparameters performed best in our case with approximately 99% accuracy with 514,736 URLs dataset. An approach from malicious URL detection to prevention is also presented in the paper by which, how futuristic access to the predicted malicious URLs can be prevented to improve security posture. The results of this paper work can be adopted in information security systems to enhance security posture. In the future for better results with different type of datasets of URLs, content-based URL features will be experimented, and some advanced soft computing techniques will be explored.

References

Alexa the Web Information Company. https://www.similarweb.com/website/alexa.com. Accessed 1, 2023

Bhattacharyya S, Bhattacharya S (2021) Practical handbook of machine learning. Career Launcher Infrastructure Pvt. Ltd. Publisher

Browser Defender. https://chrome.google.com/webstore/detail/microsoft-defenderbrowse/bkbeeeffjjeopflfhgeknacdieedcoml. Accessed 12, 2022

Chiba D, Tobe K, Mori T, Goto S (2012) Detecting malicious websites by learning IP address features, pp 29–39. https://doi.org/10.1109/SAINT.2012.14

Comodo Site Inspector. https://www.comodo.com/news/website-scanner-malwareandblacklist-monitoring-for-websites-from-comodo-site-inspector/. Accessed 12, 2022

Domain Reputation API. https://www.urlvoid.com/. Accessed 12 2022

Google Safe Browsing Diagnostic. https://transparencyreport.google.com/safe-browsing/. Accessed 12 2022

Kaggle Datasets. https://www.kaggle.com/datasets. Accessed 1, 2023

Kim BI, Im C-T, Jung H-C (2011) Suspicious malicious web site detection with strength analysis of a java script obfuscation. Int J Adv Sci Technol

Malicious_n_Non-MaliciousURL. https://www.kaggle.com/datasets. Accessed 1, 2023

Naveen INVD, Manamohana K, Verma R (2019) Detection of malicious URLs using machine learning techniques. Int J Innovative Technol Exploring Eng (IJITEE) 8(4S2)

Online Link Scan. https://opentip.kaspersky.com/. Accessed 1, 2023

Pao H, Chou Y, Lee Y (2012) Malicious URL detection based on Kolmogorov complexity estimation. In: IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology, pp 380–387. https://doi.org/10.1109/WI-IAT.2012.258

Patgiri R, Katari H, Kumar R, Sharma D (2019) Empirical study on malicious URL detection using machine learning. In: 15th international conference on distributed computing and internet technology (ICDCIT), Bhubaneswar, India, Jan 10–13, Proceedings, pp 380–388

Prasad SDV, Rao KR (2021) A novel framework for malicious URL detection using hybrid model. Turk J Comput Math Educ 12(7):68–76

Sahoo D, Liu C, Hoi SCH (2019) Malicious URL detection using machine learning: a survey. https://doi.org/10.48550/arXiv.1701.07179

Seifert C, Welch I, Komisarczuk P (2008) Identification of malicious web pages with static heuristics. In: Australasian telecommunication networks and applications conference, ATNAC 2008, pp 91–96

Sheng S, Wardman B, Warner G, Cranor LF, Hong J, Zhang C (2009) An empirical analysis of phishing blacklists. In: Proceedings of sixth conference on email and anti-spam (CEAS)

Sinha S, Bailey M, Jahanian F (2008) Shades of grey: on the effectiveness of reputation based blacklists. In: 3rd international conference on malicious and unwanted software (MALWARE), pp 57–64

Sucuri. https://sucuri.net/. Accessed 1, 2023

Unmask parasites. https://unmask.sucuri.net/. Accessed 12, 2022

Unshorten.It. http://www.unshorten.it. Accessed 1, 2023

URL shortening service. https://goo.gl/. Accessed 12, 2022

URL shortening service. https://bitly.com/. Accessed 12, 2022

VirusTotal–Home. https://www.virustotal.com. Accessed 12, 2022

WHOIS Database. https://who.is. Accessed 1, 2023

Xuan CD, Nguyen HD, Nikolaevich TV (2020) Malicious URL detection based on machine learning. Int J Adv Comput Sci Appl (IJACSA) 11(1)

A Cluster-Based Routing Protocol for Wireless Sensor Networks-Based IoT System

Irak Rigia, Rajdeep Chakraborty, Bhaskar Bhuyan, and Hiren Kumar Deva Sarma

Abstract A cluster-based routing protocol for Internet of Things (IoT) network which is based on wireless sensor networks has been proposed in this paper. The basic clustering approach is adopted from the well-known Hybrid Energy Efficient Distributed (HEED) clustering protocol. This work mainly focuses on the data forwarding model through which data originated at source node can reach the base station, which is the destination. The intermediate sensor nodes are considered to be static in nature. Achieving energy efficiency being the primary objective, the protocol follows a cluster-based hierarchical data delivery model. The proposed protocol achieves energy efficiency. Simulation results depicting the performance of the proposed protocol have been reported. Future scopes of the work have also been outlined.

Keywords Internet of Things (IoT) · Wireless sensor networks · Routing · Energy efficiency · Node clustering

1 Introduction

With the massive advancements of information and communication technology, it has become very common to build devices of microsizes which are much more capable of doing things than we can even imagine. One of the trending applications is Internet of Things (IoT). Prior to the evolution of IoT, wireless sensor network (WSN) was the trending technology. Wireless sensor network (WSN) is consisting of tiny

I. Rigia · B. Bhuyan
Department of IT, Sikkim Manipal Institute of Technology, Rangpo, Sikkim 737136, India

R. Chakraborty
Department of CSE, Chandigarh University, Chandigarh, Punjab 140413, India

H. K. D. Sarma (✉)
Department of IT, Gauhati University, Guwahati, Assam 781014, India
e-mail: hirenkdsarma@gauhati.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_49

sensor nodes capable of receiving data and communicating data. These nodes consist of sensory circuitries for sensing natural phenomena like temperature, humidity, fire, etc. These nodes are usually resource constrained, and thus, they have limited power or energy, lower computational capability and memory, and also have lower communication range and bandwidth.

Routing is essential and also an important task in the entire WSN system, for effective delivery of data in the destinations. In this paper, a cluster-based routing protocol is proposed. Energy requirement has been given utmost consideration, while the protocol was designed. The cluster-based routing protocol consists of two phases, namely cluster setup phase and data communication phase. In this protocol, aim is to develop technique to deal with how data will be sent from source node to the base station using multiple hops. The most important parameter considered is the residual energy of nodes which are participating in data forwarding. Total energy dissipation will be reduced by distributing the energy usage among those participating nodes. It will be done by periodically allowing nodes to sleep or relax which are otherwise engaged in contributing to the process of routing for quite some time. Therefore, the proposed protocol will accomplish all the goals as mentioned above.

Some of the primary factors that must be taken care of while designing an effective routing protocol are mentioned below. These are energy, processing, and storage constraints for the nodes; the type of application that the network will be used for; highly frequent changes in the topology due to the presence of mobile nodes; lack of position awareness of the sensor nodes; redundancies in the data generated by sensors; building global addressing schemes such as the traditional IP-based addressing of nodes which may create additional overhead of node identity maintenance and unique characteristics of the application based on WSN.

A typical sensor network setup is depicted below (Fig. 1). The constituting components of a sensor node are also depicted in Fig. 1. Sensor node has different constituent units such as sensing unit, processor, storage, transceiver, and power unit apart from mobilizer, position finding system, and power generator which are optional units. The sensor nodes collect data from the environment and send the same to the base station, and finally, the base station forwards the data to the user over Internet.

Clustering of nodes plays a vital role in achieving energy efficiency and overall stability in the network.

Following are some of the important objectives of clustering (Abbasi 2007; Meng 2016):

Load balancing: The total work of a sensor network is divided. The cluster heads (CHs) perform data aggregation and forwarding, and some other intra-cluster management duties. Other non-CH nodes perform the sensing task and local data forwarding task.

Fault tolerance: In some applications, nodes may be located in harsh environments, due to which some nodes may be damaged or destroyed. With clustering, there are better ways to deal with these situations.

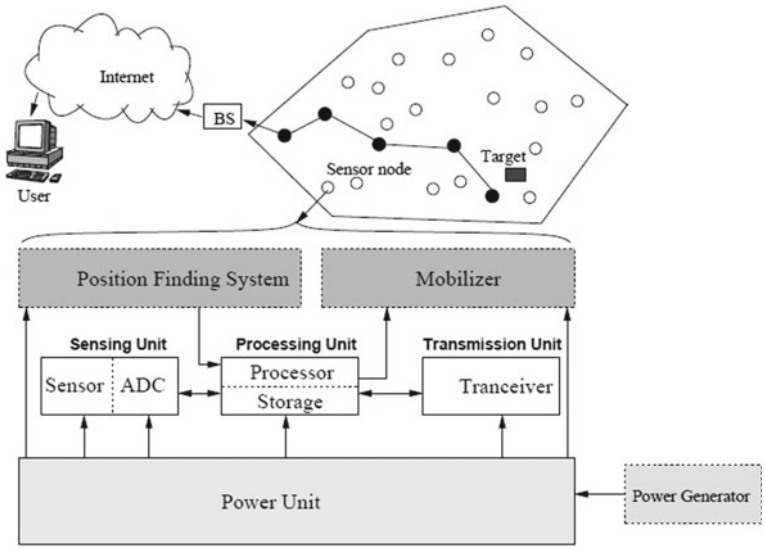


Fig. 1 Wireless sensor network setup and sensor node components

Increased connectivity and reduced delay: Connectivity among CHs is an important requirement. This factor comes into play when CHs are chosen among the sensor nodes rather than specially deployed CH nodes. In any way, data should be able to reach the BS. Now, this also brings into picture the latency of data transfer. Connectivity should not be fulfilled without considering the delay that will occur in order to transfer the data.

Minimal cluster count: The CHs are resource-rich nodes, in general. The network designer often likes to employ the least number of these nodes since they tend to be more expensive and vulnerable than ordinary sensors.

Maximal network longevity: This objective is a concern since sensor nodes are energy constrained, and the network lifetime relies on the way their energy is used.

In this paper, a cluster-based routing protocol for wireless sensor network-based IoT system has been proposed. The proposed protocol achieves energy efficiency, and in turn, it prolongs the lifetime of the network. The rest of the paper is organized as mentioned below. Section 2 reports few related works in the same direction. The proposed protocol is presented in Sect. 3. Simulation results are presented in Sect. 4. The paper is concluded in Sect. 5.

2 Related Works

There are several papers published in recent years that deal with routing in wireless sensor network-based IoT networks. The protocols such as LEACH (Heinzelman 2000), TEEN (Manjeshwar 2001), APTEEN (Manjeshwar 2002), HEED (Younis 2004), PEGASIS (Lindsey 2002) are a few early protocols developed for wireless sensor networks. All the above-mentioned protocols deal with routing and data forwarding in wireless sensor networks. Along with the growth of Internet of Things (IoT) technologies, there have been developments in the line of wireless sensor network-based IoT also. Few such routing protocols developed for WSN-based IoT are reported in (Lenka 2018, 2019; Shen 2017). Energy efficiency is important as the sensor nodes are energy constrained. The issue of energy efficiency in WSN is studied in (Al-Turjman 2019; Boukerche 2008; Dong 2016; Sarkar 2021; Shen 2017; Verma 2020). Routing in general and energy efficient routing in specific is a well-studied problem in WSN although novel solutions are still awaited for different applications. Energy efficiency in IoT networks is also an important problem to investigate. Few energy efficient routing protocols for WSN are reported in (Al-Karaki 2004; Boukerche 2008; Dong 2016; Sharma 2015, 2017a, 2017b; Tan 2015). Clustering plays a vital role in providing stability and energy efficiency to WSN and various clustering protocols developed for WSN are reported in (Abbasi 2007; Meng 2016). Security issues in IoT are also attracting attention of the researchers as security in IoT networks is a big concern. Few such solutions are reported in (Al-Turjman 2021, 2019; Deebak 2021).

3 Proposed Protocol

In most of the cluster-based routing protocols, each cluster head communicates with the base station (BS) in a single hop or direct fashion. But in this approach, if necessary, multiple cluster heads are used for delivering packets at the base station which is located far away. This is because in some situations the clusters are very far from the base station and would require maximum transmission power to send data directly, resulting in high energy dissipation. To address this issue, the proposed protocol will communicate in multi-hop fashion to reach the base station. In this protocol, total energy dissipation will be reduced by distributing the energy usage among nodes that are participating in providing a path to the base station. It will be done by periodically allowing nodes to sleep or relax.

Following are some assumptions made in this work.

- The IoT network consists of static nodes.
- Nodes can tune their transmission power levels as required.
- Interference of signals—considered to be null.
- Energy models of all the nodes are same.

- Clusters have been formed using a standard clustering protocol like HEED (Younis 2004).
- The clustering algorithm will elect Cluster Heads (CH) as well as Assistant Cluster Heads (ACH); the ACH will be one of the sensing nodes.

Different parameters and symbols used in this work are enlisted below.

- T_{CP}: Duration of clustering phase.
- T_{NO}: Duration of network operation or data forwarding phase.
- *u*: The node in the context of which the processes will be explained.
- *{M}*: Set of those nodes in the cluster neighbour table (or assistant cluster neighbour table) of *u* whose distances to the BS are less than *u*; it is sorted in increasing order of their distance to BS.
- *v*: Next hop node of *u*.
- *BS*: Base Station.
- *distance*(*i*, *j*): Distance from node *i* to *j*.
- *energy*(*i*): Residual energy of node *i*.
- *energyGap*: If the difference between the residual energy of a node selected from the cluster neighbour table and the maximum in the table is less than this value, then the node will be rejected as a next hop for a moment until it reaches the balance point.
- Each CH (if necessary, each node in the n/w) has an information which contains a sequence of selected divisions of its energy level; for example: if the maximum energy of a node is considered to be 100 J, and the number of divisions is selected to be 5 (say), then the sequence of threshold values will be {100, 80, 60, 40, 20}.
- CH nodes maintain a table called *cluster neighbour table*. This table contains information about neighbouring cluster heads of the node CH node *u*.

Structure of this table (in the CH node u) is shown in Fig. 2. There are six fields as shown in the figure. Significance of various fields are mentioned below.

Here, <Address(neighbour_CHi)> : Address of the ith neighbour; <Distance_to_me> : Distance from the ith neighbour to node u itself; <Distance_to_bs> : Distance from the ith neighbour to the BS; <Energy_level> : Energy level of node I; <Current_state> : Active or Resting; <Assistant_node_ID> : Pointer to the entry of the assistant node in the assistant neighbour table. Few principles adopted regarding this table are enlisted below.

- Node entries in this table are sorted in increasing order of <distance_to_bs>.

Address (neighbour_CHi)	Distance_to_me	Distance_to_bs	Energy_level	Current_state	Assistant_node_ID

Fig. 2 Cluster neighbour table structure

Sender ID	DestinationID=BROADCAST ADDRESS	PacketType=RELAX

Fig. 3 Relaxation request packet structure

- Similar to the above table, there is one more table that stores information about all the assistant CH nodes. This table is linked by the main cluster neighbour table's column <Assistant_node_ID>.
- All the information stored by a CH will also be stored by its assistant CH.
- *Relaxation Packet*: When the value of the residual energy of a node reaches the upper bound of an energy partition, then it broadcasts a packet to all its neighbour CH (including assistant CHs) informing about that it is going in relaxation or sleep mode. Since this information is crucial to be successfully shared, this packet will have the highest priority in order to assure that this information reaches to all its neighbour. The structure of the packet is shown in Fig. 3.

After the successful broadcast of this packet, the assistant CH of the particular CH will become active and be responsible for routing data. The node in the relaxation period will go to sleep for certain time period. When the sleep time is over, it will broadcast a relaxation packet again but with the parameter PacketType = ACTIVE; this packet will be called an *Activation packet*.

After the successful broadcast of this packet, the node will be ready to be selected for data forwarding again until its next relaxation period. Also its assistant cluster head will go to its original work. This process will repeat until the energy is completely exhausted.

Working of the protocol:

There are two phases in the protocol, namely cluster setup phase and data forwarding phase. In the context of the proposed protocol, the cluster setup phase and the data forwarding phase are discussed below in detail.

Cluster setup phase:

It is mentioned that the clusters will be formed by the standard node clustering protocol named as Hybrid Energy Efficient Distributed (HEED) (Lindsey 2002) clustering protocol.

HEED is a node clustering protocol that can be used for clustering nodes in a given wireless sensor network setup. This protocol is distributed in nature. According to this protocol, suitable sensor nodes are selected for the role of cluster head (CH) by considering energy and cost of communication both. In HEED, each node is mapped to one cluster only. It consists of multiple phases, and those are summarized below.

Initialization phase: This is the first phase of the algorithm. A certain percentage of the total number of deployed sensor nodes is set as the percentage of cluster head nodes in the field. The percentage of the cluster head nodes which is set initially (C_{prob})

is utilized in order to control the announced number of cluster head nodes among the sensor nodes in the field. Then, each of the sensor nodes sets their corresponding probabilities of getting selected as cluster head (CH_{prob}). This value is computed as: $CH_{prob} = C_{prob} * E_{residual}/E_{max}$. Here, $E_{residual}$ indicates the current energy level in the respective sensor node. Moreover, E_{max} indicates the energy level which is the maximum. Again maximum energy level represents the energy level of a battery which is fully charged. It is important to note that CH_{prob} is considered to be such a value which does not come down beyond a certain threshold p_{min}. This threshold is assumed to be inversely proportional to E_{max}.

Repetition phase: This phase (i.e. the second phase) is iterative in nature. Each sensor node passes through multiple iterations in this phase. The iterations continue till the sensor nodes identify the corresponding cluster head nodes. The condition for identifying a cluster head node is that the node must be able to communicate with the identified cluster head node consuming the minimum energy level (i.e. the required transmission power has to be the minimum incurring the minimum cost). In case a sensor node does not find a cluster head and thus does not hear from any cluster head node, then the sensor node will identify itself as the cluster head node. In this circumstance, this sensor node will send an announcement message to all the neighbouring nodes and thus informs the neighbours about this status change (i.e. from ordinary sensor node to cluster head node). The sensor nodes then double their corresponding CH_{prob} value; immediately another iteration in this phase itself starts. This phase will stop when the CH_{prob} value becomes 1. In this way there can be two types of cluster head status of respective sensor node; these two types are as follows:

Tentative status: The first status is *tentative cluster head*; this is the status when $CH_{prob} < 1$. This status (i.e. tentative) may change and the node may become a regular node (at a later stage) provided a different cluster head node is found with lower cost.

Final status: The second status is *permanent cluster head*. This status is achieved when the corresponding CH_{prob} becomes equal to one ($CH_{prob} = 1$).

These two status information are announced to the neighbouring nodes in the sensor field.

Finalization phase: This is the third and final phase. In this phase, every sensor node needs to make a decision regarding their status, and this decision is going to be considered as final. The sensor node will either identify the least cost cluster head node, or it will pronounce itself as the cluster head.

It is assumed that each cluster head will also elect its own assistant cluster head. In addition, each cluster head as well as assistant cluster head nodes will keep a table called cluster neighbour table. The entries in this table are the information about the neighbouring cluster heads. Another table of the same structure will be stored which will consist of neighbouring assistant cluster heads information. These tables will be helpful in making a routing decision.

Data forwarding phase:

In this sub-section, the process of data forwarding to the base station using inter-cluster communication has been detailed. The goal of this protocol is to maximize network lifetime by distributing the load of data routing among nodes, or more specifically cluster heads and assistant cluster heads since these are the nodes responsible for forwarding data to the base station. Various algorithms used by this protocol are discussed below in detail.

Selection process of the next hop node:

The next node is first chosen from those in the set *{M}* and if none is found to be valid, it chooses from the set *{M}* of the assistant cluster head table. To produce a path/route from source node u(CH) to the sink BS, the node(CH) u chooses a next hop node(CH) v if the following conditions are met:

$$distance(v, BS) < distance((\{M\} - v), BS) < distance(u, BS), \tag{1}$$

$$(energy(highest\_energy\_node\_in\_CHTable) - energy(v)) < EnergyGap, \tag{2}$$

$$v \text{is not in relaxation mode.} \tag{3}$$

Here, the expression ($\{M\}$-v) represents all the nodes in the set $\{M\}$ excluding v.

The process of routing is depicted in Fig. 4. Broadcast or relax packet transmissions are represented through yellow line. Various data paths for the cluster head nodes are represented through solid (for new data path) and dashed (for old data path) lines. The process of selecting the next hop is depicted in Fig. 5. The entire flow chart depicts the process of selecting the next hop in the route.

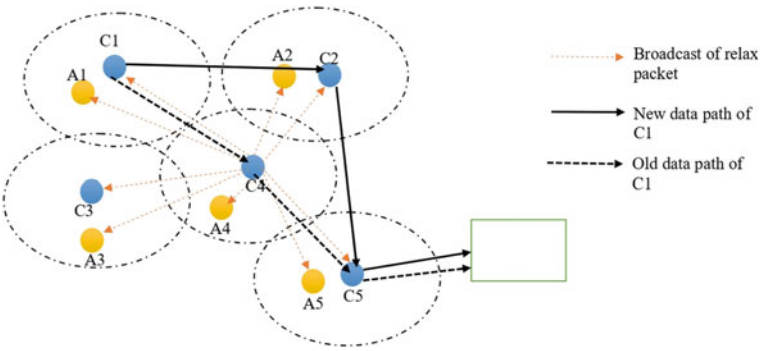


Fig. 4 Routing process

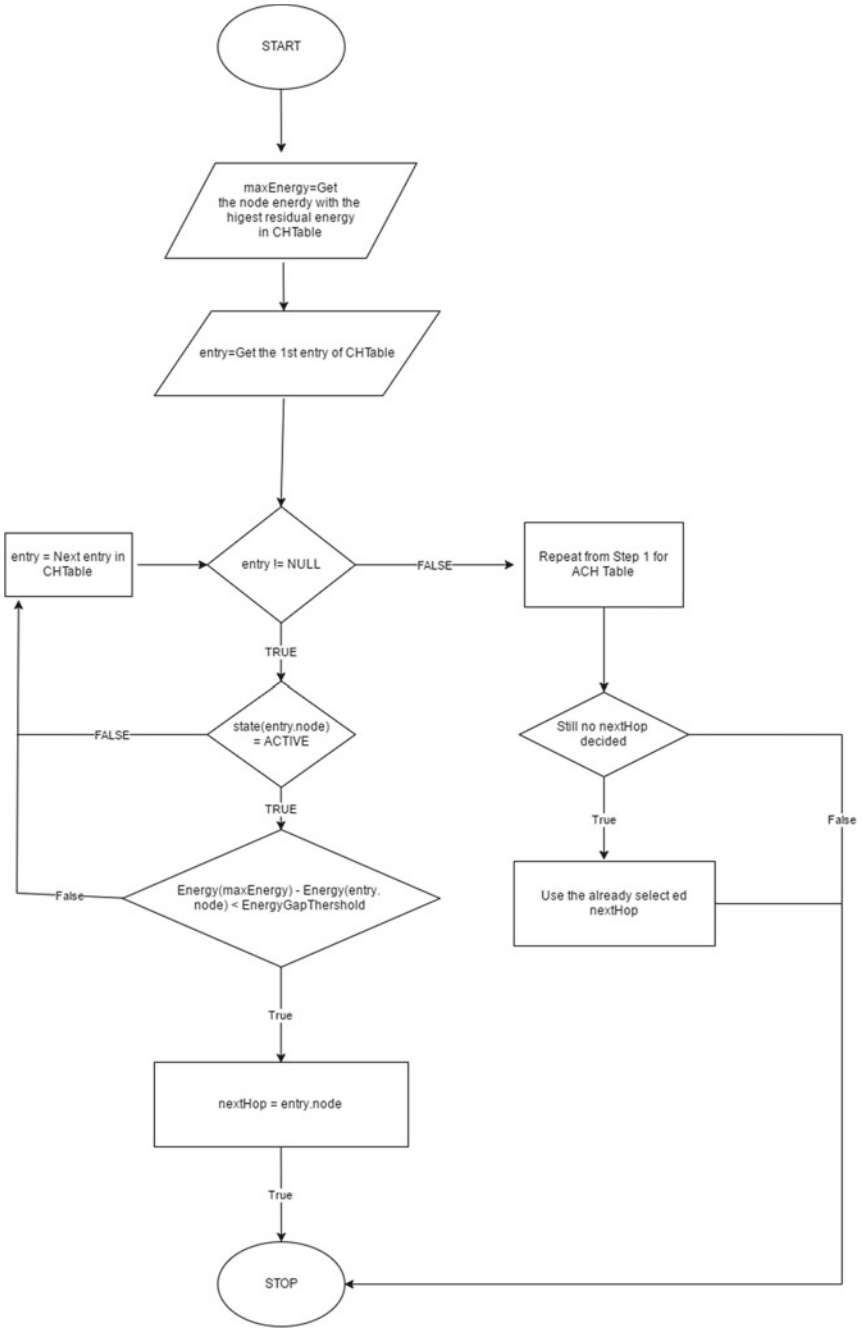


Fig. 5 Next hop selection

4 Simulation Results

The proposed protocol was simulated in C++/windows environment. Different parameters and respective values used in the simulation are mentioned in Table 1.

Following are few important principles adopted during the simulation.

- Packet drop is caused only due to congestion in a node and overflowing the packet queue.
- The simulation time does not consist of the cluster setup phase; only the data forwarding phase is simulated at one go.

Goal was to analyse how energy is dissipated throughout the network during its lifetime. Other important performance metrics were throughput, end-to-end delay, and packet drop ratio.

Analysis regarding **average residual energy** versus time is presented in Fig. 6. Average residual energy is calculated by taking the average of the residual energy of all the nodes over a periodic time interval.

It was observed that with more number of nodes in a network, the average residual energy decreases faster. This is because the CH and assistant CH nodes that are closer or one hop away from the base station are chosen often as a forwarding nodes by most nodes that further or multiple hop away from the base station. This problem can be solved by electing adequate number of CH and assistant CH near the base station.

Analysis regarding **average throughput** versus number of nodes is presented in Fig. 7. Here, the average throughput for various network sizes are shown. This result is obtained by computing the average of the throughputs of each packet delivered successfully to the base station.

Table 1 Simulation parameters

Parameter	Value
Number of energy partitions	5
Energy gap	0.0005 J
Relaxation time duration	1 s
Initial energy	2 J
Energy required for TX ($E_{\text{TX-elec}}$)	50 nJ/bit
Energy required for RX ($E_{\text{RX-elec}}$)	50 nJ/bit
Energy required for transmit amplifier $(\varepsilon_{\text{amp}})$	100 pJ/bit/m^2
Energy required per sensing data	40 nJ/bit
Transmission range	260 m
No. of nodes	20, 40, 60, 100
Area of sensor fields	(100 × 100) m, (1400 × 700) m
Simulation time	1000 s

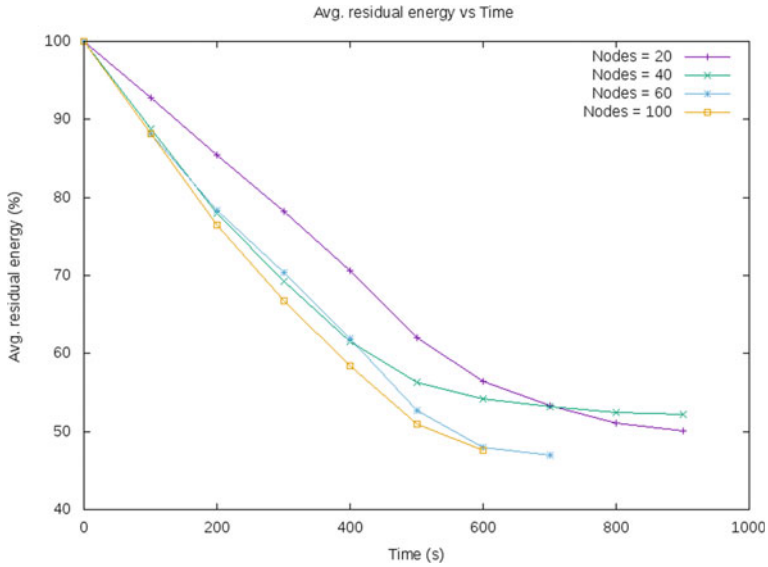


Fig. 6 Average residual energy versus time

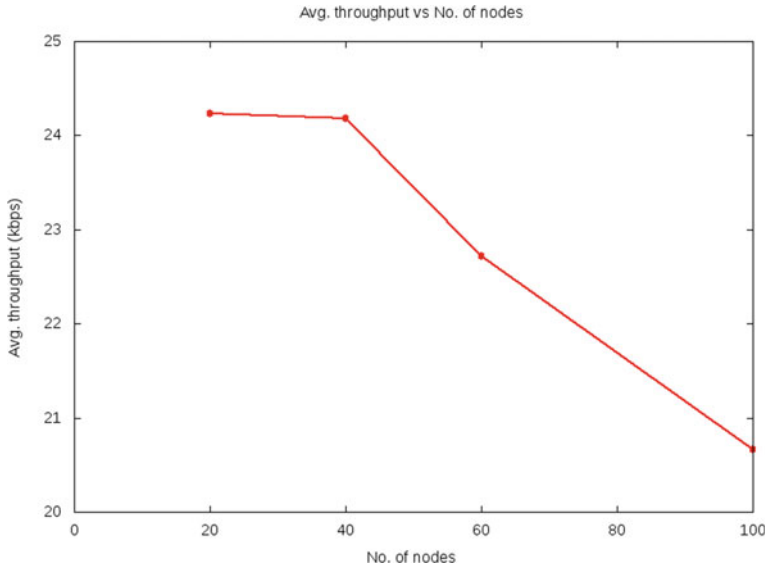


Fig. 7 Average throughput versus number of nodes

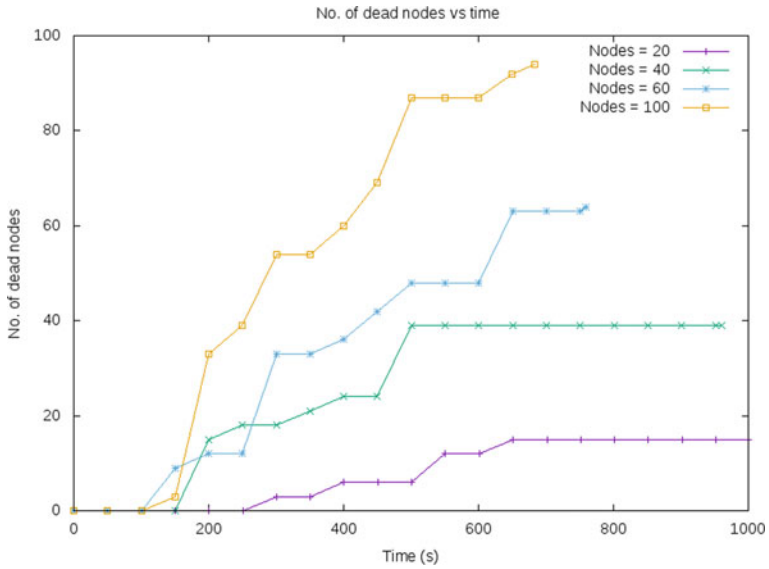


Fig. 8 Number of dead nodes versus time

It has been observed that the average throughput is low for networks with more number of nodes. From similar observation described in Sharma (2017b), the number of successfully delivered messages (packets) increases if the percentage of CHs in a network is higher. In our case, higher percentage of CHs and assistant CHs will allow more paths to be available.

Analysis regarding **number of dead nodes** versus time is presented in Fig. 8.

This graph represents the number of nodes that die out of energy with time. As we have already observed in the previous section that the total energy depletes faster with more number of nodes, this in turn explains that the number of dead nodes will increase faster with increase in the total number of nodes.

Analysis regarding **packet drop ratio** is presented in Fig. 9.

This graph shows the percentage of packets dropped for different network size. Packet drops are also caused due to congestion in nodes which are closer to the BS. This can be improved if more numbers of forwarding nodes are located near the base station.

5 Conclusion and Future Scope

A cluster-based hierarchical routing protocol for IoT network based on wireless sensor networks has been proposed in this paper. The protocol exploits HEED (hybrid, energy efficient, distributed clustering) protocol for clustering the nodes in the network. Once the clusters are formed, data are forwarded towards the base station

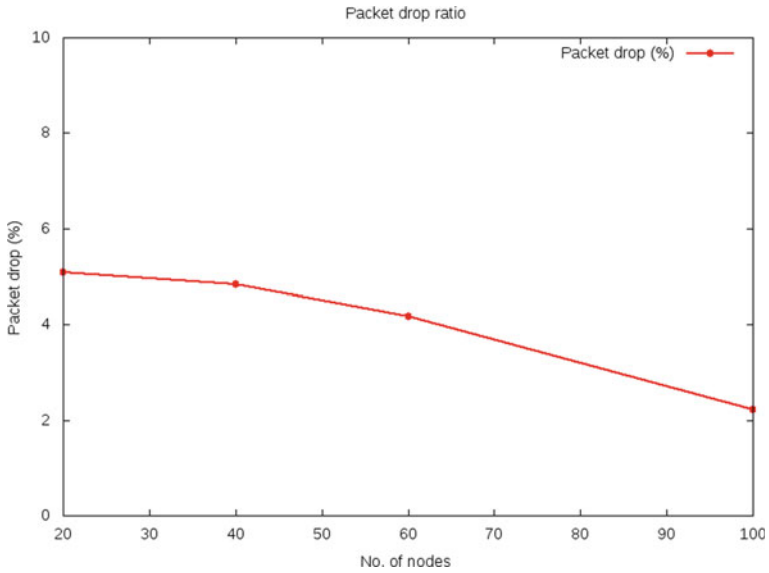


Fig. 9 Packet drop ratio analysis

considering the energy efficiency in the overall network system. Data forwarding is the major focus in this work. The proposed protocol is simulated through C++ programs written in windows environment. The performance analysis of the proposed protocol reveals its energy efficiency. In future, the proposed idea regarding data forwarding in an IoT network is planned to be implemented in a test bed consisting of various hardware units. Moreover, node mobility may also be added to the system in order to achieve more realistic scenario. Performance of the proposed protocol may be evaluated in test bed, in future.

References

Abbasi: A survey on clustering algorithms for wireless sensor networks. Computer Communications 30 (14), 2826–2841 (2007).

Al-Karaki (2004) Routing techniques in wireless sensor networks: a survey. IEEE Wirel Commun 11(6):6–28

Al-Turjman F (2019) A. Energy monitoring in IoT-based ad hoc networks: an overview. Comput Electr Eng (76):133–142

Al-Turjman FM (2019) An overview of security and privacy in smart cities' IoT communications. Trans Emerg Telecommun Technol (Spl. Issue):1–19

Al-Turjman F (2021) Seamless authentication: for IoT-big data technologies in smart industrial application systems. IEEE Trans Ind Inf 17:2919–2927

Boukerche (2008) An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. Mob Networks Appl 13(6):614–626

Deebak BD (2021) Lightweight authentication for IoT/Cloud-based forensics in intelligent data computing. Future Gener Comput Syst 116:406–425
Dong M (2016) RMER: reliable and energy-efficient data collection for large-scale wireless sensor networks. IEEE Internet of Things J 3:511–519
Heinzelman WR (2000) Energy-efficient routing protocols for wireless microsensor networks. In: Proceedings of 33rd annual Hawaii international conference on system and sciences. IEEE, Hawaii, USA
Lenka RK (2018) Building scalable cyber-physical-social networking infrastructure using IoT and low power sensors. IEEE Access (6):30162–30173
Lenka RK (2019) Building reliable routing infrastructure for green IoT network. IEEE Access (7):129892–129909
Lindsey S (2002) PEGASIS: power-efficient gathering in sensor information systems. In: Proceedings of IEEE aerospace conference. IEEE, Big Sky, MT, USA
Manjeshwar A (2001) TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Proceedings of 15th international parallel and distributed processing symposium (IPDPS). IEEE, San Francisco, CA, USA, pp 2009–2015
Manjeshwar A (2002) APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless. In: Proceedings 16th international parallel and distributed processing symposium (IPDPS). IEEE, Ft. Lauderdale, FL, USA
Meng X (2016) A grid-based reliable routing protocol for wireless sensor networks with randomly distributed clusters. Ad Hoc Netw 51:47–61
Sarkar S (2021) Green IoT: design goals, challenges and energy solutions. In: Proceedings of the 6th international conference on communication and electronics systems (ICCES). Coimbatore, India, pp 637–642
Sharma S (2015) A virtual grid-based backbone structure type scheme for mobile sink based wireless sensor networks. In: Proceedings of the 2015 international conference on advanced research in computer science engineering & technology (ICARCSET 2015). Unnao, India, pp 1–5
Sharma S (2017a) MSGR: a mode-switched grid-based sustainable routing protocol for wireless sensor networks. IEEE Access 5:19864–19875
Sharma S (2017b) Rendezvous based routing protocol for wireless sensor networks with mobile sink. J Supercomput 73:1168–1188
Shen J (2017) An efficient centroid-based routing protocol for energy management in WSN-assisted IoT. IEEE Access 5:18469–18479
Tan ND (2015) SCBC: sector-chain based clustering routing protocol for energy efficiency in heterogeneous wireless sensor network. In: Proceedings of the 2015 international conference on advanced technologies for communications (ATC). Ho Chi Minh City, Vietnam, pp 314–319
Verma G (2020) A comparative study based on different energy saving mechanisms based on green internet of things (GIoT). In: Proceedings of the 8th international conference on reliability, infocom technologies and optimization (Trends and future directions) (ICRITO). Noida, India, pp 659–666
Younis O (2004) HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans Mob Comput 3(4):366–379

A Novel Approach for Deep Learning Based Video Classification and Captioning using Keyframe

Premanand Ghadekar, Vithika Pungliya, Atharva Purohit, Roshita Bhonsle, Ankur Raut, and Samruddhi Pate

Abstract Video classification has gained a significant importance in recent years because of the enormous rise in internet usage and because of the emergence of deep learning models as a tool to automatically classify videos. In this work, an innovative approach for video classification and captioning is proposed by extracting keyframes from the videos. A transfer learning approach is used to classify the videos. A 99.76% training accuracy is observed on the ISRO dataset and 99.89% training on UCF101 dataset was achieved by using the proposed algorithm. The proposed system also discusses the traditional approach of video classification using the CNN + RNN architecture and shows that algorithm outperformed the eminent approach. Finally, an approach for video captioning using a DenseNet201 + GRU based encoder-decoder model is compared. A METEOR score of 0.29 was achieved on the ISRO dataset and a METEOR score of 0.299 on the MSR-VTT dataset.

Keywords CNN · Encoder-decoder · ISRO · Video classification · Video captioning · Keyframe extraction

1 Introduction

Today is an era where text, images, music, and videos are all digitized. On personal electronic devices like laptops and mobile phones, images and videos take up greater space. In everyday life, videos play a significant part and are just a series of images. The most popular video file format is called Audio Video Interleave (.avi) (Shankar et al. 2012). Media platforms such as blogs, forums, business networks, photo-sharing websites, and social networking sites are used to share images and videos (Ramesh and Mahesh 2018). A report estimates that 2.46 billion people used social networks globally in 2017. As of September 2017, Facebook was the most popular social network worldwide (Bhat et al. 2019). One of the significant research areas in the field

P. Ghadekar (✉) · V. Pungliya · A. Purohit · R. Bhonsle · A. Raut · S. Pate
Vishwakarma Institute of Technology, Pune 411037, India
e-mail: premanand.ghadekar@vit.edu

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_50

of computer vision is video classification. Classification is a term used in data mining to group various types of data into certain categories (Bhat and Malaganve 2021). Data is often considerably more difficult to locate and comprehend. Assisting viewers to find videos of interest more quickly is the primary goal of a video classification system. The main aim behind video classification is to categorize a particular video under a particular category, so it in turn makes it easier for interested folk to select it accordingly.

One of the central issues in computer vision is image captioning or the automatic creation of descriptions for images. Deep Learning techniques have made significant advancements in this area. However, the caption generation model must also be capable of identifying the links between the items in the image and expressing those relationships in natural language. Deep learning research faces a significant difficulty since it must be able to replicate how humans convert the received information from the visual domain to the realm of natural speech (Katiyar and Borgohain 2021). It's quite difficult for people to accurately describe a scenario present in a video or an image. Ways to combine the science of comprehending human language with the science of automatically extracting and analyzing visual information are being investigated by computer science researchers to build computers with this power. This procedure creates a wealth of possibilities in real-world application fields, including assistance for those with visual impairment, application of self-driving cars, and automatic video subtitling, video surveillance, sign language translation, human–robot interaction and many more (Oluwasammi et al. 2021).

The paper discusses a unique approach for Video Classification and Captioning on the ISRO dataset. In addition to classifying a given video to a class, it also generates a caption for that video. At the very beginning of the paper, is the Introduction (Sect. 1). Then is the Literature Review (Sect. 2) section, which contains the summary of a few papers being studied related to this topic. Methodology forms the (Sect. 3) section, which details the implementation of the proposed solution step by step. Part A mentions the steps undertaken for Video Classification. Part B mentions Image Captioning. Finally, comes the Results and Discussion (Sect. 4) section, wherein the table of comparisons for all these models, all these approaches are plotted. Conclusion (Sect. 5) of the paper, with the Future scope of the project included in it.

2 Literature Survey

Wu et al. (2017), this paper discusses in great length the methodologies for video classification as well as video captioning. It discusses image-based video classification, end-to-end CNN architectures, and explores LSTM to model the temporal dynamics. Similarly, it also enlists some approaches for video captioning which include—template based language models, sequential learning and common framework using 3D CNN and LSTM. Finally, the authors discuss the well-known benchmarks for both classification and captioning and discuss the challenges by drawing comparisons using the different evaluation metrics.

Jiang et al. (2018), in this paper, a brand-new hybrid deep learning architecture is proposed that combines relevant cues from many modalities, such as static spatial appearance data, motion patterns within a limited time window, auditory data, and long-range temporal dynamics. To calculate spatial, motion, and audio features, respectively, three distinct CNN models that operate on static frames, stacked optical flow pictures, and audio spectrograms were used. To use long-range temporal cues in movies, two LSTM models are applied with the spatial and motion data as input. With the help of a regularized feature fusion network, it is possible to recognize video semantics by using a single feature representation. Two datasets were used for the study, UCF—101 and Columbia Consumer Videos (CCV). The proposed model gives out an accuracy of 93.1% on the UCF-101 and 84.5% on the CCV.

3 Methodology

In this section, a detailed description of the Methodology has been elucidated. It has been divided into two Parts for Information Retrieval—Video Classification and Video Captioning. Figure 1 shows the detailed flowchart of the methodology. Each of the subsections further explains them in detail.

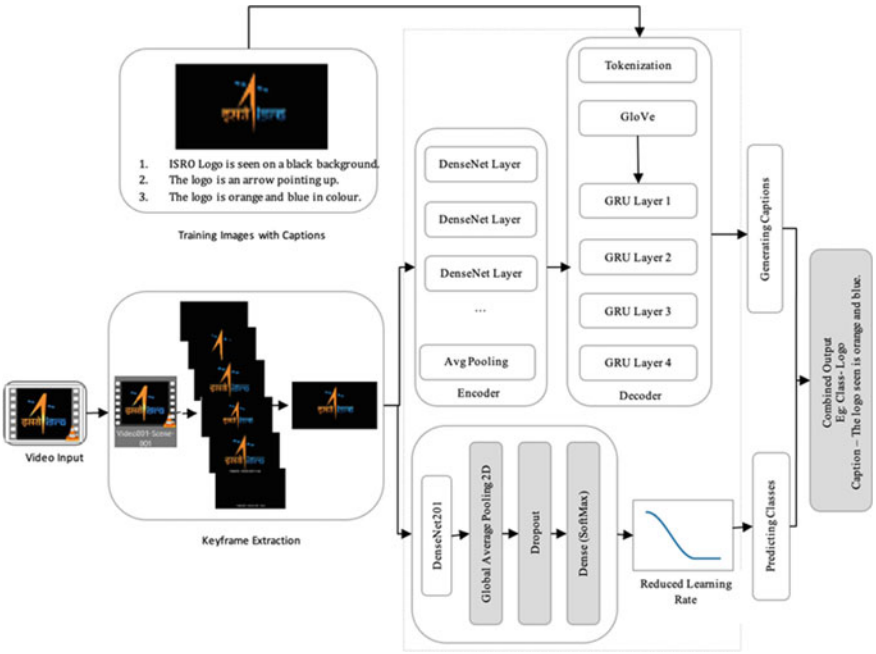


Fig. 1 Flow diagram for proposed methodology

3.1 Video Classification

Proposed Approach—Keyframe Extraction
The ISRO Dataset has 3611 videos categorized into 30 classes, including documentaries of ISRO missions and programs. The training folder contains these videos with a metadata file that tags each video to its respective class. Initially the frames from the videos are extracted and then converted into grayscale for Canny Edge Detection (Canny 1986). This is used to extract the edges of the frames. Next, the difference between each consecutive Edge Detected Frames is found. This is used to calculate the mean and the standard deviation of the differences which is then used to calculate the global threshold value. The formula for calculating the threshold value can be seen in (1).

$$\text{Threshold} = \text{Mean} + a * \text{Standard Deviation} \tag{1}$$

where "a" is a hyper parameter that can take any arbitrary value. In this case it has been set to 4. Subsequently, the threshold for every frame in the video is calculated. If the threshold is greater than the global threshold value, the frames are appended to a list. Then a median of the list is taken as the most important frame and finally stored as an image. The final time complexity for extracting keyframes from the entire training data is $O(m*n)$. Where, 'n' represents the number of videos in the training dataset and 'm' represents the number of frames extracted from a particular video. For a particular video time complexity is $O(m)$.

Figure 2 shows the sequential process to select the keyframe using proposed approach

Modelling

- *Converting Images into Required Size:*
- Before training the model with the extracted keyframes, they are converted into a shape of 224 * 224.
- *Label Smoothing and Reduce Learning Rate:*
- Label Smoothing, which is a type of output distribution regularization technique that prevents a neural network's tendency to overfit by softening the ground-truth labels in the training data to punish overconfident outputs has been used. The main goal of using this is to improve validation accuracy. The Loss Function used is Binary Cross Entropy, optimizer is Adam, and the initial learning rate is set to 0.0001. The 'ReduceLRonPlateau' Class from the Keras library has been used to keep a check on the validation accuracy, when there is no improvement seen in the validation accuracy, the learning rate is reduced.
- *Feature Extraction and Classification Model:*
- The dataset is divided into training and validation sets, the ratio used was 80:20. A Transfer Learning architecture was used for extracting the spatial features and then for classifying them. The base model used is DenseNet201 (Zhong et al. 1651) which is pre- trained on ImageNet. In the proposed system, a sequential

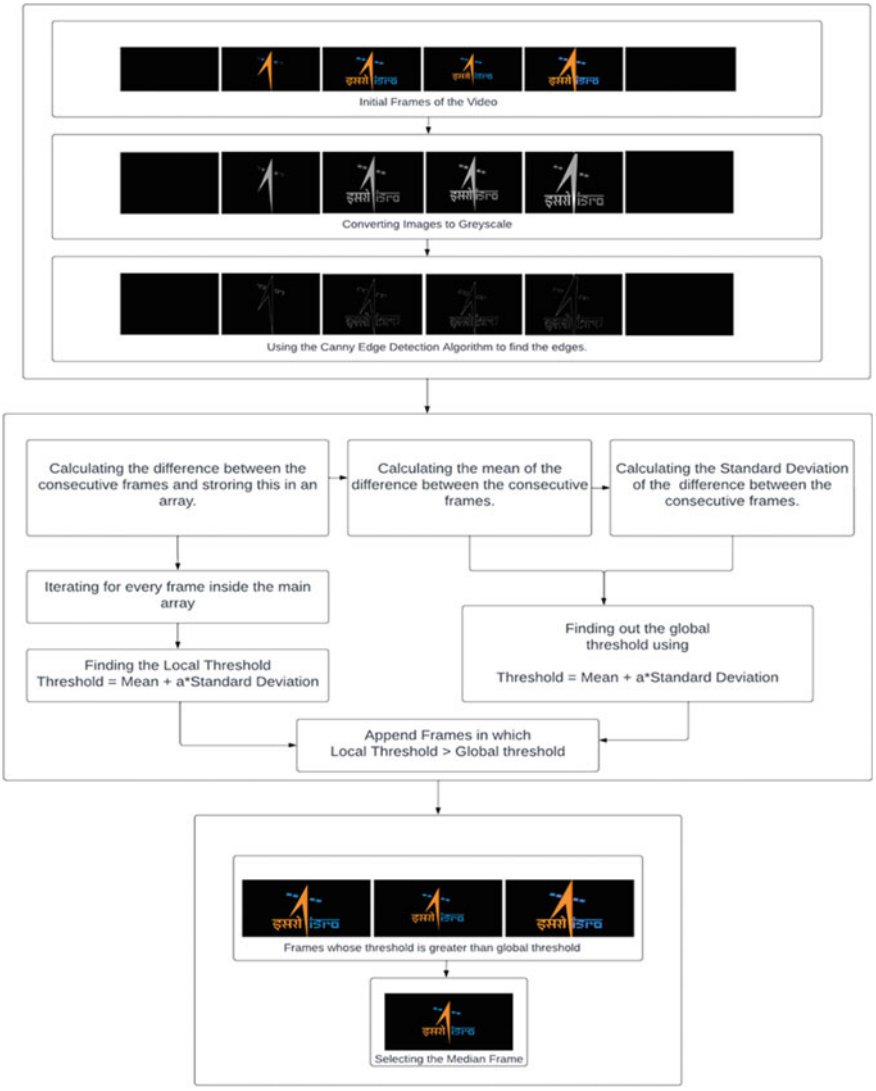


Fig. 2 Step-by-Step process to select the keyframe

model consisting of a Global Average Pooling Layer, a Dropout Layer and a Dense Layer is added. A Dropout Layer randomly sets the outgoing edges of neurons in hidden layers to 0 at each update of the training phase. The main aim is to create one feature map in the final Dense layer for each category that corresponds to the classification problem. Every feature map is averaged and a SoftMax activation function is used to predict the class probabilities. Next the model is fit on the training data using a batch size of 32, for 15 epochs. The training accuracy comes

out to be 99.76% and the validation accuracy is 76%. The validation accuracy has increased by a huge 24% as compared to the earlier CNN + RNN Model.

3.2 Video Captioning

In this section of the code, the proposed model for captioning the videos of the ISRO dataset is discussed in detail.

Data Pre-Processing Since the ISRO dataset only contains videos, a dataset for captioning was created by manually providing 3 captions for each keyframe in an excel file.

The excel file was read as a data frame and the rows of keyframes without the captions were dropped from the data frame. The next step was to clean and process the captions. This was done by removing the punctuation in the text, by removing single characters, by removing the numeric characters if present and finally converting each caption text into lower case. The dataset was then split into training and testing sets class-wise in the ratio of 85:15. This was done so that the training and testing set contained keyframes and their captions in the same ratio as above. Next, the captions are given a start mark 'ssss' and an end mark 'eeee'. The tokenization of the captions is then performed using the Keras Tokenizer Class since a model cannot be trained using textual data. The captions are embedded using GloVe which is a pre-trained model for embedding.

Encoder-Decoder Framework The proposed method in this paper implements and compares the different pre-trained convolutional neural networks architectures like—DenseNet201 and InceptionV3 for the feature extraction from the generated keyframes. Now these pretrained models are generally used for classification of images and hence the output of the penultimate layer is redirected to give a vector of numbers that summarize and encode the features of the image concisely. Accordingly, the Global Average Pooling layer is selected as the final layer of the model for feature extraction. This will result in the encoded image vectors to be a size of 1920 and 2048 for DenseNet201 and InceptionV3 respectively. Next, a decoder using Recurrent Neural Network is used to map the encoded image vectors with the word embeddings. The decoder is made up of four Gated Recurrent Unit (GRU) Layers and is used to generate the caption for each keyframe. GRU is a recurrent neural network that uses less memory and is faster than Long Short-Term Memory (LSTM). Since the encoded image features are a vector of 1920 (for DenseNet201) and 2048 (for InceptionV3) and the state-size of the GRU units is 512, image feature vectors were mapped to 512 using a fully connected layer. The fourth and last GRU layer is then connected to a Dense layer that outputs a sequence of one-hot encoded arrays. Finally, the model is compiled and trained for 50 epochs by feeding batches of training data having a batch size of 32.

Generating Captions The proposed approach uses two search techniques to generate captions such as Greedy Search and Beam Search. Greedy search is an

algorithm that iterates through the caption from left to right one word/token at a time and chooses the most probable token. This means that at any given time step t, the token having the highest conditional probability is chosen as seen in (2).

$$y_t = \arg\max_{y \in \Upsilon} P(y|y_1, \ldots, y_{t-1}) \tag{2}$$

where, y_t is generated token, $y_1, \ldots, y_{t-1}$ is the previously predicted tokens and Y is the vocabulary (Zhang et al. 2021). The pseudocode/algorithm of Greedy Search is as follows:

1. Preprocess the keyframe using the pre-trained image model.
2. Till token! = 'eeee' or number_of_tokens < = max_tokens
   a. Use argmax () to find the most likely token.
   b. Convert the token into its corresponding word.
   c. Add word to output sentence/caption.
3. Return the final output sentence/caption.

Beam Search is another algorithm, which considers k different hypotheses and generates tokens for each of the hypotheses from the vocabulary to find the most suitable token.

$y_t = (y_1, y_2, \ldots\ldots, y_t)$ be an output sequence of length t containing tokens from a vocabulary V.

Beam search reaches an approximate solution by maintaining a beam B_t containing only k (beam-size) most likely partial sequences at each decoding time-step t. At each time step t, the beam B_t is updated by retaining the k most likely sequences in the candidate set E_t generated by considering all possible next word extensions (Anderson, et al. 2016). This can be seen below in (3):

$$E_t = \{(y_{t-1}, w) | y_{t-1} \in B_{t-1}, w \in V \tag{3}$$

The pseudocode/algorithm of Caption Generation through Beam Search is as follows:

1. Preprocess the keyframe using the pre-trained image model.
2. Till token! = 'eeee' or number_of_tokens < = max_tokens
   a. Use argsort() such that k tokens with the highest probability are selected.
   b. Each subsequent word's probability is considered in relation to the words already chosen.
   c. Add word to output phrase/caption.
3. Return the final output phrase/caption.

Therefore, it is evident that beam search prioritizes word pairs that appear together to produce a caption. At each stage of the beam search, numerous possibilities are picked for various values of k (beam width). Thus, k is a hyperparameter used in

beam search. Larger values of k increase the chances of matching the target sequence better, although they require larger memory.

In the proposed system, $k = 3$ and $k = 5$ were used to generate the captions for the extracted keyframes. At the end, the model was evaluated on BLEU score and METEOR score (Chen et al. 2015).

4 Results and Discussion

Under this section the results for Video Classification as well as Caption Generation are discussed in this section.

4.1 Video Classification

The algorithm for keyframe extraction and DenseNet201 was applied on the UCF101 Dataset as well. Nine classes were chosen at random—['Archery', 'Basketball', 'Bowling', 'HorseRace', 'PlayingPiano', 'RopeClimbing', 'SkyDiving', 'HammerThrow', 'Skiing']. The training set had 941 images. The training accuracy came out to be 99.89% and the validation accuracy was 97.88%. Table 1 shows the comparison of various architectures with the proposed architecture.

Kızıltepe et al. (2019) uses an Enhanced Keyframe extraction method in which the time complexity is $O(N)$ for extracting the video. The keyframe extraction algorithm in the proposed system also has a time complexity of $O(N)$ but has performed better on the UCF101 dataset by around 6% increase in training accuracy. As UCF101 dataset is balanced as compared to the ISRO dataset, therefore the validation accuracy increased. The loss curves for ISRO and UCF101 can be seen Fig. 3 respectively.

Table 1 Training and validation accuracy of different architectures for the UCF101 dataset

Paper	Year	Architecture used	Training Accuracy (%)	Validation Accuracy (%)
Ramesh and Mahesh 2022)	2022	Customized CNN	92.77	–
Kızıltepe et al. (2019)	2021	VGG-16-ConvLSTM	99.24	97.45
Li and Wang (2019)	2022	Context-LSTM	92.2	91
Shrestha et al. (2022)	2022	3D-ConvNet + Attention	–	81.1
Wang et al. (2022)	2022	Double branch 3D CNN	95	–
Proposed System	–	DenseNet201 + Keyframe Extraction	99.89	97.88

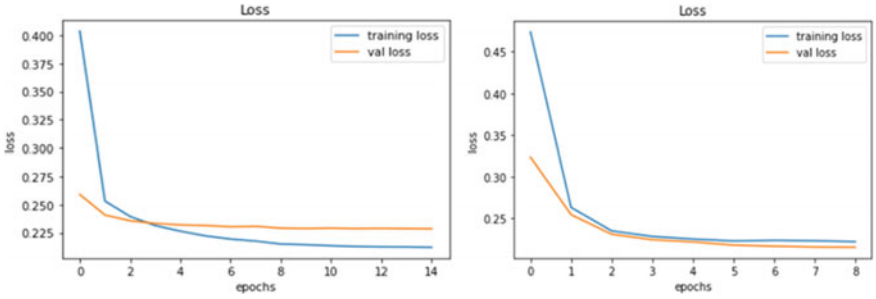


Fig. 3 Training and validation loss for ISRO and UCF101 Dataset respectively

Fig. 4 Test video from the ISRO dataset

It is evident that the validation loss decreases till around the 4th epoch and then onwards remains almost constant. For testing, the keyframe is extracted from a video using the algorithm and resized, before being given as an input to the model. The model then gives us the probabilities for all the classes. The actual testing on videos can be seen in Fig. 4.

The image is extracted as the keyframe from a video which shows a helicopter. When this image is fed as an input to the model, the output is "Helicopter". Though the helicopter was a class in minority it was correctly classified.

4.2 Video Captioning

Since DenseNet201 and InceptionV3 proved to be the best models, the same were used for training the model for Captioning as well. Greedy Search and Beam Search ($K = 3, 5$) were used to compare these models. BLEU 1, BLEU 2 and METEOR score are used as evaluation metrics.

Figure 5 illustrates the generated captions for a video scene from the ISRO dataset using DenseNet201

The predicted captions of all images were compared and evaluated using the performance metrics—BLEU1, BLUE2 and METEOR scores for the same were

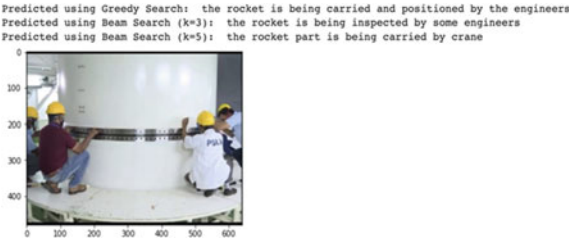


Fig. 5 DenseNet201 predicted and actual captions for greedy, Beam($k = 3$), Beam($k = 5$)

calculated. The overall comparison table of all metrics on ISRO dataset is described in Table 2.

In addition to this, the Video Captioning model was also tested on a Popular Video Captioning Dataset—MSR-VTT (Microsoft Research Video to Text). It is a video dataset with videos ranging from 10 s to 1 min. From this dataset 2 categories—'soccer' and 'Horse-Riding' category videos were chosen. The key-frame was extracted and then the same algorithm was used. The training to testing ratio considered was 80:20. Table 3 below shows the comparison of various architectures with the proposed architecture on the MSR-VTT dataset.

Table 2 Evaluation metrics for captioning on the ISRO dataset (50 epochs)

Model	Metric	Greedy Search	Beam Search (k = 3)	Beam Search (k = 5)
DenseNet201	BLEU1	0.2989	0.2724	0.2810
	BLEU2	0.1949	0.1783	0.1871
	METEOR	0.3012	0.2825	0.2972
InceptionNetV3	BLEU1	0.2767	0.2513	0.2499
	BLEU2	0.1647	0.1568	0.1553
	METEOR	0.2649	0.2559	0.2681

Table 3 Comparison table for METEOR scores with other literature (MSR-VTT dataset)

Paper	Year	Architecture used	Meteor
Hori et al. (2022)	2022	Audio visual transformers model	0.25
Xiao and Shi May (2020)	2020	Text by text dynamic attention mechanism + step-by-step learning	0.273
Yang et al. (2022)	2022	Multimodal semantic grouping-based video caption model	0.282
Zhao et al. (2022)	2022	Visual transformer and reinforcement learning	0.288
Li et al. (2022)	2022	Text-based dynamic attention model (TDAM)	0.289
Proposed system	–	DenseNet201 + GRU + Keyframe extraction	0.299

5 Conclusion

In conclusion, it can be said that extracting the keyframes and using that as an input to a Transfer Learning Model i.e., Densenet201 gave very accurate results as well as saves the computational time. Instead of processing 20 frames for a video, bringing it down to a single keyframe achieved a higher validation accuracy by over 24%. The proposed algorithm performed extremely well on the UCF101 Dataset. It achieved a training accuracy of 99.89%. Also, in the case of Captioning, both DenseNet201 and InceptionV3 nearly gave similar results in terms of performance metrics like BLEU score and METEOR score. On the MSR-VTT dataset, the METEOR score for this approach was 0.299. In future, methods for oversampling of the minority class can be implemented to overcome class imbalance and overfitting.

References

Shankar K, Mahesh K, Kuppusamy K (2012) Video segmentation on 2D images with 3D effect. Int J Comput Appl 43(8):1–4. https://doi.org/10.5120/6120-8322

Ramesh M, Mahesh K (2018) Significance of various video classification techniques and methods: a retrospective. Int J Pure Appl Mathh 118(8):523–526

Bhat P, Malaganve P, Hegde P (2019) A new framework for social media content mining and knowledge discovery. Int J Comput Appl 182(36):17–20

Bhat P, Malaganve P (2021) Metadata based classification techniques for knowledge discovery from facebook multimedia database. Int J Intell Syst Appl 13(4):38–48

Katiyar S, Borgohain SK (2021) Image captioning using deep stacked LSTMs, contextual word embeddings and data augmentation. arXiv:2102.11237

Oluwasammi A, Aftab MU, Qin Z, Ngo ST, Doan TV, Nguyen SB, Nguyen SH, Nguyen GH (2021) Features to text: a comprehensive survey of deep learning on semantic segmentation and image captioning. Complexity 2021:1–19

Wu Z, Yao T, Fu Y, Jiang YG (2017) Deep learning for video classification and captioning. In: Frontiers of multimedia research, pp 3–29

Jiang Y-G, Wu Z, Tang J, Li Z, Xue X, Chang S-F (2018) Modeling multimodal clues in a hybrid deep learning framework for video classification. IEEE Trans Multimedia 20(11):3137–3147. https://doi.org/10.1109/TMM.2018.2823900

Visualisation of Earth Observation Data and Archival System, ISRO (2022) Available: https://vedas.sac.gov.in/en/sih2022.html

Canny J (1986) A computational approach to edge detection. IEEE Trans Patt Anal Mach Intell PAMI-8(6):679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Zhong Z, Zheng M, Mai H, Zhao J, Liu X (2020) Cancer image classification based on DenseNet model. J Phys: Conf Series 1651(1):012143. https://doi.org/10.1088/1742-6596/1651/1/012143

Zhang A, Lipton ZC, Li M, Smola AJ (2021) Dive into deep learning. arXiv:2106.11342

Anderson P et al (2016) Guided open vocabulary image captioning with constrained beam search. arXiv:1612.00576

Chen X et al (2015) Microsoft coco captions: data collection and evaluation server. arXiv:1504.00325

Ramesh M, Mahesh K (2022) Sports video classification framework using enhanced threshold based keyframe selection algorithm and customized CNN on UCF101 and Sports1-M dataset. Comput Intell Neurosci 2022

Kızıltepe RS, Gan JQ, Escobar JJ (2019) A novel keyframe extraction method for video classification using deep neural networks. Neural Comput Appl 1–12

Li D, Wang R (2019) Context-LSTM: a robust classifier for video detection on UCF101. arXiv:2203.06610

Shrestha L, Dubey S, Olimov F, Rafique MA, Jeon M (2022) 3D convolutional with attention for action recognition. arXiv:2206.02203

Wang Y, Sun J (2022) Video human action recognition algorithm based on double branch 3D-CNN. In: 2022 15th international congress on image and signal processing, BioMedical engineering and informatics (CISP-BMEI), Beijing, China, pp 1–6. https://doi.org/10.1109/CISP-BMEI56279.2022.9979858

Hori C, Hori T, Le Roux J (2022) In: Low-latency streaming scene-aware interaction using audio-visual transformers, Interspeech, pp 4511–4515

Xiao H, Shi J (2020) Video captioning with text-based dynamic attention and step-by-step learning. Pattern Recogn Lett 133:305–312

Yang X, Wang X, Ye X, Li T (2022) VMSG: a video caption network based on multimodal semantic grouping and semantic attention

Zhao H, Chen Z, Guo L, Han Z (2022) Video captioning based on vision transformer and reinforcement learning. PeerJ Comput Sci 8

Li S, Yang B, Zou Y (2022) Adaptive curriculum learning for video captioning. IEEE Access 10:31751–31759. https://doi.org/10.1109/ACCESS.2022.3160451

Classification of Image with Convolutional Neural Network and TensorFlow on CIFAR-10 Dataset

Gunjan Sharma and Vijay K. Jadon

Abstract Convolution Neural Networks (CNNs), one type of deep learning model, that have been effective in classification of image and for tasks of object detection have been utilized. In this paper, a model has been proposed to classify images effectively and quickly by using CIFAR-10 dataset. Further, TensorFlow with Google Colab has been utilized for the model's building. The model is trained by providing training dataset images of 32 × 32 size to determine the parameters that best fit the data. The model is designed with limited number of CNN layers and learning rate of 0.001. No dropout technique has been used in our model. The results show that the model gives test accuracy of 98.93% and validation accuracy of 81.91%. This work finds its application in surveillance and Internet of things.

Keywords Image classification · CNN · Convolutional Neural Network · TensorFlow · Python · CIFAR-10

1 Introduction

In the recent past years, with the expansion of data in various sectors of industry, such as e-commerce, image classification is rising which has attracted the attention of technology developers (Abu et al. 2019; Chauhan et al. 2018) toward this field. Image classification of different features is in demand now a days. One of the finest examples for the same is Facebook in which images of persons are distinguished based on different parameters (Lu and Weng 2007). With just tagged photo, Facebook can now identify your face with up to 90% accuracy, classifying it into the persons' Facebook album. Although technology has made substantial progress in this sought, yet there exists a room for some further advancements (He et al. 2015). Image classification

G. Sharma (✉) · V. K. Jadon
Institute of Engineering and Technology, Chitkara University, Chitkara University, Punjab, India
e-mail: sharma.gunjan@chitkara.edu.in

V. K. Jadon
e-mail: vijay.jadon@chitkara.edu.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_51

is generally understood to be the process of feeding an image into a model created using a particular method, which then outputs the category or the likelihood that the image corresponds to that specific class (Angra et al. 2022). Supervised learning is the method by which we assign a certain class to a given image (Kembuan et al. 2020). The way the machine (computer) perceives an image and the way we perceive it differ significantly. We are able to visualize the image and assign it a size and color to it. The machine, on the other hand, just gets to observe numbers. Pixels are the name for the visible numbers (He et al. 2015). Deep learning has become one of the dominant approaches for learning this image classification technology. It is a subset of Artificial Intelligence that can behave and think just like a human. In order to make the system itself more effective and quicker during the "training" session, hundreds or even thousands of input data are typically set into the system (Ciregan et al. 2012). To make the "training" session more effective, the system is typically configured with hundreds or even thousands of input data (Jingyi et al. 2021) (Fig. 1).

Convolutional Neural Networks (CNNs), a sub-set of machine learning, are an image recognition and categorization architecture that is widely used now a days. Its great success in decision-making and image recognition applications has been proven (Failed 2015). Images are high dimensional vectors, and to describe these

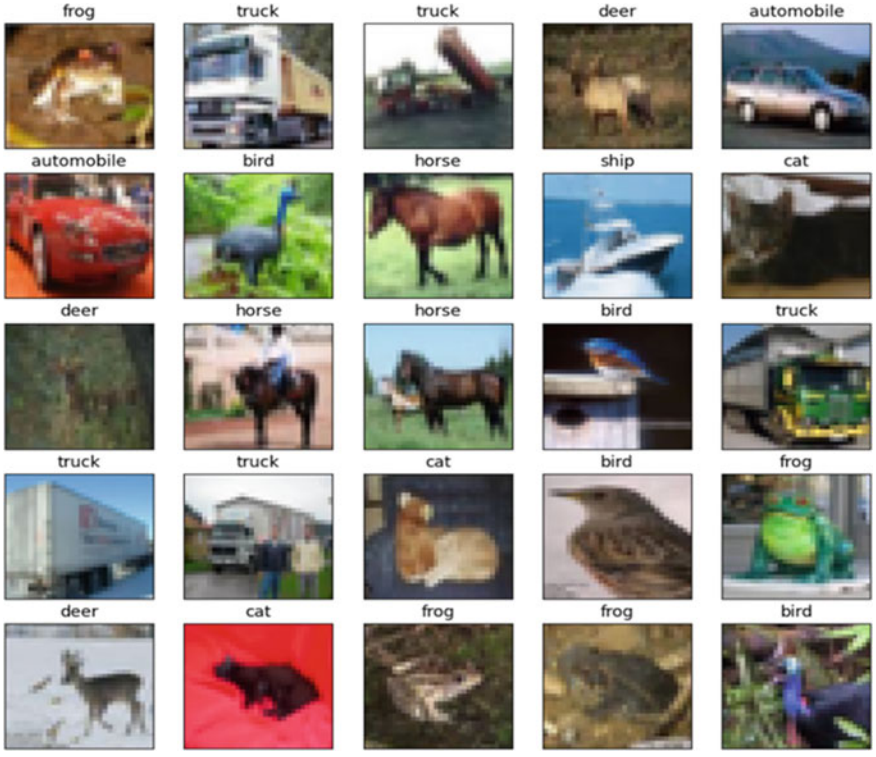


Fig. 1 Sample images of CIFAR-10 dataset

vectors, several other criteria would be needed. Instead of identifying an object and categorizing it appropriately to its class, the image classification problem is comprised many unknown patterns (Zeiler and Fergus 2014a). The CNN algorithm is categorized as a deep learning approach because it uses neural networks which consist of many hidden layers to combine and extract high-level features from 2D inputs. CNN is equipped for picture categorization and processing, all thanks to this functionality (Xie et al. 2015). Utilizing Convolutional Neural Network (CNN) architecture and TensorFlow, we hope to meet the demand for image identification systems (Bragg, et al. 2019).

TensorFlow is an open-source Python programming package used to create the picture classification model (Kothari 2018). Google Colaboratory, an open-source platform for machine learning research environment, is used to conduct this study. Additionally, we employ the Rectified Linear Unit (ReLU) activation function for solving the vanishing gradient issue and a classifier named SoftMax (Kembuan et al. 2020).

2 Related Work

According to Krizhevsky et al., using only supervised algorithms, deep Convolutional Neural Networks can achieve record-breaking performance on a very challenging dataset (Krizhevsky et al. 2017). Five of the eight learned layers in there proposed model, AlexNet, were convolutional layers and three of those layers were completely connected. To reduce overfitting, data augmentation and dropout techniques were used. Error rates for this architecture were 37.5% and 17.0%, respectively. Experiments show that removing just one convolutional layer causes performance to drop by 2%. However, in this, there were no experiments done to test the effects of varying the number of layers, neurons, or filter size. Instead, the impact of removing one convolutional layer was tested.

According to Sudharshan and Raj (2018), the topic of object recognition and classification from a set of images is difficult in the area of image analysis and computer vision. As a result, they implemented the Convolutional Neural Networks using TensorFlow on the Keras platform and it supports the CIFAR-10 dataset to classify the images of airplane and bird for aviation purpose.

Chherawala et al. (Chherawala et al. 2015) developed a vote-weighted Recurrent Neural Networks' (RNNs) framework for evaluating the importance of feature sets. Weighted votes are combined with the model to determine significance, and RNN is used to implement the model. To recognize handwriting, it takes features from the Alex word images.

Chauhan et al. (Chauhan et al. 2018) implemented CNN model for image recognition on MNIST dataset and CIFAR-10 dataset had been used for object detection using CPU unit. Different techniques like dropout, data augmentation were used in their work. The accuracy for both the models was different as for MNIST dataset, the accuracy was 99.6%, and for CIFAR-10 dataset, it was 80.17%.

3 Network Model and Tools Used

3.1 Dataset

CIFAR-10 dataset, which contains 60,000 images with a 32 × 32 pixel size overall, has been used for training our model. The dataset is separated into ten categories, including "Automobile," "Airplane," "Bird," "Cat," "Deer," "Frog," "Dog," "Ship," "Horse," and "Truck."(Doon et al. 2018). Each category comprises 6000 unique photos. There are 10,000 test photos and 50,000 training images altogether. This dataset is ideal for this task because the images in this dataset are small and labeled and have no clamor in it (Krizhevsky and Hinton 2009). Firstly, to train the designed CNN's weights, a training set has been used. The validation set is chosen randomly from the set to test the generalizability of the model (Sudharshan and Raj 2018). Therefore, on the basis of loss function value, the model weights are chosen to be saved on the validation data. Finally, at the test stage, judgments concerning the class on the test set are made using the trained model's saved weights (Jingyi et al. 2021). Here are some images from this CIFAR-10 dataset.

3.2 Environment Development

In this experiment, CNN was run on GPU using the Python language and the TensorFlow program. Python, Scikit-learn, TensorFlow, and Keras packages are also installed.

The Python programming language makes it simple to solve problems through coding flexibility (Kembuan et al. 2020). It is a platform-independent and fast scripting language which is having garbage collection mechanism. The most fruitful and reliable Python machine learning repository is Scikit-learn. Through a Python interface, it offers a variety of efficient tools for statistical data and machine learning modeling, including classification, regression, and clustering. This Python-based library is built upon Matplotlib, NumPy, and SciPy (Ciregan et al. 2012). Feature extraction and cross-validation, supervised learning algorithms, non-supervised learning algorithms are some most prominent features of this library.

3.3 Convolutional Neural Network (CNN)

In order to access structured arrays of data, such as characterizations or portrayals, CNN, a deep learning neural network, has been created (Zeiler and Fergus 2014b). CNN does not require any preprocessing. It can be applied directly to an underdone image. There are various layers in CNN model like convolutional, pooling, activation, and output layers (He et al. 2015) (Fig. 2).

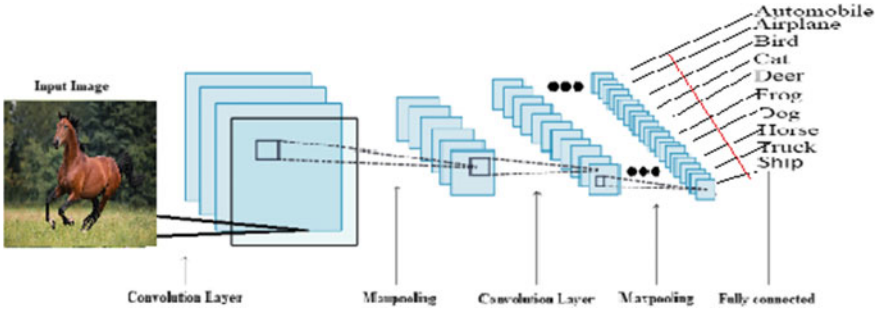


Fig. 2 Framework of CNN model

After receiving an input image, the CNN model goes through a series of pooling and convolutional layers. Fully connected layers then carry out this process to produce the result. Different operations are applied as part of the learning process in each layer (Kembuan et al. 2020). The layers known as pooling layers are in charge of reducing the height and width of feature maps while maintaining the depth (Zeiler and Fergus 2014a). The CNN model supports two different types of pooling layers: average pooling and maximum pooling.

Each element's value in the filtered portion of the image is maximized by pooling layer (Ciregan et al. 2012). Max pooling is the best technique for removing prominent features and is therefore thought to be most efficient (Table 1).

CNN model contains an activation function that makes the decision whether or not to activate a neuron (Nair and Hinton 2010). Basically, the activation function is a simply mathematical model that converts an input into an output that will fall within a particular range. This common activation function in CNN is "ReLU." The function can be represented by:

$$f(s) = \max(0, s), \tag{1}$$

where $f(s) = 0$ if $s < 0$
$= 1$ if $s > = 0$.

Table 1 Proposed CNN model parametersle captions should be placed above the tables

Layer name	Layer parameter
Input image	32 × 32 pixels
Conv1	Nodes = 32, Filter = 3 × 3, ReLU
MaxPooling	Pool size = 2
Conv2	Nodes = 32, Filter = 3 × 3, ReLU
MaxPooling	Pool size = 2
Dense layer	Nodes = 64, ReLU
Output	Fully connected 10 classes, SoftMax

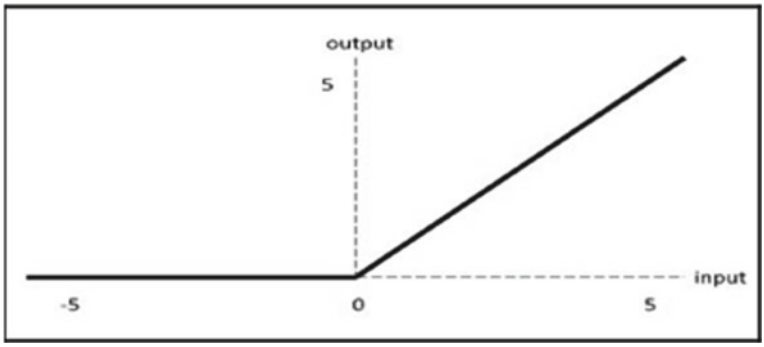


Fig. 3 Graph for ReLU function

ReLU is a nonlinear activation function that is utilized in multi-neural network. The output will be zero if the input is negative, and the input will be output immediately if it is positive which means that it will revert the same positive input value as output (Fig. 3). The rectifier activation function (ReLU) is used to make the neural network nonlinear (Sudharshan and Raj 2018).

4 Methodology

In image classification, an image must be given a label in order to be classified from a list of categories. This suggests that a model should always be developed to analyze the incoming images and produce labels that categorize the images, with the labels always being chosen from the predefined set of possible categories.

Figure 4 shows the framework or steps to be followed for image classification using CNN (Carter et al. 2021). Throughout this process, there are seven phases. With Python serving as its programming language and TensorFlow as its open-source software, each phase is covered. Then, the procedure is carried out to continue gathering some of the photographs (inputs), and ultimately, all images will be classified according to their groups. After categorizing the images, the model will be trained, and after that, it will be tested for random image.

4.1 Data Preprocessing

This step includes downloading the dataset from Keras. It is already having training and test images in it. TensorFlow and Python are used to create the code (Zhou et al. 2010). TensorFlow, a deep learning framework that is free and open source, increases the level of designer control over each neuron, which is referred to as a "node" in

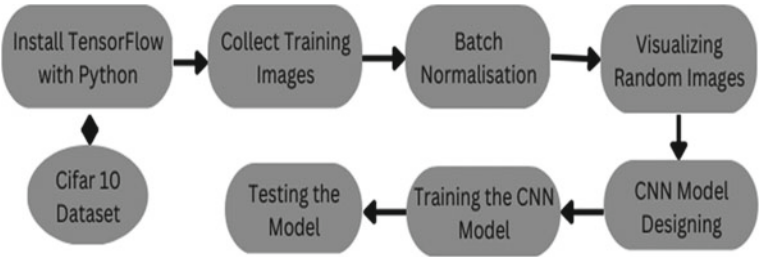


Fig. 4 Block diagram of architecture

TensorFlow. In order to achieve the best execution, we can alter the weight. RGB values (red, green, and blue numbers ranging from 0 to 255) can be used to represent the colors of an RGB image (Doon et al. 2018). The information included in these photos will then be decoded and converted into floating point tensors using the Image Data Generator class provided by tf.keras (Tokozume et al. 2018). Given that small input values are preferred for processing by neural networks, we have modified the tensor scale from values between 0 and 255 to values between 0 and 1.

4.2 CNN Model Creation and Training

Table 2 depicts the configuration of the Convolutional Neural Network employed in this study. Pooling layers are a technique for reducing the dimension of feature maps which means the height and width of image while keeping the depth constant (Chherawala et al. 2015). The features in the image's area where the filter was applied to had their highest values obtained by max pooling.

Table 2 Configuration of CNN

Layer (type)	Output shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max pooling2d (MaxPooling2D >	(None, 15, 15, 32)	0
conv2d_l (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_l (MaxPooling 2D)	(None, 6, 6, 32)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 64)	73792
dense_l (Dense)	(None, 10)	650
Total params: 84,586 Trainable params: 84,586 Non-trainable params: 0		

Max pooling is thought to be more efficient since it is more effective at extracting dominating features (Obaid et al. 2020). A max pool layer is included in each of the model's convolution layers. In this model, we have used two convolutional layers. The resulting feature mappings are transformed into the single one-dimensional vector using the "Flatten" layer in the CNN model (Kingma and Ba 1412). In order to employ layers with full connectivity after several convolutional/maxpool layers, the flattening step is required. We have utilized the SoftMax activation function in the last layer of our model, i.e., dense layer. Each class's output in SoftMax is constrained to fall between 0 and 1. The dense layer neuron receives one output from the prior layer, which is delivered to all the neurons of every layer.

4.3 Training and Testing of the Model

The training data are provided to the model in the first stage; in this, the train images along with the train labels are provided here. The model then gains the ability to link labels to the photos. The test data and test images are then predicted using the model. Lastly, in the final step, we use test labels to confirm the anticipated categories (Doon et al. 2018).

5 Test Result

5.1 Validation and Training Accuracies

In our CNN model, we have used SGD optimizer and learning rate of 0.001. After 25 iteration or epoch, the accuracy reached to 98.93% and validation accuracy reached at 81.91%. Figure 5a depicts the accuracy vs. epoch graph and Fig. 5b shows validation accuracy vs. epochs' graph.

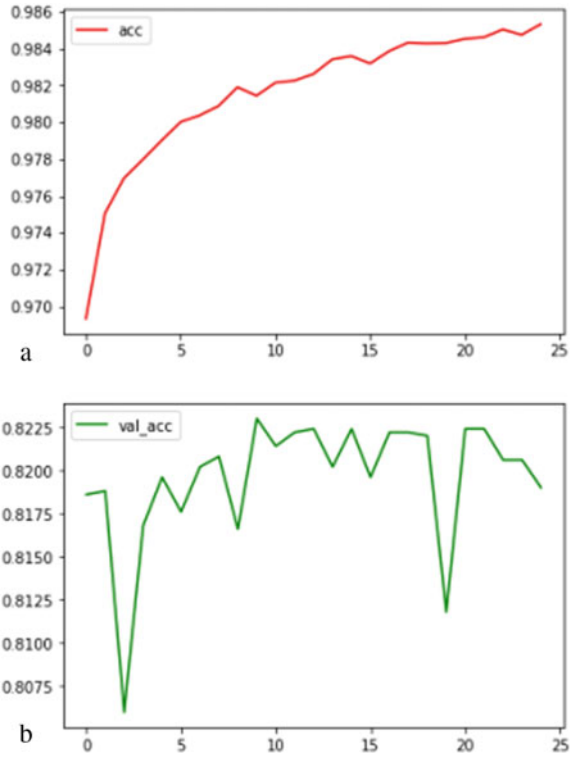


Fig. 5 **a** Accuracy versus epochs. **b** Validation accuracy versus epochs

5.2 Validation and Training Losses

Loss is not a percentage, in contrast to accuracy. It displays the overall amount of errors made for each example in the training or validation sets. The initial loss for the model trained and validation losses for the dataset is shown in Figs. 6.a and 6.b. From the graph, we can notice that the value for loss is continuously decreasing on loss on training data and the value of validation loss lies between 1.26 and 1.34.

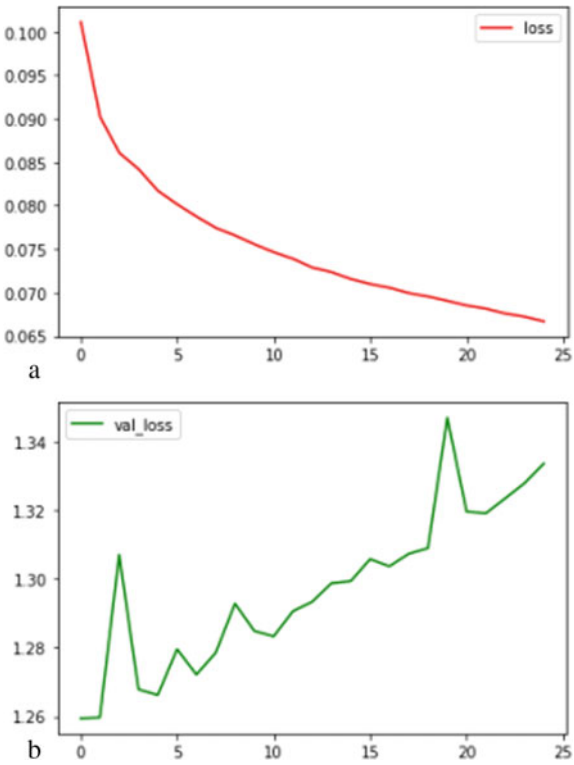


Fig. 6 **a** Loss versus epoch. **b** Validation loss versus epoch

5.3 Test Accuracy

We have randomly given 20 images to test the model, and the outcome was accurate each time. Every time, the model can appropriately classify the image. Figure 7a can be seen as example for test accuracy in which the model has to predict the image no 789 and as the results show that the model has predicted the image at no. 789 to be of ship. Figure 7b shows the predicted image at index 789 that clearly depicts that the model has classified the image accurately.

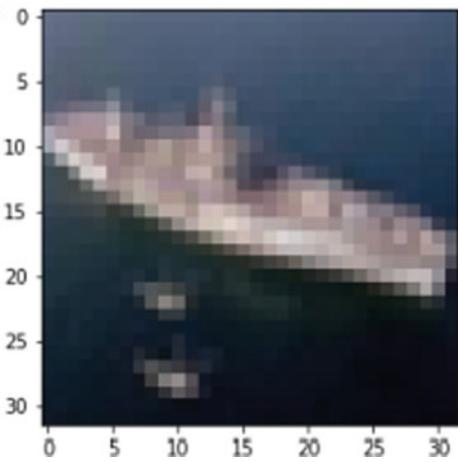
Fig. 7 **a** Prediction of image. **b** Predicted image at index 789

a
```
imag = np.argmax(predict[789])
class_names[imag]

    'ship'

plt.figure()
plt.imshow(x_test[789])
```

b



6 Conclusion

Here, we have used a Convolutional Neural Network to classify photos into ten categories. The network was trained using the CIFAR-10 dataset as a baseline. For updating weights, the SGD optimization method provides the most accurate image categorization for our model. The CNN model is prepared with two Conv2D layers, two pooling layers, one flatten layer, one dense layer, and finally, an output dense layer. The learning rate is also kept very low and the model is trained for only 25 epochs. Image Data Generator has been used for preprocessing of data/images. We have tried to get the best accuracy in less number of epochs. The model has shown very good results in categorizing the images. These results can vary by using regularization and dropout method. In future, we will try to apply other techniques to achieve 100% accuracy with our CNN model.

References

Abu MA, Indra NH, Rahman A, Sapiee NA, Ahmad I (2019) A study on image classification based on deep learning and tensorflow. Int J Eng Res Technol 12(4):563–569

Angra S, Sharma B, Sharma KD (2022) Amalgamation of virtual reality, augmented reality and machine learning: a review. In: 2022 2nd international conference on advance computing and innovative technologies in engineering (ICACITE), IEEE, pp 2601–2604

Bragg D et al (2019) Sign language recognition, generation, and translation: an interdisciplinary perspective. In: The 21st international ACM SIGACCESS conference on computers and accessibility, pp 16–31

Carter B, Jain S, Mueller JW, Gifford D (2021) Overinterpretation reveals image classification model pathologies. Adv Neural Inf Process Syst 34:15395–15407

Chauhan R, Ghanshala KK, Joshi R (2018) Convolutional neural network (CNN) for image detection and recognition. In: 2018 first international conference on secure cyber computing and communication (ICSCCC), IEEE, pp 278–282

Chherawala Y, Roy PP, Cheriet M (2015) Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network model. IEEE Trans Cybernetics 46(12):2825–2836

Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: 2012 IEEE conference on computer vision and pattern recognition, IEEE, pp 3642–3649

Doon R, Rawat TK, Gautam S (2018) Cifar-10 classification using deep convolutional neural network. In: 2018 IEEE Punecon, IEEE, pp 1–5

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034

Jingyi Y, Rui S, Tianqi W (2021) Classification of images by using TensorFlow. In: 2021 6th international conference on intelligent computing and signal processing (ICSP), IEEE, pp 622–626

Kembuan O, Rorimpandey GC, Tengker SMT (2020) Convolutional neural network (CNN) for image classification of indonesia sign language using tensorflow. In: 2020 2nd international conference on cybernetics and intelligent system (ICORIS), IEEE, pp 1–5

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980

Kothari JD (2018) A case study of image classification based on deep learning using TensorFlow. Int J Innovative Res Comput Commun Eng 6(7):3888–3892

Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90

Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images

Lu D, Weng Q (2007) A survey of image classification methods and techniques for improving classification performance. Int J Remote Sens 28(5):823–870

Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814

Obaid KB, Zeebaree S, Ahmed OM (2020) Deep learning models based on image classification: a review. Int J Sci Bus 4(11):75–81

Sudharshan DP, Raj S (2018) Object recognition in images using convolutional neural network. In: 2018 2nd international conference on inventive systems and control (ICISC), IEEE, pp 718–722

Tokozume Y, Ushiku Y, Harada T (2018) Between-class learning for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5486–5494

Xie L, Hong R, Zhang B, Tian Q (2015) Image classification and retrieval are one. In: Proceedings of the 5th ACM on international conference on multimedia retrieval, pp 3–10

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, Part I 13. Springer, pp 818–833

Zeiler MD, Fergus R (2014a) Visualizing and understanding convolutional networks. European conference on computer vision. Springer, pp 818–833

Zhou X, Yu K, Zhang T, Huang TS (2010) Image classification using super-vector coding of local image descriptors. In: Computer vision–ECCV 2010: 11th European conference on computer vision, Heraklion, Crete, Greece, September 5–11, 2010, proceedings, part V 11. Springer, pp 141–154

A Comparative Inspection and Performance Evaluation of Distinct Image Fusion Techniques for Medical Imaging

Harmanpreet Kaur, Renu Vig, Naresh Kumar, Apoorav Sharma, Ayush Dogra, and Bhawna Goyal

Abstract Fusion in medical imaging refers to the synthesis of medical images obtained from various imaging technologies by utilizing an algorithm to combine the benefits or complementarities of each image to produce a more insightful image. This can greatly improve the accuracy with which diseases are detected and treated for positive health of the patient. In this study, a comparative analysis among widely used image fusion methods in transform domain using a medical image dataset of varying imaging modalities is carried out to determine the finest strategy for future study and give medical researchers new direction. The outcomes of the experiments are evaluated using a range of well-known performance evaluation criteria. Laplacian pyramid provided the best values of $Q^{AB/F}$ in all three datasets, GF-based strategy is most frequently favored by the metrics Q_S, Q_C, Q_Y while Q_P gives the best results for ASR. The hybrid method NSST-MSMG-PCNN gave the second-highest values for all three datasets. Thus, the qualitative and quantitative tests performed on a conventional dataset of multimodal medical images reveal that fusion process produces better quality images and both are necessary for a consistent comparison of fusion methods.

Keywords Medical image fusion · Healthy lives · Multimodality · Biomedical imaging technology · Image quality evaluation indexes

H. Kaur (✉) · R. Vig · N. Kumar · A. Sharma
Electronics and Communication Engineering Department, UIET, Panjab University, Chandigarh 160014, India
e-mail: harman.phdpub@gmail.com

A. Dogra
Chitkara University Institute of Engineering and Technology, Chitkara University, Chandigarh, Punjab, India

B. Goyal
Department of UCRD and ECE, Chandigarh University, Gharuan, Mohali, Punjab, India

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_52

1 Introduction

Image fusion (IF) is the practice of amalgamating imagery from distinct sources into a composite image utilizing cutting-edge image processing methods. The ability to combine heterogeneous and complementary data from different sensors enhances the performance for visualization or computer processing tasks. This leads to several advantages of image fusion like greater reliability, classification of interpretation, reduced ambiguity, increased geographical and temporal coverage, and more accurate data. Image fusion is perceived as an important assistant in different research areas such as medical imaging, military, nighttime operation, remote sensing, multispectral imaging, computer vision, and material sciences. The significant amount of scientific material that has been published in the previous ten years demonstrates the extraordinary expansion that has occurred in the field of medical imaging. These reflect increasing capability of medical imaging systems and significant growth of imaging modalities in clinical usage. Each of the medical imaging modality has its own distinctive traits and it's laborious to encapsulate all the details/features from a sole modality. Major imaging modalities used for fusion process include computer tomography (CT), magnetic resonance imaging (MRI), X-ray, ultrasound (USG), angiography, positron emission tomography (PET), and single photon emission computed tomography (SPECT). CT images provide facts of dense tissues like bones with less distortion, whereas MRI provides the best view of soft tissues. PET/SPECT is frequently utilized to examine the blood flow; i.e., how tissues and organs are functioning, whereas USG creates images of the inside of the body. This motivates the need of merging the supplementary information from different modalities. In this regard, medical image fusion (MIF) is a technique of amalgamating diverse imaging modalities to produce a more representative image which improves diagnostic accuracy. So multimodal medical image fusion (MMIF) gives better visual representation of the medical images which assists clinicians in diagnostics and treatment planning of various human organs brain, lungs, liver, stomach, breast, mouth, pelvic, head and neck, bones, and soft tissues (El-Gamal et al. 2016; James and Dasarathy 2014).

The expansion of several image fusion techniques over the last few years can be indexed into three processing levels: (a) pixel level, (b) feature level, and (c) decision level. The pixel-level fusion or signal-level fusion, the bottommost level defines the method of creating a single fused image by combining the pixel-by-pixel data from several images. Feature-level fusion, also called object-level fusion, extracts features of interest (pixel intensities, edges on tenure features) using advanced image processing methods such as segmentation, morphological operation, and region characterization. A composite set of features is then created by combining the region or feature. The topmost level of fusion is the decision-level or symbol-level fusion. In this level, classifier systems are used to assign each detected object of the input images to a class known as decision from a set of pre-defined classes. The statistics is then combined at the decision level to increase the likelihood that the objects will be correctly classified. Several parameters like information content, sensitivity

to noise, computational complexity, preprocessing, and difficulty in implementation help comparing various fusion levels (Hermessi et al. 2021).

Section 2 contains a discussion on revolutionary multimodal medical image fusion (MMIF) methods along with advantages and disadvantages. In Sect. 3, image quality evaluation indexes are discussed and comparison on various image fusion methods is presented. Finally, Sect. 4 draws conclusion by providing a concise summary and analysis of the findings.

2 Background and Literature Review

Numerous surveys and review papers have assembled the MMIF methodologies and classifications. For instance, a thorough literature panorama of MIF methods including historical background and recent developments is provided (Hermessi et al. 2021). Theories underlying various fusion techniques are investigated to address each class's restrictions, and an experimental analysis is performed to further guide the discussion. A detailed review of multimodal medical imaging modalities, openly available multimodal datasets, classification of MIF algorithms and image quality assessment metrics, and associated disorders is presented (Azam 2022). Further, a detailed survey on pixel-level image fusion methods and major application domains are presented (Li et al. 2017). The authors highlighted interesting ideas and challenges associated in the field of IF giving a constructive direction to the researchers aiming to work in this research area. An overview of several image fusion approaches along with pros and cons of each is given (Kaur et al. 2021, 2019). Additionally, challenges and issues associated with diverse application areas are discussed and widely used performance evaluation metrics with/without ideal reference image are highlighted. The authors provided a thorough and focused study, covering and analyzing almost all the key mechanisms for the fusion of infrared and visible images (Ma et al. 2019). The research work presented by Tawfik et al. (2021) is anticipated to lay a solid platform for the researchers engaged in the study of medical image fusion. In another study, many problems with medical image fusion and various accomplishments made in the related sector are discussed in detail (Yadav and Yadav 2020). Further, numerous experiments are carried out to give an unbiased comparison of the conventional methodologies found in the literature.

Based on the survey, the existing pixel-level IF methods/techniques are classed into spatial domain (SD) and transform domain (TD) methods. The process of fusion is done directly on the source images using SD methods, and the resulting fused images have excellent spatial quality. The well-known spatial domain methods are averaging, non-negative matrix factorization (NMF), weighted average (WA), principal component analysis (PCA), linear discriminant analysis (LDA), intensity-hue-saturation (IHS), canonical correlation analysis (CCA), Brovey transform (BT), and independent component analysis (ICA). Despite being simple to use, spatial domain approaches have spectrum degradation. As image features exist at distinct scales in real world, the methods in spatial domain are not appropriate for medical image

fusion. Hence, researchers have gained more interest in multi-scale decomposition methods. These strategies consist of three steps: 1. Image Transform: using multi-scale decomposition (MSD) techniques, source images of contrasting modalities are decomposed into different coefficients (low frequency and high frequency) at various resolutions and spatial directions. 2. Image Fusion: decomposed coefficients are fused together to attain newer coefficients using a certain fusion rule 3. Inverse Transform: inverse MSD enforced to the fused coefficients produces the final fused/merged image. Pyramid transform (PT) and wavelet transform are bilateral fundamental approaches used for multi-scale or multi-resolution image fusion. Laplacian pyramid (LP), ratio of low pass pyramid (RLP), Gaussian pyramid (GP), contrast pyramid (CP), gradient pyramid (GrP), and morphological pyramid (MP) are few examples of image fusion methods based on pyramid transform.

On the other hand, wavelet transforms like discrete wavelet transform (DWT), discrete cosine transform (DCT), rotating WT (RWT), lifting WT (LWT); complex wavelet transforms (CWT) like dual-tree CWT (DTCWT) and Daubechies complex wavelet transform (DCxWT); advanced MSD transforms like curvelet transform (CVT), bandelet transform (BLT), ripplet transform (RPT), contourlet transform (CT), shearlet transform (ST); and shift invariant version of contourlet transform and shearlet transform called non-subsampled contourlet transform (NSCT) and non-subsampled shearlet transform (NSST) respectively are frequently used wavelet transform based image fusion methods. Salient feature-based methods like guided filter (GF) and its versions are running hot for its unique advantage of separating low-, middle-, and large-scale details separately. Recent years have seen the successful application of sparse representation (SR) techniques based on sparse coding and dictionary learning (DL) to produce fused images. Researchers are also interested in implementing hybrid/combination methods which have further enhanced the performance of fusion process (Hermessi et al. 2021; Azam 2022; Li et al. 2017; Kaur et al. 2021, 2019; Ma et al. 2019; Tawfik et al. 2021; Yadav and Yadav 2020; Chen et al. 2021). The comparison among various transform domain methods is given in Table 1.

Hence, image fusion is an approach in which the fused (resultant) image reserves all the extraneous and redundant data and details from the input images, which is extremely beneficial for functions comprising human perception and image processing. The fused result must adhere with the subsequent specifications to achieve the objective: (a) the most complimentary and important information/data from the primary source must be conserved in the merged image, (b) the fusion procedure shouldn't generate any artificial information that could distract the advanced image processing software or the human observer, (c) it must avoid imperfect conditions, such as noise and misregistration.

Table 1 Comparison among various pixel-level IF methods

Method	Advantages	Disadvantages
SD	Retain pixel originality, low complexity, and faster processing	Introduces spectral degradation in the resultant image
PT	Works directly on pixel values	Have poor SNR, do not provide spatial orientation selectivity, blocking effect occurs where the input images are significantly dissimilar
DWT	Preserves time and frequency details of an image	Shift variant, lack of directionality, not good at edges and textured regions
DTCWT	Eliminate shift-sensitive property, provide phase information	Computationally costly and requires large memory
DCxWT	Computationally faster, preserves edge and phase information	Low contrast, cannot describe shapes and contours from input images
LWT	Integral transform, non-uniform sampling, and adaptive design	Has problem of the spatial resolution
RPT	Provide details regarding the orientation of linear edges	Lack translation invariance causing pseudo-Gibbs phenomena
CT	Can preserve geometric regularity along edges	Reduced contrast, coupled with a lack of shift invariance, results in pseudo-Gibbs phenomena encompassing singularities
CoT	More efficiently provides edges and distinct singularities along curves	
ST	Directional selectivity (no limit on number of directions) and computational efficient	
NSCT	Can lessen the impact of incorrect registration and the pseudo-Gibbs phenomena	Complex to implement, high time and cost requirement
SR	Enhances resistance to misregistration and decreases visual artifacts	High computational complexity, less ability in preserving details
GF	Translation-variant, fast linear time filter method	May have halos near some edges, time-consuming
DL	Capable of modeling complex interactions among data, can automatically extract distinguishing characteristics from data	Higher complexity of the method and time-consuming
NN	Strong fault tolerance, anti-noise, adaptability	Needs large amount of training data

3 Comparative Analysis

This section offers insight into pair of medical image modalities from benchmark dataset, compares the outcomes with seven IF methods that have recently been published, and evaluates the merged images using seven image quality evaluation indexes. The pairs of three medical image datasets are acquired from

http://www.med.harvard.edu/aanlib/, an online resource for central nervous system imaging. Several transform domain methods like discrete wavelet transform (DWT), dual-tree complex wavelet transform (DTCWT), Laplacian pyramid (LP), curvelet transform (CVT), adaptive sparse representation (ASR), guided filter (GF), and hybrid method (NSST-MSMG-PCNN) are used for the comparative analysis of the IF process. To evaluate the performance, seven quality measures: edge strength ($Q^{AB/F}$), fusion loss ($L^{AB/F}$), fusion artifacts ($N^{AB/F}$), fusion similarity metric (Q_C), fusion quality index (Q_S), and indexes based on structural similarity (Q_Y) and phase congruency (Q_P) are used.

3.1 Image Quality Evaluation (IQE) Indexes

The outcome of the image fusion process is compared and evaluated both subjectively/qualitatively and objectively/quantitatively. The qualitative/subjective assessment makes use of personal inspection of an individual and has no rigorous mathematical models to determine image quality. However, it is challenging to do a qualitative analysis for a variety of reasons, such as expert's experience, viewing circumstances, and trained subjects. Consequently, a quantitative/objective evaluation of fusion quality is required in addition to the visual inspection to accurately measure an algorithm's effectiveness. The proportion of information contained in image details and features that are transmitted out of the primary source to the final fused image serves as a barometer for the performance of an IF method. An objective evaluation is done using following metrics (Xydeas and Petrović V, 2000, 2005; Yang et al. 2008; Piella and Heijmans 2003; Liu et al. 2008; Cvejic et al. 2006).

Edge strength. It calculates how much edge data from the input source images was transmitted to the merged image. To determine edge strength $s(m, n)$ and orientation value $o(m, n)$ of each pixel in the source images (A, B) and the fused image (F), a sobel edge operator is employed. Using these parameters, when comparing a source image to a fused image, the relative edge strength $S^{AF}(m, n)$ and orientation value $O^{AF}(m, n)$ are then defined as

$$S^{AF}(m,n) = \begin{cases} \frac{s_F(m,n)}{s_A(m,n)}, & \text{if } s_A(m,n) > s_F(m,n) \\ \frac{s_A(m,n)}{s_F(m,n)}, & \text{otherwise} \end{cases} \tag{1}$$

$$O^{AF}(m,n) = \frac{||o_A(m,n) - o_F(m,n)| - \pi/2|}{\pi/2} \tag{2}$$

These estimates are further utilized to determine edge strength $Q_s^{AF}(m, n)$ and orientation value $Q_o^{AF}(m, n)$ amidst the input source images and the final fused image which are then combined to determine edge information preservation coefficient using the following equation.

$$Q^{\mathrm{AF}}(m,n) = Q_s^{\mathrm{AF}}(m,n) * Q_o^{\mathrm{AF}}(m,n) \tag{3}$$

Normalized fusion performance is measured as a weighted sum of edge information preservation values for both input source images $Q^{\mathrm{AF}}(m,n)$ and $Q^{\mathrm{BF}}(m,n))$ and is defined as

$$Q^{\frac{\mathrm{AB}}{F}} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left(Q^{\mathrm{AF}}(m,n)w_A(m,n) + Q^{\mathrm{BF}}(m,n)w_B(m,n)\right)}{\sum_{m=1}^{M}\sum_{n=1}^{N}(w_A(m,n) + w_B(m,n))} \tag{4}$$

where (m,n) represent the image location, (M, N) defines the size of image, and the weight factors $w_A(m,n)$, $w_B(m,n))$ describe the perceptual relevance of respective input image pixel. The dynamic range of $Q^{\mathrm{AB}/F}$ is [0, 1]. It should be near to 1 for best possible value; i.e., higher value of fusion metric reflects better preservation of edge information in the fused image.

Fusion loss. The quantum of information lost during fusion is quantified as fusion loss. Although not present in the resultant image, this information is present in the input images. A loss of input information is designated by a weaker gradient strength in the resultant fused image. The loss is defined as

$$L^{\frac{\mathrm{AB}}{F}} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} r(m,n)\left[\left(\left(1 - Q^{AF}(m,n)\right)w_A(m,n) + ((1 - Q^{BF}(m,n)\right)w_B(m,n)\right)]}{\sum_{m=1}^{M}\sum_{n=1}^{N}(w_A(m,n) + w_B(m,n))} \tag{5}$$

where

$$r(m,n) = \begin{cases} 1, & \text{if } s_A(m,n) > s_F(m,n) \text{or} \quad s_B(m,n) > s_F(m,n) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Fusion artifacts or fusion noise. These are visual elements that add to the fused image, but which did not correlate to any input features during the fusion process. It is an erroneous data that adversely affect the value of the fused image and possesses major ramifications in some fusion applications and is represented as

$$N^{\frac{\mathrm{AB}}{F}} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} N(m,n)(w_A(m,n) + w_B(m,n))}{\sum_{m=1}^{M}\sum_{n=1}^{N}(w_A(m,n) + w_B(m,n))} \tag{7}$$

$$N(m,n) = \begin{cases} 2 - Q^{\mathrm{AF}}(m,n) - Q^{\mathrm{BF}}(m,n), & \text{if } s_F(m,n) > s_A(m,n) \text{and} \quad s_B(m,n) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $N(m,n)$ are local estimates of fusion artifacts evaluated at positions where fused gradients are sturdy than the input.

Fusion similarity metric. It assesses the degree to which batches of pixels in the fused image and the source image are similar in terms of spatial position. The metric is represented as

$$Q_C(A, B, F) = \sum_{w \in W} \left[\begin{array}{c} \text{sim}(A, B, (F|w))Q(A, (F|w)) + \\ (1 - \text{sim}(A, B, (F|w))Q(B, (F|w)) \end{array} \right] \tag{9}$$

$$\text{sim}(A, B, (F|w)) = \begin{cases} 0, & \frac{\sigma_{\text{AF}}}{\sigma_{\text{AF}}+\sigma_{\text{BF}}} < 0 \\ \frac{\sigma_{\text{AF}}}{\sigma_{\text{AF}}+\sigma_{\text{BF}}}, & \frac{\sigma_{\text{AF}}}{\sigma_{\text{AF}}+\sigma_{\text{BF}}} \geq 1 \\ 1, & \frac{\sigma_{\text{AF}}}{\sigma_{\text{AF}}+\sigma_{\text{BF}}} \geq 1 \end{cases} \tag{10}$$

$$\sigma_{\text{AF}} = \frac{1}{M-1} \sum_{m=0}^{M} (x_m - \overline{x})(y_m - \overline{y}) \tag{11}$$

where w is the analysis window, F is the final fused image, A and B are the input images, and W is the family of all windows. A higher score denotes a higher degree of spatial similarity amidst the source images and the final fused image.

Fusion quality index. It gauges the extent to which the prominent facts from source images are accurately transmitted to the fused image without deformity. The metric is computed as

$$Q(A, B, F) = \frac{1}{W} \sum_{w \in W} [s(w)Q(A, (F|w)) + (1 - s(w)Q(B, (F|w))] \tag{12}$$

$$s(w) = \frac{s(A|w)}{s(A|w) + s(B|w)}, \tag{13}$$

where $s(A|w)$, $s(B|w)$ are variance (or local saliencies) of images A, B respectively within window w. It depends upon contrast, sharpness, or entropy. $s(w)$ indicates the importance of source image A over B and $s(w) \in [0, 1]$. $Q(A, (F|w))$, $Q(B, (F|w))$ is the quality index over a window for input source images and the final fused image. In contrast to HVS, this index treats different quality measurements within each window equally.

Weighted fusion quality index, a different variant, offers better weightage to those windows where the input images with higher value of variance are achieved. It is then represented as

$$Q_S(A, B, F) = \sum_{w \in W} \delta(w)[s(w)Q(A, (F|w)) + (1 - s(w)Q(B, (F|w))] \tag{14}$$

where $\delta(w)$ is the normalized version of $\delta'(w)$ and $\delta'(w) = \max(s(A|w), s(B|w))$. Taking into consideration the applicability of edge information, the metric is modified to edge-dependent fusion quality index. It is figured with the edge images (A', B', F') in contrast to original image (A, B, F) and β is a constant that weighs the edge

contribution information. It is given as

$$Q_{S'}(A, B, F) = Q_S(A, B, F) * Q_S\left(A', B', F'\right)^{\beta} \tag{15}$$

Based on structural similarity. It makes use of local similarities among input source images as a similarity measure to evaluate various local areas. The grade of the composite image is determined by how near the $Q_Y(A, B, F)$ value is to 1. It is defined as

$$Q_Y(A, B, (F|w)) = \begin{cases} s(w)\text{SSIM}(A, (F|w)) + (1 - s(w))\text{SSIM}(B, (F|w)), \\ \text{if SSIM}(A, B|w) > 0.75 \\ \max\{\text{SSIM}(A, (F|w)), SSIM(B, (F|w))\}, \\ \text{if SSIM}(A, B|w) < 0.75 \end{cases} \tag{16}$$

The overall fusion metric is calculated as

$$Q_Y(A, B, F) = \frac{1}{W} \sum_{w \in W} [Q_Y(A, B, (F|w))] \tag{17}$$

Based on phase congruency (absolute measure). It is based on phase congruency measurement and its principal moments which impart an absolute measurement of image features. Assuming L blocks in an image, the average over the entire image is obtained as

$$Q_P(A, B, F) = \frac{1}{L} \sum_{l=1}^{L} P'_{\text{ab}}(l) \tag{18}$$

where P'_{ab} is the product of three correlation coefficients and is given as

$$P'_{\text{ab}} = \left(P_p\right)^a (P_{\text{ma}})^b (P_{\text{mi}})^c \tag{19}$$

Here a, b, c is exponential parameter which can be adjusted, ma and mi are the maximum and minimum moments, p is the phase congruency measurement.

3.2 Experimental Result and Analysis

The results obtained with well-established IF approaches are exhibited in visual and tabular form. The visual results for all the three datasets are provided in Figs. 1a–i, 2a–i and Fig. 3a–i. The initial medical image is a pair of CT and MRI images as indicated in Fig. 1a–b. It can be found that the obtained fused images as displayed in Fig. 1c–i

have more details and greater visual representation than the single modal image. The second pair as shown in Fig. 2a, b also consists of CT and MRI images, and Fig. 2c–i shows that the fusion results have better feature quality and contrast compared to the source images. The third pair accommodate a pair of T1 weighted MRI image (provide information on the body's anatomical structures) and T2 weighted MRI image (optimally show fluid and abnormalities) as shown in Fig. 3a, b. It can be ascertained from Fig. 3c–i that the features in the fused image have been enhanced with a good mixture of supplementary data from an individual image used an input. Besides the visual comparison, different image fusion methods compared under different IQE indexes are also tabulated in Table 2 corresponding to three sets of medical images.

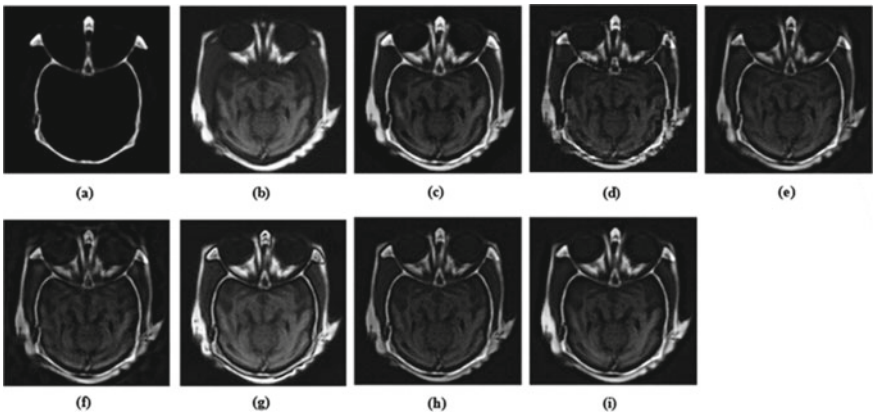


Fig. 1 Performance comparison of MMIF: **a**, **b** dataset 1 (CT-MRI); fused images using **c** LP; **d** DWT; **e** DTCWT; **f** CVT; **g** ASR; **h** GFS; **i** NSST-MSMG-PCNN methods

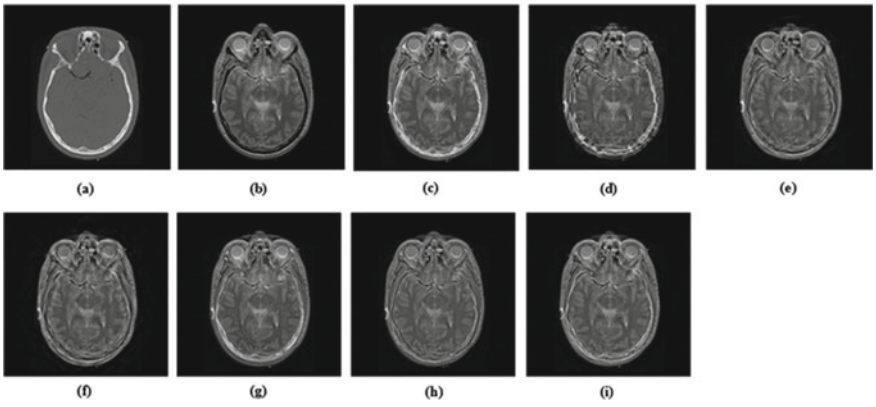


Fig. 2 Performance comparison of MMIF: **a**, **b** dataset 2 (CT-MRI); fused images using **c** LP; **d** DWT; **e** DTCWT; **f** CVT; **g** ASR; **h** GFS; **i** NSST-MSMG-PCNN methods

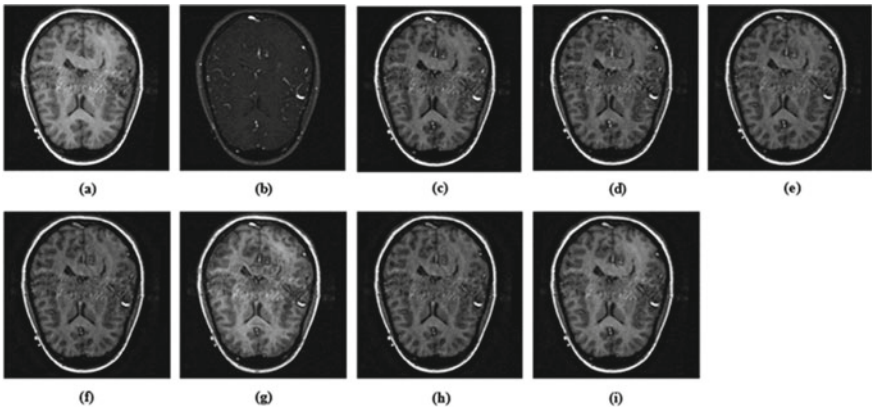


Fig. 3 Performance comparison of MMIF: **a**, **b** dataset 3 (MRT1-MRT2); fused images using **c** LP; **d** DWT; **e** DTCWT; **f** CVT; **g** ASR; **h** GFS; **i** NSST-MSMG-PCNNmethods

Table 2 Performance evaluation of various IF methods for three sets of medical images

	IQE indexes	Image fusion methods						
		LP	DWT	DTCWT	CVT	GF	ASR	NSST-MSMG-PCNN
Dataset 1 (MRI/ CT)	Q_P	0.538	0.259	0.403	0.321	0.498	**0.564**	0.54
	Q_S	0.528	0.564	0.614	0.566	**0.891**	0.715	0.765
	Q_C	0.482	0.535	0.533	0.445	**0.776**	0.648	0.671
	Q_Y	0.606	0.569	0.59	0.502	**0.857**	0.706	0.728
	$Q^{AB/F}$	**0.921**	0.811	0.856	0.842	0.901	0.896	0.887
	$L^{AB/F}$	0.068	0.13	0.129	**0.141**	0.085	0.1	0.104
	$N^{AB/F}$	0.012	0.06	0.016	0.017	0.014	**0.004**	0.01
Dataset 2 (MRI/ CT)	Q_P	0.438	0.288	0.401	0.362	0.464	**0.554**	0.473
	Q_S	0.795	0.799	0.811	0.709	**0.882**	0.87	0.861
	Q_C	0.428	0.508	0.513	0.332	**0.726**	0.61	0.637
	Q_Y	0.638	0.66	0.678	0.509	**0.87**	0.785	0.775
	$Q^{AB/F}$	**0.808**	0.748	0.765	0.756	0.772	0.74	0.777
	$L^{AB/F}$	0.172	0.199	0.216	0.22	0.22	**0.256**	0.207
	$N^{AB/F}$	0.02	0.053	0.019	0.024	0.008	**0.004**	0.016
Dataset 3 (MR-T1/ MR-T2)	Q_P	0.55	0.425	0.433	0.366	0.408	**0.62**	0.495
	Q_S	0.577	0.652	0.657	0.608	**0.801**	0.779	0.78
	Q_C	0.493	0.588	0.592	0.523	**0.783**	0.713	0.726
	Q_Y	0.597	0.646	0.659	0.597	**0.847**	0.782	0.777
	$Q^{AB/F}$	**0.878**	0.853	0.864	0.853	0.856	0.862	0.855
	$L^{AB/F}$	0.084	0.081	0.107	0.107	0.123	**0.134**	0.12
	$N^{AB/F}$	0.038	0.066	0.028	0.04	0.021	**0.004**	0.024

It is to be noted that edge strength, fusion loss, and fusion artifacts are complementary and signify a thorough assessment of the information fusion process, i.e., $Q^{AB/F} + L^{AB/F} + N^{AB/F} = 1$. Laplacian pyramid performs the best (high $Q^{AB/F}$) in all three datasets indicating a fused image rich in visual content, i.e., better edge information. An interesting result that shows for CVT in dataset 1 and ASR in dataset 2 and 3 respectively is an elevated fusion loss ($L^{AB/F}$). This results in an image with great deal of the contrast and features lost reflecting the weakest overall fusion performance. Further, ASR has fewer fusion artifacts ($N^{AB/F}$) in all the three datasets while DWT has the highest, thereby affecting the fused image adversely. The fact that the fusion algorithms perform differently on the various images is not surprising. In all the cases, the GF-based strategy is most frequently favored by the metrics Q_S, Q_C, Q_Y while Q_P gives the best results for ASR. Q_P is a feature-oriented method. Utilizing the phase congruency approach and based on the feature measurement, this index can effectively determine lines, edges, and other features reliably. The hybrid method NSST-MSMG-PCNN gave the second-highest values for metrics Q_S, Q_C, Q_Y, Q_P for all three datasets.

Consequently, it can be concluded from the outcomes that the hybrid method produces an image that successfully restores the essential as well as complementary aspects of both source images. Instead of ranking these approaches in a broad sense, this study determined the viability and validity of the image quality evaluation (IQE) indexes, or how these metrics function for diverse image modalities.

4 Conclusion

An inherent challenge of MMIF is to efficiently merge the most pertinent data and image details without information loss. Although numerous strategies have been put out, it still proves difficult to deliver and retain information effectively. As a result, the quality evaluation of MMIF is becoming more and more crucial in the disciplines of image analysis and for therapeutic purposes. In addition to achieving the goal of image quality management, an excellent objective evaluation approach may also direct the enhancement of image fusion methods/algorithms to determine the most effective algorithm for fusing images from various modalities. The results obtained in this paper are encouraging. It is found that the IF algorithms rely on the type of image being combined and the application for which it is being done. Also, these methods offer new opportunities for applications that enhance images. Further, by employing a particular assessment metric, a single fusion method might not always yield the same results on different images. Hence, in the future a novel pixel-level fusion algorithm will be developed which enables more precise analysis of multimodality medical images.

References

Azam MA et al (2022) A review on multimodal medical image fusion: compendious analysis of medical modalities, multimodal databases, fusion techniques, and quality metrics. Comput Biol Med 144. https://doi.org/10.1016/j.compbiomed.2022.105253

Chen J, Chen L, Shabaz M (2021) Image fusion algorithm at pixel level based on edge detection. J Healthcare Eng. article id 5760660. https://doi.org/10.1155/2021/5760660

Cvejic N, Łoza A, Bull D, Canagarajah N (2006) A similarity metric for assessment of image fusion algorithms. Int J Signal Process 2(3):178–182

El-Gamal FEZA, Elmogy M, Atwan A (2016) Current trends in medical image registration and fusion. Egyptian Inf J 17(1):99–124. https://doi.org/10.1016/j.eij.2015.09.002

Hermessi H, Mourali O, Zagrouba E (2021) Multimodal medical image fusion review: theoretical background and recent advances. Signal Process 183:1–28. https://doi.org/10.1016/j.sigpro.2021.108036

James AP, Dasarathy BV (2014) Medical image fusion—a survey of the state of the art. Inf Fusion 19(1):4–19. https://doi.org/10.1016/j.inffus.2013.12.002

Kaur H, Koundal D, Kadyan V (2021) Image fusion techniques: a survey. Arch Comput Methods Eng 28(7):4425–4447. https://doi.org/10.1007/s11831-021-09540-7

Kaur H, Koundal D, Kadyan V (2019) Multimodal medical image fusion: comparative analysis. ICCSP, 0758–0761. https://doi.org/10.1109/ICCSP.2019.8697967

Li S, Kang X, Fang L, Hu J, Yin H (2017) Pixel-level image fusion: a survey of the state of the art. Inf Fusion 33:100–112. https://doi.org/10.1016/j.inffus.2016.05.004

Liu Z, Forsyth DS, Laganiere R (2008) A feature-based metric for the quantitative evaluation of pixel-level image fusion. Comput Vision Image Understanding 109:56–68. https://doi.org/10.1016/j.cviu.2007.04.003

Ma J, Ma Y, Li C (2019) Infrared and visible image fusion methods and applications: a survey. Inf Fusion 45:153–178. https://doi.org/10.1016/j.inffus.2018.02.004

Piella G, Heijmans H (2003) A new quality metric for image fusion. Proc Int Conf Image Process 2:173–176

Tawfik N, Elnemr HA, Fakhr M, Dessouky MI, Abd El-Samie FE (2021) Survey study of multimodality medical image fusion methods. Multimed Tools Appl 80(4):6369–6396. https://doi.org/10.1007/s11042-020-08834-5

Xydeas CS, Petrović V (2000) Objective image fusion performance measure. Electron Lett 36(4). https://doi.org/10.1049/el:20000267

Xydeas CS, Petrović V (2005) Objective image fusion performance characterization. IEEE international conference on computer vision. https://doi.org/10.1109/ICCV.2005.175

Yadav SP, Yadav S (2020) Image fusion using hybrid methods in multimodality medical images. Med Biol Eng Comput 58(4):669–687. https://doi.org/10.1007/s11517-020-02136-6

Yang C, Zhang JQ, Wang XR, Liu X (2008) A novel similarity-based quality metric for image fusion. Inf Fusion 9:156–160. https://doi.org/10.1016/j.inffus.2006.09.001

Early Detection and Classification of Waterlogging Stress in Broccoli Plants Prior to Visual Symptom Appearance Through Electrophysiological Signal Analysis

Kavya Sai, Neetu Sood, and Indu Saini

Abstract Plants experience a variety of abiotic stresses as a result of environmental changes. These stresses differ significantly in plant electrophysiology's potential and frequency. The persistent physiological status of the plant is interpreted by electrical impulses in the plants. In this study, broccoli plants are cultivated in a soilless substrate and electrical signals are acquired from both healthy and waterlogging stress states. Using the BIOPAC®MP150 setup, the short-term electrophysiological signals were recorded for eight days in both the stress (4 days) and healthy (4 days) states. Using the fast Fourier transform, signals are converted into the frequency domain (FFT). Five window lengths are used to extract frequency domain characteristics. K-nearest neighbors (KNN) classifier is used for binary classification. Results were reported with a classification rate of up to 98.8%. The findings showed that plant electrical signals identified stress more accurately before its manifestation.

Keywords Brassica oleracea · Fourier transform · Abiotic stress · Electrophysiological signals · Feature extraction · Machine learning · Plant electrical activity

1 Introduction

Plants are multicellular, segmented organisms without a brain or other central processing unit to coordinate the environmental information compiled in each segment. Nevertheless, signaling pathways' reactions provide a biological foundation for learning, memory, calculation, and problem-solving (Primary and Perception: Biocommunication with Plants, Living Foods, and Human CellsHedrich 2012; Dicke et al. 2003; Gagliano et al. 2016). For interplant communication, plants use a range

K. Sai (✉) · N. Sood · I. Saini
Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab 144011, India
e-mail: kavyasy.ec.17@nitj.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_53

of useful signals, which are electrical activities relating to fluctuations in membrane potential, termed action potential (AP), variation potential (VP), and system potential (SP). A non-destructive stimulus can cause an action potential (AP), which is an electrical pulse that propagates quickly while retaining constant amplitude (cold, heat, and mechanical). The resting membrane potential changes in AP are characterized by a spike and are independent of the stimulus's intensity (Dziubí et al. 2001). AP is unlike from VP. VP is deliberate and the waves are triggered by numerous environmental stimuli. Contingent on the strength of the stimuli, changes in potentials can occur over either a long or short distance. (Vodeneev et al. 2015; Dziubinska et al. 2003; Pyatygin et al. 2008). The strength of the stimulus received is either directly or inversely proportional to the intensity of variation, VP has a changeable shape, amplitude, and time period. (Fromm and Lautner 2007; Sukhova and Sukhov 2021; Li et al. 2021). The plasma membrane's hyperpolarization or the apoplastic voltage's depolarization is reflected by the term "system potential" (SP). The magnitude and interval of SP likewise rely on the strength of the provocation, but a key distinction is the proton pump is sustainedly activated to cause membrane hyperpolarization, which initiates the SP (Zimmermann et al. 2009). Electrophysiology or plant electrical signal acquisition is obliged to acquire data from many essential plant activities like ecological stress inconsistency, translocation of ions, and transpiration in the form of continuous signals. Plants endure numerous varieties of stress from both living and non-living factors (Tuteja 2007; Zhang et al. 2022; S A 2017; Atkinson and Urwin 2012). Abiotic stress is brought on by environmental factors that are not alive, such as a flood, a drought, an extreme temperature, excessive irrigation, excess nutrients, or swift winds. Living things like fungi, bacteria, parasites, weedy plants, and dangerous insects are what create biotic stress (Nejat and Mantri 2017). Two distinct techniques are employed to assess the electrophysiological condition of the plants: invasive and non-invasive. Invasive approaches involve cutting or splitting the plant tissue to obtain the signal (Luo et al. 2021; Yaddanapudia and Soodb xxxx). Non-intrusive techniques utilize exterior measurements by gently implanting ultra-thin metal electrodes into plant tissues or by employing sticky gel electrodes over the plant's leaves (Meder et al. 2021). Different window lengths and domains can be used to examine the electrophysiological signals obtained from plants, including the frequency domain, time domain, and time–frequency domain. A widely used approach from earlier literature research includes signal preprocessing, feature extraction, and classification. Researchers presented a preprocessing method using an IIR filter that uses wavelet packet transform (WPT) to enhance filter parameters for plant signals and remove low-frequency drift (Das et al. 2015). By developing several ways to create a classification model, statistical feature extraction achieves a Mahalanobis distance-based classification of chemical stress induced in tomato plants (H_2SO_4, NaCl, Ozone) in various quantities (Chatterjee et al. 2018, 2015; Chatterjee 2017). Scientific studies on tomato plants under drought stress and biotic stress caused by infesting spider mites have presented a classification rate of up to 98% to classify stress utilizing distinctive windowing approaches. Also, iron deficiency in tomato plants is classified using electrophysiological signals while reporting a 98% classification rate using gradient boosted tree machine learning classifier (Tran and

Camps 2021; Najdenovska et al. 2021a, b; Tran et al. 2019). In soybean databases, low light, osmotic, and cold stressors are automatically classified using machine learning techniques together with interval arithmetic in two different approaches, where one is image-based and the second is signal-based. (Pereira et al. 2018). Additionally, work presented on a similar database, by enhancing the electrophysiological signals of plants in the spectral domain, where spectral components are combined and binary classification reaches accuracies up to 98%, allows for the categorization of abiotic stress. Also, different nutrient deficiencies are classified through empirical mode decomposition and sample space reduction together with machine learning classifiers reporting an average accuracy of 98.8% in tomato plants (Sai et al. 2022a, b).

The authors of this paper attempted to classify waterlogging stress in broccoli plant species. By transforming the signal data into the frequency domain, surveyed feature extraction, feature grouping, followed by the classification are accomplished. The subsequent sections provide a comprehensive explanation of the proposed methodology.

2 Material and Methods

The Biopac®MP150 equipment is used to acquire the electrophysiological signals from the plants. For the simultaneous acquisition of data from two plants, two channels have been set up in the Acknowledge 4.2 software version of BIOPAC MP150. For the data collection, two young broccoli (Brassica oleracea) plants with 3–4 leaves and a height of 14 cm were considered for the study. Atmospheric humidity varies from 25 to 45% during the day to 65% to 85% at night. The temperature is maintained between 12ºC and 15ºC during the day and 5 ºC and 9 ºC at night. A non-soil medium made of vermiculite, perlite, and coco peat, combined in a 3:2:1 ratio was used to cultivate the plants. NPK fertilizer, which contains potassium (16%), nitrogen (16%), and phosphorus (16%), by volume, is treated for plant growth.

2.1 Proposed Workflow

The overall workflow is portrayed in Fig. 1. Signals obtained from experiments designed for data acquisition in both stressed and healthy conditions. A step-by-step process is proposed for signal preprocessing, frequency domain conversion, feature extraction, and classification.

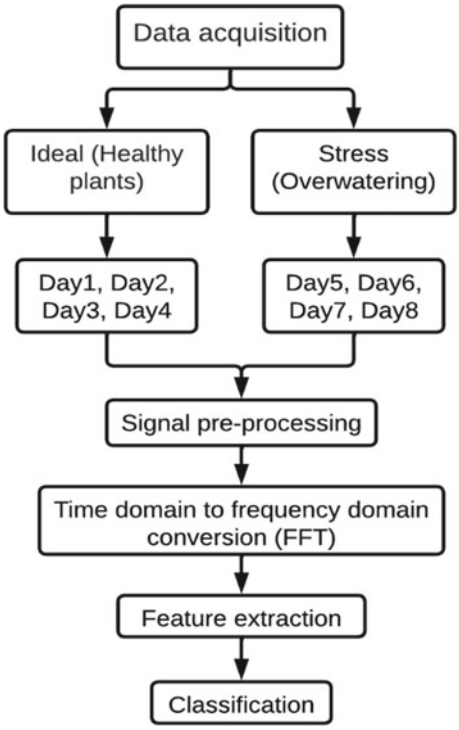


Fig. 1 Illustration of work methodology

2.2 Data Acquisition

Signals from healthy plants were obtained using a BIOPAC MP150 for four successive days; subsequently, the transplantation of healthy plants was done into a soilless medium made of coco peat, vermiculite, and perlite (3:2:1) that had been excessively watered to create waterlogging stress. Followed by, the signals are acquired for four days successively after transplanting plants into a stress environment, as illustrated in Fig. 2. Non-invasive sticking gel electrodes made of AgCl are placed at two nodes (shoot, stem above the root) of the plant: Positive terminal of MP150 probe is connected to shoot electrode and the negative terminal is connected to an electrode placed on the stem just above the root, and the third node of the probe is placed in the ground. Using two channels of the BIOPAC®MP150, electrophysiological signals from two plants were recorded at a sampling rate of 125 Hz. The healthy data is acquired for the first four days 1, 2, 3, and 4, and stressed data is acquired for days 5, 6, 7, and 8. For developing a classification model, days 1 and 5 data are set as a combination of healthy and stressed classes. Similarly, days 2, 6; 3, 7; and 4, 8 are combined for processing and classification.

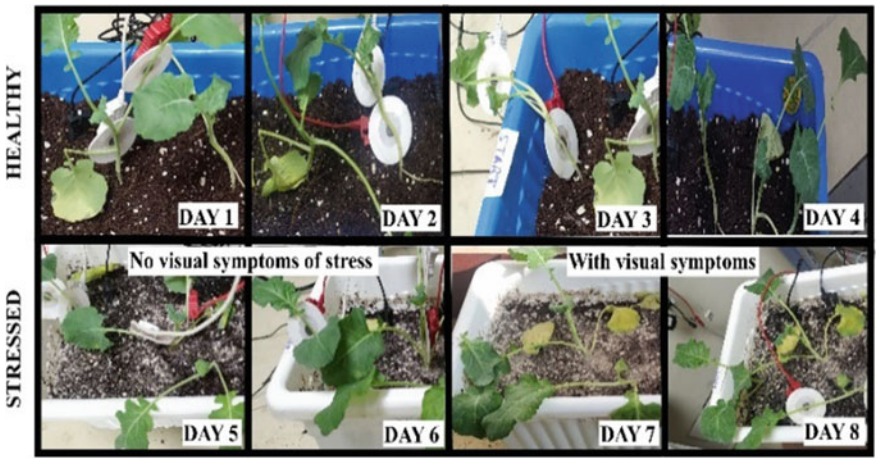


Fig. 2 Plants illustrating healthy and stressed states during data acquisition, no noticeable visual symptoms on days 5 and 6 while days 7 and 8 presented visual symptoms

2.3 Signal Preprocessing

The initial raw signal data acquired from the device is preprocessed to check for motion artifacts and powerline interference. The device's noise and other artifacts are ignored during the acquisition by using a high-pass filter at a frequency of 0.5 Hz and a gain of 500. For each plant, data is recorded for 20 min duration, by acquiring 1,50,000 samples of data for both healthy and stressed environments.

2.4 Time to Frequency Conversion (FFT)

Fast Fourier transform (FFT), a technique that computes the discrete Fourier transform (DFT) of a sequence, is initially used to convert time realm signals collected from broccoli plants into the frequency domain. The discrete Fourier transform procedure known as the fast Fourier transform (FFT), where the base-2 logarithm, decreases the number of computations required for N points from $2N^2$ to 2N log N (Dass et al. 2013; Akin 2002). The time series data is transformed into a frequency domain representing various frequencies over amplitude. Instead of using the entire dataset to extract characteristics, the electrical signal frequencies are then separated into chunks with substantially shorter durations (windows). In this experiment, five different window spans with samples of 200 (W1), 500 (W2), 750 (W3), 1000 (W4), and 2000 (W5) are taken into consideration. The extraction of features in five window spans followed by binary classification is ensured to analyze signals with a range of window lengths.

2.5 Feature Extraction

To classify the delineated signal for each of the five different window lengths, Matlab® R2021b was utilized to compute seven signal features in the frequency domain. In addition to the core descriptive features like maximum (f1), minimum (f2), median (f3), and mean (f4), the statistical features in a spectral domain such as the bandwidth of power (f5), spectral entropy (f6), and signal-to-noise ratio (SNR) (f7) are included to detect and classify changes occurred in the signal data pre and post-stress. Equations (1) and (2)'s parameters minimum (min) (f2) and maximum (max) (f1) acquire the minimum and maximum potentials amid the specified samples. As demonstrated in Eq. (3), the median (mn) (f3) calculates the intermediate value of the potentials organized in either rising or downward order. The mean (m) (f4) is calculated using Eq. (4) as the total number of samples divided by the sum of all samples. Power bandwidth (pb) (f5), which is determined using Eq. (5), is the variance in frequency between the positions where the spectrum decreases by at least three decibels below the reference. The power dispersal of a frequency range generated using Eq. (6) is evaluated by spectral entropy (se) (f6). The signal-to-noise ratio (SNR) (f7), which is determined by Eq. (7), is the ratio of the power of the signal to the power of the noise specified in decibels.

$$f_1 = \min(x(n)) \tag{1}$$

$$f_2 = \max(x(n)) \tag{2}$$

$$f_3 = \frac{1}{2}\sum_{l=0}^{n} x(l) \tag{3}$$

$$f_4 = \frac{1}{n}\sum_{l=1}^{n} x(l) \tag{4}$$

$$f_5 = F_m - F_n \tag{5}$$

$$f_6 = E[-\sum_{l=0}^{n} p(l)\log 2p(l)] \tag{6}$$

$$f_7 = 10.\log_{10}\left(\text{Power}_{\text{signal}}/\text{Power}_{\text{noise}}\right) \tag{7}$$

In order to make an analysis infer, classification is accomplished by utilizing both individual features and features that are taken into account in clusters. The mean, median, and power bandwidth (G1) are included in the first set of clustered attributes, while the signal-to-noise ratio (SNR), entropy, minimum, and maximum are included in the cluster (G2).

2.6 Developing a Classification Model

Initially, the data is experimented with various classifiers suggested in the previous literature studies. The k-nearest neighbor (kNN) classification technique outperformed supplementary algorithms including support vector machines (SVMs), decision trees (DT), naïve Bayes (NB), ensemble classification, and discriminant analysis, according to experiments employed on data. Although kNN and DT both classifiers produced good classification rates in preliminary studies, the kNN classifier is utilized to standardize the classification model. For the purposes of validation (training) and testing the model, the data is divided into two subsets. The validation subset contains 20% of the values, and the validation model possesses 80% of the total data. The feature extraction from each dataset is used to create the training and testing datasets. The accuracy of the classifier model developed during the training phase is assessed using the test data. Each window length's characteristics are formed into training and testing datasets. Subsequently, binary classification is performed.

3 Results

Models were able to distinguish between the plant's healthy state and stressed state using the kNN machine learning algorithm and reported specific performance measures for both individual and aggregated feature clusters. Individual features are initially employed to perform binary classification; subsequently, the features are sequentially separated into two groups which are grouped as G1 and G2. The features f1, f2, f3, and f4, are clustered into G1 while G2 contains features f5, f6, and f7 as discussed in the earlier section. Table 1 notifies the observed classification rates attained by various classifiers in different groups and window lengths. W2 and W3 window lengths have higher classification rates for both feature groups than other window lengths. Accuracies of both the groups in the Day 4-Day 8 data are higher than in the Day 1-Day 5 data. When comparing Day3-Day7, and Day4-Day8 outcomes to Day1-Day5 findings, Day2-Day6 results are modest, according to an average interpretation. Figure 3 demonstrates a clear visual comparison. The test data has a mean classification rate of 86.08% for respective features analyzed individually and 91.86% for feature groupings. The accuracy levels attained by different features in various window lengths for the appropriate day groups are shown in Fig. 3. According to Fig. 3a, window length W1 has a reduced classification rate for features with f3 and f4 and a better classification rate for features with f6 and f7.

For the features extracted using W3 as the window length for Day1-Day5, findings showed a satisfactory classification rate for each individual feature. Overall, accuracy has deteriorated for features f3 and f4 across all window lengths. The classification results reported for window length W3 is satisfactory in all phases (4 days) of the study, according to the observations in Fig. 3a–d, while window length W1 displayed weaker performance metrics. In the initial days, the window

Table 1 Day-wise classification rates achieved when features are considered in groups for different window lengths

Days	Groups	W1 (%)	W2 (%)	W3 (%)	W4 (%)	W5 (%)
Day 1–Day 5	G1	96.3	98.25	88.95	88.55	90.40
	G2	94.1	98.15	93.85	92.05	89.65
Day 2–Day 6	G1	98.3	97.65	98.75	95.10	92.05
	G2	91.3	96.60	91.90	87.55	90.65
Day 3–Day 7	G1	84.3	98.10	98.75	94.25	90.20
	G2	85.3	97.55	98.95	89.15	84.05
Day 4–Day 8	G1	92.6	97.55	98.65	94.15	93.70
	G2	94.6	98.60	97.55	91.05	95.45

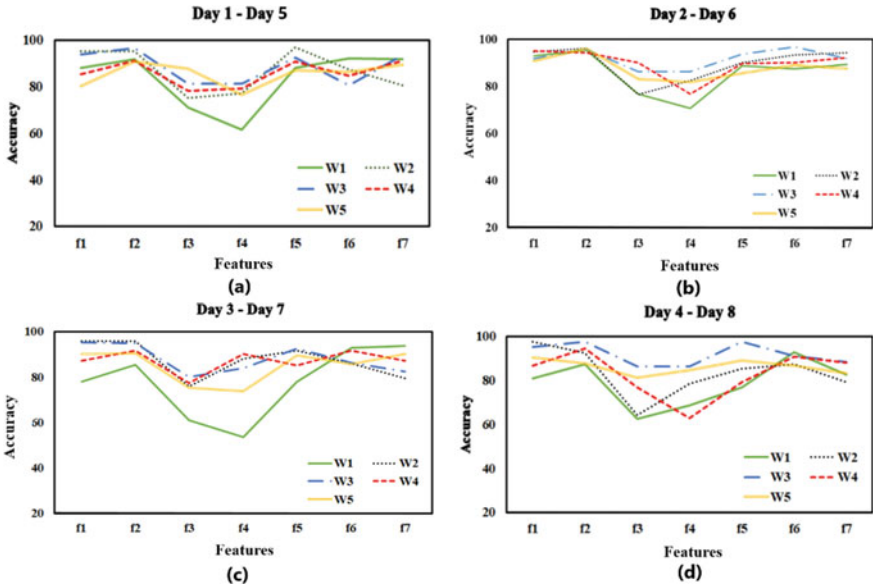


Fig. 3 **a**. Specific feature classification rate for healthy state (Day 1) and stressed state (Day 5) **b** Illustration of the classification rate of individual features for a healthy state (Day 2) and stressed state (Day 6) **c** Illustration of the classification rate for individual features in a healthy state (Day 3) and stressed state (Day 7) **d** Illustration of the classification rate for individual features in a healthy state (Day 4) and stressed state (Day 8)

length W4 reported satisfactory performance (Fig. 3a–c), and a decreased performance in Fig. 3d. Mean, median, and power bandwidth, the first three measures, demonstrated a gradual increase, whereas power bandwidth and SNR, the third and fourth features, depicted increased variation through the duration of the days. Additionally, for five window lengths, feature f6 achieved the greatest classification rate throughout all experiments. In all window lengths, the features f2 and f5 reported

decent performance. For features f3 and f4, the window length W1 has demonstrated a sharp decreasing gradient. For all window lengths throughout the duration of the days, the individual feature performance of f7 likewise produced an excellent classification accuracy. The fluctuation in performance measures over window lengths of W2 to W5 is reduced in the Day1-Day5 and Day2-Day6 databases. From Day 1 to Day 4, the third window length (W3) exhibits the least amount of variance. For the features f3, f4, and f5, the initial window length (W1) exhibits the greatest fluctuation over the course of the four days. The window length, W5, exhibits only slight variation. Before the onset of visual signs of stress, features f1 and f2 in Fig. 3 a, b demonstrated excellent performance measures. Due to increased stress, possible deviation increased every day. There have been no reports of any visual signs in plants in the early days following stress provision. On Day 3, visible symptoms (yellowing leaves) start to manifest. Increased potential separability is a key factor in improving classification performance. Figure 4 displays the average performance of all features while considering them as groups for classification. With an average accuracy of 91.76%, the grouping of the features resulted in better outcomes than the individual features.

The graphical illustration in Fig. 4 demonstrates that signal data might be more accurately categorized both before and after the onset of visible signs of stress. Up to 90% classification accuracy is attained in almost all feature groups in various window lengths. Additionally, more than 90% of the grouped aspects are accurate after the onset of the visual symptoms. In Fig. 4, it is evident that the classification rate increased from Day 1 to Day 4 as a result of potential separability. 90% accuracy in diagnosing plant stress before the onset of visible signs is a substantial improvement for crop protection techniques. Observations from the presented analysis indicate the data's ability to be classified among both early and intermediate symptom stages.

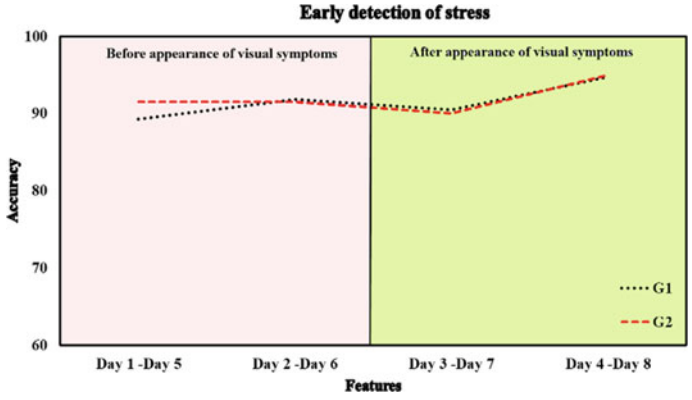


Fig. 4 Day-wise grouped feature classification rate illustrated before and after the appearance of stress symptoms

4 Conclusion

Studies previously have not reported to develop a model to classify the waterlogging stress from a healthy state using signal analysis in the frequency domain. The study of electrophysiological signals in plants is still in its adolescence. In this work, the model by kNN classifier was trained and tested using individual features, and the overall classification accuracy was 86.08%. Additionally, grouping features to enhance classification performance had a 91.86% average accuracy. The accuracy results from the experiment were fivefold cross-validated. A broader research focus can be established by investigating different possibilities for assessing the electrophysiological signals of plants in several stresses for various species. The ability to identify plant stress before the emergence of visual symptoms seems to be a major significant advancement in crop protection.

References

Akin M (2002) Comparison of wavelet transform and FFT methods in the analysis of EEG signals. J Med Syst 26:241–247

Atkinson NJ, Urwin PE (2012) The interaction of plant biotic and abiotic stresses: from genes to the field. J Exp Bot 63:3523–3544

Chatterjee SK (2017) An approach towards plant electrical signal based external stimuli monitoring system

Chatterjee SK, Das S, Maharatna K, Masi E, Santopolo L, Mancuso S, Vitaletti A (2015) Exploring strategies for classification of external stimuli using statistical features of the plant electrical response. J R Soc Interface. https://doi.org/10.1098/rsif.2014.1225

Chatterjee SK, Malik O, Gupta S (2018) Chemical sensing employing plant electrical signal response-classification of stimuli using curve fitting coefficients as features. Biosensors (Basel). https://doi.org/10.3390/bios8030083

Das S, Ajiwibawa BJ, Chatterjee SK, Ghosh S, Maharatna K, Dasmahapatra S, Vitaletti A, Masi E, Mancuso S (2015) Drift removal in plant electrical signals via IIR filtering using wavelet energy. Comput Electron Agric 118:15–23

Dass S, Holi M, Rajan S (2013) A comparative study on FFT, STFT and WT for the Analysis of auditory evoked potentials

Dicke M, Agrawal AA, Bruin J (2003) Plants talk, but are they deaf? Trends Plant Sci 8:403–405

Dziubí H, Tr K, Ebacz ̨ Zawadzki T (2001) Transmission route for action potentials and variation potentials in Helianthus annuus L

Dziubinska H, Filek M, Koscielniak J, Trebacz K (2003) Variation and action potentials evoked by thermal stimuli accompany enhancement of ethylene emission in distant non-stimulated leaves ofVicia faba minorseedlings. J Plant Physiol 160:1203–1210

Fromm J, Lautner S (2007) Electrical signals and their physiological significance in plants. Plant Cell Environ 30:249–257

Gagliano M, Vyazovskiy VV, Borbély AA, Grimonprez M, Depczynski M (2016) Learning by association in plants. Sci Rep. https://doi.org/10.1038/srep38427

Hedrich R (2012) Ion channels in plants. Physiol Rev 92:1777–1811

Li JH, Fan LF, Zhao DJ, Zhou Q, Yao JP, Wang ZY, Huang L (2021) Plant electrical signals: a multidisciplinary challenge. J Plant Physiol 261:153418

Luo Y, Li W, Lin Q, Zhang F, He K, Yang D, Loh XJ, Chen X (2021) A morphable ionic electrode based on thermogel for non-invasive hairy plant electrophysiology. Adv Mater 33:2007848

Meder F, Saar S, Taccola S, Filippeschi C, Mattoli V, Mazzolai B, Mazzolai B, Saar S, Taccola S (2021) Ultraconformable, self-adhering surface electrodes for measuring electrical signals in plants. https://doi.org/10.1002/admt.202001182

Najdenovska E, Dutoit F, Tran D, Plummer C, Wallbridge N, Camps C, Raileanu LE (2021a) Classification of plant electrophysiology signals for detection of spider mites infestation in tomatoes. Appl Sci (switzerland) 11:1–16

Najdenovska E, Dutoit F, Tran D, Rochat A, Vu B, Mazza M, Camps C, Plummer C, Wallbridge N, Raileanu LE (2021) Identifying general stress in commercial tomatoes based on machine learning applied to plant electrophysiology. Appl Sci (Switzerland). https://doi.org/10.3390/app11125640

Nejat N, Mantri N (2017) Plant immune system: crosstalk between responses to biotic and abiotic stresses the missing link in understanding plant defence. Curr Issues Mol Biol 23:1–16

Pereira DR, Papa JP, Saraiva GFR, Souza GM (2018) Automatic classification of plant electrophysiological responses to environmental stimuli using machine learning and interval arithmetic. Comput Electron Agric 145:35–42

Backster C, Primary perception: biocommunication with plants, living foods, and human cells

Pyatygin SS, Opritov VA, Vodeneev VA (2008) Signaling role of action potential in higher plants. Russ J Plant Physiol 55:285–291

S A (2017) Molecular strategies for development of abiotic stress tolerance in plants. Cell Cellular Life Sci J. https://doi.org/10.23880/cclsj-16000113

Sai K, Sood N, Saini I (2022b) Abiotic stress classification through spectral analysis of enhanced electrophysiological signals of plants. Biosyst Eng 219:189–204

Sai K, Sood N, Saini I (2022) Classification of various nutrient deficiencies in tomato plants through electrophysiological signal decomposition and sample space reduction. Plant Physiol Biochem. https://doi.org/10.1016/J.PLAPHY.2022.07.022

Sukhova E, Sukhov V (2021) Electrical signals, plant tolerance to actions of stressors, and programmed cell death: Is interaction possible? Plants. https://doi.org/10.3390/PLANTS10081704

Tran D, Camps C (2021) Early diagnosis of iron deficiency in commercial tomato crop using electrical signals. Front Sustain Food Syst. https://doi.org/10.3389/fsufs.2021.631529

Tran D, Dutoit F, Najdenovska E, Wallbridge N, Plummer C, Mazza M, Raileanu LE, Camps C (2019) Electrophysiological assessment of plant status outside a Faraday cage using supervised machine learning. Sci Rep. https://doi.org/10.1038/s41598-019-53675-4

Tuteja N (2007) Abscisic acid and abiotic stress signaling. Plant Signal Behav 2:135–138

Vodeneev V, Akinchits E, Sukhov V (2015) Variation potential in higher plants: mechanisms of generation and propagation. Plant Signal Behav. https://doi.org/10.1080/15592324.2015.1057365

Yaddanapudia KS, Soodb N, Sainic I, A testimony of inter-plant communication through electrophysiological signal analysis

Zhang H, Zhu J, Gong Z, Zhu J-K (2022) Abiotic stress responses in plants. Nat Rev Genet 23:104–119

Zimmermann MR, Maischak H, Mithöfer A, Boland W, Felle HH (2009) System potentials, a novel electrical long-distance apoplastic signal in plants, induced by wounding. Plant Physiol 149:1593–1600

Detection of Parkinson's Disease Based on Biological Features Using Deep Neural Models

Nikita Aggarwal, Barjinder Singh Saini, and Savita Gupta

Abstract The correct and timely finding of Parkinson's disease (PD) is a very challenging issue because this disease is diagnosed when 60 to 80% of neurons get vanished. Basically, it arises due to the death of dopamine neurons of the substantia nigra. To avoid misdiagnose, this experimental study developed a deep learning-based model for detecting PD at its early stages. For classification, the various biological features have been taken based on CSF, urine, plasma, and serum. Also evaluated the performance of the proposed DNN model with other highly implemented supervised learning classifiers (SVM, Naïve Bayes, and extreme gradient boosting). The obtained results reveal that the developed DNN-based classifier gives better metric results as compared to other classifiers, i.e., 97.43%, 97.36%, 98.87%, and 98.18% accuracy, precision, recall, and F1-score, respectively. Henceforth, this developed DNN classifier may support practitioners or physicians to identify the disease in its early phases.

Keywords Biological features · Feature engineering · Machine learning · Deep learning · Parkinson's disease

N. Aggarwal (✉) · B. S. Saini
Department of Electronics and Communication, Dr. B.R.Ambedkar, National Institute of Technology, Jalandhar, India
e-mail: aggarwal.nikita.91@gmail.com

B. S. Saini
e-mail: sainibs@nitj.ac.in

S. Gupta
Department of Computer Science, University Institute of Technology, Chandigarh, India
e-mail: savita2k8@yahoo.com

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_54

1 Introduction

PD becomes the 2nd multitudinous neurodegenerative disorder (Pahuja and Prasad 2022). Generally, it appears due to the deficit of neurotransmitters (dopamine) in the substantia nigra (part of basal ganglia) (Booij et al. 1997), and this disorder affects over 10 million individuals. People older than 60 years are more likely to be afflicted with the disease (Aggarwal et al. 2021a). Basically, tremors, rigid muscles, slowed movement; speech changes, and impaired posture are the common symptoms of PD (Aggarwal et al. 2021b). Therefore, the detection of the disease in its early stages is a confusing and immense issue and sometimes problematic for highly knowledgeable physicians to diagnose precisely (Prashanth et al. 2017) due to the overlying of manifestations with other alike disorders like multiple system atrophy, dementia with Lewy bodies, hydrocephalus, and so on (Hiner 2006). To avoid these issues of misdiagnosis and early detection, the ability to quantify disease development and evaluate the effectiveness of treatments using biological characteristics has been shown to be more objective.

So, there is a need to obtain the automatic system as a biomarker to categorize PD in healthy people. In order to categorize data, many deep learning (DL) and machine learning (ML)-based methods have been established. These learning models are actively being researched for use in the analysis of medical data and may even be able to identify PD patients in the earliest stages (Shiiba et al. 2020). These learning models are widely used in many medical fields like pattern and speech recognition, natural language processing, signal and image processing, etc.

A lot of work has been done in the detection of PD patients from healthy people by using one or a combination of various feature sets like biological, clinical (motor and non-motor), extracted attributes from imaging, and so on. Hence, some examples in the investigation of PD from healthy people are presented in literature which have taken biological features lonely or with some combination such as in the study (Pahuja et al. 2020), the authors applied a total of five biological features (plasma, RNA, urine, and serum) to develop an automatic model based on stacked auto-encoder-based classification model for the detection of PD patients from healthy people. In (Leger et al. 2020), the authors used biological as well as non-motor features. The data was obtained from PPMI, and the metric performance was assessed on ten cross-validations. Similarly, (Pahuja and Prasad 2022) used MRI imaging-based extracted features, biological features, and SBR features from SPECT imaging. The authors developed the CNN- based model for classification. The authors of Wang et al. (2020) used non-motor, SBR features, and biological to develop a deep learning-based model.

The chief contributions of this research are

1. The study collected the nine biological attributes that consist of the different tests of cerebrospinal fluid (CSF), urine, serum, and plasma for binary classification between healthy people and PD patients.

2. Developed the DL-based deep neural network (i.e., DNN) model for differentiating the disorder and also evaluated the performance of the developed model with other extremely used ML classifiers (XGBoost, SVM, and Naïve Bayes) in various studies.

The rest of the manuscript is segregated into various sections as the detail of the database and methods which have been implemented for this experimental research work in Sect 2. Section 3 provides the classification results after pre-processing of data and also evaluated the performance metrics with existing ML classifiers. The last Sect. 4 presents the conclusion of this experimental work.

2 Detail of Data and Methods Used

2.1 Data

The data was collected on 24 January 2023 from the PPMI database (Marek et al. 2018). This organization mainly focuses on finding biomarkers for PD. This experiment used data based on biological features that consist of the different tests of cerebrospinal fluid (CSF), urine, serum, and plasma. The detail of demographic features and the explanation of each biological feature for PD and healthy people are given in Tables 1 and 2, respectively. A total of 221 entities have been considered of which 142 and 79 are male and female, respectively.

Table 1 Demographic detail (age in Mean ± Standard deviation) for healthy and PD

Category	Sample size (n = 376)	Age	Sex(F/M)
PD	131	62.30 ± 10.05	42/89
Healthy	90	60.40 ± 11.05	37/53

Table 2 Description of nine biological features (Mean ± SD) for Healthy people and PD

S. No.	Biological features	Healthy people	PD patients
1	Aβ 1–42	1035.54 ± 542.05	919.75 ± 437.34
2	α-syn	1702.14 ± 777.52	1516.30 ± 619.70
3	Ptau 181	16.65 ± 6.98	14.25 ± 5.77
4	tTau	191.50 ± 72.38	169.07 ± 61.09
5	Total Cholesterol	186.97 ± 39.54	188.67 ± 44.13
6	Triglycerides	122.32 ± 62.79	114.56 ± 56.28
7	NfL	13.19 ± 6.38	13.07 ± 5.72
8	IGF-1	140.60 ± 56.17	135.70 ± 54.8460
9	Total di-22:6-BMP	11.60 ± 10.80	15.95 ± 12.80

A total of nine biological features are considered and the category of each biological characteristic is given below in which they belong.

1. Amyloid beta peptide 1–42 (Aβ 1–42), Alpha-synuclien (α-syn), Phosphorylated tau at threonine 181 (ptau181), and Total tau (tTau) are the biomarkers of CSF.
2. Total Cholesterol and Triglycerides are the biomarkers of plasma.
3. Neurofilament light (NfL) and serum-insulin-like growth factor 1 (IGF-1) are the biomarkers of serum.
4. Di-22:6-Bis (monoacylglycerol) phosphate (total di-22:6-BMP) is the biomarker of urine.

2.2 Visualization of Data

To visualize the biological features of healthy people and PD, the box plots (Fig. 1a–i) have been generated. In essence, these graphs are utilized to show the variations, middle value, and distribution shape of a dataset. The box plot's borders represent the 25th and 75th percentiles of the information, while the box's center line represents the median. The expanded whiskers are used to plot the changes outside of the 25th and 75th percentiles (presented as outliers). If the number of outliers is high, there will be variability in the data. The Z-score has been applied uniformly (complete data) to overcome the problem of outliers.

2.3 Methodology

Basically, the data on biological features is collected from the PPMI database. A total of nine features were extracted from four main features, i.e., CSF, urine, serum, and plasma. Then proposed a deep learning-based model after comparing the results of the proposed model with other existing classifiers. The flow of work is shown in Figs. 1 and 2.

2.4 Classifiers

The function of ML and DL-based classifiers will be deliberated in the parts that follow. In addition, the results of each classifier's classification were subjected to a tenfold cross-validation test, during which they were compared with those of healthy people and PD patients.

Naïve Bayes (NB). This classifier obeys the Bayes Theorem. The supposition of NB classifier demonstrates that, there is conditional independence between each pair of characteristics that contribute to the value of the class parameter *Cy*. Therefore, for

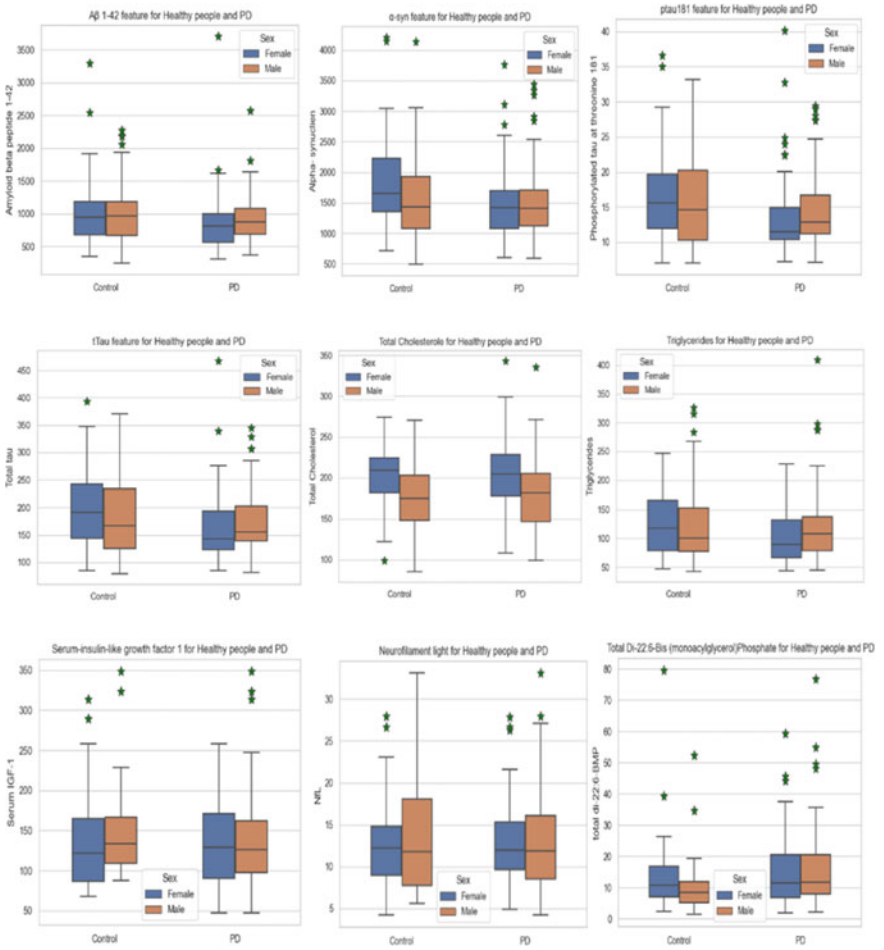


Fig. 1 Represented the box plots of nine biological features for healthy people and PD

classification, the Gaussian NB is applied, and the conditional probability $P(ai|Cy)$ is supposed to be

$$P(a_i|C_y) = \frac{1}{\sqrt{2\pi\sigma_{C_y}^2}} \exp\left(-\frac{(a_i - \mu_{C_y})^2}{2\sigma_{C_y}^2}\right).$$

2. Support vector machine (SVM). The chief aim of this classifier is to discover the hyperplane in a higher dimensional that utilizes a kernel function to separate data points according to their possible classes (Aggarwal et al. 2023). In this experiment, a radial bias function kernel has been used.

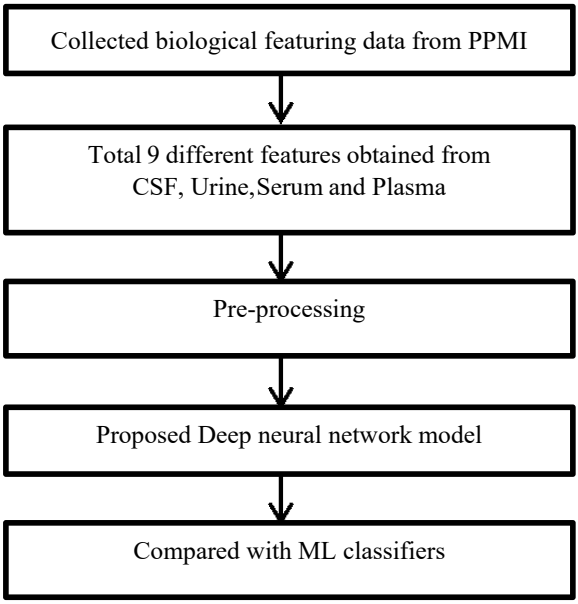


Fig. 2 Methodology of research work

3. Extreme gradient boosting (XGB). The major goal of this model's use of gradient-boosted decision trees was speed and performance. It assists in optimizing all hardware and memory resources for tree-boosting approaches. This is capable of carrying out the three main gradient boosting techniques: stochastic, regularized, and gradient boosting. In contrast to other libraries, it also enables the insertion and fine-tuning of regularization parameters.
4. Deep neural networks (DNN). These DL-based models have many hidden layers in their design. The input of features is given in the input layer, i.e., 9 features and the output of data is taken from the output layer. The hidden layers will process the data. Therefore, in this experiment, the number of neurons in the 1st, 2nd, 3rd, and 4th dense layers are 200, 100, 50, and 2 respectively that have been taken and evaluated the result at 50 epochs for iteration.

3 Results and Discussion

3.1 Metrics

For the assessment of the performance of various classifiers of ML for classification, the accuracy, recall, precision, and F1-score measure have been employed:

- $\text{Accuracy} = \frac{\text{Tr\_Pos+Tr\_Neg}}{\text{Tr\_Neg+Tr\_Pos+Fa\_Pos+Fa\_Neg}}$
- $\text{Recall} = \frac{\text{Tr\_Pos}}{\text{Tr\_Pos+Fa\_Neg}}$

Table 3 Performance given by all classifiers (in percentage)

Classifier	Accuracy	Precision	Recall	F1-score
NB	95.43	95.72	96.48	97.09
SVM	97.18	96.15	98.31	97.14
XGB	96.73	97.18	97.48	97.29
Proposed DNN	97.43	97.36	98.87	98.18

- $\text{Precision} = \frac{\text{Tr\_Pos}}{\text{Tr\_Pos+Fa\_Pos}}$
- $F1 - \text{Score} = \frac{2\text{Tr\_Pos}}{2\text{Tr\_Pos+Fa\_Pos+Fa\_Neg}}$

where True_Pos is true positive, False_Pos is false positive, True_Neg is true negative, and False_Neg is false negative. The data is divided into testing (consisting of 20% of the data) and training (consisting of 80% of the data) goals. Additionally, all of the tests were carried out using Python version 3.9 on an Intel(R) Xeon(R) W-2255 CPU operating at 3.70 GHz 3.70 GHz, with 128 GB of RAM, a 64-bit operating system, and an × 64-based processor.

3.2 Results

Table 3 illustrated the performance of binary classification for the detection of PD in healthy people. It has been found that the developed DNN classifier showed provided better metric outcomes as compared with other supervised models, i.e., 97.43%, 97.36%, 98.87%, and 98.18% in accuracy, recall, F1-score, and precision respectively as compared to other ML classifiers.

The second highest results are provided by the SVM classifier, i.e., 97.43%, 98.87%, 98.18%, and 97.36% of accuracy, recall, $F1$-score, and precision, respectively. On other hand, NB and XGB classifiers are also performed well then the results provided in the literature.

3.3 Comparison with Existing Literature

No doubt, DL classifiers outperform than ML classifiers in many situations. Hence, some examples in the investigation of PD from healthy people are presented in literature which have taken biological features lonely or with some combination such as in the study (Pahuja et al. 2020), the authors applied a total of five biological features (plasma, RNA, urine, and serum) to develop an automatic model based on stacked auto-encoder-based classification model for the detection of PD patients from healthy people. In (Leger et al. 2020), the authors used biological as well as non-motor features. The data was obtained from PPMI, and the metric performance

Table 4 Comparison of developed DNN model with related literature

References	Technique	Accuracy (%)
Pahuja and Prasad (2022)	CNN-based model	97.1
Pahuja et al. (2020)	Stacked auto-encoder	93.33
Leger et al. (2020)	XGB	88.2
Wang et al. (2020)	Ensemble DNN model	96.68 ± 1.06
Proposed	DNN model	97.43

was assessed on ten cross-validations. Similarly, (Pahuja and Prasad 2022) used MRI imaging-based extracted features, biological features, and SBR features from SPECT imaging. The authors developed the CNN-based model for classification. The authors of Wang et al. (2020) used non-motor, SBR features, and biological to develop a deep learning-based model (Table 4).

4 Conclusion

Generally, PD is a rapidly rising neuro-disorder, and it occurs due to the death of transmitter neurons of substantia nigra that is why the onset identification of the disorder is much imperative. Hence, it has been detected that the developed DNN classifier has the capability to distinguish disease in its onset phases. The proposed DNN model is applied to nine biological features of CSF, urine, plasma, and serum. It is found that the DNN-based model provided better metric outcomes as compared with other models, i.e., 97.43%, 98.87%, 98.18%, and 97.36% in accuracy, recall, F1-score, and precision, respectively. The complexity and iteration time of the developed model is less due to the simple design of the model. In the future, these DNN-based models can be upgraded by involving some more clinical attributes.

Acknowledgements PPMI—a public-private partnership—is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners, including Abbott Laboratories, GE Healthcare, Avid Radiopharmaceuticals, Bristol-Myers Squibb, Covance, Élan, Genentech, GlaxoSmithKline, Eli Lilly and Company, Merck and Co., Meso Scale Discovery, Pfizer, Hoffmann-La Roche, UCB (Union Chimique Belge), Biogen Idec, and Piramal.

References

Aggarwal N, Saini BS, Gupta S (2021) The impact of clinical scales in Parkinson's disease: a systematic review. Egypt J Neurol Psychiatry Neurosurg 57(1). https://doi.org/10.1186/s41983-021-00427-9

Aggarwal N, Saini J, Saini BS, Gupta S (2021) Different classification approaches for early detection of parkinson's disease. Proc Int Conf Women Res Electron Comput Wrec:84–88. https://doi.org/10.21467/proceedings.114.12

Aggarwal N, Saini BS, Gupta S (2023) Multi-class and binary classification of Parkinson's disease and SWEDD variants using SBR features derived from SPECT imaging. In: 2023 6th international conference on information systems and computer networks (ISCON). Mathura, India, pp 1–5. https://doi.org/10.1109/ISCON57294.2023.10112104

Booij J et al (1997) [123I]FP-CIT SPECT shows a pronounced decline of striatal dopamine transporter labelling in early and advanced Parkinson's disease. J Neurol Neurosurg Psychiatry 62(2):133–140. https://doi.org/10.1136/jnnp.62.2.133

Hiner BC (2006) Differential diagnosis of Parkinson's disease: a new blood test? Clin Med Res 4(4):246–247. https://doi.org/10.3121/cmr.4.4.246

Leger C, Herbert M, DeSouza JFX (2020) Non-motor clinical and biomarker predictors enable high cross-validated accuracy detection of early PD but lesser cross-validated accuracy detection of scans without evidence of dopaminergic deficit. Front Neurol 11(May):1–19. https://doi.org/10.3389/fneur.2020.00364

Marek K et al (2018) The Parkinson's progression markers initiative (PPMI)—establishing a PD biomarker cohort. Ann Clin Transl Neurol 5(12):1460–1477. https://doi.org/10.1002/acn3.644

Pahuja G, Prasad B (2022) Deep learning architectures for Parkinson' s disease detection by using multi-modal features. Comput Biol Med 146(January):105610. https://doi.org/10.1016/j.compbiomed.2022.105610

Pahuja G, Nagabhushan TN, Prasad B (2020) Early detection of parkinson's disease by using SPECT imaging and biomarkers. J Intell Syst 29(1):1329–1344. https://doi.org/10.1515/jisys-2018-0261

Prashanth R, Dutta Roy S, Mandal PK, Ghosh S (2017) High-accuracy classification of parkinson's disease through shape analysis and surface fitting in 123i-ioflupane SPECT imaging. IEEE J Biomed Heal Inf 21(3):794–802. https://doi.org/10.1109/JBHI.2016.2547901

Shiiba T, Arimura Y, Nagano M, Takahashi T, Takaki A (2020) Improvement of classification performance of Parkinson's disease using shape features for machine learning on dopamine transporter single photon emission computed tomography. PLoS ONE 15(1):1–12. https://doi.org/10.1371/journal.pone.0228289

Wang W, Lee J, Harrou F, Sun Y (2020) Early detection of Parkinson's disease using deep learning and machine learning. IEEE Access 8:147635–147646. https://doi.org/10.1109/ACCESS.2020.3016062

EEG-based Binary Classification of Brain State of Activities Level Using a Single-Sensor Headset

Rakesh Kumar Rai and Dushyant Kumar Singh

Abstract Electroencephalography (EEG) is a non-invasive procedure that is used to measure and record the electrical activity of the brain. EEG uses scalp electrodes to measure brain activity, commonly used to diagnose epilepsy, brain tumors, sleep disorders, and stroke. The goal of this paper is to classify the brain state activities of the individual based on the EEG readings obtained through the Neuro Sky Headset. In this paper, several classification methods were utilized in order to accurately predict the subject's brain state activities, including logistic regression, linear SVC, nonlinear SVC, and gradient boosting. The analysis in terms of accuracy, precision, recall, and F1-score is done to figure out performance of these methods on brain activity data. The gradient boosting algorithm yielded a maximum accuracy of 81%.

Keywords EEG · Brain signals · Machine learning · Classification · Brain state activity

1 Introduction

Electroencephalography (EEG) is defined as a graphical recording of neural electrical activity in the brain. It is measured as the potential differences caused by ionic current within the neurons. Brain computer interfaces (BCI) are devices that directly observe and interpret EEG signals to control machines and automata (Chao et al. 2018). EEG technologies for their non-invasive, high temporal resolution, and low acquisition cost properties are frequently used for EEG signal collection. Motion picture EEG signals, which record brain activity during user-imagined motions, have several applications in the fields of neuro-rehabilitation (Cantillo-Negrete et al. 2018), game control

R. K. Rai (✉) · D. K. Singh
Department of Computer Science and Engineering, MNNIT, Allahabad, Prayagraj, UP 211004, India
e-mail: rakesh.2021rcs54@mnnit.ac.in

D. K. Singh
e-mail: dushyant@mnnit.ac.in

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_55

(Liao et al. 2012), and motion restoration for the paralyzed (Tariq et al. 2018), and are the focal points of ongoing research. Additionally, EEGs are applied in the diagnosis and investigation of epilepsy, a condition typified by recurrent seizures. In recent years, EEG research has branched out into exploring different areas, such as drowsiness (Lim et al. 2014), cognitive stress (Lim and Chia 2015), automated wheelchairs (Mirza et al. 2015), and more, all utilizing BCI devices. An overview of the various deep learning architectures available for the task of object detection and classification has been provided (Ansari and Singh 2019). The EEG output is a graphical representation of electrical signals in the brain, showing brainwave voltage over time or frequency which can be viewed on a computer (Alnafjan et al. 2017).

Liang et al. (2022) state that current deep learning-based methods in this area require a lot of labeled data. Categorizing EEG signals takes time and a skilled pathologist. Only a portion of the training data is labeled in their consistency-based semi-supervised seizure prediction model (CSSPM). Their approach is based on the consistency regularization principle, which says a resilient model should produce consistent results for the same inputs despite perturbations. Using stochastic augmentation and dropout, they consider the neural network as a whole as a stochastic process, and they impose a consistency constraint to minimize present prediction discrepancies. LSSVM has been applied in the detection of epilepsy through various studies, including those of Bajaj and Pachori (2012), Joshi et al. (2014), Li and Wen (2009), and Sharma and Pachori (Sharma and Pachori 2015). Furthermore, Li and Wen (2011) applied it in a clustering-based approach for EEG signal classification, and Guler and Ubeyli (2007) compared its results with those of probabilistic neural networks (PNNs) and multilayer perceptron neural networks (MLPNNs), finding that SVM yielded better accuracy while classifying EEG signals. N.C et al. (2022) proposed that most multivariate machine learning research has focused on identifying children with psychiatric conditions among typically developing children. They used two machine learning benchmarks to do this. The initial problem is to predict the seven most common DSM-V psychiatric disorders, of which each person may have more than one symptom. The second method involves predicting the severity of psychiatric symptoms using behavioral and cognitive measures.

Different methods for recognizing humans have been created and divided into two categories based on their features and the situations in which they can be used (Singh and Kushwaha 2016). Ma et al. (2020) suggest that with the development of brain science and biomedical engineering, as well as EEG signal analysis techniques, the use of EEG signals to monitor human health has become a prominent research area. This paper creates a depth factorization machine model to evaluate the EEG signal, allowing us to use EEG data to forecast the binomial state of the eyes based on user interaction analysis. The discovery shows that eye health can be used to determine weariness and overall health. According to MC Guerrero et al. (Subasi and Erçelebi 2005), EEG frequency data was used to classify epileptic patients with traditional classification methods. Fourier analysis was employed to acquire features, and Python was utilized to implement the methods. After a comparison of metrics and performance, it was found that artificial neural networks had the highest accuracy of 86%. Bhardwaj et al. (2022) proposed to use hybrid genetic programming (HGP)

to extract features and classify EEG data. They utilized a NeuroSky MindWave 2 dry electrode unit with a single channel for recording EEG signals.

The main contributions of this work are:

(1) To extract the enhanced features from the preprocessed EEG dataset using Fast Fourier Transform (FFT).
(2) To measure the performance of logistic regression, linear SVC, nonlinear SVC, and gradient boosting in predicting a person's brain state in terms of accuracy, precision, recall, and F1-score.

The rest of this article is laid out in the following way: Sect. 2 covers the experimental setup, which includes data collection, data preparation and data preprocessing, as well as exploratory data analysis. Section 3 looks into the proposed methodology, which involves feature engineering and the creation and training of a machine learning model. Lastly, the last section of this article discusses the results and potential future scope.

2 Experimental Setup

2.1 Data Collection

The MIDS cohort at UC Berkeley's School of Information gathered EEG data from 30 participants during an experimental session. Participants were split into two groups of 15 and exposed to a 5 min film. Each group viewed the movie in a 15-person group setting and EEG activity was recorded using Mind Wave headsets at 1 s intervals. During the movies, participants were assigned tasks such as blinking, relaxation, instruction, mental math, listening to music, watching a Doritos commercial (comedy), thinking of stuff, and color counting as mentioned in Table 1. The brain waves that have been used in this experiment are delta, theta, low alpha, high alpha, low beta, high beta, low gamma, and mid gamma.

Table 1 Data description

Stimuli	Brain waves
Blinking, relaxation, instruction, mental math, listening to music, watching a Doritos commercial (comedy), thinking of stuff, and color counting	Delta, theta, low alpha, high alpha low beta, high beta, low gamma, mid gamma

2.2 Data Preparation and Preprocessing

First, we removed any irrelevant details from the EEG data table. We excluded the date and time columns since the creators of the dataset had already taken care of the timing synchronization (aside from 'Indra time'). Any rows with a signal quality of 100 or more were eliminated because these data packets were not beneficial. (In other words, a lower signal quality was preferable). We removed all of the records from the table that were collected when the participant was absent. Group 1 was shown stimulus 1, while Group 2 stayed outside when their headsets were in place. Consequently, this generated unidentifiable information that could not be utilized. Even though this information is not available, it will be easy to set a baseline since each stimulus begins with a baseline-setting activity. EEG data and subject metadata were merged based on the ID column, forming a data frame with 20 columns. Unnecessary labels were removed, converting the problem to a binary classification of math vs relax. The dataset was then left with 1870 samples and 15 columns.

2.3 Exploratory Data Analysis

EDA is the process of exploring and analyzing datasets in order to gain insights and discover relationships between variables. To begin with, the occurrence of every category, i.e., math and relax was inspected to determine if the dataset was balanced or not, and the following conclusion was reached.

Figure 1 shows that the dataset is balanced, with 936 samples for math label and 934 samples for relax label; therefore no sampling techniques are required.

Figure 2 shows the relationship between the EEG categorical features (math and relax) and the EEG power (delta, theta, low alpha, high alpha, low beta, high beta, low gamma, and mid gamma). Each scatter plot is either blue or orange, with blue indicating the relax label and orange representing the math label.

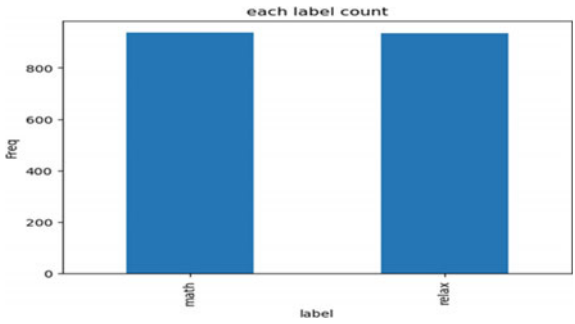


Fig. 1 Frequency of math and relax label

Fig. 2 Scatter plot of each categorical label, i.e., math and relax with respect to the egg power features, i.e., 'delta', 'theta', 'low_alpha', 'high_alpha', 'low_beta', 'high_beta', 'low_gamma', 'mid + gamma'

3 Methodology

This paper initially presents the proposed flow diagram as illustrated in Fig. 3, then examines the various machine learning models used for predicting the subject's action in this study. The evaluated models include logistic regression, linear SVC, nonlinear SVC, and gradient boosting. To enable precise measurements to be used early in the process, precise and efficient models for predicting the user's activity are sought.

3.1 Features Engineering

Feature engineering is the process of extracting features from raw data using domain knowledge in order to create more relevant and useful information to feed into a machine learning algorithm. This can include creating new features from existing data, combining existing features, or selecting a subset of existing features. First, the values of the browser_latency feature were converted to float, and the values of the categorical features were converted from object to categorical. After this, the categorical features (math and relax) were taken, and their values were converted

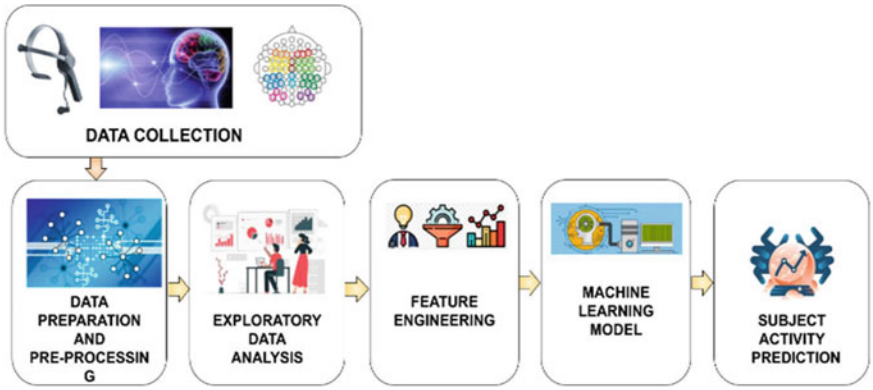


Fig. 3 Proposed process flow diagram

to one hot encoded value (n ->0.1), and labels were stored in a new variable, their values got converted to 0 and 1, and the label feature from the dataset was dropped. And for the egg_power (delta, theta, low alpha, high alpha, low beta, high beta, low gamma, and mid + gamma), the median for each list was taken, and it was saved as a new feature in the data frame called 'egg features mean'. For the last attribute with raw values, a better representation was generated. Each cluster of 512 raw values provided by the device was roughly taken, and Fast Fourier Transform (FFT) was applied to produce the power spectrum.

The Fast Fourier Transform (FFT) is a popular tool used in EEG signal processing to gain insight into the frequency-domain representation of the signal. By using the FFT, the power spectrum of the EEG signal can be obtained by computing the squared magnitudes of the FFT coefficients. This power spectrum can then be used to identify dominant frequency bands, such as alpha (8–12 Hz), beta (12–30 Hz), and gamma (>30 Hz), which are associated with different cognitive processes. In EEG signal processing, the Fast Fourier Transform (FFT) is typically used to gain insight into the frequency-domain representation of the signal. This can be beneficial in understanding the power spectrum of the signal, which can provide insight into the neural activity taking place.

The Fast Fourier Transform is an effective way to calculate the discrete Fourier transform of a finite set of samples of a continuous function. Mathematically, the DFT can be defined as a set of Q complex numbers $p[q]$, where q ranges from 0 to Q-1, is expressed as:

$$P[k] = \sum q = 0 \text{ to } Q - 1 \; p[q] \exp\left(-\frac{2\pi ikq}{Q}\right) \tag{1}$$

where k also ranges from 0 to Q-1.Here, $\exp(-2\pi ikq/Q)$ in the DFT represents a frequency component at k/Q Hz in the signal. The DFT uses this term to calculate the complex amplitudes of the frequency components present in the signal.

Subsequently, the output was logarithmically constrained to create feature vectors of 1870 dimensions, and the median of each vector was extracted to be added as a new feature in the dataset. Eventually, the raw_values feature was removed from the dataset. Any additional features that were not going to be used in the training process were eliminated, such as 'id', 'raw_values', 'eeg_power', 'Unnamed: 0', 'indra_time', 'reading_time', 'updatedAt', and 'createdAt'. The values were then normalized by calculating the mean and standard deviation.

3.2 Machine Learning Models

The concept of machine learning originated from the need to enable computers to acquire multiple skills, such as recognizing brain state activity, facial features, handwriting, and speech, through the use of algorithms and statistical models (Rai et al. 2023). The dataset was divided into two sections: a training set that contained 80% of the data and a testing set that contained the remaining 20%. Logistic regression, linear support vector classifier, nonlinear support vector classifier, and gradient boosting classifier were used to build the machine learning model.

3.2.1 Logistic Regression

As a type of supervised learning, logistic regression produces a discrete class label that may have a binary value (e.g., 0 or 1). Mathematically, it is often written as:

$$r = \text{logistic}(pTQ + s) \tag{2}$$

where p is a vector of weights, Q is a vector of features, s is a bias, and logistic is the sigmoid function.

3.2.2 Linear SVC

Linear support vector classifier is a supervised machine learning algorithm that can solve binary classification problems. It aims to maximize the margin between two classes of data points in order to find a linear boundary between them. It can be defined as follows:

$$y(k) = \text{sign}(pkq + s) \tag{3}$$

where p represents weights of the features, s represents bias, and q represents feature vector.

3.2.3 Nonlinear SVC

A nonlinear support vector classifier is a type of classifier that classifies data points using support vector machines (SVMs). When there is no linear separation between the data points, this type of classifier is used. The SVM works by mapping the data points to a higher dimensional space and then linearly separating the points in that space.

3.2.4 Gradient Boosting

Gradient boosting classifiers are supervised machine learning algorithms that can be used to solve classification and regression problems. It brings together weak learners (individual trees) into a single ensemble model. The weak learners are fitted sequentially, with each attempting to correct the mistakes of the one before it. The predictions from each of them are then added together to form the final prediction. While adding new models, it employs a gradient descent algorithm to minimize loss. All these models were made using the training dataset and analyzed using metrics such as accuracy, precision, recall, and F1-score.

4 Result and Discussion

Logistic regression, linear SVM, nonlinear SVC, and gradient boosting were used to classify the MIDS dataset. With XG boosting, accuracy in the high 80 s was achieved, but further analysis revealed that most of the models accurately predicted the user's actions. Unclassified data was removed, and numerous incorrectly classified labels were merged, while all 12 math variables were combined into a single label.

4.1 Models' Performance

The model's performance was assessed using various evaluation metrics including the confusion matrix, accuracy, precision, recall, and F1-score presented in Table 2.

Table 2 Model performance evaluation

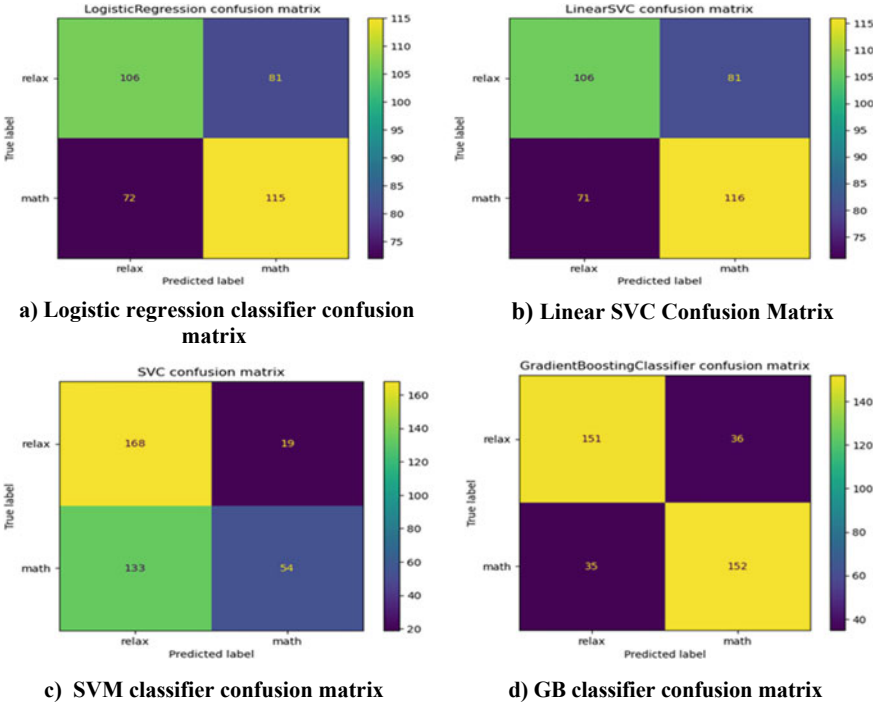
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic regression	59	59.5	59	59
Linear support	59.35	60.5	60.5	59.5
Nonlinear SVC	59.35	65	59.5	55.5
Gradient boosting	81	81	81	81

Table 2 shows that the values of accuracy, precision, recall, and F1-score for logistic regression, linear SVC, and nonlinear SVC are low which means that the model is not performing very well and may need to be adjusted or improved. Low accuracy values show that these models are making many incorrect predictions, while low precision values indicate that these models are returning too many false positives. These three models also have low recall values which means that the model is not detecting all of the relevant items, while their low F1-score shows that these models are not balancing precision and recall well. Gradient boosting has the highest value of accuracy, precision, recall, and F1-score among the models used which shows that the model is more effective than the rest of the three models in predicting the subject class. High values of these measures show that the model is operating effectively and can be relied upon to predict outcomes with better accuracy.

In Fig. 4a, the logistic regression classifier predicted 221 labels correctly out of a total of 374 labels, with 106 being for relaxation and 115 for math, yielding an accuracy of 59%. Figure 4b showed that the linear support vector classifier also predicted 222 labels correctly, with an accuracy of 59.35%, which is almost identical to the logistic regression classifier. Figure 4c showed that the nonlinear support vector classifier also had 222 labels correctly predicted out of 374, with an accuracy of 59.35%, which is in line with the other two classifiers. Lastly, Fig. 4d shows that the gradient boosting classifier performed well, predicting 303 true labels (151 for relax and 152 for math) out of 374 labels with an accuracy of 81%. It can be concluded that the gradient boosting model was the most accurate of the models listed in the table. It had a high precision value and was able to accurately predict the subject's action.

5 Conclusion

We presented an effective method for accurately distinguishing between users' math and relaxation activities. We first removed any irrelevant data from the collected dataset before conducting an exploratory data analysis to analyze trends, discover patterns, and validate hypotheses using statistical summaries and graphical visuals. The various machine learning algorithms were utilized to train the models and the results were examined through a range of evaluation metrics such as accuracy, precision, recall, and F1-score. The highest accuracy of 81% on the MIDS dataset for correctly identifying the user's action was achieved by the gradient boosting algorithm among all the models tested. Going forward, a variety of machine learning and deep learning techniques will be explored and their hyper parameters will be tuned in order to further enhance the accuracy of the predictions. Additionally, class labels beyond math and relax can be taken into account for the categorization.



a) Logistic regression classifier confusion matrix

b) Linear SVC Confusion Matrix

c) SVM classifier confusion matrix

d) GB classifier confusion matrix

Fig. 4 **a** Logistic regression classifier, **b** Linear SVC, **c** Nonlinear SVC and **d** Gradient boosting classifier

References

Alnafjan A, Hosny M, Al-Ohali Y, Al-Wabil A (2017) Review and classification of emotion recognition based on eeg brain-computer interface system research: a systematic review. Appl Sci 7:1239. https://doi.org/10.3390/app7121239

Ansari MA, Singh DK (2019) Review of deep learning techniques for object detection and classification. In: Verma S, Tomar R, Chaurasia B, Singh V, Abawajy J (eds) Communication, networks and computing. CNC 2018. Communications in computer and information science, vol 839. Springer, Singapore. https://doi.org/10.1007/978-981-13-2372-0_37

Bajaj V, Pachori RB (2012) Classification of seizure and nonseizure EEG signals using empirical mode decomposition. IEEE Trans Inf Technol Biomed 16(6):1135–1142. https://doi.org/10.1109/TITB.2011.2181403

Bhardwaj H et al (2022) Personality prediction with hybrid genetic programming using portable EEG device. Comput Intell Neurosci 2022. https://doi.org/10.1155/2022/4867630

Cantillo-Negrete J et al (2018) Motor imagery-based brain-computer interface coupled to a robotic hand orthosis aimed for neurorehabilitation of stroke patients. J Healthcare Eng 2018

Chao H, Zhi H, Dong L, Liu Y (2018) Recognition of emotions using multichannel EEG data and DBN- GC-based ensemble deep learning framework. Comput Intell Neurosci 2018. https://doi.org/10.1155/2018/9750904

Guler I, Ubeyli ED (2007) Multiclass support vector machines for EEG-signals classification. IEEE Trans Inf Technol Biomed 11(2):117–126

Joshi V, Pachori RB, Vijesh A (2014) Classification of ictal and seizure-free EEG signals using fractional linear prediction. Biomed Signal Process Control 9:1–5

Li Y, Wen PP (2011) Clustering technique-based least square support vector machine for EEG signal classification. Comput Methods Prog Biomed 104(3):358–372

Liang D et al (2022) Semisupervised seizure prediction in scalp EEG using consistency regularization. J Healthc Eng 2022. https://doi.org/10.1155/2022/15730

Liao LD et al (2012) Gaming control using a wearable and wireless EEG-based brain-computer interface device with novel dry foam-based sensors. J Neuroeng Rehabil 9:1–12

Lim A, Chia W (2015) Analysis of single-electrode EEG rhythms using MATLAB to elicit correlation with cognitive stress. Int J Comput Theory Eng 7:149–155. https://doi.org/10.7763/IJCTE.2015.V7.947

Lim CKA, Chia WC, Chin SW (2014) A mobile driver safety system: Analysis of single-channel EEG on drowsiness detection. In: 2014 International conference on computational science and technology (ICCST). Kota Kinabalu, pp 1–5. https://doi.org/10.1109/ICCST.2014.7045175

Ma P, Gao Q (2020) EEG signal and feature interaction modeling-based eye behavior prediction research. Comput Math Methods Med 2020. https://doi.org/10.1155/2020/2801015

Mirza IA et al (2015) Mind-controlled wheelchair using an EEG headset and arduino microcontroller. In: 2015 International conference on technologies for sustainable development (ICTSD). Mumbai, India, pp 1–5. https://doi.org/10.1109/ICTSD.2015.7095887

N. C et al (2022) A benchmark for prediction of psychiatric multimorbidity from resting EEG data in a large pediatric sample. Neuroimage 258(June):119348. https://doi.org/10.1016/j.neuroimage.2022.119348

Rai RK, Giri P, Singh I (2023) Machine learning for speech recognition. In: Dutta P, Chakrabarti S, Bhattacharya A, Dutta S, Shahnaz C (eds) Emerging technologies in data mining and information security. Lecture notes in networks and systems, vol 490. Springer, Singapore. https://doi.org/10.1007/978-981-19-4052-1_17

Sharma R, Pachori RB (2015) Classification of epileptic seizures in EEG signals based on phase space representation of intrinsic mode functions. Expert Syst Appl 42(3):1106–1117

Singh D, Kushwaha D (2016) Analysis of face feature based human detection techniques. Int J Control Theory Appl 9:173–180

Siuly YL, Wen P (2009) Classification of EEG signals using sampling techniques and least square support vector machines. Rough sets and knowledge technology. Springer, Berlin, pp 375–382

Subasi A, Erçelebi E (2005) Classification of EEG signals using neural network and logistic regression. Comput Methods Programs Biomed 78:87–99. https://doi.org/10.1016/j.cmpb.2004.10.009

Tariq M, Trivailo PM, Simic M (2018) EEG-based BCI control schemes for lower-limb assistive-robots. Front Hum Neurosci 12:312

Text Analysis for Information Retrieval Using NLP

Kuldeep Vayadande, Harshal Baru, Abhishek Kashid, Aniket Kulkarni, Prathamesh Londhe, and Atharv Vanjari

Abstract The application of natural language processing (NLP) methods in textual analysis for information retrieval is examined in this work. In this research, an overview of the significance and function of text processing in information retrieval comes first, then comes text pre-processing techniques such as stop-word deletion, stemming, and tokenization. In addition, various NLP methods, including sentiment, component identification, and named entity recognition, are being researched. This research then examines various text representation methods, including language models, TF-IDF, and bag-of-words. The act of analyzing disorganized text and turning it into valuable data for analysis to get a quantifiable figure that contains some essential information is known as "text analytics." Businesses are using text analysis more and more frequently. It aids in the analysis of unstructured data, such as customer reviews, as well as the discovery of patterns and the forecasting of trends. The technologies for text analysis that are offered for transforming text information into useful data for analysis include systems, libraries, analysis, automated process programmers, data collection, and extraction-based tools, to name just a few. The fundamentals of textual data, various text mining approaches, and the most widely used text analysis tools will all be covered in this research. We look at Naive Bayes, deep learning, and support machines for text classification. This research examines

K. Vayadande (✉) · H. Baru · A. Kashid · A. Kulkarni · P. Londhe · A. Vanjari
Information Technology, Vishwakarma Institute of Technology, Pune, India
e-mail: kuldeep.vayadande1@vit.edu

H. Baru
e-mail: harshal.baru20@vit.edu

A. Kashid
e-mail: abhishek.kashid20@vit.edu

A. Kulkarni
e-mail: aniket.kulkarni20@vit.edu

P. Londhe
e-mail: prathamesh.londhe20@vit.edu

A. Vanjari
e-mail: atharv.vanjari20@vit.edu

G. Mehta et al. (eds.), *Innovations in VLSI, Signal Processing and Computational Technologies*, Lecture Notes in Electrical Engineering 1095,
https://doi.org/10.1007/978-981-99-7077-3_56

the use of natural language processing (NLP) techniques in textual analysis for information retrieval. The research covers information retrieval systems, natural language processing (NLP) methods, captioning, phrase-based systems, text clustering, text similarity metrics, and text pre-processing. The description of numerous techniques and their use in information retrieval is the focus. The proposed system analyzes different parameters such as reading time, ease of reading, readability score, no of paragraphs, avg words per paragraph, total sentences in longest paragraph, avg words per sentence, longest sentence, words in longest sentence, frequency of "and" word, compulsive hedgers, intensifiers, vague words. All this processing will be done using NLP. The paper's conclusion discusses the value of textual analysis for information retrieval as well as its potential moving forward.

Keywords Natural language processing · Text pre-processing · Lexical analysis · Semantic analysis

1 Introduction

In today's rapidly developing information age, text summaries have developed into a crucial tool for accompanying and comprehending text material. The Internet, which has a wealth of information in numerous formats, surrounds the world today. Yet, a human being cannot read all this information. It is necessary to limit the enormous amount of data, which grows daily, so that people can simply understand it and find it helpful and productive (Gupta et al. 2016).

Text analysis for information retrieval using natural language processing (NLP) involves computational techniques to analyze huge amounts of unstructured text data, such as documents, web pages, or social media posts. Enhancing the precision and relevance of information retrieved from various sources is the aim of text analysis for information retrieval. Extractive and abstractive summarization are the two sorts of procedures that may be used to summarize textual material. Both approaches aim to condense the material to provide a clear knowledge of the subject without including any extraneous information. Recent years have seen a fast increase in data volume due to technological advancements.

It takes a long time possibly decades to find materials and put them to practical use in practical applications. To accelerate this process, we must utilize and exploit the knowledge about minerals that has been amassed through time through meticulous scientific method. Texts, academic publications, articles, guidebooks, sites, etc., serve as a huge data bank from which it is possible to mine with already knowledge. Yet, as most of the scientific information is semi- or unstructured and is presented as a combination of language, paragraphs with cross references, picture captions, and tables, it might be difficult to extract relevant information from these materials. Such information must be manually extracted, which requires a lot of time and resources and depends on the judgment of a subject-matter expert.

Understanding natural language is among the key tasks in computer–human interaction that are accomplished with the aid of NLP. Even though fundamental NLP operations like language grammar analysis, segment labeling, decoding, filtering, etc., are necessary to completely comprehend and produce natural language and are necessary before conducting the complicated NLP functions like addressed identification (NER), syntactic interpretation, compassion, interpretation software, and organic language production, among others. Because there is not enough transfer learning data available, NER is one of the most difficult tasks in NLP (Mahmood et al. 2017).

NLP techniques can be used to pre-process text data, such as removing stop words, stemming, or lemmatizing words, and normalizing the text to make it more easily searchable. Moreover, named things such as persons, locations, and organizations may be identified and extracted using NLP, and to label words with their part-of-speech tags. As a branch of computing, natural language analysis (NLA) offers an alternative method for automating extraction of information from text.

In addition to these pre-processing steps, NLP techniques can be used to perform more advanced text analysis, such as sentiment analysis, which involves identifying the emotional tone of the text, and topic modeling, which involves identifying the underlying topics or themes in the text.

The number of people using the Internet has drastically expanded because of the advancement of communication and information technologies. It shifts people's traditional information and news consumption patterns to digital ones, which results in comfort and efficiency for both information presenters and presenters (Mahmood et al. 2017).

Meaningful structured data in the form of groupings, links, entities, activities, and various more categories are retrieved from unstructured data using an information extraction (IE) approach. The data is mined for details in order to get it ready for examination. As a result, the data analysis is enhanced by the correct and efficient translation of large amounts of data all throughout the IE process. Word, voice, and film are just a few of the diverse sorts of data that have been handled using various techniques.

The volume of items in the contemporary technological age is textual, and this information is growing quickly as a result of web users' migration to media platforms, where they often exchange new textual material (Mahmood et al. 2017).

Academics who study literature and language have long been intrigued by computers' capacity to peruse and organize vast quantities of material. Prior to the invention of the computer, searching through an author's full body of work (for example) required turning the pages by hand, but as soon as the text was converted to a machine-searchable format, it could be rapidly browsed and analyzed. Literary academics are mainly interested in how a writer or writers use important terms or phrases, how frequently, when words became commonly used, etc. This is what we usually refer to as "quantitative analysis," since it emphasizes recurrence and location.

In fact, these practices date back decades to a time before computers. For instance, in the late thirteenth century, Dominican monks produced a concordance of the

Vulgate. This tool enabled them to look up familiar phrases and see where they appeared in the text.

Natural language processing methods are used in information retrieval to find and extract pertinent information from vast amounts of unstructured text data. Here are some key steps involved in the process:

Text pre-processing: This involves cleaning and normalizing the text data, such as removing stop words, punctuation, and special characters, and converting the text to lowercase.

Tokenization: This involves breaking down the text into indiviual words or tokens, which can be used as units of analysis.

Named entity recognition: This entails locating and extracting identified items from the text that may be utilized to enhance knowledge, such as individuals, groups, and places.

Part-of-speech tagging: Each token must be marked with the appropriate component of speech, such as a noun, verb, or adjective, which can be used to identify the structure and meaning of the text.

Semantic analysis: This involves using techniques such as topic modeling and sentiment analysis to identify the underlying topics or themes in the text and to determine the attitude of the writer or speaker toward the topic.

Information retrieval: This involves using various algorithms and techniques to match user queries to relevant text documents or passages, based on factors such as keyword frequency, semantic similarity, and relevance score.

Some common techniques used for information retrieval using NLP include the following:

Vector space models: These represent documents as vectors in a high-dimensional space and use measures such as cosine similarity to determine a document to a query.

Latent semantic indexing: This involves identifying latent semantic structures in the text data and using these to improve the accuracy of information retrieval.

Natural language understanding: This involves using advanced NLP techniques to better understand the meaning and intent behind user queries and to retrieve more relevant results.

This project is implemented in Python; it uses different Python libraries such as Click, colorclass, nltk, Pyphen. This system works with homer which is Python package that can assist in improving the readability, clarity, and utility of your writing. It offers details about the entire content as well as specific paragraphs. It provides information on readability, paragraph and sentence length, avg words per sentence, longest sentence, words in longest sentence, and other factors. Additionally, it looks for specific ambiguous terms. It keeps note of how frequently “and” words appear in the text.

2 Literature Survey

In the paper by Krallinger (2008), it explains the most typical medium for the official exchange of information in this contemporary civilization is text. Even if extracting relevant information from texts is not a simple operation, having a business intelligence tool that can do it quickly and inexpensively is essential in today's world. Text analytics is a new and intriguing track of study which tackles the problem and provides the intelligence tool. The methodologies, applications, and difficult problems in text mining have been provided in this work as an overview. Basic text mining techniques have been the focus of attention. The techniques include natural language processing, information extraction, and application domains have been briefly reviewed.

In the paper by Thompson (2009), it explains that, by automatically choosing instances of intriguing biological occurrences from big document collections, information extraction (IE), a text mining component, makes it easier to uncover new information. Understanding the semantic behavior and syntactic of these terms is vital since verbs and nominalized verbs are typically the focus of events. In the biomedical sector, the GREC is a singular resource since it offers not only the fundamental links between entities but also a wide range of additional crucial characteristics, such as their context, chronology, method, and environmental elements. It therefore strives to support the creation of information and technologies that are specifically related to biology.

In the paper by Wu (2010), it explains aiming to extract semantic relations from natural language text, information extraction (IE) systems are constrained by the availability of training data because the majority of these systems rely on supervised learning of relation-specific instances. Open IE solutions, such as Text Runner, aim to manage the web's infinite number of relationships. This study introduces WOE, a novel open IE method that combines a heuristic evaluation of Wikipedia info boxes and relevant material with self-supervised learning over unlexicalized features. The two operating modes for WOE are a pattern classifier trained on dependency path patterns and a CRF extractor trained on shallow features like POS tags.

In the paper by Gharehchopogh (2011), the paper explains that databases have been developed by businesses and other organizations as a result of the recent advancements in information systems and technology. Different sorts of data are obtained based on the objectives and organizational structure of the company. Structural data that was gathered using the unstructured model of the data utilizing text and web mining techniques was included in order to compare the structural models that were created. The findings imply that text and web mining approaches are more successful when using the obtained model. This discovery also highlights how the quality of information may be removed from an unstructured data model to accommodate unexpected outcomes. The usage of this data, which has the potential to be used in study, will surely enhance.

In the paper by Shaidah Jusoh (2012), this paper explains the most typical medium for the official exchange of information in this contemporary civilization is text.

Even if extracting relevant information from texts is not a simple operation, having a business intelligence tool that can do it quickly and inexpensively is essential in today's world. Text analysis is an all new and intriguing ground of study who tackle the problem and provide the intelligence tool. The methodologies, applications, and difficult problems in text mining have been provided in this work as an overview. Basic text mining techniques have been the focus of attention. The techniques include information recovery and natural language processing (NLP).

In the paper by Chiticariu (2013), this paper explains the majority of contemporary academic research. This field works under the presumption that statistical machine learning is the most effective strategy for handling information extraction issues. While rule-based IE rules the business sphere, academics often reject it as being outdated. To the practitioners in the industry, we argued the value of rule-based methods. Authors suggested approaches for resolving the divide by drawing inspiration from SQL's and the database community's achievements. In particular, author demand the standardization of an IE rule language and lay forth a bold research agenda for NLP academics who want to work on issues that are highly valuable and of broad interest in the field.

In the paper by Verma (2014), it explains the examination of data which is found in natural language text is known as text mining. In order to gather potentially useful business insights, a company may use text-based material, such as emails, word documents, and postings on social media channels. Several well-known scientific fields, including data analysis, case-based reasoning, statistics, information recovery, and machine learning (ML), employ their methodology in the research and applications of text mining, a burgeoning subject. This study investigates a technique for extracting structured databases using an information extraction system that has self-taught.

In the paper by Dr. S. Vijayarani. (Mohan 2015), paper explains finding or extracting meaningful information from text is a process called text mining. It is an interesting field of research since it seeks to understand unstructured texts. Data analysis and the knowledge recovery or discovery in textual databases are its official names (KDT). In modern applications like text comprehension, KDT is becoming more significant. Finding or extracting meaningful information from text is a process called text mining. It searches through big databases for intriguing patterns. Many pre-processing methods are used, including stemming and stop-word elimination. In-depth information on text mining pre-processing approaches is provided in this article.

In the paper by Gupta (2016), this paper says, in today's rapidly developing information era, text summary has developed into a key tool for assisting and comprehending textual material. Today's world is surrounded by an atmosphere of the Internet, which includes a wealth of information in many formats. However, a person cannot read all this data. A huge quantity of data, which keeps growing every day, needs to be condensed so that people can readily understand it and find it helpful and productive. Text summarization provides pertinent and exact information from the content to cut down on reading time. Implementing abstractive summaries of single and many papers has been suggested in this research. Tokenization, POS-tagging, chunking, and parsing are the first pre-processing steps for the document.

In the paper by Ahsan Mahmood (2017), paper clarifies that text-based information makes up the vast bulk of data in the current information age, and this data is expanding incredibly, rapidly as more and more Internet users flock to social media, where they often share new text-based information. The main use of this research is to produce a data retrieval system using Sahih Bukhari's hadith data. Authors have used a number of techniques and procedures to accomplish this. In this attempt, they can use the Sanad and Matn parts of the Hadith, FST to extract entities from the data, and NER to categories the named entities into several categories based on the tagged data.

3 Comparison Table

See Table 1.

4 Methodology

An application that organizes, stores, retrieves, and evaluates information from document libraries, especially text information, is known as information retrieval (IR).

The method of data collection:

The system is developed for retrieving the information from the text such as sentences and paragraphs from a text file. Generally, the system is developed such that it can take a text file as input and take the file for analysis as a whole. As the file will have a number of paragraphs and number of sentences, the output will be generally for the whole file.

The illustration below demonstrates how a user in need of information must create a request in the format of a natural language search. The proposed scheme will then fetch the pertinent output and return the output. The output will take the form of papers containing the necessary information.

The step-by-step processes are as given below:

- Indexing the documents collection.
- Translating the query in the same way as represented in the document content.
- The description of each document and that of the query is compared.
- The results are listed in the order of relevancy.

Table 1 Table of comparison

Sr.No.	Authors	Year	Techniques	Remark/outcome
1	Martin Krallinger, Alfonso Valencia and Lynette Hirschman (Krallinger et al. 2008)	2008	Text mining	Future uses of text mining techniques based on labor-intensive manual literature searches and curation pipelines could be feasible, but only if the conclusions drawn from automatically generated text-based outputs are reliable and pertinent
2	John McNaught, Paul Thompson, Sophia Ananiadou, Syed A Iqbal, (Thompson et al. 2009)	2009	Annotated GREC technique	The authors have created a special method that concentrates on both verbs and nominalized verbs for annotating sentence-bound gene regulatory events
3	Fei Wu, Daniel S. Weld (Fei and Weld 2010)	2010	Natural language processing	Authors conducted their studies using three corpora: the WSJ from Penn Treebank, Wikipedia, and the whole Web They chose 300 phrases at random from each dataset
4	Gharehchopogh Turkey and Khalifelu (2011)	2011	Natural language processing	The effectiveness of text and web mining techniques employing the resulting model is demonstrated. This finding also demonstrates that the unexpected outcomes may be included into an unstructured data model to reduce the quality of the information
5	Shaidah Jusoh and Hejab M. Alfawareh (Jusoh and Alfawareh 2012)	2012	Text mining	The fundamental methods of text mining have been the emphasis. Some of the approaches include information extraction and natural language processing. An immediate evaluation of application domains has been conducted
6	Laura Chiticariu, Yunyao Li, Frederick R. Reiss (Chiticariu and Yunyao Reiss 2013)	2013	Big data analytics	They have persuaded business professionals of the value of rule-based methods. They have recommended strategies for resolving the gap. they demand that an IE rule language be standardized
7	Tanu Verma, Renu, Deepti Gaur (Verma et al. 2014)	2014	Text mining	They have provided a method for extracting structured databases using an autonomously learnt information extraction system

(continued)

Table 1 (continued)

Sr.No.	Authors	Year	Techniques	Remark/outcome
8	Ms. Nithya, Ms. J. Ilamathi (Mohan 2015)	2015	TF/IDF algorithms	From a big amount of data, relevant information may be extracted via data mining. Many research challenges are handled and solved using data mining approaches
9	Honey Gupta, Sheetal Chaudhari, Aveena Kottwani (2016)	2016	Text summarization	Topical analysis, phrase frequency, and POS labeling are all included in the text summary. All of these are utilized to produce a thorough synopsis of the papers' content
10	Ahsan Mahmood, Wahab Khan, Zahoor-ur-Rehman (2017)	2017	Named entity recognition (NER)	Named entity recognition has been utilized by them (NER). Catboats, speech recognition are a few examples of NER's applications. They suggest a framework for knowledge extraction to extract named entities

4.1 Data Analysis

The system will analyze different parameters such as number of words in a paragraph, most repeated sentence, word, and the minimum grade of the reader who can conveniently read the sentence. The processing will be done using NLP. To explain how the algorithm works, let's take an example. For text summarizing, NLP takes. Most common words repeated in the paragraph create a sentence that is grammatically correct with these words. The same is used for text analysis and image retrieval as well.

The system's article/essay stats may be used to analyze research papers, journals, stories, and other types of content. The program provides the minutes spent reading, the mean trying to read ease, and the Dale Chall readability ratings. Total sentences, paragraphs, and words. The average number of words and phrases used in each paragraph is also shown. Also, the quantity and list of obsessive intensifiers, hedgers, and ambiguous terms are displayed.

Statistics about the sentences and words in the paragraph, as well as the average amount of words each sentence, the paragraph's longest sentence, and readability ratings (Flesch reading ease and Dale Chall readability scores). Also, it provides warnings when there are more than five sentences each paragraph and more than 25 words per phrase.

5 Proposed System

See Figs. 1 and 2.

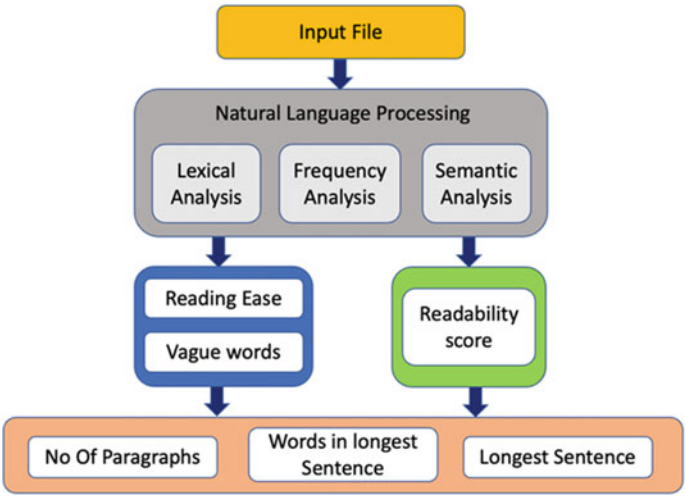


Fig. 1 Working of NLP

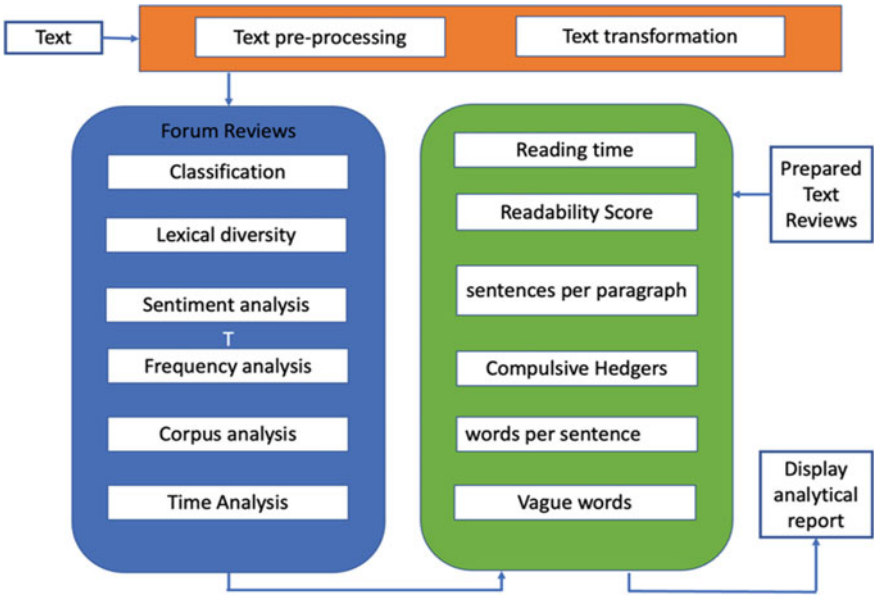


Fig. 2 Workflow of system

Table 2 Calculation of evaluation parameters (I)

Performance evaluation on dataset	Text file with different paragraphs for calculating performance parameters				
	P1 (Paragraph 1)	P2 (Paragraph 2)	P3 (Paragraph 3)	P4 (Paragraph 4)	P5 (Paragraph 5)
Mean square error (MSE)	0.0400	0.0100	0.0169	0.0225	0.0144
Root mean square deviation (RMSE)	0.20	0.10	0.130	0.150	0.120
Mean absolute error (MAE)	0.2000	0.1000	0.1300	0.1500	0.1200

Table 3. Calculation of evaluation parameters (II)

Performance evaluation on dataset	Text File With different paragraphs for calculating performance parameters				
	P1 (Paragraph 1)	P2 (Paragraph 2)	P3 (Paragraph 3)	P4 (Paragraph 4)	P5 (Paragraph 5)
Percentage deviation (PD)	2.1739	1.1494	4.6428	1.2145	0.5424
Average percentage deviation (APD)	1.9434	1.9434	1.9434	1.9434	1.9434
Standard deviation (SD)	0.1414	0.0707	0.0919	0.1060	0.0849

6 Table of Analysis

See Tables 2 and 3.

7 Result and Discussion

The technology was put to the test in real time, and the results seem promising. We had encouraging results ten times with various inputs. The lengthiest phrases, intensity, intonations, vague words, compulsive words, average duration, reading ease, legibility rating, number of passages, overall number of sentences per passage, overall number of sentences in the longest passage, average number of phrases, and average number of phrases are all precisely calculated (Figs. 3, 4, and 5).

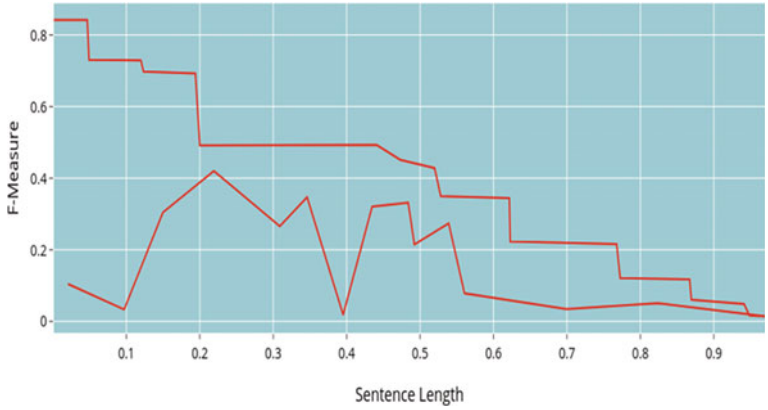


Fig. 3 Graph of F-measure and sentence length

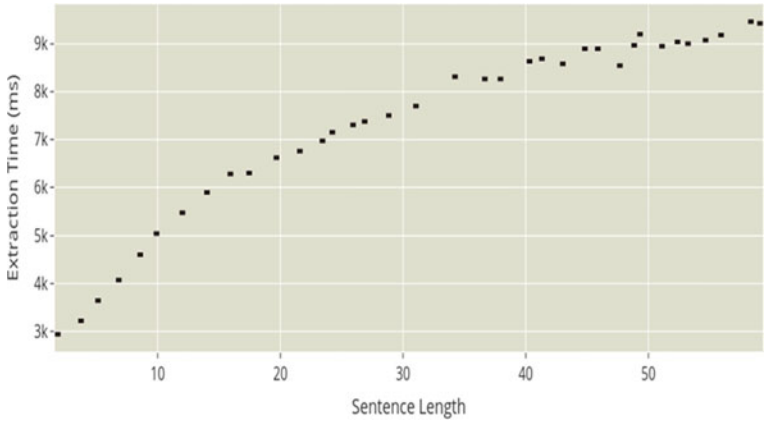


Fig. 4 Graph of extraction time and sentence length

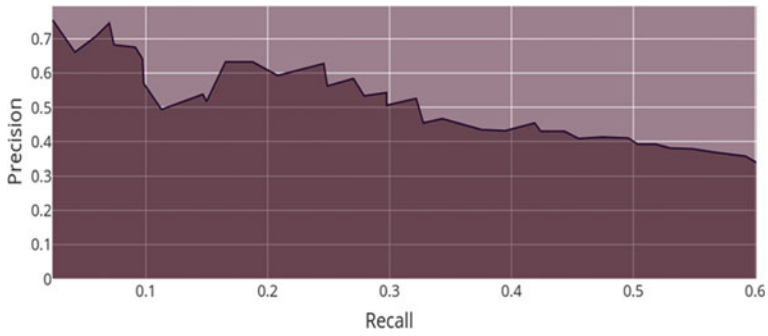


Fig. 5 Graph of precision and recall

8 Scope of Research

With several areas of concentration and research issues, the field of text processing for feature extraction using natural language processing (NLP) is fairly large and diverse. Listed below are some prospective topics of study for content analysis for NLP-based information retrieval:

Document classification: Following text pre-processing, it's normal practice to group documents according to their subject matter or content. The objective of this study might be to create new theories or methods for more precise document categorization.

Finding the subjects or patterns that are inherent in a group of documents is known as "topic modeling." The creation of innovative algorithms or frameworks for more precise and effective subject modeling may be the main goal of this research.

Text summarization: Condensing lengthy passages of text into more manageable summaries is a crucial text analysis activity. The goal of this research might be to create new methods for text summarization that are more precise and effective.

Multilingual NLP: Even as the volume of international text data rises, there is a growing need for NLP methods that can handle several languages. The goal of this research might be to create new multilingual NLP algorithms or models that are more accurate and effective.

According to the individual study issue and application, there are several potential areas of concentration within the large field of textual analysis for information extraction using NLP.

9 Conclusion

In conclusion, natural language processing (NLP) is a crucial tool for processing and flow of thoughts from unstructured text data. Text analysis for information retrieval is a subset of this technique. Text analysis helps us to spot trends, extract insights, and obtain data that would be difficult to access. Techniques used include tokenization, named entity identification, component labeling, sentiment classification, and topic modeling. Information retrieval systems based on NLP are often employed in a wide range of applications, such as search engines, recommendation systems, and chatbots. These systems are powered by algorithms that match user queries to relevant text documents or passages based on factors such as keyword frequency, semantic similarity, and relevance score. However, developing effective text analysis for information retrieval using NLP requires domain expertise and knowledge of the specific problem domain. The methodology for text analysis is iterative and requires ongoing evaluation and refinement to improve the accuracy and relevance of the results. Overall, text analysis for information retrieval using NLP is a powerful tool that has the potential to transform the way we process and extract meaning from

unstructured text data. As NLP techniques continue to evolve, we can expect to see even more innovation.

10 Future Scope

The field of text analysis for information retrieval using natural language processing (NLP) is constantly evolving, and there are several exciting areas of future scope for this technology. Here are some potential areas for future development:

Improved accuracy and relevance: NLP models for text analysis are continuously improving in accuracy and relevance, with the development of new techniques such as deep learning and transfer learning. We can expect to see even more accurate and relevant results from text analysis in the future.

Multilingual text analysis: As the world becomes more connected, there is an increasing need for text analysis to be able to handle multiple languages. NLP models are already being developed that can handle multiple languages, and this trend is expected to continue.

Context-aware text analysis: Text analysis that takes into account the context in which the text was written or spoken is an area of growing interest. By analyzing the context, NLP models can produce more accurate and nuanced results.

Integration with other technologies: NLP for text analysis can be integrated with other technologies, such as voice recognition and image analysis, to create more sophisticated systems that can analyze multiple types of data. Ethical and responsible text analysis: As text analysis becomes more powerful and pervasive, there is a growing need for ethical and responsible use of the technology. This includes ensuring the privacy of individuals and avoiding biases and discrimination in the results produced by NLP models.

Overall, the future of text analysis for information retrieval using NLP is very promising, with the potential to revolutionize the way we extract meaning from unstructured text data. With ongoing research and development, we can expect to see even more powerful and sophisticated NLP models that can analyze text data in increasingly complex and nuanced ways.

References

Chiticariu L, Li Y, Reiss FR (2013) Rule-based information extraction is dead! Long live rule-based information extraction systems! EMNLP 2013—2013 conference on empirical methods in natural language processing, proceedings of the conference, pp 827–832

Gharehchopogh FS, Khalifelu ZA (2011) Analysis and evaluation of unstructured data: text mining versus natural language processing. In: 2011 5th international conference on application of information and communication technologies (AICT). Baku, Azerbaijan, pp 1–4. https://doi.org/10.1109/ICAICT.2011.6111017

Gupta H, Kottwani A, Gogia S, Chaudhari S (2016) Text analysis and information retrieval of text data. In: 2016 International conference on wireless communications, signal processing and networking (WiSPNET). Chennai, India, pp 788–792. https://doi.org/10.1109/WiSPNET.2016.7566241

Jusoh S, Alfawareh H (2012) Techniq TechnTechn techniques, applications and challenging issue in text mining. IJCSI Int J Comput Sci Issues 9:431–436

Krallinger M, Valencia A, Hirschman L (2008) Linking genes to literature: text mining, information extraction, and retrieval applications for biology. Genome Biol 9(Suppl 2):S8. https://doi.org/10.1186/gb-2008-9-s2-s8

Mahmood A, Khan HU, Rehman ZU, Khan W (2017) Query based information retrieval and knowledge extraction using Hadith datasets. In: 2017 13th international conference on emerging technologies (ICET). Islamabad, Pakistan, pp 1–6. https://doi.org/10.1109/ICET.2017.8281714

Mohan V (2015) Preprocessing techniques for text mining—an overview

Thompson P, Iqbal SA, McNaught J et al (2009) Construction of an annotated corpus to support biomedical information extraction. BMC Bioinf 10:349. https://doi.org/10.1186/1471-2105-10-349

Verma T, Renu R, Gaur D (2014) Tokenization and filtering process in rapidminer. Int J Appl Inf Syst 7:16–18. https://doi.org/10.5120/ijais14-451139

Wu F, Weld DS (2010) Open information extraction using wikipedia. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Uppsala, Sweden. Association for Computational Linguistics, pp 118–127