

Advances in Science, Technology & Innovation
IEREK Interdisciplinary Series for Sustainable Development

D. Jude Hemanth *Editor*

Machine Learning Techniques for Smart City Applications: Trends and Solutions

Advances in Science, Technology & Innovation

IEREK Interdisciplinary Series for Sustainable Development

Editorial Board

Anna Laura Pisello, Department of Engineering, University of Perugia, Italy

Dean Hawkes, University of Cambridge, Cambridge, UK

Hocine Bougdah, University for the Creative Arts, Farnham, UK

Federica Rosso, Sapienza University of Rome, Rome, Italy

Hassan Abdalla, University of East London, London, UK

Sofia-Natalia Boemi, Aristotle University of Thessaloniki, Greece

Nabil Mohareb, Faculty of Architecture - Design and Built Environment,
Beirut Arab University, Beirut, Lebanon

Saleh Mesbah Elkaffas, Arab Academy for Science, Technology and Maritime Transport,
Cairo, Egypt

Emmanuel Bozonnet, University of La Rochelle, La Rochelle, France

Gloria Pignatta, University of Perugia, Italy

Yasser Mahgoub, Qatar University, Qatar

Luciano De Bonis, University of Molise, Italy

Stella Kostopoulou, Regional and Tourism Development, University of Thessaloniki,
Thessaloniki, Greece

Biswajeet Pradhan, Faculty of Engineering and IT, University of Technology Sydney,
Sydney, Australia

Md. Abdul Mannan, Universiti Malaysia Sarawak, Malaysia

Chaham Alalouch, Sultan Qaboos University, Muscat, Oman

Iman O. Gawad, Helwan University, Egypt

Anand Nayyar , Graduate School, Duy Tan University, Da Nang, Vietnam

Series Editor

Mourad Amer, International Experts for Research Enrichment and Knowledge Exchange
(IEREK), Cairo, Egypt

Advances in Science, Technology & Innovation (ASTI) is a series of peer-reviewed books based on important emerging research that redefines the current disciplinary boundaries in science, technology and innovation (STI) in order to develop integrated concepts for sustainable development. It not only discusses the progress made towards securing more resources, allocating smarter solutions, and rebalancing the relationship between nature and people, but also provides in-depth insights from comprehensive research that addresses the **17 sustainable development goals (SDGs)** as set out by the UN for 2030.

The series draws on the best research papers from various IEREK and other international conferences to promote the creation and development of viable solutions for a **sustainable future and a positive societal** transformation with the help of integrated and innovative science-based approaches. Including interdisciplinary contributions, it presents innovative approaches and highlights how they can best support both economic and sustainable development, through better use of data, more effective institutions, and global, local and individual action, for the welfare of all societies.

The series particularly features conceptual and empirical contributions from various interrelated fields of science, technology and innovation, with an emphasis on digital transformation, that focus on providing practical solutions to **ensure food, water and energy security to achieve the SDGs**. It also presents new case studies offering concrete examples of how to resolve sustainable urbanization and environmental issues in different regions of the world.

The series is intended for professionals in research and teaching, consultancies and industry, and government and international organizations. Published in collaboration with IEREK, the Springer ASTI series will acquaint readers with essential new studies in STI for sustainable development.

ASTI series has now been accepted for Scopus (September 2020). All content published in this series will start appearing on the Scopus site in early 2021.

D. Jude Hemanth
Editor

Machine Learning Techniques for Smart City Applications: Trends and Solutions

 Springer

Editor

D. Jude Hemanth
Karunya Institute of Technology and Sciences
Coimbatore, India

ISSN 2522-8714 ISSN 2522-8722 (electronic)
Advances in Science, Technology & Innovation
IEREK Interdisciplinary Series for Sustainable Development
ISBN 978-3-031-08858-2 ISBN 978-3-031-08859-9 (eBook)
<https://doi.org/10.1007/978-3-031-08859-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Cover illustration: Tostphoto/Shutterstock.com

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

With the significant advancement in technologies, it has been always a desire of the human race to focus towards a comfortable lifestyle. Apart from the lifestyle, technologies play a major role in conserving the natural resources which aid in a better environment for the human beings to live in this world. With this background, the concept of smart cities has gained significant attention in today's scenario. Though the "smart city" concept is popular, it is still a distant dream in the context of under-developed and developing countries. Though many advanced technologies are being explored in the context of smart cities, there are still scope for improvement in the different sub-concepts of smart cities which can help the rural regions also. In other words, it can be mentioned that advanced technologies are not fully explored in the context of smart cities. Another aspect is the necessity for the innovative/novel technologies which can make the different dimensions of smart cities practically feasible. Machine learning is one of the prime advanced technologies which guarantee high performance for any practical set-up/system. The combination of machine learning and smart cities can be one of the solutions for making this dream a reality. This book specially focuses on the application of different machine learning techniques on the sub-concepts of smart cities such as smart energy, smart transportation, smart waste management, smart health and smart infrastructure. The objective of this book is to bring out innovative solutions in the above-said issues to alleviate the pressing difficulties of human society. A brief introduction about each chapter is as follows.

Chapter 1 is aligned closely with smart education sector in which a deep learning model is used to predict the behaviour of young adults in the context of their participation in public life. This chapter also emphasizes the need for the proper involvement of young adults in the development of any cities. Chapter 2 deals with the combination of smart infrastructure and deep learning. A deep learning model is used in this work to analyse the electrical theft in smart grid systems. The security issues of smart grid systems are dealt in detail which are an inherent part of smart cities. The focal point of Chap. 3 is smart health in which a machine learning-based assistive system is developed for the disabled community. This chapter highlights the necessity for maintaining the quality of living of all the people in a smart city. The availability of different machine learning-based methods for sign language recognition is described in Chap. 4. The contents of this chapter are focussed towards smart health for a hearing-impaired person.

Chapter 5 covers the smart surveillance sector in combination with machine learning techniques. A smart city is expected to be under surveillance to avoid any unnecessary events such as the accidents. Machine learning assists in this context by analysing the behaviour of the drivers which is one of the prime reasons for the unpleasant events. Smart drones-based city surveillance is the prime objective of Chap. 6. Several machine learning algorithms are developed to classify and identify the different types of vehicles in a live traffic system. Chapter 7 deals with an artificial neural network-based smart tracking system. The inclusion of the concepts of antenna into machine learning adds significant weightage to the practical feasibility of the proposed system in a smart city scenario. Ensemble machine learning

classifiers are used to detect the pedestrians in Chap. 8 with the help of infrared images. This method is particularly useful for a night vision system which falls under the category of smart transportation systems.

Chapter 9 deals with the broad spectrum of smart health and machine learning techniques. Machine learning techniques are used to diagnose and manage the sleep pattern of the elderly people. It may be noted that a significant portion of the whole population in a smart city comprises the elderly people. Smart traffic management is the emphasis of Chap. 10. In this work, machine learning techniques are used to predict the time-dependent traffic pattern which can avoid the congestions and other bottlenecks in a traffic system. Chapter 11 talks about the emergency department allocation in hospitals during pandemic situations. Machine learning models are used to efficiently allot the emergency rooms to the patients who need the most. This proposed model will avoid any chaotic situations in the hospital environment. Smart energy is the main concept of Chap. 12. In this work, the application of machine learning models for generation and distribution of wind energy is dealt in detail. Energy sources are an integral part of a successful smart city set-up.

Chapter 13 deals with the smart waste management system using machine learning techniques. Efficient categorization and management of trash in coastal areas is the main topic of this work. Deep learning architectures are used for this experimental work. Smart classroom in a university set-up is discussed in Chap. 14. The integration of machine learning with the education sector is the need of the hour in today's online education system. Smart health monitoring protocols in educational institutions among students are well described in Chap. 15. This type of machine learning-based system will be useful especially during the pandemic situation. Automation of industrial applications such as path navigation is the prime focus of Chap. 16. Autonomous robots along with the machine learning techniques are well described in this chapter. This type of models is often deployed in autonomous robots-based industries.

I am thankful to the authors and reviewers for their contributions towards this book. My special thanks go to Dr. Mourad Amer and Dr. Nabil Khelifi (Editors) for the opportunity to organize this edited volume. Finally, I would like to thank Ms. Toka Mourad Frihy who coordinated the entire proceedings. This edited book covers the fundamental concepts and application areas in detail which is one of the main advantages of this book. Being an interdisciplinary book, I hope it will be useful to a wide variety of readers and will provide useful information to professors, researchers and students.

Coimbatore, India
January 2022

Dr. D. Jude Hemanth

Contents

Applying Deep Learning to Predict Civic Purpose Development: Within the Smart City Context	1
Hyemin Han	
Convolution Neural Network Scheme for Detection of Electricity Theft in Smart Grids	13
Matthew Palmer, G. Jasper Willsie Kathrine, and S. Jebapriya	
Helping Hand: A GMM-Based Real-Time Assistive Device for Disabled Using Hand Gestures	23
S. Gnanapriya and K. Rahimunnisa	
A Review on Hand Gesture and Sign Language Techniques for Hearing Impaired Person	35
Safyzan Salim, Muhammad Mahadi Abdul Jamil, Radzi Ambar, and Mohd Helmy Abd Wahab	
DriveSense: Adaptive System for Driving Behaviour Analysis and Ranking	45
Sankar Behera, Bhavya Bhardwaj, Aurea Rose, Mohammad Hamdaan, and M. Ganesan	
Classification and Tracking of Vehicles Using Videos Captured by Unmanned Aerial Vehicles	59
Jorge E. Espinosa, Jairo Espinosa, and Sergio A. Velastin	
Tracking Everyone and Everything in Smart Cities with an ANN Driven Smart Antenna	75
Herman Kunsei, Paul R. P. Hoole, K. Pirapaharan, and S. R. H. Hoole	
Wavelet-Based Saliency and Ensemble Classifier for Pedestrian Detection in Infrared Images	95
R. Newlin Shebiah and S. Arivazhagan	
A Survey of Emerging Applications of Machine Learning in the Diagnosis and Management of Sleep Hygiene and Health in the Elderly Population	109
B. Banu Rekha and A. Kandaswamy	
Smart City Traffic Patterns Prediction Using Machine Learning	123
David Opeoluwa Oyewola, Emmanuel Gbenga Dada, and Muhammed Besiru Jibrin	
Emergency Department Management Using Regression Models	135
S. Kezia, A. Hepzibah Christinal, D. Abraham Chandy, and M. James Graham Steward	
Machine Learning in Wind Energy: Generation to Supply	143
Bhavya Bhardwaj and M. Ganesan	

Multi-class Segmentation of Trash in Coastal Areas Using Encoder-Decoder Architecture	155
S. Surya Prakash, V. Vengadesh, M. Vignesh, and Satheesh Kumar Gopal	
Learning Analytics for Smart Classroom System in a University Campus	171
Tasneem Hossenally, U. Kawsar Subratty, and Soulakshmee D. Nagowah	
Predictive Analytics for Smart Health Monitoring System in a University Campus	187
Zakia N. S. H. Mohung, B. Unayza Boodoo, and Soulakshmee D. Nagowah	
SysML-Based Design of Autonomous Multi-robot Cyber-Physical System Using Smart IoT Modules: A Case Study	203
Qasem Abu Al-Haija	
Vulnerabilities and Ethical Issues in Machine Learning for Smart City Applications	221
K. Martin Sagayam, Roopa Jeyasingh, J. Jenkin Winston, and Tony Jose	



Applying Deep Learning to Predict Civic Purpose Development: Within the Smart City Context

Hyemin Han

Abstract

We applied the deep learning method, which has been developed in the fields of computer and data science for accurate prediction, to predict political purpose development during emerging adulthood. We tested whether deep learning more accurately predicted Wave 2 political purpose with Wave 1 predictors compared with traditional regression. A convolutional neural network consisting of two dense and dropout layers was trained to predict the outcome variable. For comparison, we also estimated a multinomial logistic regression model. The result demonstrated that deep learning outperformed traditional regression in general while effectively minimizing overfitting. Moreover, from exploratory analysis, we found that deep learning might be able to model the non-linear relationship between the predictors and outcome variables. Based on the findings, we discussed the implications of the present study within the context of improving citizens' lives in smart cities.

Keywords

Deep learning • Political purpose • Multinomial logistic regression

1 Introduction

In the present study, we applied the deep learning method, a relatively novel method for prediction and pattern classification in computer science, for predicting the development of purpose to engage in civic activity, political activity in particular, among young adults in emerging adulthood. We

intended to present the prediction process and result as a simple illustrative example demonstrating the potential values and benefits of using computational methods in educational and psychological research. Participating in political activities during adolescence and early adulthood is essential in promoting continued political engagement throughout one's life (Youniss et al., 2002). Thus, it is necessary to empirically examine which factors significantly influence political purpose, a sense of purpose to continuously engage in political activities (Malin et al., 2015), during this period. Several previous studies have examined the aforementioned factors (Malin et al., 2017) reported that demographical and education-related factors were significantly associated with changes in civic engagement and purpose during emerging adulthood. (Han et al., 2021) showed that presence of strong moral and political identity positively predicted development of political purpose.

Although the previous studies examined the factors that facilitate the development of political purpose during emerging adulthood, (e.g., Crocetti et al., 2014; Han et al., 2021; Malin et al., 2017) several points related to model prediction shall be reconsidered and improved. Recent methodological advances in data science, such as the wide employment of the deep learning method, have demonstrated that novel methods can more accurately predict outcomes compared with traditional regression-based methods in educational and psychological research (Han et al., 2020). This advantage becomes more significant when the prediction is involved in diverse and complex predictors (Awad & Khanna, 2015). Also, previous studies that used traditional regression methods did not take into account the possibility of overfitting (Han & Dawson, 2021; Kim et al., 2016; McNeish, 2015). This occurs when we attempt to predict outcome variables of interest with limited amount of data. Although a trained model may well explain and predict outcomes within the boundary of the used data, the model might be “overfitted” to the data and not be able to perform in reality outside of the boundary (Han & Dawson, 2021;

H. Han (✉)
Educational Psychology Program, University of Alabama,
Box 872301 Tuscaloosa, AL 35487, USA
e-mail: hyemin.han@ua.edu

McNeish, 2015). Overfitting might significantly limit the generalizability of findings (Han & Dawson, 2021; Yarkoni & Westfall, 2017). Hence, it would be worth examining whether use of the novel method in predicting change in political purpose, which is associated with diverse factors, improved our ability to predict the change.

To improve prediction accuracy and better model the relationship between predictors and change in political purpose, we applied the deep learning method in the present study. First, we compared performance in terms of both prediction accuracy and robustness against overfitting between deep learning and traditional multinomial methods in predicting political purpose with the Civic Purpose Project (CPP) dataset (Malin et al., 2015). For the comparison between the methods, we used the variables that were initially analyzed in a previous study on political purposes (Han et al., 2021). Second, to explore the relationship between predictor and outcome variables, we generated simulated data, predicted the outcome variable, and plotted the relationship. At the same time, we intend to introduce the novel prediction method in data science to the field for improving analysis methodology by sharing all program codes to readers via the Open Science Framework (OSF; <https://osf.io/mn4eh/>) (Brandt et al., 2014).

Related to the improvement of smart cities, which is the main focus of the current edited volume, examining the developmental trajectories of political purpose with an advanced computational method, the deep learning method, will provide scholars, educators, and policymakers with useful insights (Han et al., 2019, 2020). We are primarily interested in how to improve citizens' quality of life by implementing smart city solutions based on computational methods, including but not limited to, smart transportation, smart surveillance, smart health solutions, smart infrastructure, and smart energy. Although the improvement of materialistic aspects of a smart city is crucial in promoting citizens' quality of life (Fischer & Amekudzi, 2011; Ülengin et al., 2001), it is undeniable that such citizens' active engagement in civic activities is fundamental to making the city a better place (Portney, 2005). If this is the case, then accurately understanding how such civic engagement is developed over time and which factors are significantly associated with its development becomes an important task to complete (Brown, 1992; Yeager et al., 2015). Such understanding would provide us with more concrete evidence-based ideas about how to design our education as well as policies addressing civic engagement among citizens. In this process, the application of advanced computational methods, the deep learning method, in particular, is deemed to increase the accuracy of modeling and prediction (e.g., (Han et al., 2020)). So, it would be worthwhile to carefully consider how to apply the methods in the context of research on political purpose development. The prediction

of purpose development with the deep learning method in the present study will serve as an illustrative example proposing the potential implications and benefits of using computational methods in educational and developmental psychology research.

2 Methods

2.1 Dataset

We analyzed the dataset initially collected for the CPP, which was conducted by members at the Stanford Center for Adolescence. All data files and documentation for the present study were downloaded from the Institute for Social Research at the University of Michigan repository (<https://doi.org/10.3886/ICPSR36561.v1>). Further details about the samples and measures are available in the PDF files in the repository. The original data was collected from three Californian high schools. Wave 1 was conducted when participants were 12th graders. Wave 2 was conducted two years thereafter.

In the present study, we analyzed data collected from 476 participants (60.3% female, 39.1% male, 0.6% unknown; Wave 1 Age mean = 16.83 years, SD = 0.52 years; 40.10% Latinx, 34.20% Asians, 9.00% multiethnicity, 6.10% Caucasians, 5.30% other, 4.60% African Americans, 0.6% unknown) who completed the surveys for both waves.

2.2 Analyzed Variables

We used variables related to political purpose, political and civic identity, and demographics surveyed in the CPP. The questionnaire form and further details about the scoring procedures are available in the CPP page and (Han et al., 2021).

Political Purpose Status. The main variable of the present study, political purpose status, was determined by three surveyed variables, i.e., civic intent, participation in political activity, and beyond-the-self motivation. These three variables were collected with civic purpose questionnaires that were administrated during both Wave 1 and Wave 2 (Malin et al., 2017). The full questionnaire used for the current study is downloadable via the Institute for Social Research at the University of Michigan repository (<https://pcms.icpsr.umich.edu/pcms/performDownload/1c7d9501-45ad-4482-8c01-793b31e21174>). Further details about how each variable score was calculated are available in Han et al. (2021) for readers' information.

First, civic intent was measured by asking whether motivations associated with civic activities were significantly meaningful to participants. For the civic intent

assessment, responses to the five items in “20. Life Goals (Purp_goals),” Purp_goal_1, Purp_goal_2, Purp_goal_5, Purp_goal_7, Purp_goal_9, in the full questionnaire were averaged. Second, political engagement was assessed with the questionnaires by examining to what extent each participant actively participated in political activities. Political engagement was calculated by averaging responses to the six items in “31. Civic Action (Yllacts),” Yllacts_6, Yllacts_10, Yllacts_14, Yllacts_18, Yllacts_20, Yllacts_22. Third, presence of beyond-the-self motivation was assessed by examining whether a participant regarded motivation to contribute to beyond-the-self beings, instead of oneself, as one of the most important motives to engage in civic activities. The motive items were presented in “33. Motivation for Political Involvement (Polmotives).” Participants were asked to choose three most important motives that let them engage in civic activity. If they selected one of six beyond-the-self motive items (i.e., Polmotives_1, Polmotives_4, Polmotives_7, Polmotives_8, Polmotives_10, Polmotives_12) as at least one of the three most important motives, then they were deemed to possess beyond-the-self motivation (Yes/1). If they selected all three self-oriented motives, then they were deemed to be self-oriented (No/0).

Wave 1 and Wave 2 political purpose status was classified into one of these five categories: drifting, dreamer, dabbler, self-oriented, and purposeful. We classified each participant using their responses to the civic purpose questionnaires, which measured civic intent, political engagement, and presence of beyond-the-self motivation. We employed a theoretical classification method developed in previous mixed-method studies (see Table 1 for the details) (Malin et al., 2015, 2017).

Political and moral identity. Political identity references to what extent political concerns are central to one’s self-identity (Porter, 2013). In the same way, moral identity is defined as the centrality of morals and morality to identity (Aquino et al., 2009). These variables were measured at Wave 1 with items included in the CPP questionnaire. Participants were asked to what extent each value was important to their identity. The list of presented values, as well as the questions, are available via <https://pcms.icpsr.umich.edu/pcms/performDownload/1c7d9501-45ad-4482-8c01-793b31e21174>. For assessment, the value items presented in

“41. Civic Identity (civID)” were used. Among the presented values, three value items (i.e., civID_3, civID_8, civID_11) were used to assess political identity, and six (i.e., civID_4, civID_5, civID_9, civID_10, civID_13, civID_14) were used to assess moral identity.

Participants’ perceived centrality of each value item was anchored to a four-point Likert scale, “not at all central to my identity”—“very central to my identity.” Political and moral identity scores were calculated with the quantified centrality of the aforementioned presented values (see Fig. 1 for additional visualized information about how the centrality was measured in a quantitative manner). The political identity score was acquired by averaging participants’ responses to the three political identity items. In the same way, the moral identity score was calculated in terms of the mean of the six items related to moral identity.

Demographics. Demographic information was surveyed with a questionnaire in the CPP. The analyzed demographic variables included gender (male/female), ethnicity (Asian or Asian American, including Chinese, Indian, Korean, and others/Black or African American/Hispanic or Latino, including Mexican American, Colombian, Nicaraguan, and others/White, Caucasian, European American; not Hispanic/American Indian or Native American/ Mixed; Parents are from different groups/Other), born in the US (yes/no), mother and father born in the US (yes/no/unknown), college boundedness (yes/no), and socioeconomic status (SES; 1 (lowest) to 9 (highest)). These variables were treated as categorical variables.

2.3 Deep Learning

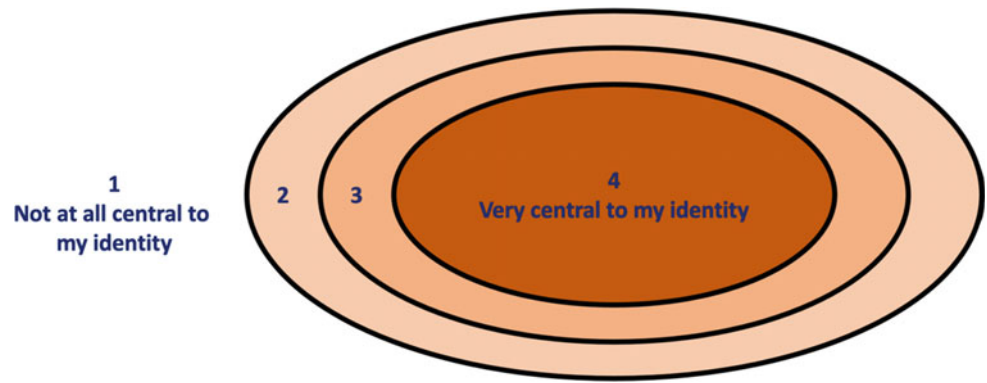
As predictors for model training, we used Wave 1 political purpose status, political and moral identity, and demographics. The dependent variable used in the process was Wave 2 political purpose status. Categorical variables, i.e., Wave 1 political purpose status, gender, ethnicity, born in the US, mother, and father born in the US, college boundedness, were converted into dummy variables.

Deep learning is one of the widely used methodologies in machine learning for pattern recognition and prediction in the field of computer science (Han et al., 2020). Because

Table 1 Political purpose classification

Category	Civic intent (≥ 4 in 5-point Likert scale?)	Political engagement (top quartile?)	Beyond-the-self motivation (present?)
Drifting	No	No	–
Dreamer	Yes	No	–
Dabbler	No	Yes	–
Self-oriented	Yes	Yes	No
Purposeful	Yes	Yes	Yes

Fig. 1 Visual representation of the political and moral identity measures



deep learning utilizes a neural network consisting of multiple layers for training, it is assumed to have a deep structure, so it is called “deep” learning. The neural network, which includes numerous simulated neurons, is trained with input values and then adjusted by comparing the predicted outcomes and actual outcomes for accuracy improvement (LeCun et al., 2015).

For deep learning, we employed the convolutional neural network (CNN) implemented in *tensorflow* and *keras* with Python (Chollet et al., 2015). For the outcome prediction, we trained the CNN consisting of one input layer, two levels of dense layers (+ dropout layers), and one output layer (see Fig. 2). The input was entered into the network in the shape of (23,) and then forwarded to the first dense layer with 512 neurons (512,). The output from the first dense layer was forwarded to the second dense layer with 1,024 neurons (1024,). Each neuron employed the Rectified Linear Units (ReLU) function, which is defined as $y = \max(x, 0)$, the activation function. For optimization, we employed Adam optimizer with the *keras*’s default learning rate, 0.01 (Kingma & Ba 2014). To prevent model overfitting (further technical and mathematical details are explained in the Appendix), first, we added two dropout layers (dropout rate = 0.50) at the ends of these two dense layers (Srivastava et al., 2014). Second, we also applied L1 (λ_1) and L2 (λ_2) regularization to minimize overfitting (Zou & Hastie, 2005). In the present study, we used $\lambda_1 = 0.0001$ and $\lambda_2 = 0.001$. Finally, the results from the second dense-dropout layers were forwarded to the output layer that calculated the probability of each of the five Wave 2 political purpose statuses. We produced the final output by examining which status showed the highest probability. These parameters were employed in a previous study that applied the deep learning method in the prediction of outcomes of educational interventions, which is relevant to the context of the current simulation (Han et al., 2020). We made minor adjustments to the parameters so that they were appropriate to examine the analyzed data.

At the beginning of each epoch, the randomly selected training dataset was entered into the CNN. Once the

predicted outcome was produced, we calculated the loss function to evaluate model performance. The loss function was calculated in the format of categorical cross-entropy, which is related to how the predicted probabilities and actual outcomes are different from each other (Siriak et al., 2019). When the loss function calculated with the validation dataset continuously increased for ten epochs, the learning process was “early” stopped to prevent overfitting. Finally, the calculated loss function was used to adjust the weights in the neurons in the dense layers through backpropagation to improve model performance in the next epoch (Chollet et al., 2015). The training procedure continues for up to 1000 epochs if no early stopping occurred.

We employed fivefold validation while training the models to examine whether overfitting occurred (Browne, 2000). Before starting model training, the whole dataset ($N = 476$) was shuffled and randomly separated into five parts, P1 to P5. Only 4/5 of the whole dataset, P1 to P4, was used for model training while P5 was reserved for validation. Then, the validation accuracy of the trained model was tested with the validation dataset, P5.

Similarly, the traditional multinomial logistic regression model was estimated only with the training dataset and then its validation accuracy was evaluated with the reserved validation dataset. Traditional multinomial logistic regression was performed to compare the performance of deep learning versus traditional analysis methods in terms of accuracy within the context of modeling and predicting political purpose development during emerging adulthood (Kwak & Clayton-Matthews, 2002). We used an R package, *nnet*, to perform multinomial logistic regression analysis (Ripley & Venables).

2.4 Evaluation of Model Performance

We compared performance between deep learning and traditional multinomial logistic regression. We examined two aspects of the performance: the prediction accuracy

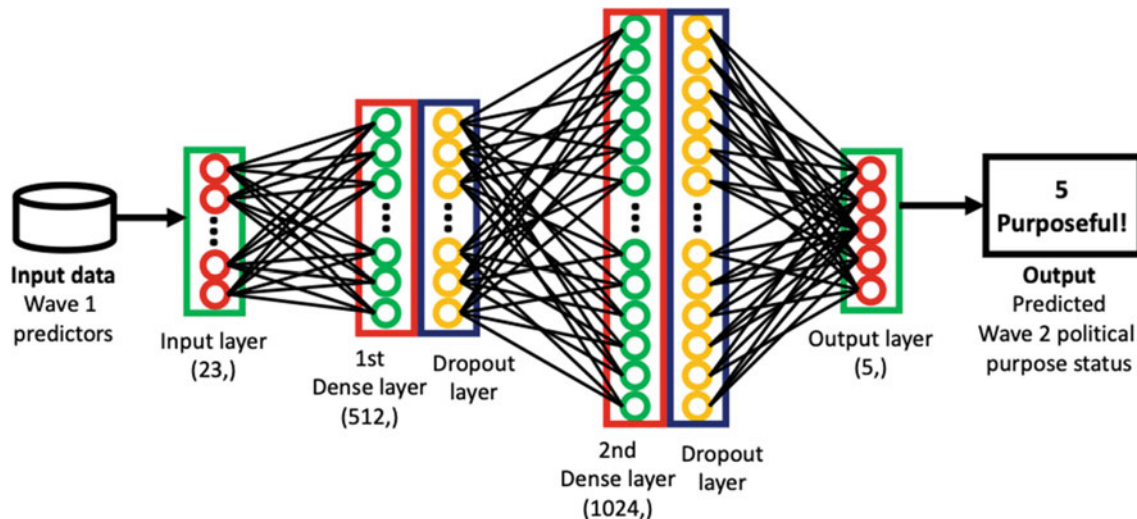


Fig. 2 Overview of the convolutional neural network used in the present study

calculated with cross-validation and area under the curve (AUC). First, we investigated prediction accuracy with cross-validation to examine whether the model was able to accurately predict Wave 2 political purpose within as well as without the boundary of data used for training (Browne, 2000; Han et al., 2020). For this purpose, we employed fivefold validation. The same process was repeated four more times while using the different parts of the data for validation. We calculated the prediction accuracy with both training and validation datasets. Prediction accuracy was defined as the ratio of the participants whose Wave 2 political purpose status was accurately predicted to the number of the whole participants in data. This fivefold cross-validation was conducted for both deep learning and traditional multinomial logistic regression. We iterated this process 1,000 times to collect prediction accuracy data while randomly shuffling the dataset.

Second, to evaluate sensitivity and selectivity of each method, we plotted a receiver operating characteristic (ROC) curve and calculated the responsive AUC during 1,000 iterations (Maroco et al., 2011). A ROC curve illustrates whether a binary classifier can sensitively differentiate two classes while preventing false positives. The x-axis of the curve represents the false positive rate ($1 - \text{selectivity} = \text{False Positive} / \text{False Positive} + \text{True Negative}$), and the y-axis denotes the true positive rate ($\text{sensitivity} = \text{True Positive} / \text{True Positive} + \text{False Negative}$). Then, AUC can be obtained by calculating the area below an ROC curve in a provided plot. $\text{AUC} > 0.50$ means that the classifier performs better than random. In an ideal case, $\text{AUC} = 1.00$. Because we performed multinomial logistic regression to predict Wave 2 political purpose, which has five different

categories, we examined five different ROC curves for both deep learning and traditional multinomial logistic regression. We treated each of the aforementioned five pair comparisons similar to binary classification. The ROC curves were created with the whole dataset and AUCs were calculated accordingly.

For statistical testing, we employed both frequentist and Bayesian two-sample *t*-tests. Bayesian testing was employed to conduct additional evidence-based inference (Han et al., 2018; Wagenmakers et al., 2018a). Bayesian statisticians have proposed that results from Bayesian analyses are more suitable for testing whether evidence supports hypotheses of interest in a direct manner, which cannot be feasibly done with the frequentist approach employing *p*-values (Wagenmakers et al., 2018b). We compared the training and validation dataset prediction accuracy and AUC between deep learning and traditional multinomial logistic regression. More specifically, we conducted the comparisons with the calculated mean of prediction accuracy (M) and standard deviation of the accuracy (SD) across 1000 trials. M and SD were calculated as follows:

$$M = \frac{\sum X_i}{N} \quad (1)$$

$$\text{SD} = \sqrt{\frac{\sum (X_i - M)^2}{N}} \quad (2)$$

where N was the total number of trials, X_i was the resultant prediction accuracy from each trial (i).

While interpreting the analysis results, we assumed that those reporting $p < 0.05$ and Bayes Factor ≥ 3.00 indicated

significant differences (Han, 2020). For additional information regarding the effect sizes of the tested differences, we also calculated the effect sizes in terms of Cohen's D .

$$t = \frac{M_1 - M_0}{\sqrt{\frac{(N_1-1)SD_1^2 + (N_0-1)SD_0^2}{N_1 + N_0 - 2} \sqrt{\frac{1}{N_1} + \frac{1}{N_0}}}} \quad (3)$$

$$95\%CI = (M_1 - M_0) \pm t \sqrt{\frac{(N_1 - 1)SD_1^2 + (N_0 - 1)SD_0^2}{N_1 + N_0 - 2} \sqrt{\frac{1}{N_1} + \frac{1}{N_0}}} \quad (4)$$

$$D = \frac{M_1 - M_0}{\sqrt{\frac{SD_1^2 + SD_0^2}{2}}} \quad (5)$$

The tested statistics were calculated as shown above. T -statistics for comparisons were calculated by (3), where M_1 , SD_1 , and N_1 were the mean accuracy, standard deviation of accuracy, number of trials for deep learning; M_0 , SD_0 , and N_0 were the mean accuracy, standard deviation of accuracy, number of trials for the traditional regression. The 95% confidence intervals for the difference between the mean accuracy of deep learning versus the traditional regression (95% CI) were calculated with (4). Cohen's D s indicating the effect sizes of differences were calculated following (5). Because Bayes Factors were obtained through iterative computational processes, not by analytic formulae, we used an R package, *BayesFactor* (Morey et al., 2021), for their calculation.

2.5 Examining the Relationship Between Predictors and Predicted Outcomes

To examine the relationship between predictors and the outcome variable, we generated a large simulation dataset, predicted the outcome variable with the trained model, and created plots demonstrating the relationship. We particularly focused on three predictors, Wave 1 political and moral identity, and SES. Since these variables are continuous, examining their relationship with the outcome variable can clearly show how deep learning-based prediction works.

First, we generated 100,000 data entries based on the distributions of the predictors in the real dataset. We investigated the overall distribution of each predictor and then created the dataset that followed the distributions with a random number generator. Second, we predicted Wave 2 political purpose with the generated data entries; in this process, we used the deep learning model that showed the highest mean prediction accuracy. For comparison, we predicted the same outcome variable with multinomial logistic

regression as well. Third, we plotted the relationship between the continuous Wave 1 predictor, i.e., political and moral identity as well as SES, and the Wave 2 predicted outcome variable in 100% stacked area charts. In these charts, the x-axis represented the value of the predictor of interest and the y-axis represented the relative proportion of each of the five Wave 2 political purpose statuses (summed to 100%) at the location of the x-value. We visually investigated the differences between the charts generated with the deep learning and multinomial logistic regression models.

3 Experimental Results and Discussions

Our deep learning procedures were executed on a MacBook Pro 16" equipped with 2.5 GHz Intel Core i7 (four cores) and 16 GB 1600 MHz DDR3 RAM. Deep learning ($M = 70.29\%$, $SD = 3.23\%$) showed significantly better training dataset accuracy compared with traditional multinomial logistic regression ($M = 62.03\%$, $SD = 1.54\%$), $t(7175.5) = 163.13$, $p < 0.001$, 95% CI [8.15 8.35], Cohen's $D = 3.26$; Bayesian t-test also strongly supported the non-zero effect, Bayes Factor = $1.29e + 2815$. The validation dataset prediction accuracy was also significantly better in deep learning ($M = 53.49\%$, $SD = 4.93\%$) compared with traditional regression ($M = 53.22\%$, $SD = 4.92\%$), but the effect was very small, $t(9998) = 2.70$, $p < 0.01$, 95% CI [0.07 0.46], Cohen's $D = 0.05$; however, Bayesian t-test reported that the effect was merely anecdotal, Bayes Factor = 0.86. Furthermore, AUCs were significantly better in deep learning compared with traditional multinomial logistic regression for all five statuses (see Fig. 3 and Table 2).

For additional information, Table 3 demonstrates the average true positive rate for each purpose status for each prediction method when the average false positive rate was set to 0.10 (10%). As shown, in all statuses, the average true positive rate was higher when deep learning was performed compared with when traditional multinomial logistic regression was performed. In this process, the false positive rate and true positive rate were defined as follows:

$$\text{False Positive Rate (FP)} = \frac{\text{The number of false positive cases}}{\text{The number of predicted negative cases}} \quad (6)$$

$$\text{True Positive Rate(TP)} = \frac{\text{The number of true positive cases}}{\text{The number of predicted positive cases}} \quad (7)$$

Together with the ROC curves and AUC comparisons, the results suggest that deep learning is capable of producing better sensitivity and selectivity compared with traditional regression in the present study.

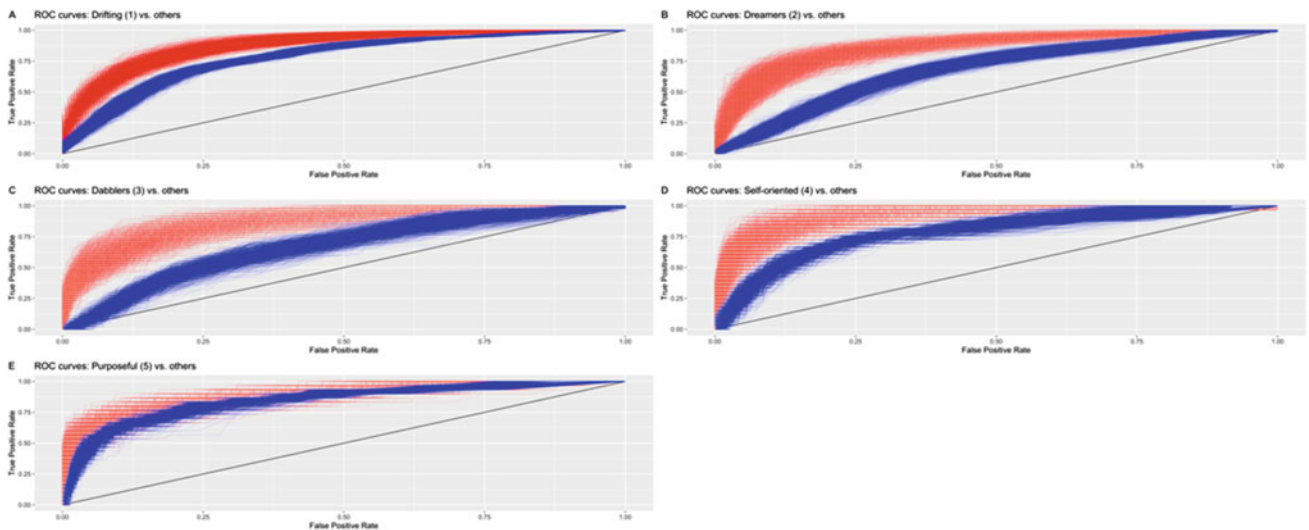


Fig. 3 ROC curves in five different purpose statuses. Red: deep learning. Blue: traditional multinomial logistic regression

Table 2 Results from AUC comparisons

Status	Deep learning		Traditional multinomial logistic regression		<i>t</i>	95% CI	Cohen's <i>D</i>	Bayes Factor
	<i>M</i>	SD	<i>M</i>	SD				
1: Drifting	0.88	0.02	0.78	0.00	197.25***	[0.10 0.10]	8.82	1.08e + 1307
2: Dreamer	0.86	0.02	0.68	0.01	269.43***	[0.18 0.18]	12.05	3.38e + 1567
3: Dabbler	0.86	0.03	0.67	0.01	199.90***	[0.19 0.19]	8.94	1.17e + 1318
4: Self-oriented	0.93	0.02	0.79	0.01	162.96***	[0.14 0.14]	7.29	1.44e + 1151
5: Purposeful	0.88	0.02	0.84	0.01	56.83***	[0.04 0.04]	2.54	1.12e + 415

Table 3 Average true positive rates when the threshold was set to adjust the false positive rate = 0.10 in the ROC curve

Status	Deep learning		Traditional multinomial logistic regression	
	<i>M</i>	SD	<i>M</i>	SD
1: Drifting	0.63	0.05	0.38	0.04
2: Dreamer	0.64	0.07	0.20	0.04
3: Dabbler	0.63	0.08	0.21	0.05
4: Self-oriented	0.79	0.07	0.45	0.05
5: Purposeful	0.69	0.06	0.61	0.04

With this illustrative example, we examined the potential values of the deep learning method in educational and developmental psychology research, as well as that addressing the improvement of smart cities in terms of citizens' quality of life through promotion of civic and political engagement. According to the results, deep learning showed significantly improved prediction performance compared with traditional regression. Although there was a concern regarding overfitting, which has been a significant issue in

the field, the validation dataset prediction accuracy of deep learning was at least slightly better than traditional regression. One caveat is that although the resultant *p*-value was < 0.01, the estimated effect size and Bayes Factor did not suggest presence of a strong effect. However, at least one point is obvious: deep learning did cause overfitting worse than traditional regression while significantly improving prediction accuracy with the training dataset. In addition, given the significantly higher AUCs in all five categories,

deep learning showed significantly better sensitivity and selectivity compared with traditional regression (Maroco et al., 2011). The results presented in Table 3, the average true positive rates, also support this point. This suggests that deep learning is more robust against both false positives and false negatives (Parikh et al., 2008). Hence, deep learning can improve overall prediction performance while effectively preventing overfitting.

Furthermore, Fig. 4 demonstrates the generated 100% stacked area charts. Particularly in the cases of Wave 2 political purpose predicted by political identity and SES, the charts generated with the deep learning model suggested possible non-linearity in the relationship between the predictors and predicted outcome while the multinomial logistic regression model showed a linear relationship.

From the 100% stacked area charts generated with the simulated dataset (see Fig. 4), we found that deep learning can model a potential non-linear relationship between the predictors and outcome variables. Although we could not quantitatively test the non-linearity, two predictors, political identity and SES, showed apparent non-linear relationships with Wave 2 political purpose. Two statuses, dreamer and dabbler, showed non-linear increases when political identity ≥ 2 . In addition, the proportion of the Wave 2 purposeful status was maintained from low to middle SES in the deep learning prediction, while the proportion showed monotonic decrease for traditional regression. We also found a rapid increase in the proportion of dreamers when SES 4 in the deep learning prediction, which was not found from traditional regression.

These results were somehow consistent with what has been reported regarding the relationships between one's background variables and purpose development from

previous qualitative studies. For instance, in the case of SES, it did not show a monotonical positive relationship with purpose as it would be found from linear regression. Instead, qualitative researchers have reported that purpose development can be significantly prominent among urban low-SES populations, which were deemed to be less purposeful previously (Ballard et al., 2015; Bronk et al., 2020). Hence, the deep learning method that enables us to model non-linearity in the relationship between variables might allow us to examine complex associations between background variables and variables of interest in human development that could not be well discovered with conventional linear regression.

4 Concluding Remarks

In the present study, we examined an illustrative example of the application of the deep learning method in predicting political purpose development during emerging adulthood. When predicting and simulating outcome variables is the main research goal, the deep learning method would be worth employing given its improved accuracy. For instance, if a researcher intends to predict developmental outcomes of populations with limited amounts of available longitudinal data, deep learning can produce more accurate prediction outcomes compared with traditional regression while minimizing overfitting with several techniques, such as cross-validation, regularization, and use of dropout layers (Browne, 2000; Srivastava et al., 2014). Furthermore, given that deep learning allows use of multiple candidate predictors, it is a better method compared with previous simulation methods tested in predicting intervention outcomes among

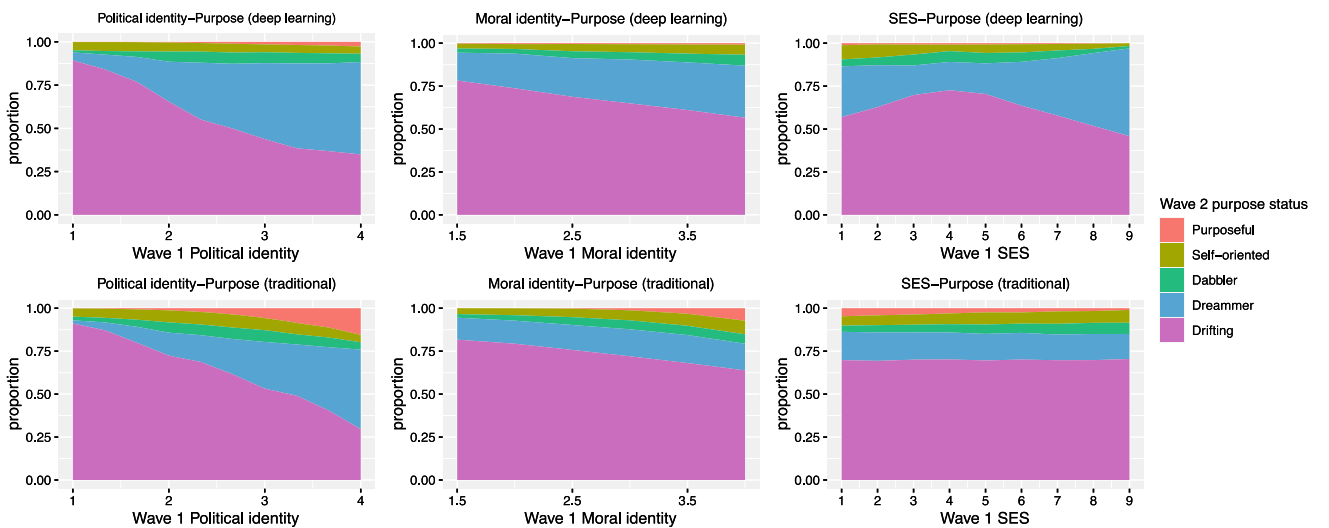


Fig. 4 Proportions of Wave 2 political purpose statuses predicted by Wave 1 political and moral identity, and SES. Top: deep learning. Bottom: traditional multinomial logistic regression

students such as Markov Chain and Evolutionary Causal Matrices, which only take into account a limited number of predictors (Han et al., 2016; Han et al., 2018). Deep learning will be able to provide researchers and even educators and policymakers who are interested in predicting future outcomes with the aforementioned methodological benefits (Han et al., 2019).

Furthermore, by more accurately modeling data, the deep learning method will be able to contribute to future follow-up research projects. Although further interpretations are necessary, the aforementioned findings from the simulated data analysis are interesting since the non-linear pattern could not be discovered in previous studies that employed traditional regression methods. The identified pattern from deep learning may provide researchers with useful insights about how to understand the overall nature of data and interesting research questions worth pursuing in future studies (Baker, 2010). The practical implications of the identified pattern can be considered with the researchers' domain knowledge and heuristics in their fields (Han, 2019).

As mentioned in the introduction, the findings from the present study provide useful insights related to our efforts to improve smart cities with computational methods, which constitutes the basis of the current edited volume. Given citizens' quality of life in smart cities could not be improved solely through developing physical and materialistic aspects of the cities, considering how to promote citizens' civic and political engagement is fundamental in research addressing the aforementioned quality of life. In fact, researchers have argued that civic and political engagement is strongly required for sustainable development of cities as well as citizens' quality of life (Portney, 2005). Hence, by providing a way to model and predict developmental outcomes with improved accuracy, the deep learning method could become a useful tool in the related fields, including but not limited to educational and developmental psychology. Researchers will be able to better identify associations between personal and environmental factors and developmental outcomes with models created by the deep learning method. Also, because such associations are likely to be complex and non-linear in some instances, the deep learning method will be a viable method to take into account the aforementioned complexity and non-linearity in modeling and prediction processes. By better examining factors significantly predicting civic and political development using deep learning, researchers, educators, and policymakers will be able to learn how to model education and policies in their smart cities to influence the identified factors in an effective way in order to promote civic and political development.

Although several potential benefits of the deep learning method in research on adolescence and emerging adulthood were presented, several limitations warrant further research.

First, whether deep learning also performs well when a non-categorical variable (e.g., continuous variable) is a dependent variable was not tested. Given that the method has been optimized for binary or categorical classification, whether it also performs well with a continuous dependent variable has not been thoroughly examined. Second, the interpretation and use of the prediction model might not be fully suitable for ordinary research projects in the field. In many cases, researchers are interested in inference, such as examining whether a specific predictor of interest shows a significant non-zero effect on a dependent variable. Although the deep learning method is ideal for generating accurate prediction models, it cannot properly address the aforementioned point since it does not usually provide indicators for statistical analysis (e.g., *p*-values, Bayes Factors, etc.) used for inference. Third, we only compared deep learning versus traditional regression with one set of parameters and settings. We did not compare different types of deep learning methods in the present study because we were primarily interested in demonstrating how the deep learning method, in general, can be applied in a human development study and how the method performed differently compared with traditional regression. Future studies shall explore the best options and settings when using deep learning for a better methodological comparative analysis.

Appendix: Supplementary Methods

We employed dropout layers and L1 and L2 regularization of each neuron to address the issue of overfitting. First, by using the dropout layer, it is possible to randomly deactivate a certain proportion of neurons, i.e., 0.50, so that the trained model becomes less complex and less likely to be overfitted (Srivastava et al., 2014). Second, we also used L1 and L2 regularization in each neuron. The regularization is based on elastic-net regression, which penalizes unnecessarily large coefficients (Zou & Hastie, 2005). In the case of deep learning, the regularization behaves in a way that penalizes unnecessarily large neuron weights. In predicting a certain outcome variable y ,

$$L(\lambda, \beta) = \sum_{i=1}^N (\hat{y}_i - y_i)^2 + \lambda_2 |\beta|^2 + \lambda_1 |\beta|_1$$

where \hat{y} is the predicted outcome value, $|\beta|_1 = \sum_{j=1}^n |\beta_j|$, and $|\beta|^2 = \sum_{j=1}^n \beta_j^2$ (Zou & Hastie, 2005). The regularization process tends to minimize $L(\lambda, \beta)$, so large weights, β s, are penalized. Thus, the model becomes simpler, and overfitting can be minimized.

To evaluate the model performance and determine when to do early stopping, a categorical cross-entropy was calculated at the end of each epoch. The categorical cross-entropy, which was used to evaluate the current error at the end of each epoch, is defined as follows:

$$\text{Loss} = - \sum y_i \cdot \log \hat{y}_i$$

where y is the vector of actual outcomes (e.g., when Wave 2 political purpose status is purposeful, $y = [00001]$), and \hat{y} is the vector of predicted probabilities.

References

- Aquino, K., Freeman, D., Reed, A., Lim, V. K. G., & Felps, W. (2009). Testing a social-cognitive model of moral behavior: The interactive influence of situations and moral identity centrality. *Journal of Personality and Social Psychology, 97*, 123–141.
- Awad, M., & Khanna, R. (2015). *Efficient learning machines*. Apress. <https://doi.org/10.1007/978-1-4302-5990-9>
- Baker, R. S. J. (2010). Data mining for education. In Baker, E. (Ed.), *International encyclopedia of education* (3rd ed.). Elsevier.
- Ballard, P. J., Malin, H., Porter, T. J., Colby, A., & Damon, W. (2015). Motivations for civic participation among diverse youth: More similarities than differences. *Research in Human Development, 12*, 63–83. <https://doi.org/10.1080/15427609.2015.1010348>
- Brandt, M. J., IJzerman, H., Dijksterhuis, A., Farach, F. J., Geller, J., Giner-Sorolla, R., Grange, J. A., Perugini, M., Spies, J. R., & van 't Veer, A. (2014). The replication recipe: What makes for a convincing replication? *Journal of Experimental Social Psychology, 50*, 217–224. <https://doi.org/10.1016/j.jesp.2013.10.005>
- Bronk, K. C., Mitchell, C., Hite, B., Mehoke, S., & Cheung, R. (2020). Purpose among youth from low-income backgrounds: A mixed methods investigation. *Child Development, 91*. <https://doi.org/10.1111/cdev.13434>
- Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The Journal of the Learning Sciences, 2*, 141–178. https://doi.org/10.1207/s15327809jls0202_2
- Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology, 44*, 108–132. <https://doi.org/10.1006/jmps.1999.1279>
- Chollet, F., et al. (2015). Keras. <https://github.com/fchollet/keras>
- Crocetti, E., Erentaitė, R., & Žukauskienė, R. (2014). Identity styles, positive youth development, and civic engagement in adolescence. *Journal of Youth and Adolescence, 43*, 1818–1828. <https://doi.org/10.1007/s10964-014-0100-4>
- Fischer, J. M., & Amekudzi, A. (2011). Quality of life, sustainable civil infrastructure, and sustainable development: Strategically expanding choice. *Journal of Urban Planning and Development, 137*, 39–48. [https://doi.org/10.1061/\(ASCE\)UP.1943-5444.0000039](https://doi.org/10.1061/(ASCE)UP.1943-5444.0000039)
- Han, H., Soyly, F., & Anchan, D. M. (2019). Connecting levels of analysis in educational neuroscience: A review of multi-level structure of educational neuroscience with concrete examples. *Trends Neuroscience Education, 100113*. <https://doi.org/10.1016/j.tine.2019.100113>
- Han, H. (2019). Neuroscience of morality and teacher education. In Peters, M.A. (ed.), *Encyclopedia of teacher education*. Springer. https://doi.org/10.1007/978-981-13-1179-6_190-1
- Han, H. (2020). Implementation of Bayesian multiple comparison correction in the second-level analysis of fMRI data: With pilot analyses of simulation and real fMRI datasets based on voxelwise inference. *Cognitive Neuroscience, 11*, 157–169. <https://doi.org/10.1080/17588928.2019.1700222>
- Han, H., Ballard, P. J., & Choi, Y.-J. (2021). Links between moral identity and political purpose during emerging adulthood. *Journal of Moral Education, 50*, 166–184. <https://doi.org/10.1080/03057240.2019.1647152>
- Han, H., & Dawson, K. J. (2021). Improved model exploration for the relationship between moral foundations and moral judgment development using Bayesian Model Averaging. *Journal of Moral Education. https://doi.org/10.1080/03057240.2020.1863774*
- Han, H., Lee, K., & Soyly, F. (2016). Predicting long-term outcomes of educational interventions using the Evolutionary Causal Matrices and Markov Chain based on educational neuroscience. *Trends in Neuroscience and Education, 5*, 157–165. <https://doi.org/10.1016/j.tine.2016.11.003>
- Han, H., Lee, K., & Soyly, F. (2018b). Simulating outcomes of interventions using a multipurpose simulation program based on the evolutionary causal matrices and Markov chain. *Knowledge and Information Systems. https://doi.org/10.1007/s10115-017-1151-0*
- Han, H., Lee, K., & Soyly, F. (2020). Applying the deep learning method for simulating outcomes of educational interventions. *SN Comput. Sci., 1*, 70. <https://doi.org/10.1007/s42979-020-0075-z>
- Han, H., Park, J., & Thoma, S. J. (2018a). Why do we need to employ Bayesian statistics and how can we employ it in studies of moral education?: With practical guidelines to use JASP for educators and researchers. *Journal of Moral Education, 47*, 519–537. <https://doi.org/10.1080/03057240.2018.1463204>
- Kim, M. -H., Banerjee, S., Park, S. M., & Pathak, J. (2016). *Improving risk prediction for depression via Elastic Net regression—Results from Korea National Health Insurance Services Data*. AMIA Annual Symposium Proceedings. AMIA Symposium, pp. 1860–1869
- Kingma, D., & Ba, J. (2014) *Adam: A method for stochastic optimization*. <http://arxiv.org/abs/1412.6980>
- Kwak, C., & Clayton-Matthews, A. (2002). Multinomial logistic regression. *Nursing Research, 51*, 404–410. <https://doi.org/10.1097/00006199-200211000-00009>
- LeCun, Y., Bengio, Y., Hinton, G., Goodfellow, I. J., & Courville, A. (2015). Deep learning. *Nature, 521*, 436–444. <https://doi.org/10.1038/nature14539>
- Malin, H., Ballard, P. J., & Damon, W. (2015). Civic purpose: An integrated construct for understanding civic development in adolescence. *Human Development, 58*, 103–130. <https://doi.org/10.1159/000381655>
- Malin, H., Han, H., & Liauw, I. (2017). Civic purpose in late adolescence: Factors that prevent decline in civic engagement after high school. *Developmental Psychology, 53*, 1384–1397. <https://doi.org/10.1037/dev0000322>
- Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I., & de Mendonça, A. (2011). Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC Research Notes, 4*, 299. <https://doi.org/10.1186/1756-0500-4-299>
- McNeish, D. M. (2015). Using Lasso for predictor selection and to assuage overfitting: A method long overlooked in behavioral sciences. *Multivariate Behavioral Research, 50*, 471–484. <https://doi.org/10.1080/00273171.2015.1036965>
- Morey, R. D., Rouder, J. N., Jamil, T., Urbaneck, K., & Ly, A. (2021). Package ‘BayesFactor.’ <https://cran.r-project.org/web/packages/BayesFactor/BayesFactor.pdf>
- Parikh, R., Mathai, A., Parikh, S., Chandra Sekhar, G., & Thomas, R. (2008). Understanding and using sensitivity, specificity and

- predictive values. *Indian Journal of Ophthalmology*, 56, 45–50. <https://doi.org/10.4103/0301-4738.37595>
- Porter, T. J. (2013). Moral and political identity and civic involvement in adolescents. *Journal of Moral Education*, 42, 239–255. <https://doi.org/10.1080/03057240.2012.761133>
- Portney, K. (2005). Civic engagement and sustainable cities in the United States. *Public Administration Review*, 65, 579–591. <https://doi.org/10.1111/j.1540-6210.2005.00485.x>
- Ripley, B., & Venables, W. Package ‘nnet.’ <https://cran.r-project.org/web/packages/nnet/nnet.pdf>
- Siriak, R., Skarga-Bandurova, I., & Boltov, Y. (2019). *Deep convolutional network with long short-term memory layers for dynamic gesture recognition*. 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). pp. 158–162. IEEE. <https://doi.org/10.1109/IDAACS.2019.8924381>
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. <https://doi.org/10.1214/12-AOS1000>
- Ülengin, B., Ülengin, F., & Güvenç, Ü. (2001). A multidimensional approach to urban quality of life: The case of Istanbul. *European Journal of Operational Research*, 130, 361–374. [https://doi.org/10.1016/S0377-2217\(00\)00047-3](https://doi.org/10.1016/S0377-2217(00)00047-3)
- Wagenmakers, E. -J., Love, J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Selker, R., Gronau, Q. F., Dropmann, D., Boutin, B., Meerhoff, F., Knight, P., Raj, A., van Kesteren, E. -J., van Doorn, J., Šmíra, M., Epskamp, S., Etz, A., Matzke, D., de Jong, T., van den Bergh, D., Sarafoglou, A., Steingroever, H., Derks, K., Rouder, J.N., Morey, R. D. (2018a). Bayesian inference for psychology. Part II: Example applications with JASP. *Psychonomic Bulletin & Review*, 25, 58–76. <https://doi.org/10.3758/s13423-017-1323-7>
- Wagenmakers, E. -J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Love, J., Selker, R., Gronau, Q. F., Šmíra, M., Epskamp, S., Matzke, D., Rouder, J. N., Morey, R. D. (2018b). Bayesian inference for psychology. Part I: Theoretical advantages and practical ramifications. *Psychonomic Bulletin & Review*, 25, 35–57. <https://doi.org/10.3758/s13423-017-1343-3>
- Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 53, 174569161769339. <https://doi.org/10.1177/1745691617693393>
- Yeager, D. S., Fong, C. J., Lee, H. Y., & Espelage, D. L. (2015). Declines in efficacy of anti-bullying programs among older adolescents: Theory and a three-level meta-analysis. *Journal of Applied Developmental Psychology*, 37, 36–51. <https://doi.org/10.1016/j.appdev.2014.11.005>
- Youniss, J., Bales, S., Christmas-Best, V., Diversi, M., McLaughlin, M., & Silbereisen, R. (2002). Youth civic engagement in the twenty-first century. *Journal of Research on Adolescence*, 12, 121–148. <https://doi.org/10.1111/1532-7795.00027>
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of Royal Statistical Society Series B Statistical Methodology*, 67, 301–320. <https://doi.org/10.1111/j.1467-9868.2005.00503.x>



Convolution Neural Network Scheme for Detection of Electricity Theft in Smart Grids

Matthew Palmer, G. Jasper Willsie Kathrine, and S. Jebapriya

Abstract

Smart Grid has revolutionized the field of electricity, giving overall stability and reliability to the electric grid. The smart grid has made significant changes to the conventional grid system to include RTU and intelligent AI-based framework for autonomous operation in real-time. The security issues in smart grid introduced threats to the secure operation of the grid. In this paper, an analysis of threats identified for electricity theft and secure communication of smart grid has been discussed. To address electricity theft in an electrical grid this paper introduces an improved methodology to implement electric theft detection (ETD) using deep learning CNN model. The implementation uses two different approaches of data processing to determine the suitable processing technique for the proposed model. During the fine-tuning of the proposed model, the model obtained highest precision rate as 0.948 and recall as 1 for ETD, and an overall accuracy of 92.16% was obtained. The performance analysis of the proposed model has been discussed and also been compared with the existing methodologies for ETD.

Keywords

Electricity theft detection • Smart grid • Cyber-physical system • Deep learning • CNN

M. Palmer · G. Jasper Willsie Kathrine (✉) · S. Jebapriya
Department of Computer Science Engineering, Karunya Institute of Technology and Sciences, Coimbatore, 641114, India
e-mail: kathrine@karunya.edu

M. Palmer
e-mail: matthewpalmer@karunya.edu

S. Jebapriya
e-mail: jebapriya@karunya.edu

1 Introduction

Electricity is crucial for every field of work and the field of electricity has developed significantly over the years. The traditional grid system used to transmit electricity in a conventional manner has been improving with extension of smart devices and sensors in the grid. The electricity grid had major developments over the years to revolutionize the grid technology for effective operation and eventually led to Smart Grid. Smart grid emphasis more on reliability, sustainability, and security of the real-time operation and control. Smart grid also insists active participation of customers for operation of the system, as the generation side requires consumption data to predict demand for the customer side to ensure efficient operation (Greer et al., 2019).

Smart Grid is reliable grid system allowing bidirectional flow of power by implementing control and automation over electric grid (Musleh et al., 2019). According to NIST smart grid has seven domains consisting Bulk Generation, Transmission, Distribution, Customer, Market, Service Provider, and Operation Domain as shown in Fig. 1. Smart Grid has more significance than the traditional grid due to the addition of key elements such as Supervisory Control and Data Acquisition (Scada), Advanced Metering Infrastructure (AMI), IoT-based Smart Grid, and Plug-in Hybrid Electric Vehicles (PHEV) (Yu & Xue, 2016).

Traditional electric grid has been utilized for more than few decades, and the engineering advancements have introduced HVDC, HVAC, FACTS on the transmission side; and SCADA along with RTU in both generation and transmission sides. The advancements have been more emphasized on the technical aspects of the grids, however, the birth of Industrial 4.0 in every engineering sector has also paved its way into the electricity grid system. The basic factors such as Voltage, Current, Frequency (incase AC system), and reactive power during transmission and distribution were properly analyzed and calculated during generation. Any of the factors being miscalculated would lead to

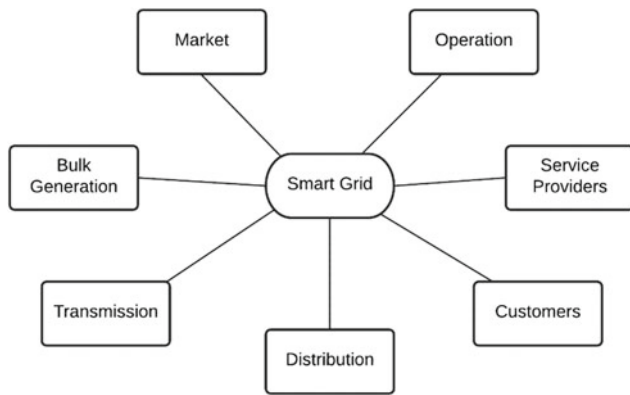


Fig. 1 NIST Conceptual model consisting seven modules constituting the operation of Smart Grid

brownouts or even blackouts in a region. The technological assistance from sensors connected in a network can provide real-time changes in a grid, so whenever there is a need to certain parameters to balance the operation of the grid, it can be done more effectively in smart grid. The smart grid real-time data collection also leads to new possibilities such as integration of big data and AI to the grid, which can help to monitor and predict the forecast of usage patterns of customers and to take critical automated decisions in a cringe (Rajesh et al., 2021).

Smart Grid revolutionized the traditional grid system to decentralized network and made technological developments to the grid. The addition of technology realized the digital control over the electric grid, and it added more assets, also resulted in the electrical grid more susceptible to cyber-attacks. Cyber-attacks vary from the type and attack coverage which could lead to small accidents or can lead to catastrophic incidents. The imminent danger from new assets poses a threat to the implementation of smart grid, so implementation of smart grid is being limited to partial. Threats in the SG can lead to Denial of Service (DoS), LDoS (Low-rate DOS), False Data Injection, Man in the middle (MiTM) attack, Spoofing, Sniffing, Time Synchronization, and other cyber-related attacks on the grid (Gunduz & Das, 2020).

AMI sends usage patterns and other power-related data to the control center or central monitoring system for demand prediction and supply allocation. AMI can also be used to detect electricity theft in the consumer side. Electricity Theft can be due to faulty meters (Jakaria et al., 2019) Improper maintenance of distribution equipment (Abdulahman et al., 2019), Rogue electrical connection (Yao et al., 2019), Dishonest meter reading, and Tampering energy meter (Singh et al., 2019).

Smart grid architecture is decentralized and blockchain also supports decentralized operation. Blockchain implementation in smart grid ensures security during transactions

and other data collection since the data for electrical generation are critical information. Data manipulation or obstructions attacks are deterred by adopting blockchain in smart grid (Zhuang et al., 2020). Blockchain is categorized as Private, public, and consortium. Private blockchain can be used for Local Energy Markets in smart grid for customer usage statistics and transactions (Mengelkamp, et al., 2018). Tiered blockchain architecture can be used for secure data exchange which allows decentralized direct communication between circuit relays and prevents single point of failure (Sikeridis, et al., 2020).

Thus, one of the prominent developments were seen is blockchain integration in smart grid, the other key improvements in development of smart grid would be Electricity Theft Detection (ETD) for smart grid. The relative developments for securing smart grid from electricity theft is discussed in the following section. The paper then mentions the steps that can be taken to prevent ETD in prosumer area and also explains the implementation of the model proposed in this paper. The paper finally analyzes the results from the model and a comparative analysis was shown to prove the significance of the proposed model.

2 Background and Related Work

Smart Grid Security focuses on confidentiality, integrity, availability, accountability of the grid system to the users (El Mrabet et al., 2018). In a simpler implementation of smart grid, the security focus can be primarily on proper authenticated usage of electricity by customers and the major concern in electricity revenue would be lost due to electricity theft. Electricity theft is an act of stealing electricity illegally which is a criminal offense, and the traditional grid system could not determine the possibility of electricity theft in the system. The continuous monitoring by a human at all times is not realistic, so using remote terminal units (RTU) and sensors connected in WAN could be a possible solution, and the placements of sensors can be done by adopting Enhanced Grid Sensor Placement algorithm (Patil & Sankpal, 2019). The data from RTUs can be monitored and collected in a control center for smart grid operation, and any abnormalities in usage can be detected using event-based Graph Signal Processing for Non-Intrusive Appliance Load Monitoring (NALM) (Zhao et al., 2016). Energy theft and defective meters can be detected by using Linear Regression Model as proposed by Yip Sook-Chin et al. (2017).

In 2019, ETD models using CNN-LSTM and CNN with pallier algorithm to protect the consumer privacy was also proposed (Hasan et al., 2019). To protect against the electricity theft attacks W. Li et al., proposed three-stage model by combining MLP, RNN, LSTM, with GRU, and the proposed model produced an overall accuracy of

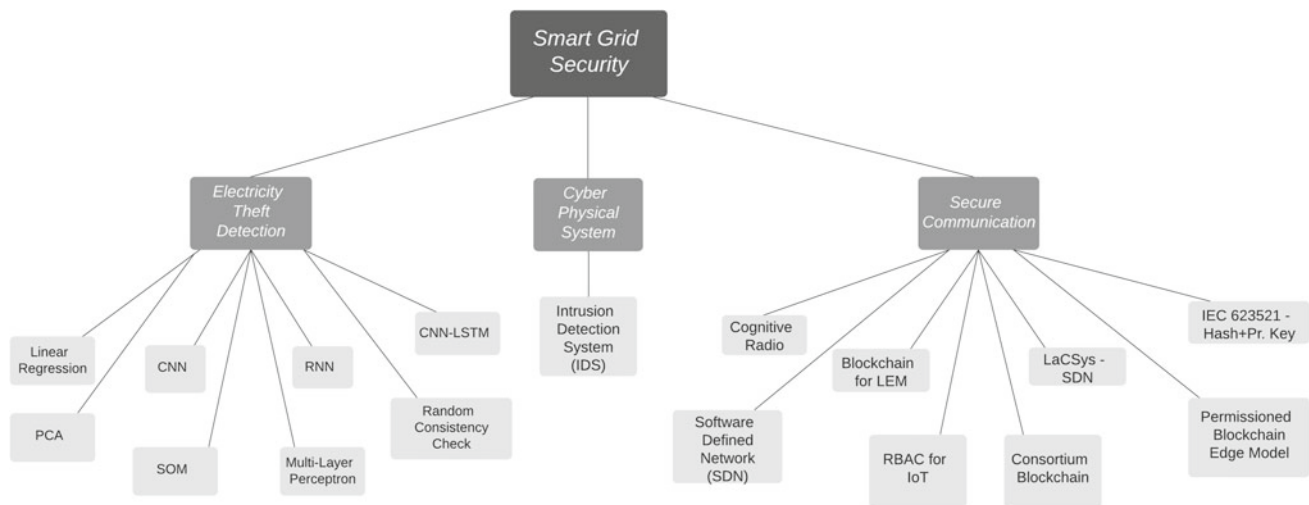


Fig. 2 Security proposals for Smart Grid in three aspects of SG operation (Micheli et al., 2019)

99.96% (Li et al., 2019). These proposals made significant improvements for detecting and protecting against electricity theft.

The other major concern in providing security to the SG is in the cyber-physical level, as SG takes advantage of Information Communication Technology (ICT) and its operation is based on Industrial 4.0, the system works based on integration of technology to the existing traditional system. The real-time operation units can be vulnerable to existing attack methods, and the risk of a successful attack on smart grid raises a threat to the reliability of the system. Usually, any attack on the smart grid would require access to the network, and an effective mitigation technique would be installing intrusion detection system either host or network based in smart grid (Sun et al., 2016).

The AMI meters used in smart grid can also be tampered for false data injection, and to detect that AMIDS was proposed by McLaughlin et al., (2013). On implementing security to SG on cyber-physical aspects, the system should be able to provide Comprehensive Situation Awareness, Secure Communication, Big Data, Optimized Decision-making Capacity, and Distributed collaborative control abilities (Jinghong et al., 2017).

The next concern when providing security in SG would be communication protocols and the methods of communication in smart grid. Cognitive Radio for Smart grid on HAN, WAN, and NAN was proposed by Le et al. (2016). The SDN enables SG can be secured by using Lattice-based Crypto System (LaCSys) (Chaudhary et al., 2018). The IoT devices in the smart grid can also be protected using OTP-based secure multicast and setting privileges using role-based access control, this could possibly avoid any rogue devices in the network and can stop privilege escalation attacks (Abreu et al., 2020). Finally addressing the

energy markets in smart grid, the markets are decentralized in nature and involve multiple parties in any process. Blockchain implementation on Local Energy Markets (LEM) was proposed to address the decentralized market (Zhao et al., 2016).

Smart Grid Security can be implemented in phases of operations in smart grid, however, when considering the prosumer side of implementation, identified three areas for securing. As shown in Fig. 2, Electricity Theft Detection was also identified as one of the major areas for securing operations in smart grid. The model implementation is further discussed in the following sections.

3 Model Implementation Details

The previous proposals conclude that artificial intelligence can be used for securing electricity grid and it could play a major role in the operation of cyber-based modules on the smart grid. Artificial Intelligence has paved the way for Machine Learning, and Machine Learning further resulted in deep learning. Deep Learning models include CNN, RNN, Reinforced Learning, and LSTMs. Most deep learning models are implemented for Images, Videos, and speech recognition. However, it is also implemented for data depending on time spatial, time-series, and data depending on complex parameters. Learning the in-depth parameters on which the data attributes are linked with can produce precise and accurate results.

This section explains the models used for ETD in this paper and provides technical parameters of implementation. This paper uses CNN for detecting theft, and CNN is mostly applied for image classification and 3D image analysis. CNN can be classified based on input dimensions depending on

the type of data and CNN could have kernel moving in one or two or three directions.

In this paper 1D CNN is implemented for electricity theft detection, as the dataset is used for electricity theft is in one-dimensional. CNN used for image processing and classification requires, and as for quantitative data, CNN requiring that is one-dimensional input would be suited for this task.

3.1 Convolutional Neural Network

Convolution is a process of summarizing a tensor or matrix or vector into smaller ones. The idea of the convolution layer is to summarize, or we can say convolve the dimensions in a tensor. If a vector a has n elements, we can convolve it using weight vector w with m elements. This would result in a shorter summarized vector b .

$$b_i = \sum_{j=m-1}^0 a_{i+j} * w_j \quad (1)$$

where $i = [1, n - m + 1]$.

3.2 Batch Normalization

This step is carried out to solve the problem of internal covariate shift since each layer should adapt itself to a new distribution from every activation. Batch Normalization normalizes inputs for each layer by calculating mean, variance, normalizing input layers, and scale and shift.

$$y_i = \gamma \bar{x} + \beta \quad (2)$$

where γ and β are learned during training, \bar{x} is the normalized input layers.

3.3 Max-Pooling and Flatten

This step, down samples by taking maximum vector over the steps and for each stride it is constrained to a pool size. Flatten is used to convert into one-dimensional array as input to the next layer. The flatten is applied on output layer of convolutional network which is connected with a fully connected layer.

3.4 Dropout

An easy method to prevent overfitting is to use dropout, implementing dropout causes reducing the network or nodes

present in a layer. It produces a sparse activation for a given layer and makes the layer to learn sparse representation in autoencoder models.

3.5 Dense Layer

Dense layer is also termed as fully connected layer. Each neuron in dense layer relates to all the neurons in the previous layer and thus forms dense structured layer when all neurons in dense layer are connected with the previous layer. Thus, the neuron in the dense layer provides learning features from the previous layers in a combination of features.

3.6 Optimizers and Loss Function

After building the model, during compilation state, optimizers and loss functions are added to improve the performance of the model.

Stochastic Gradient Descent: SGD is an optimizer function, and it takes each sample $x^{(i)}$ and labels $y^{(i)}$, and for that, the parameters are updated. It avoids redundancy by performing one update at a time and it can be fast by adapting this method. It causes the function to fluctuate heavily as it updates frequently with high variance. In the formula given below θ is the parameter (i.e., weights), η is the learning rate, ∇_{θ} is gradient of loss function,

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^i; y^i) \quad (3)$$

Binary Cross-Entropy: Binary cross-entropy is best to use when dealing with output prediction $\{0, 1\}$. The loss function uses cross-entropy to calculate a summary of difference between actual and predicted probability distribution.

3.7 Activation

Activation function is placed as node in between the next layer and previous layer, this helps the neurons that need to be activated for the next layer.

Rectified Linear Unit (ReLU): One of the widely used layers is ReLU and its significance is due to it does not activate all the neurons in a layer. So, this makes it as efficient for computation as few neurons only get activated and also it does not saturate at the positive region.

$$\text{ReLU} = \max(0, x) \quad (4)$$

Tanh Function: Tanh is a monotonic and differentiable function. The function is similar to sigmoid function, but this

function can map the negative values strongly as negative and same for zero values. The range of this function is $(-1, 1)$.

SoftMax: This function transforms values into probability distribution and this function would be ideal for multi-class classification tasks. The class with high confidence score is the predicted class. So, for given vector z , the SoftMax can be calculated as,

$$\text{Softmax} = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (5)$$

4 Proposed Model Implementation

The model requires a set of usage data to recognize the pattern of usage among the customers, and a labeled dataset would be helpful for the model to learn swiftly. As told before Voltage (V), Current (I), Power Factor (P.F) are some of the parameters that can be observed in consumer/prosumer side. The usage of customers usually would be recorded in Electrical Power (P) having Watts as unit and Power is calculated as $P = V * I$. The power readings show the consumption of the customers from the electric grid which is reflected by the action of the customers, thus any abnormalities in the usage pattern can be used an indication of anomaly. The anomalies can be inferred as an act of theft, as usage patterns was already been analyzed in the traditional grid system for generation purposes. Thus, we can conclude that power readings of customers can be used as data for the deep learning model (Fig. 3).

4.1 Data Attributes

The important attributes of the power usage of users are considered as: Consumption period of users, Periodicity of the usage. The next the attribute is pattern of usage, it is common to assume that the users living in similar locations would have similar usage patterns, there can be slight difference in amount of usage however the patterns would not tend to vary much. These attributes would eventually help the model to identify the essential features need to be extracted from the data.

4.2 Data Preprocessing

The dataset used for the detection model is from State Grid Corporation of China (<http://www.sgcc.com.cn/>) (Zheng et al., 2018). The Chinese dataset contains electricity usage of 42,372 customers and has records from Jan. 1, 2014 to Oct. 31, 2016 for 1,035 days. The dataset was padded with one day of power usage to make it as 1036 days of usage. The dataset had many missing values, and to fill out the missing values the process used a formula given in equation (6). That is if the preceding and the succeeding values are present the mean was taken to fill the missing value.

$$f(x_i) = \begin{cases} 0, & x_i \in NaN, -1 \leq i \leq 1; \\ \frac{x_{(i-1)} + x_{(i+1)}}{2}, & x_i \in NaN; \\ x_i, & x_i \notin NaN. \end{cases} \quad (6)$$

In case if the preceding and succeeding value is not present, the missing value would be filled with zero. The dataset also contained outliers, and it was necessary for the data to be processed for the model. To handle the outliers,

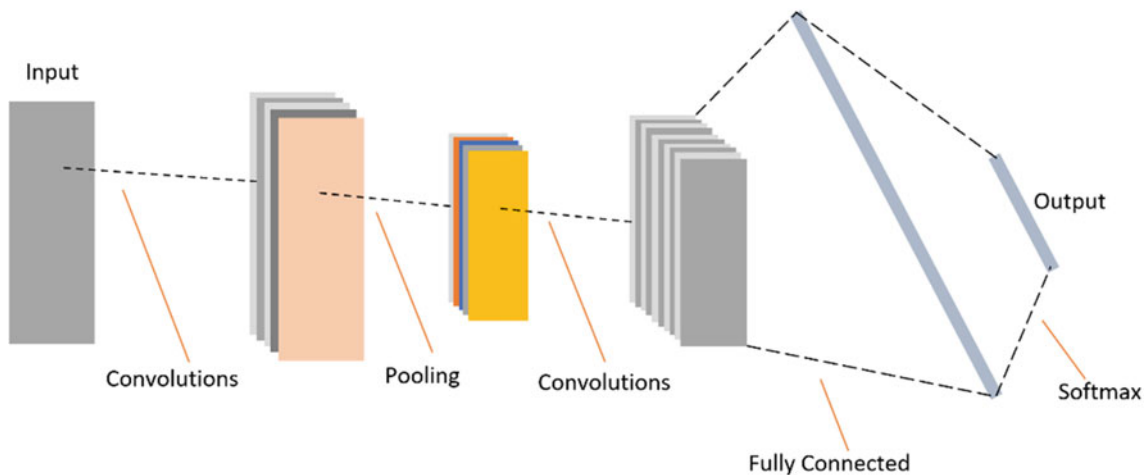


Fig. 3 Convolutional neural network model

we used 3-sigma rule, $\Pr(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 0.9973$, where μ is the mean and σ is the standard deviation. A threshold value ε was determined, and any value exceeding the threshold value is replaced by ε , where $\varepsilon = \mu + 2\sigma$. The dataset was then scaled using a Standard scaler,

$$z = (x - u)/s \quad (7)$$

where u is mean, and s is standard deviation. The scaled data was then normalized using sci-kit learn MinMaxScaler. The data was prepared for the model, and the input tensor were taken from the processed data.

4.3 Model Design

The CNN model used for ETD was given with an input shape of (1036,1) with 128 filters and kernel size of 3. The model is composed of two convolutional layers followed with batch normalization and max-pooling layer with pool size of 2. After the convolutional layers flatten was added and finally fully connected was added with four dense layers and intermediate dropout layers (Table 1, Fig. 4).

The model compilation used stochastic gradient descent for optimizer, and loss was calculated using binary cross-entropy. The model was trained on variable epochs and

Table 1 CNN model parameters

Model parameters	Values
Filter size	128
Kernel size	3
Batch size	64
Pool size	2
Input shape	(1036,1)

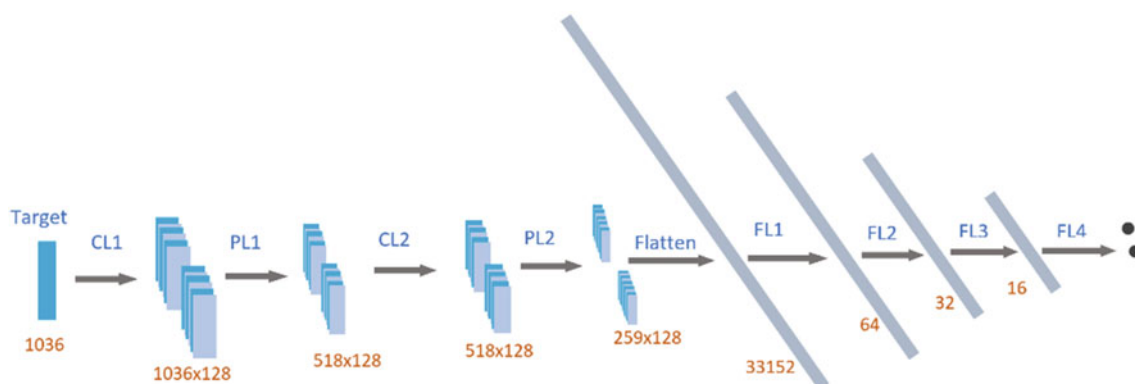


Fig. 4 Proposed CNN model for ETD in Smart Grid

batch size for hyperparameter tuning. The results of training and evaluation of the model is given in the next section.

5 Results

The dataset was split into 70% for training and 30% for validation and testing, making 15% of dataset for validation and 15% for testing. The batch size and number of epochs were changed during parameter tuning to identify the model's best performance.

The test result provided in Fig. 5a shows comparison of training and validation accuracy of standardized dataset. The training took 50 epochs to train and validate the model to determine accuracy and loss of standardized and normalized datasets. The training dataset and testing dataset used 30% of the original dataset, using split function from sci-kit learn.

Figure 5b used the same standardization process for processing dataset trained for 50 epochs to show the comparison of loss between training and validation.

From Fig. 5c, the training and validation accuracy of the model using normalized dataset can be observed. The figure shows the increase in accuracy is minimal and not much of increase in accuracy can be seen.

Figure 5d shows the results of normalized dataset which has been trained and validated with the same number of 50 epochs. The model trained with normalized dataset has greater loss in accuracy when compared with standardized dataset.

Model Testing

The model was tested for performance using the standardized dataset. The results are shown in Fig. 6a, b.

The testing result for the model trained with standardized dataset shown in Fig. 6a has a steep rise in accuracy, the accuracy of the validation sets is also similar to the training accuracy.

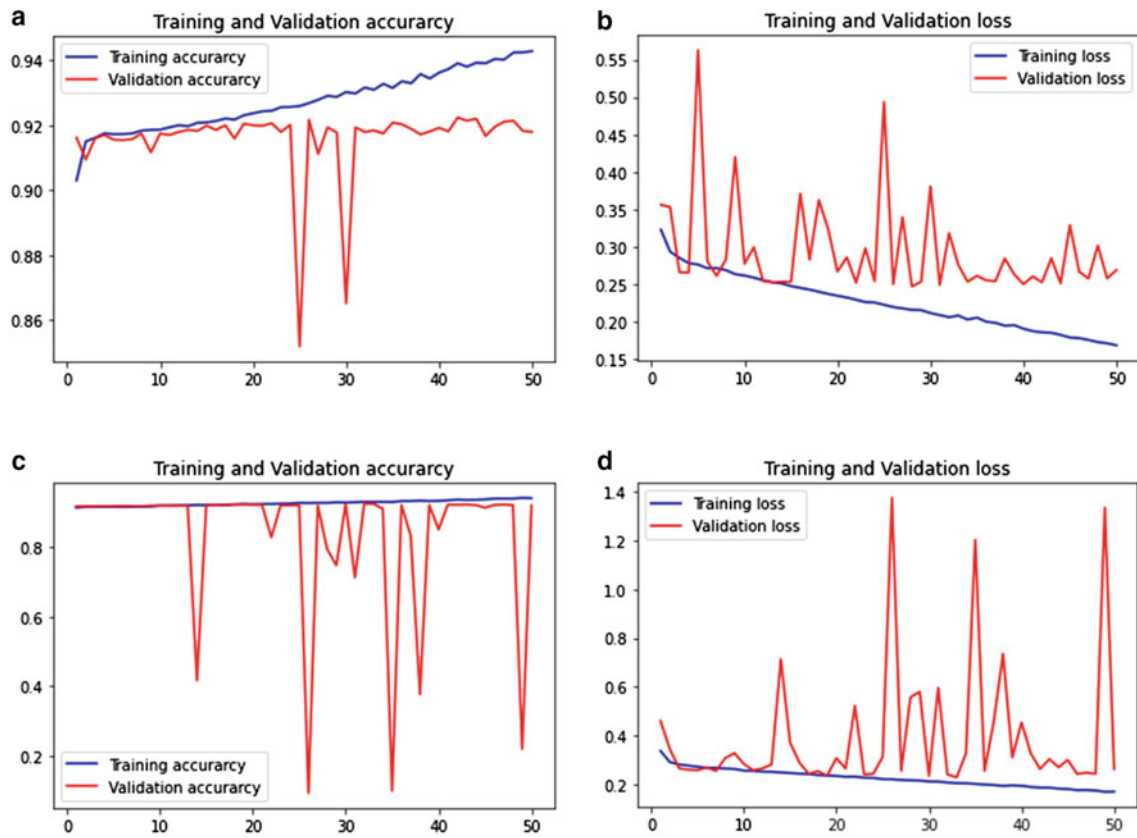


Fig. 5 **a** Training and Validation accuracy of standardized dataset **b** Training and Validation loss of standardized dataset **c** Training and Validation accuracy of normalized dataset **d** Training and Validation loss of normalized dataset

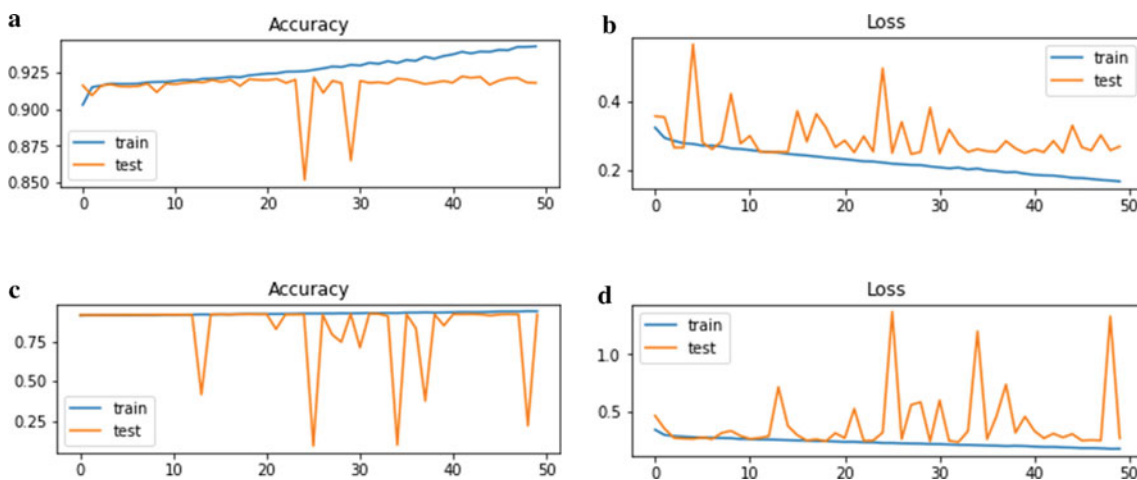


Fig. 6 **a** Accuracy of the model for standardized dataset **b** Loss of the model for standardized dataset **c** Accuracy of the model for normalized dataset **d** Loss of the model for normalized dataset

Figure 6b shows the loss of model trained with standardized dataset, the loss in accuracy is reduced with increase in epochs and loss is relatively gets low.

In Fig. 6c, the testing accuracy of the model trained with normalized dataset is given. The training and testing of the model are similar to results got for validation set.

Table 2 Experimental Results from model training and testing

	CNN			CNN with normalized data		
	20	30	50	20	30	50
Epochs	20	30	50	20	30	50
Accuracy (%)	91.45	91.74	91.22	91.86	91.92	92.16
Precision	0.91	0.948	0.9316	0.928	0.922	0.9217
Recall	1	0.038	0.9751	0.986	0.995	0.9996
F1-score	0.955	0.0736	0.9529	0.956	0.9575	0.9591

The testing loss of the model trained with normalized dataset is given in Fig. 6d. The testing loss is considerably high when compared to the model trained with standardized dataset.

Table 2 consists of all the experimental results of the model used for ETD and the performance was tuned to identify the optimum parameters that would be the most suited for the task.

The parameters required to fine-tune the model are precision (8), recall (9), and F1-score (10). These parameters would help to determine the tuning of the model depending on the values calculated from the parameters.

Precision (8) is used to determine the actual positive rate among the predicted positive of a model, the reason for considering precision is that the model should not consider legitimate user (actual negative) as a theft user.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (8)$$

Recall (9) calculates the actual positives of the model, that is in theft detection, if a theft usage (Actual positive) is predicted as (Predicted Negative), the model would not be able to detect the theft usage. Since, our model has used electricity theft detection, it is crucial that the Recall to be high.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (9)$$

F1-score (10) of a model gives the balance between precision and recall. F1-score is primarily used when there is an uneven class distribution

$$F1 - \text{score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

Accuracy (11) of the model gives the number of correct predictions of all the predictions made. Accuracy is critical to any model being deployed in a real-time scenario.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (11)$$

6 Discussion

This paper is an implementation of the proposed CNN-based electricity theft detection scheme which has then been compared with the existing scheme. The detection of theft is done based on the trained CNN model proposed in this paper, the model was trained using standardized dataset can be used, as it has overall better performance than the normalized dataset. The trained model would be ideally installed in the Intrusion Detection System (IDS) or any other specialized theft detection module in the smart meter. The theft detection model would be present in the control unit of the prosumer side, where the meter usage data are collected, the data would be analyzed using the theft detection model for presence of anomalies.

From Table 2, we can understand that the accuracy is better when using normalization on the data. The other metrics of the model performance was also comparably good, however, the recall rate of the model trained using standardized data was better. The model performance can be improved if the number of epochs were increased, and the possibility of overfitting could also be possible. From Fig. 6a, b we can conclude the model trained was not overfitted, as the training and validation characteristics were not much deviated from each other values. The model accuracy for model with standardized dataset is 91.74% and for model trained with normalized dataset is 92.16%. The trained model can be deployed in a real-time scenario to detect any abnormalities in the usage patterns of customers. Thus, the CNN model would be used as electricity theft detection model in smart grid.

The data-preprocessing was tested with different models for ETD to evaluate the performance of each model. ETD using neural network gave an accuracy of 76%, and for Extra Tree Classifier 86% accuracy was obtained as shown in Fig. 7. The proposed method was compared with other existing proposals such as CNN-LSTM and DBSCAN as shown in Fig. 8. The CNN-LSTM model proposed in (Hasan et al., 2019) uses LSTM to process the time-series data of the smart meter. Multiple Linear Regression Model (MLRM) uses P2P computing to process the data and the method was

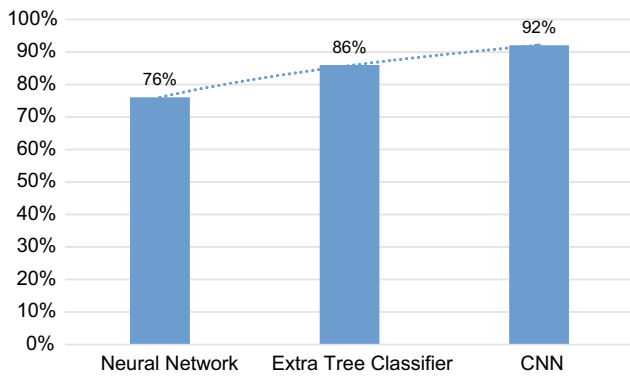


Fig. 7 Accuracy of models used for ETD

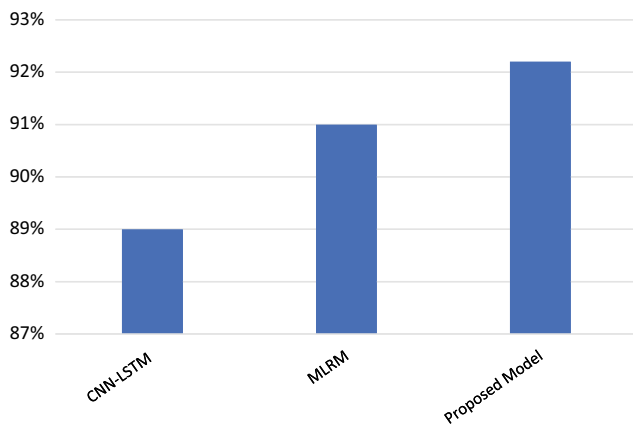


Fig. 8 Comparison of existing methods for ETD

tested with other models such as Decision Tree, SVM, and Logistic Regression (Micheli et al., 2019). The proposed ETD model performs much better when compared with the model proposed in (Hasan et al., 2019). The model can be further optimized with fine-tuning the layers used in this deep learning model so that much more reliability and robustness can be achieved. To remotely implement ETD, a model requiring less computation and high reliability is required. The model proposed in this paper can satisfy the mentioned requirements for implementing ETD remotely to detect electricity theft in real-time scenarios.

7 Conclusion

Smart grid is a revolutionary advancement in the electrical grid technology. Smart grid ensures the overall stability and provides a reliable fail-proof system for energy transmission and distribution. This paper discussed some threats for secure operation of smart grid and electricity theft. This paper defined the implementation of an improved deep learning method for ETD. The paper used two preprocessing

techniques to observe the model's performance and identified the most suited preprocessing technique for ETD model. The paper presented complete implementation of CNN deep learning model for ETD and an analysis of performance metrics shows that the model proposed in this paper obtained an accuracy of 92.16% with precision score of 0.92. The work can be further extended by improving the overall performance of the model to increase the reliability of the model.

References

- Abdulrahman, M. W., Olatunji, O., Salman, M., Toyin, N., & Salisu, S. (2019, December). Electricity theft detection by sources of threats for smart city planning, *I(2)*, 52–60
- Abreu, V., et al. (2020). *Identity and access management for IoT in smart grid*. International Conference on Advanced Information Networking and Applications. Springer, Cham.
- Chaudhary, R., et al. (2018). *LaCSys: Lattice-based cryptosystem for secure communication in smart grid environment*. 2018 IEEE International Conference on Communications (ICC). IEEE.
- El Mrabet, Z., et al. (2018). Cyber-security in smart grid: Survey and challenges. *Computers & Electrical Engineering*, 67, 469–482.
- Greer, C., Wollman, D. A., Prochaska, D. E., & Boynton, P. A. (2019). *NIST framework and roadmap for smart grid interoperability*. Accessed August 11, 2019 [Online].
- Gunduz, M. Z., & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. *Computer Networks*, 169, 107094
- Hasan, M. N., Toma, R. N., Nahid, A. A., Islam, M. M. M., & Kim, J. M. (2019). Electricity theft detection in smart grid systems: A CNN-LSTM based approach. *Energies*, 12, 3310.
- Jakaria, A. H. M., Rahman, M. A., & Moula Mehedi Hasan, M. G. (2019). *Safety analysis of AMI networks through smart fraud detection*. 2019 IEEE Conference on Communications and Network Security (CNS), Washington DC, DC, USA, 2019, pp. 1–7.
- Jinghong, G., Ziwei, H., Yan, Z., Tao, Z., Yajie, L., & Fuxing, Z. (2017). *An overview on cyber-physical systems of energy interconnection*. 2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC), Singapore, pp. 15–21
- Le, T. N., Chin, W.-L., & Chen, H.-H. (2016). Standardization and security for smart grid communications based on cognitive radio technologies—A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 19(1), 423–445.
- Li, W., Logenthiran, T., Phan, V., & Woo, W. L. (2019). A novel smart energy theft system (SETS) for IoT-based smart home. *IEEE Internet of Things Journal*, 6(3), 5531–5539.
- McLaughlin, S., Holbert, B., Fawaz, A., Berthier, R., & Zonouz, S. (2013). A multi-sensor energy theft detection framework for advanced metering infrastructures. *IEEE Journal on Selected Areas in Communications*, 31(7), 1319–1330.
- Mengelkamp, E., et al. (2018). A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science-Research and Development*, 33(1–2), 207–214.
- Micheli, G., et al. (2019). Big data analytics: an aid to detection of non-technical losses in power utilities. *Computational Management Science*, 16(1), 329–343.
- Musleh, A. S., Chen, G., & Dong, Z. Y. (2019). A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grid*, 11(3), 2218–2234.
- Patil, Y. S., & Sankpal, S. V. (2019). *EGSP: Enhanced grid sensor placement algorithm for energy theft detection in smart grids*. IEEE

- 5th International Conference for Convergence in Technology (I2CT). Bombay, India, pp. 1–5.
- Rajesh, G., Raajini, X.M., & Dang, H. (2021). *Industry 4.0 interoperability, analytics, security, and case studies* (1st ed.). CRC Press. <https://doi.org/10.1201/9781003048855>
- Sikeridis, D., et al. (2020). *A blockchain-based mechanism for secure data exchange in smart grid protection systems*. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). IEEE.
- Singh, S. K., Bose, R., & Joshi, A. (2019). Energy theft detection for AMI using principal component analysis based reconstructed data. *IET Cyber-Physical Systems: Theory & Applications*, 4(2), 179–185
- Sun, Chih-Che., Liu, Chen-Ching., & Xie, Jing. (2016). Cyber-physical system security of a power grid: State-of-the-art. *Electronics*, 5(3), 40.
- Yao, D., Wen, M., Liang, X., Fu, Z., Zhang, K., & Yang, B. (2019). Energy theft detection with energy privacy preservation in the smart grid. *IEEE Internet of Things Journal*, 6(5), 7659–7669.
- Yip, S.-C., et al. (2017). Detection of energy theft and defective smart meters in smart grids using linear regression. *International Journal of Electrical Power & Energy Systems*, 91, 230–240.
- Yu, X., & Xue, Y. (2016). Smart grids: A cyber-physical systems perspective. *Proceedings of the IEEE*, 104(5), 1058–1070.
- Zhao, B., Stankovic, L., & Stankovic, V. (2016). On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. *IEEE Access*, 4, 1784–1799.
- Zheng, Z., Yang, Y., Niu, X., Dai, H.-N., & Zhou, Y. (2018, April). Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, 14(4), 1606–1615.
- Zhuang, P., Zamir, T., & Liang, H. (2020). Blockchain for cyber security in smart grid: A comprehensive survey. *IEEE Transactions on Industrial Informatics*.



Helping Hand: A GMM-Based Real-Time Assistive Device for Disabled Using Hand Gestures

S. Gnanapriya and K. Rahimunnisa

Abstract

In recent years, the emergence of computer vision (CV) and machine learning techniques for human to communicate with computer and vice versa has paved the way to fulfill the needs of people with physical disabilities and the elderly, by breaking the communication barrier they face. The vision-based hand gesture recognition system eliminates the need for additional devices and the physical contact with the device to identify the hand gesture shown by the subject to express their willingness. The assistive device uses webcam or the video camera connected with the laptop or desktop placed in front of the disabled person's bed. The hand gesture shown by the disabled person is segmented using the machine learning-based probabilistic Gaussian mixture model (GMM) algorithm by eliminating the background. The hand features are extracted using Hough transform, as it can tolerate changes in orientation and hand shape deformation as the disabled person may not always give a clear view of any shown gesture. The recognition of hand gesture is carried out by comparing the extracted features with the most likely used pre-stored features used by the disabled and elderly like the need for food, water, medicine, wash room, poser change and wheel me out. The prediction of hand gesture is carried out using the K-NN algorithm. Once predicted, the audio file corresponding to the recognized gesture stored in the database is played since the family members or the caretaker may not be with the disabled person every second or minute for immediate attention. The overall gesture prediction accuracy of the proposed system is found to be 95%, and the frequency distribution of gesture usage per day is found to be uniformly distributed with the chi-square test.

S. Gnanapriya (✉) · K. Rahimunnisa
Easwari Engineering College, Chennai, India
e-mail: gnanapriya.s@eec.srmrmp.edu.in

K. Rahimunnisa
e-mail: rahimunnisa.k@eec.srmrmp.edu.in

Keywords

Hand gesture recognition • Disabled • Gaussian mixture model (GMM) • Background elimination • K-nearest neighbor (K-NN) • Hough transform (HT)

1 Introduction

Elderly people and to be the disabled feel depressed and discomfort as they are not in position to take care of themselves. They need to wait until the caretaker or their loved ones come and attend to them. The advancements in current trends and technologies like network of physical devices using Internet of things (IoT), automatic analytical model building using machine learning (ML), deep neural network that depict human brain with deep learning (DL), artificial intelligence (AI), digital image processing (DIP), computer vision and human computer interaction make them express their feelings in an easy way.

The materialization of IoT encourages the expansion of healthcare monitoring system. The proposed system in Tan and Abdul (2018) measures the blood pressure, temperature and pulse rate. The output consists of readings that are then transferred to cloud storage and diagnosed by the doctor. The system also addresses diabetics and kidney disease. The output electrical signals from the sensors are processed by Arduino and the computer-on-module for wearable devices. Intel Edison is used to transfer the data to cloud store. The graphical user interface (GUI) provides the data visualization chat box for doctors consultation. The system performance is analyzed by comparing the sensor reading with the conventional devices and transmission delay with respect to various Internet speeds, and it is found that the delay is acceptable even with low an Internet speed of 64kbps. The paper (Mohana et al., 2020) proposed IoT-based hand glove for helping the paralyzed persons to communicate with their attender using the hand gestures.

The personal digital assistant (PDA) proposed in Lin et al. (2006) detects fall of stroke in the patient and older people. The subject is restricted to wear the cloth attached with the microsensor to detect the fall location and send messages through Bluetooth technology to send stored help related voice message to the family member or caregiver. The devised indoor mobile robot (Pineiro et al., 2020) detects the fall in elderly and takes medicines to the victim using the shortest path and also intimates the caregiver. The Robot automatically senses the incident. It is designed using mechanical and electronics devices with sensors and actuators. Communication between electronics devices and sensors and actuators happens through message queuing telemetry transport (MQTT) IoT protocol. The proportional integral derivative controller uses the wall following technique to keep the robot away from the wall. The SLAM technique with Kalman filter is used for improving the location mapping accuracy by predicting and minimizing error covariance using the recursive nature of the Kalman filter. Depth first search (DFS) and breadth first search (BFS) are adopted to determine the shortest distance from the subject. From the experimentation results and discussion, it is obvious that BFS performed better as per the DFS; the robot moves through each vertex of the room to reach the destination. Human-computer interaction system helps people with motor disabilities to lead a partially independent life.

The proposed approach (Sriram & Palaniswamy, 2019) enables people wheeled on a mobile robot get their work done faster. The interaction between human and mobile robot happens with the help of hand gestures. The convex hull and convexity defect are used to detect hand and recognize the hand gestures by subtracting the background scene. The gesture is recognized through the camera attached to the robot. The robot is controlled by the XBee radio frequency (RF) module which provides the user interface and transfers data to and from the user. The advantage of the proposed system is that the users can acquire object in a home environment and reach out to the object location of their interest to get their work done by themselves. The drawback of the proposed system is the cost and living place of the needy, as the person has to possess a power geared wheel chair and spacious house to move around. The increase in the average life of people increased the elderly population who need attention. This attention is naturally given with the advancement in gesture-based human computer interactive technology.

The system proposed by the author (Teimourikia et al., 2014) uses both static and dynamic inputs collected using a Kinect device to collect over time. Both RGB and depth information is taken for processing. NiTE skeletal tracking is used to segment the hand by using depth thresholding; this eliminates a cluttered or dynamic background. Then, the

algorithm Voronoi diagram is used to extract the segmented portion of the hands skeletal features. The extracted skeletal feature is used for hand pose estimation and recognition. The statistical Markov model that undergoes the Markov process called the hidden Markov model (HMM) is used to analyze the continuous hand poses corresponding to the shown gesture. To recognize the user shown gesture, the pre-stored pose bank is used. The effectiveness of the system was evaluated with both young and older users. The system is trained for 21 gestures using young and old users and tested with both young and old users. It is evident that the system performs better with young users than older ones, as users as the older users considered for testing had hand deformity, arthritis and hand tremor.

In paper (Oudah, 2020), the motion sensing input device Kinect placed in front of the elderly or the disabled patient's bed recognizes the shown hand gesture sign and the meaning to the caretaker through microcontroller for mobile communications (GSM). The Kanade-Lucas-Tomasi (KLT) algorithm processes the Kinect input to detect the facial feature points in a live video. The video frame is then converted from RGB space to YUV space to extract the skin region by applying a skin threshold; this also removes the background. The extracted skin region is enhanced by using a morphological operator. This now acts as the mask to subtract the face detected using the KLT algorithm. The count of fingers in the extracted hand palm in the image is used to identify the shown gesture. The gestures were identified as water, meal, toilet, help and medicine corresponding to the number of fingers raised in shown hand gesture input. Each gesture is tested with 44 sample inputs. The advantage of the proposed system is that it is portable, but the limitation is that the system is trained only for daylight and it needs a Kinect device. From all these ideas that have been proposed in the past, the vision-based hand gesture recognition assistive device if found to be affordable can be used by all groups of people, as it requires only limited hardware, is easy to use and portable.

2 Related Work

The three main stages in hand gesture recognition as addressed by researchers in literature are segmentation, feature representation or extraction and classification.

2.1 Hand Segmentation

Segmenting the hand from other background objects from an image improves the processing and recognition time, but this is a complex process because of the change in illumination condition, complex background and object occlusion that

happens in the image. Automatic hand segmentation (Coogan et al., 2006) is achieved using the skin threshold, but this skin threshold extracts both hand and face regions leading to occlusion between the face region and the two hands. The Kalman filter tracks occlusion by tracking the vertex of the bounding box around the hand and face region for each frame. The static gestures are recognized using the principle component analysis (PCA) subspace. This subspace is created for each gesture by applying PCA on an image. This, from the resultant orthogonal eigenvectors, forms the subspace. The gesture classification is carried out using the distance measure. The dynamic signs are recognized using the hidden Markov models (HMMs), and the probability distribution over the image gives the feature vector consisting of two parts, where the first part holds the class information of the static sign, and the second part holds the position of the gesture in the frame.

Paper (Zheng & Zheng, 2015) uses the improved Gaussian mixture model to detect the dynamic hand sign from the video frames. The intensity value of each frame at every time step ' t ' is obtained as the weighted mean value of the Gaussian distribution. The all M numbers of Gaussian distribution corresponding to the gesture are sorted in the ascending order based on their weight. The first one is considered as the background and compared with successive distributions; if any frame matches, then it becomes the background, else it becomes the foreground. Background removal (Sinha et al., 2019) is carried out using the Gaussian mixture model that uses a separate skin threshold for the face and the hand region in YCbCr space.

Paper (Sharma, 2020) uses a Kinect camera to capture a depth image as input and uses a depth map to fix the depth threshold to segment the hand, but this includes the forearm along with the hand palm. To segment the forearm, the center of the palm is found using the Gaussian blur; this gives the region with dense intensity as the center. A circle drawn with the center, and intersection point in the third and fourth quadrant is the wrist and the region after the wrist point is eliminated. From the background removal suggested by researchers, it is evident that the Gaussian mixture-based approach is best for background or foreground removal.

2.2 Hand Feature Extraction

Hand features play a major role in gesture classification. To classify the gestures effectively, hand features need to be extracted. In literature, various approaches are used for precise gesture feature extraction. The paper (Mesbahi et al., 2018) uses MOG2 (mixture of Gaussian) for background subtraction and Canny edge for object contour detection. The convexity approach is used to detect and extract hand features using the convex hull over contour. The extracted

features include the number of vertices and the angle between the fingers, which is then used for classifying the gestures corresponding to classes one, two, three, four and five.

Data augmentation that addresses changes in orientation, size and position of objects is achieved by paper (Sahoo et al., 2018) using the feature selection technique discrete wavelet transform (DWT). The proposed work also addresses changes in image illumination conditions by computing the mean of each channel of the image and multiplying it with the individual pixel values using the gray world algorithm. Hand segmentation is carried out using skin thresholding. DWT generates four types of features like approximation, vertical, horizontal and diagonal features. To select the best feature out of these four, F-ratio is used, which considers the variance between samples and variance within samples to select the appropriate feature. The classification of gestures is done with the singular value decomposition (SVM).

The nonlinear edge preserving bilateral algorithm is used by Li et al. (2017) to reduce noise and fill holes in the input with the weighted average of intensity values from the nearby pixels. Feature extraction is carried out using convolutional neural network (CNN); though it is a feed forward network during training, it uses back propagation. The CNN given features are classified using SVM. In Wadhawan and Kumar (2020), the deep learning algorithm CNN performs both feature extraction and classification automatically using the CNN Alex Net architecture with the Adam optimizer and ReLU activation function. The system is trained to classify 100 classes, and the architecture performance is evaluated by changing the layers and optimizers. The system is evaluated using F1 score, precision and recall, and it is observed that the system performance is found to be the same as human interpretation.

The combination of CNN for static images and recurrent long short-term memory (LSTM) for dynamic images is used by Mittal et al. (2019) to recognize words and sentences by considering fingertip positions and palm centers from the real-time dynamic gesture inputs collected using the leap motion controller. Feature extraction is carried out using CNN, and classification is carried out by the LSTM, the recurrent neural network using ReLU activation function and Adadelta optimizer.

The objective of the authors (Patil & Subbaraman, 2018) is to recognize the dynamic gesture that changes with time by extracting the spatiotemporal feature in real time that is robust against data augmentation techniques like changes in orientation, size, position and light variation in images. The spatial features are extracted using Fourier descriptors, geometrical information of input gestures and the temporal features using the Hough transform. These two features were combined and classified using the artificial neural network

(ANN) with two hidden layers which uses tansig and purelin transfer function, respectively, trained for 100 epochs, with a learning rate 0.1. As the Fourier descriptor is used to extract spatial features, it eliminated the need for object segmentation. The extraction of temporal features detects the hand edges in successive frames, and this is then transformed from the spatial (x, y) space to the (ρ, θ) Hough space. An accumulator array $A(\rho, \theta)$ is constructed for all edge points of the image; from this accumulator array, the maximum value is determined as the local maximum. The computed local maxima of successive selected frames form the temporal features. The proposed system was tested using the benchmark hand posture databases, with images of variations in color, illumination, orientation and view, and it is found that the Sebastian dataset outperforms the Cambridge hand gesture database. The advancement in cameras enables us see objects even at night.

The idea proposed by Liao et al. (2018) uses both the depth image and RGB color image to recognize the gesture. The generalized Hough transform (GHT) extracts features of any arbitrary shape, and the depth image eliminates the need for background subtraction. GHT feature descriptors are extracted from both the depth and RGB color images. The extracted hand features are used to train the double channel convolutional neural network (DC-CNN). During preprocessing, the Sobel operator detects the hand contour and picks a reference point and draws lines from the reference point to the hand contour determining the gradient angle ϕ in a reference table called the R-Table; this table is used when the image is zoomed. Hand segmentation is done by comparing the pixel values of the RGB image with a pre-set threshold of the depth image. Double channel CNN receives both the depth image and RGB image segmented using the Hough transform. The extracted features from both the images are fused together for recognizing the gestures. The system was trained with a self-collected American Sign Language (ASL) dataset corresponding to 24 static signs of the English alphabets taken under seven different background and lighting conditions. It is observed that the proposed DC-CNN improved the learning rate and system recognition accuracy. Paper (Rerkgamsanga et al., 2016) classified the object of interest in maritime domain using the generalized Hough transform. The object edge coordinates extracted by the generalized Hough transform are extracted to construct the coordinate table; these features are then used to train the neural network classifier. It is found that the system classifies both geometrical shapes and maritime domain objects.

The Turkish paper (Onat & Ümer Üzdil, 2015) used the color and shape features of traffic signs to extract the traffic signs by eliminating the other objects in the image. The edges of the extracted signs are detected using the canny edge detector. The Hough transform feature extraction

technique extracts the feature descriptor from the gray scale edge detected image. The extracted feature maps are used to train the singular value decomposition (SVD) classifier to classify the multiple classes of traffic signs. From literature, it is found that the convexity approach fully depends on the hand shape and finger angle. DWT (Sahoo et al., 2018) and CNN (Wadhawan & Kumar, 2020) handle data augmentation with a high recognition rate, and LSTM (Mittal et al., 2019) handles continuous gestures. The Hough transform is found to tolerate hand shape deformation, change in hand orientation and provide quality equivalent to template matching and to provide a feature vector to any arbitrary shape of object with no additional hardware requirement. The quality of the results produced by Hough is almost the same as template matching with a few computational resources, and the computational cost of the Hough transform is $O(n_e l)$ (Aguado, 2008) where n_e indicates the number of lines, and l indicates the length of the lines.

2.3 Hand Gesture Recognition

The classification technique plays a major role, as it directly prompts and guides the user to perform related actions. In (Bhattacharya et al., 2012), the Kinect sensor's data stream is classified using multiclass SVM and decision tree. Classification is done by comparing class probabilities. The class probability for continuous gestures is collected by using sliding window-based estimation. Each position of the slide along the time line probabilities is computed. The summation of these values predicts the gesture. The paper (Wadhawan & Kumar, 2020) concludes that deep learning-based CNN classifier's classification accuracy is better than that of other machine learning techniques like SVM, K-NN and ANN. Though CNN (Gnanapriya et al., 2020) gives accurate prediction, it requires a high end system with special hardware for processing. Also SVM, ANN and CNN need training for prediction. This is overcome by the simplest machine learning-based K-NN classifier, which just uses various traditional distance measures for classification (Munib et al., 2007) and whose accuracy depends on the taken K value.

3 Proposed System

The webcam positioned at either end of the disabled person's cot as shown in Fig. 1, captures the input for processing. The probabilistic Gaussian mixture model (GMM) removes the background objects and extracts the hand region. The hand features are then extracted using Hough transform (HT). Since the GMM cluster data is based on each point in feature space, the position of the camera is not a constraint, and

similarly, the use of the Hough transform extracts shape features which overcome the change in orientation, viewing direction of the hand and camera facing position. This is then processed by the laptop or desktop attached to it. Finally, gesture classification is carried out using the machine learning-based K-nearest neighbor algorithm. The proposed machine learning-based hand gesture recognition system for disabled people is shown in Fig. 2.

3.1 Gaussian Mixture Model (GMM)

GMM, an unsupervised clustering technique, is used to extract the foreground by eliminating the background of an image when the regions of an image are inseparable. If the input image has no complex backgrounds, K-means clustering will do. If the input image has a complex background, the clusters overlap, and GMM is opted for separation (Marshland, 2015).

3.1.1 Procedure

- i. Input Image $I(x, y)$
- ii. Generate finite number of Gaussian distribution for each data point using the probabilistic model.
- iii. Each Gaussian had its own mean, variance (for one-dimensional data, and for multidimensional data, it is covariance), and weight, where weight is the probability of the data point.
- iv. The output of this algorithm is the summation of the values of all M Gaussians.

$$f(x) = \sum_{m=1}^M \alpha_m \varphi(x; \mu_m, \Sigma_m) \quad (1)$$

where m —number of Gaussian Model
 α_m —Weight of m Gaussian



Fig. 1 Possible webcam positions to capture hand gesture

μ_m —mean of m Gaussian Function
 Σ_m —covariance of m Gaussian Function

$$\sum_{m=1}^M \alpha_m = 1 \quad (2)$$

- v. Weight is calculated using the maximum likelihood estimation, and likelihood function calculates the conditional probability of a data point belonging to a particular cluster, where the probability that a particular data point x_i belonging to class m is given by

$$p(x_i \in c_m) = \frac{\text{Probability } x_i \text{ belong to class } m}{\text{sum of values expected by all Gaussians}} \quad (3)$$

- vi. In the subsequent iterations, the mean, variance and weight are updated using the expectation maximization algorithm for GMM, where the expectation step calculates the probability for each data point, and the maximization step calculates the mean, variance and weight.
- vii. Repeat until convergence.

After segmentation, the binary mask corresponding to the background segmented image is created by toggling the bits. The binary mask for the segmented image is shown in Fig. 3. This is then given as input to the feature extraction stage.

3.2 Hough Transform for Feature Extraction

Hough transform (HT) is a feature extraction technique (Liao, et al., 2018; Patil & Subbaraman, 2018; Sheshkus, et al., 2016) that extracts shapes in an image; initially, it is implemented to extract the lines in an image, and then, it is extended to the circle and ellipse. The performance of the HT is same as template matching. The Hough transform defines mapping from the image space to the Hough space. HT is used to find the lines in an image; in a Cartesian coordinate system, the line is defined as

$$y = mx + c \quad (4)$$

where m is the slope, and c is the y intercept.

As these two parameters are used to describe a line, the problem is that it cannot represent vertical lines since the slope is the same along every point on the line. Therefore, HT uses the polar coordinate system to represent a line, and it is defined as

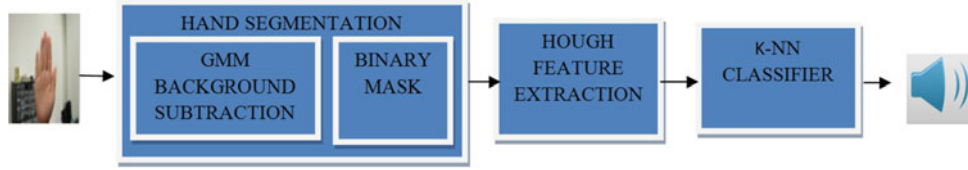


Fig. 2 Architecture of the proposed assistive system for disabled



Fig. 3 Input, binary mask of background segmented image

$$r = x \cos \theta + y \sin \theta \quad (5)$$

where

r —distance from the origin to the nearest point on the line, where $r \geq 0$.

θ —angle between x the axis and the line connecting the origin to the nearest point,

where $0 \leq \theta \leq 360$.

This is shown in Fig. 4. In Hough space, a line is represented as (θ, r) . In the Cartesian coordinate, the lines pass through each data point of an image in any direction. Each line through the point (x_i, y_i) is considered as a vote for many points in Hough space. All lines correspond to a point in (θ, r) space that satisfies the line equation in the polar coordinate. Accumulator array is a 2D matrix that is initialized to

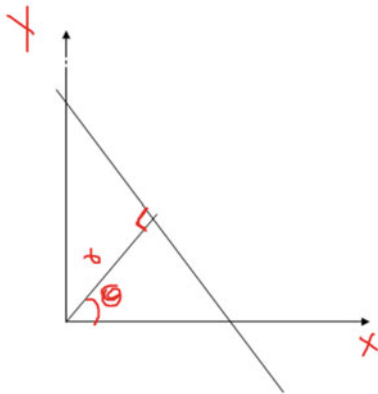


Fig. 4 Line parameters in polar coordinate

hold the count of votes. Any (θ, r) that corresponds to (x, y) is incremented by 1 i.e., $A[\theta, r] = A[\theta, r] + 1$.

3.2.1 Procedure

- i. Input data points (x_i, y_i) .
- ii. Divide θ , into finite number of bins of equal angle (instead of considering every angle).
- iii. Initialize accumulator array $A[\theta, r]$ to hold count of vote.
- iv. For every data point (x, y) , use HT to draw multiple lines, in all directions and compute (θ, r) .
- v. Check if (θ, r) aligns with any particular (x, y) point and increment that particular accumulator array matrix by 1.
- vi. At the end, the bin that has the maximum corresponding (θ, r) is considered as the local maxima.

The proposed Hough transform for feature extraction computes standard Hough features from an input binary image. The Hough transform does the mapping of Hough lines in (x, y) space to (θ, r) space. The sample Hough lines and their coordinates in (x, y) space are shown in Fig. 5. The Hough matrix or accumulator array contains the frequency of occurrence of (θ, r) for the given input image in (x, y) space for every point (x_i, y_i) as shown in Fig. 6. From this Hough matrix, the mean of 15 Hough peaks is taken to determine the local maxima. The position of the Hough peaks in the Hough matrix is shown in Fig. 7. From these positions, the mean value is taken as the extracted feature. The mean Hough peak is shown in Fig. 8. This is then given as input to the K-NN classifier.

3.3 K-NN Classifier

The machine learning-based lazy classification algorithm K-nearest neighbor, stores the given reference data in memory and uses it for finding the difference when a new instance x occurs (Wan et al., 2017).

- i. Input data (local maxima)
- ii. Input K
where K —number of nearest neighbor to consider.



Fig. 5 Hough features in (x, y) space

H x												
279x360 double												
1	172	173	174	175	176	177	178	179	180	181	182	183
191	2	2	3	3	5	6	3	2	3	3	0	1
192	4	3	3	4	2	0	3	4	2	2	5	2
193	3	3	3	1	4	5	4	2	4	4	0	3
194	3	4	4	5	3	1	2	3	2	2	6	2
195	4	4	2	1	3	5	3	3	4	3	0	4
196	2	3	4	5	5	1	3	3	2	3	6	3
197	5	2	2	1	1	5	3	3	6	3	0	3
198	1	4	3	4	4	1	3	3	0	3	6	3
199	4	2	3	3	1	5	2	2	5	3	0	3
200	0	2	2	3	5	0	3	3	0	2	5	2
201	2	2	2	2	1	6	4	2	5	3	0	3
202	3	2	2	2	3	0	2	5	0	2	5	2
203	2	3	3	3	1	4	3	1	6	5	0	3
204	3	3	3	2	4	3	4	3	0	1	6	1
205	2	2	2	3	3	2	1	4	7	3	0	5
206	2	1	3	2	2	3	4	1	0	4	7	4
207	2	3	1	2	2	2	1	4	5	1	0	3
208	2	1	3	2	2	2	2	1	0	4	5	4
209	2	3	1	2	2	2	2	3	4	1	0	1
210	3	1	3	2	2	2	2	1	0	3	4	3

Fig. 6 Sample Hough features in (θ, r) space

- iii. Use distance metric to find the difference between the data and all the reference data
- iv. Select K-nearest neighbor and its associated class based on the difference.
- v. The majority class of K reference data is considered as the predicted class.

The six Gestures, their associated classes and local maxima, taken with respect to the three samples under each category, are given in Table 1.

The classification accuracy of the system is analyzed with both the Euclidean distance and Manhattan distance (Machine Learning Matery).

$$\text{Euclidean distance } d(x, y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (6)$$

$$\text{Manhattan distance } d(x, y) = \sum_{i=1}^N |x_i - y_i| \quad (7)$$

where N is the number of parameters considered.

x_i observed local maxima.

y_i reference local maxima.

Since the single parameter (mean Hough peak) is considered for analysis, the performance of both the distance measures is found to be the same. On prediction of the shown gesture, the audio file associated with the gesture is played and amplified by speakers placed in various rooms of the house, so as to enable the caretaker or the family member to attend to the disabled or elderly knowing the need and urgency of the caller.

The proposed system obtains three Hough peaks for each class on different hand orientation and used as reference Hough peak as shown in Table 2. The slight variation of each Hough peak for the same class is due to the hand orientation with respect to the camera position and the disabled persons hand deformation percentage. This system overcomes changes in orientation of the shown input gesture; to substantiate the above sentence, an example of K-NN prediction for the sample change in orientation of

Fig. 7 Hough peaks coordinate positions

x	y
192	358
89	12
114	1
80	1
104	63
99	55
201	324
115	9
191	324
124	11
108	69
120	1
84	6
130	27
161	94

gesture medicine shown in Fig. 8c is classified with the K-NN algorithm with the number of the nearest neighbor $K = 5$. As a case of proof, the distance between the obtained Hough peak for the gesture medicine (0.6093) and the reference Hough peak is computed using Euclidean distance.

As the content shown in bold font in Table 2 has the majority class of the selected five nearest neighbor as medicine, the gesture is predicted as medicine, and the corresponding audio file is played.

4 Result and Discussion

The self-created hand gesture dataset consisting of six gesture class is used for both training and testing the assistive system. As the disabled or elderly cannot be in a position to use both the hands or the hand signs available in the dictionary to show the gestures, simple common gestures that could be shown with one hand are used here. The system is tested with 20 test images under each category based on the data available, in Table 1. It is found that the recognition accuracy of gesture food, water, medicine and wheel out is quite good compared to wash room and takeout. For a few samples, medicine is predicted as wash room, and water is predicted as take out. The prediction accuracy of the system is given in Table 3. The rotational invariant property of HT and use of either hand for showing gestures is evident from Fig. 8, where the mean Hough peak is approximately the same for any orientation and hand.

The classification accuracy in percentage of the proposed method is compared with the state-of-the-art methods (Oudah, 2020, 2021). The recognition rate of classes of the proposed system and the classes of the state-of-the-art approach are almost on par with each other, but at no additional cost of hardware, in the case of the proposed system. Unlike the proposed vision-based approach for gesture recognition, the existing approaches used depth image captured using Kinect sensor to recognize the gesture. The comparative analysis is depicted in Table 4.

The system is also tested with the chi-square test (Vani & Ananthalakshmi, 2020) to ensure that the use of each hand gesture is uniformly distributed on any taken day.

H0—frequency of occurrence of all gestures is the same.

H1—not same.

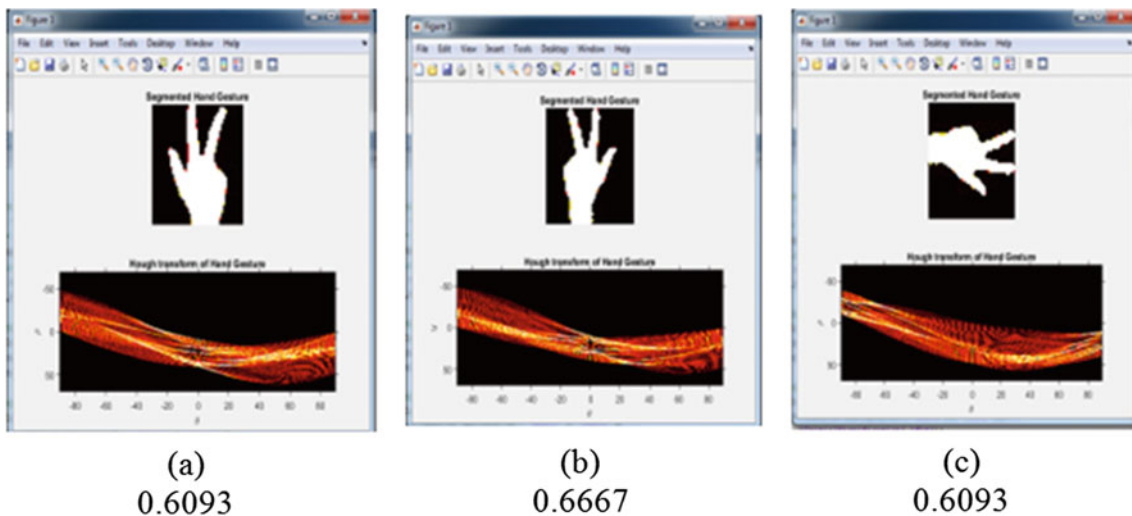


Fig. 8 Mean Hough peaks of gesture medicine shown with different hands and orientations (a–c)

Table 1 Gesture and its Class with local maxima for K-NN Prediction







Gesture	Class	Mean Hough peak (Local maxima)
	Food	0.3978
		0.4050
		0.4086
	Water	0.4910
		0.5233
		0.5054
	Medicine	0.6093
		0.6666
		0.6667
	Wash room	0.7240
		0.7419
		0.7814
	Poser change	0.8387
		0.9391
		0.8495
	Wheel out	0.4588
		0.4444
		0.4480

Table 2 Example change in orientation gesture prediction using K-NN

Class	Reference Hough peak	Euclidean distance $d(x, y)$ between reference and obtained peak	$K = 5$ Nearest neighbors
Food	0.3978	0.2115	
	0.4050	0.2043	
	0.4086	0.2007	
Water	0.4910	0.1183	
	0.5233	0.086	0.086
	0.5054	0.1039	0.1039
Medicine	0.6093	0	0
	0.6666	0.0573	0.0573
	0.6667	0.0574	0.0574
Wash room	0.7240	0.1147	
	0.7419	0.1326	
	0.7814	0.1721	
Poser change	0.8387	0.2294	
	0.9391	0.3298	
	0.8495	0.2402	
Wheel out	0.4588	0.1505	
	0.4444	0.1649	
	0.4480	0.1613	

Table 3 Prediction accuracy of proposed method

Gesture class	No. of samples taken under each category	Prediction accuracy in % [proposed]
0	20	100
1	20	95
2	20	95
3	20	90
4	20	100
5	20	90
	Overall %	95%

Table 4 Comparison of prediction accuracy with state-of-the-art methods

Gesture class	Kinect sensor (Oudah, 2020)	Depth threshold (Oudah et al., 2021)	Kinect v2 embedded system (Oudah et al., 2021)	SCNN, depth metadata (Oudah et al., 2021)	Proposed method
0	100	100	94	98.46	100
1	93.18	84.62	96	100	95
2	95.45	80	94	100	95
3	86.36	76.92	96	81.54	90
4	93.18	75.38	96	98.36	100
5	100	81.54	–	–	90
Overall %	94.69	83.07	95.2	95.53	95

Table 5 Gestures usage prediction per day with χ^2 test

Class	O	E	O-E	(O-E) ²
Food	4	5	-1	1
Water	7	5	2	4
Medicine	3	5	-2	4
Wash room	7	5	2	4
Poser change	4	5	-1	1
Wheel out	1	5	-4	16
				sum = 30

Assume that the expected frequency of occurrence of each gesture as 5, and the actual frequency of occurrence of each gesture as given in Table 5

$$\chi^2 = \sum \frac{(o - E)^2}{E} \quad (8)$$

where o —observed frequency.

E —expected frequency.

$$= \frac{30}{6} = 5$$

Calculated $\chi^2 = 6$.

Tabulated χ^2 at 5% level of significance with degree of freedom $(n-1) = 5$ is 12.592.

since calculated $\chi^2 <$ tabulated χ^2 .

i.e., $6 < 12.592$, we accept H_0 .

The frequency of using each gesture by the disabled or elderly over a day is uniformly distributed.

5 Conclusion

The use of the Hough transform for feature extraction overcomes the limitations like the use of the right or left hand, change in orientation and deformation in hand shape

and provides accuracy almost equal to template matching. Since shape deformation is addressed by the proposed approach, the people who suffer from arthritis could also be benefited from the proposed approach. The prediction performance of 100% could be achieved if the system is trained and validated with personalized gestures of the hand shape of deformed people, as elderly people generally have finger shape deformation.

References

- Bhattacharya, S., Czejdo, B., & Perez, N. (2012). *Gesture classification with machine learning using kinect sensor data*. Third International Conference on Emerging Applications of Information Technology, pp. 348–351. IEEE.
- Coogan, T., et al. (2006). *Real time hand gesture recognition including hand segmentation and tracking*. International Symposium on Visual Computing, pp. 495–504. Springer.
- Gnanapriya, S., Rahimunnisa, K., Karthika, A., Gokulnath, M., & Logeshkumar, K. (2020) Indian sign language based static hand gesture recognition using deep learning. *Journal of Critical Reviews*, 7, 2805–2811.
- Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Tao, B., & Xu, H. L. (2017). Hand gesture recognition based on convolution neural network. *Cluster Computing*, 22, 2719–2729.
- Liao, B., et al. (2018). *Hand gesture recognition with generalized Hough transform and DC-CNN using realsense*. eighth International Conference on Information Science and Technology (ICIST), pp.84–90. IEEE.
- Lin, C.-S., et al. (2006). A PDA based wearable system for real-time monitoring of human falls. *IETE Journal of Research*, 52, 403–416.
- Machine Learning Mastery. <https://machinelearningmastery.com/distance-measures-for-machine-learning>
- Marshland, S. (2015). *Machine learning an algorithmic perspective* (2nd ed.). Chapman & Hall/CRC Machine Learning & Pattern Recognition Series.
- Mesbahi, S. C., Mahrzaz, M. A., Riffi, J., & Tairi, H. (2018). *Hand gesture recognition based on convexity approach and background subtraction*. International Conference on Intelligent Systems and Computer Vision, pp. 1–5, IEEE.
- Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. (2019) A modified-LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, 19, 7056–7063.
- Mohana, M., et al. (2020). An IOT based automated communication system for paralyzed patients using simple hand gestures. *European Journal of Molecular & Clinical Medicine* 7(4), 2681–2686.
- Munib, Q., Habeeb, M., Tahruri, B., & Al-Malik, H. A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Systems with Applications*, 24–37.
- Nixon, M., & Aguado, A. (2008). *Feature extraction and image processing* (2nd ed.). Academic Press.
- Onat, E., & Üzdil, U. (2015). *Traffic sign classification using Hough transform and SVM*. 23rd Signal Processing and Communications Applications Conference (SIU), pp. 2161–2165.
- Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand gestures for elderly care using a microsoft Kinect. *Nano Biomedicine Engineering*, 12, 197–204.
- Oudah, M., Al-Naji, A., & Chahl, J. (2021). Elderly care based on hand gestures using kinect sensor. *Computers*, 10.
- Patil, A. R., & Subbaraman, S. (2018). A spatiotemporal approach for vision-based hand gesture recognition using Hough transform and neural network. *Signal, Image and Video Processing*, 13, 143–421.
- Pinheiro, P. R. et al. (2020) Integration of the mobile robot and internet of things to monitor older people. *IEEE Access*, 8, 138922–138933.
- Rerkngamsanga, P. et al. (2016). *Generalized Hough transform for object classification in the maritime domain*. 11th System of Systems Engineering Conference (SoSE), pp. 1–6. IEEE.
- Sahoo, J. P., Ari, S., & Ghosh, D. K. (2018). *Hand gesture recognition using DWT and Fratio-based feature descriptor*. *IET Image Processing*, 12(10), 1780–1787.
- Sharma, P. (2020). Radhey Shyam Anand: Depth data and fusion of feature descriptors for static gesture recognition. *IET Image Processing*, 14, 909–920.
- Sheshkus, A., et al. (2017). *Combining convolutional neural networks and hough transform for classification of images containing lines*. Ninth International Conference on Machine Vision (ICMV 2016), Vol. 10341. International Society for Optics and Photonics.
- Sinha, K., et al. (2019). *A computer vision based gesture recognition using hidden markov model*. Innovations in Soft Computing and Information Technology, pp. 55–67. Springer.
- Sriram, K. N. V., & Palaniswamy, S. (2019). *Mobile robot assistance for disabled and senior citizens using hand gestures*. International Conference on Power Electronics Applications and Technology in Present 6 Energy Scenario (PETPES), pp. 1–6. IEEE.
- Tan, E. T., & Abdul Halim, Z. (2018). Health care monitoring system and analytics based on internet of things framework. *IETE Journal of Research*, 65, 653–660.
- Teimourikia, M., Saidinejad, H., Comai, S., & Salice, F. (2014). *Personalized hand pose and gesture recognition system for the elderly*. International Conference on Universal Access in Human-Computer Interaction, pp. 191–202. Springer International Publishing Switzerland.
- Vani, V., & Ananthalakshmi, S. R. (2020). Soft computing approaches for character credential and word prophecy analysis with stone encryptions. *Soft Computing*, 24, 12013–12026.
- Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. *Neural Computing and Applications*, 32, 7957–7968.
- Wan, B., Wu, R., Zhang, K., & Liu, L. (2017) *A new subtle hand gestures recognition algorithm based on EMG and FSR*. 21st International Conference on Computer Supported Cooperative Work in Design, pp. 127–132. IEEE.
- Zheng, Y., & Zheng, P. (2015). *Hand segmentation based on improved Gaussian mixture model*. IEEE International Conference on Computer Science and Applications, pp. 168–171.

A Review on Hand Gesture and Sign Language Techniques for Hearing Impaired Person

Safyzan Salim, Muhammad Mahadi Abdul Jamil, Radzi Ambar, and Mohd Helmy Abd Wahab

Abstract

Sign language has been used by deaf communities for over three centuries. It serves as a platform for messaging and communicating. New sign languages are emerging in deaf communities all over the world. Movement and orientation of hands, arms body and facial expression can be used to represent one's thought in sign languages. However, only a small percentage of public is aware of sign language. As a result, those who use sign language for everyday communication may have difficulty in communicating with normal people. Hearing aid devices have been introduced as a consequence of remarkable technology advancements to assist the hearing-impaired community in communicating with others. Hearing aid devices also would help individuals who have not entirely lost their hearing loss, while others who have hearing impairments will have to rely on sign language to communicate with one another. This paper will discuss review on hand gesture studies in providing sign language used in the hand gesture and sign language recognition process. It is hoped that this study may provide readers with a direction on the field of gesture and sign language recognition for further future work with regards to hearing impaired subjects.

Keywords

Hand gesture • Hearing impaired • Sign language • Recognition techniques • Smart health

Acronyms

2D	2 Dimensional
3D	3 Dimensional
API	Application Programming Interface
ARM	Advanced RISC Machine
ASL	American Sign Language
BLE	Bluetooth Low Energy
CNN	Convolution Neural Network
Faster R-CNN	Faster Region-based Convolution Neural Network
FRF	Random Forest Regression
IMU	Inertial Measurement Unit
IR 4.0	Industry Revolution 4.0
KCF	Kernelized Correlation Filters
KNN	K-nearest Neighbor
LMC	Leap Motion Controller
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
R-CNN	Recursive-Convolutional Neural Network
RISC	Reduced Instruction Set Computer
RNN	Recurrent Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
SDK	Software Development Kit
sEMG	Surface Electromyography
SL	Sign Language
SLR	Sign Language Recognition
SVM	Support Vector Machine
VGG	Visual Geometry Group

S. Salim · M. M. A. Jamil (✉) · R. Ambar · M. H. A. Wahab
Biomedical Engineering Modelling and Simulation Research Group, Department of Electronic Engineering, Faculty of Electrical And Electronic Engineering, Universiti Tun Hussein onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: mahadi@uthm.edu.my

1 Introduction

Research on hand motion recognition has been a popular research topic since two decades ago (Hirota & Tagawa, 2016; Kaur & Rani, 2016). Nevertheless, this topic of research is still

relevant because of the abundance of data derived from the measurement of human body motions using various equipments such as cameras and smart sensors (Chan et al., 2015; Ciotti et al., 2016; Lu et al., 2014; Zhou et al., 2014; Jeong & Cho, 2016). Researchers are focusing on developing controller that utilizes hand motions to control electronic devices. An area where hand motion recognition is much needed is sign language recognition. Similar to normal person, in order to live independently, deaf people need to communicate and interact with normal people. Therefore, an efficient gesture recognition system can significantly improve their quality of life.

To help with deafness and other communication disorders, there are different types of off-the-shelf hearing aid devices available, including behind-the-ear, in-the-ear and canal aids. Although hearing aid devices are beneficial, users may face issues such as discomfort and hearing background noise when using them.

As a result, experts have been working on a variety of ways for translating sign language motions. In general, two common ways are known: vision-based systems and wearable gadgets. To identify hand and finger movements, vision-based systems use image processing techniques such as feature extraction.

Several experiments have been conducted on sign language translation utilizing a vision-based system. Vision-based systems have the advantage of not requiring users to be tied to sensory devices, which can be messy and inconvenient. However, vision-based systems are challenging to design since establishing algorithms for feature and movement needs complicated and expensive computations.

2 Findings on Hand Gesture and Sign Language Studies

This paper compiled selected works done by fellow researchers between the years of 2016–2021 on hand gesture and sign language. It covers type or mode of operation, algorithm applied, methodology and hardware used. The review is separated based on type of study, i.e., vision-based approach, sensor-based approach and other approach as well.

2.1 Vision Based Study

A high number of visual sequences in sign language will contribute problem to accuracy and stability of hand locating in sign language recognition. According to He (2019), this is due to the effect of light and complex background. In this study, the author applied Faster R-CNN and 3D CNN as the solution to the mentioned.

FR-CNN is used in order to detect either the sign language video or part of hand in a picture. It integrates the

RPN module for better performance in terms of locate gesture effectively even though facing interference of skin color and background, movement blurs, and hand occlusion. 3D CNN is used because of its ability in improving the feature extraction and classification of sign language accuracy.

Ahmed et al. (2016) has developed a system that not only would be able to function as sign-to-speech engine but as speech-to-sign engine too, a sign language to speech conversion system. It is a software-based solution that used Microsoft Kinect 2.0's Continuous Gesture Builder to record and train the hand gesture. The input from Kinect will be compared with the pre-recorded dataset and for each matched gesture. Sentences comes from the matched gestures before being into speech. The Visual Gesture Builder from Kinect's SDK, which uses AdaBoostTrigger and RFRProgress detection technologies, was used to train gesture detection databases (Gonçalves et al., 2015).

During gesture recording session, information such as 3D coordinate points, depth, body heat and infrared scan were gathered for tagging purposes. The gesture will be stored at gesture database and will be used as a data dictionary to compare the incoming gesture. Any incoming gesture that matches with the gesture in database, the keyword of the gesture is added in the sentence before being speak out by the computer.

Tamiru et al. (2018) proposed a technique on controlling of a vision-based wireless mobile robot hand using hand gesture. A set of 100 hand gesture images has been recorded and it will be compared with the input in real time. The noise has been removed by applying median filtering together with histogram equalization to match with the output image with the database images. Even though it has went through the segmentation process, it was found out that the result was not satisfied. A morphological filtering technique was used to get a clearer contour of the hand image before the gesture was able to be recognized. This step produces a better efficiency. The final step to recognize the hand sign is to apply the 2D cross-correlation coefficient technique. Then the robot will be able to identify the direction.

Chung et al. (2019) used a webcam to instantly track the hand region of interest (ROI) in order to identify the hand gesture. The hand detection block will do the tasks such as skin color detection method and morphology to remove the unnecessary background information from the hand image. By applying background subtraction, the author will be able to detect the ROI of the hand.

Next process is to monitor the observed ROI to avoid background influences on artifacts or noise affecting the ROI by applying Kernelized Correlation Filters (KCF) (Oh et al., 2017). The final process is to apply the deep Convolution Neural Network (CNN) (Ji et al., 2017) technique, mainly AlexNet (Alom et al., 2019) and VGGNet (ElBadawy et al., 2017) to identify the multiple hand image. The process of

tracking and recognition is carried out continuously in order to achieve an instantaneous effect until the hand leaves the camera range.

In 2016, Naglot and Kulkarni (Naglot & Kulkarni, 2016) proposed a system for ASL recognition using Leap Motion Controller (LMC). The normalized training images are passed through feature extraction where only finger and palm dataset will be used for classification process (Kajan et al., 2015). The author used multi-layer perceptron (MLP) algorithm for gesture recognition and backpropagation is chosen for training. The proposed method has shown that the recognition rate is 96.15%.

Fasihuddin et al. (2018) reported on limited tools and resources on Arabic Sign Language (ArSL) assistive learning tools for hearing impairment people. The author also used LMC for capturing hand gesture. The author applied k-nearest neighbor (KNN) algorithm which provides the highest accuracy of sign recognition compared with other algorithms.

Kolivand et al. (2021) presented a faster method that improved the accuracy of SLR system and resulting faster sign language recognition even with plain background or clutter background. The author used a three-dimensional depth-based sensor camera as the input since it is capable to capture depth-based image and scan the hand pose at the same time. The depth-based image carries many information producing derivation of many methods.

The author also applied K-fold cross-validation during training phase in order to produce high accuracy of results over large amount of dataset. Furthermore, the validation and training processes can be used for this procedure.

Bantupalli and Xie (2018) outlined the usage of a smartphone to record multiple different hand gesture as the dataset. The author applied two different approaches to classification: the prediction from Softmax Layer and the output of the global Max Pooling layer (Tripathi et al., 2015) splits the dataset into segments extract features and classifies using Euclidean Distance and K-Nearest Neighbors. This is to overcome issues when applying Neural Network to segmentize the video.

The smartphone's camera is used again for gesture detection process. Using the video Inception model is applied to extract spatial features from the video stream for SLR. The proposed method managed to clock 99% of accuracy when applying inception method. The only drawback of the design is, that the system cannot be tested with different facial features and skin tone without having train.

2.2 Sensor Based Study

O'Connor et al. (2017) presented a gesture-tracking-glove attached with strain sensors, 6 degree of freedom micro-electro-mechanical-sensor and microcontroller. The

system is not only able to translate American Sign Language alphabet into text on a computer or over a smart phone, it is also capable to communicate with virtual hand. The low cost piezoresistive sensors measures the knuckle later converted into voltages. The microcontroller will process them and mapped the voltages into series of binaries. Each binary key represents a specific ALS. The translated ASL is transmitted to the smartphone via BLE.

Wu et al. (2016) implemented inertial measurement unit (IMU) and sEMG sensors in their multi-model wearable ASL recognition system. A large set of well-established features were extracted from the sensors during training phase. The associated measurements were put in one feature vector before fed into information gain to produce the optimum features subset. This study has tested Decision Tree, Support Vector Machine (LibSVM), Naïve Bayes and Nearest Neighbor classifiers to the selected subset. LibSVM scored the highest accuracy with 96.16% and has shown a consistent result (95.16%) to the prior study.

Lee et al. (2020) model achieved 99.81% on average recognition rate for dynamic hand gesture. The smart wearable system consists of 6 IMU which applied to fingers and back palm for capturing its movement during sign language gestures. The information from the IMUs were passed through the preprocessing segment for noise removal, feature extraction and normalization. Then, the author adopted the recurrent neural network (RNN) with the LSTM layer in classifying the gestures.

In 2017, Patil (2017) extended the application of biomedical signal to a hand gesture recognition and controlling system. The surface electromyography (sEMG) that attached to both hands will emit nerve value than processed by the Arm microcontroller that maps with the preset gesture value. The captured muscle movement is channeled to conditioning circuit (Haroon & Malik, 2016) to smooth the output and coupled with ARM microprocessor to recognize the input and Visual Basic to display the output, the author is able to produce desired output.

2.3 Hybrid Approaches

Hafit et al. conducted a study on developing mobile apps that will translate the Malaysian sign language into text. The apps were developed by using native camera plugin, Ionic Framework, Angular Firebase and Google Cloud Vision API. When the image is captured from the smartphone camera, it will be uploaded to Firebase storage. The API will detect the contents of the image. It will then map to the text that match with the image and display back at the smartphone. The authors have to upload samples of MSL to Firebase database that will be mapped with the image

uploaded to the Firebase. This app was specifically designed for normal people to communicate with the disabilities.

Haron et al. mentioned in her paper that hearing impaired individuals should deserve the same right with normal people in using mobile e-learning apps. Thus, she and her team developed mobile apps that's specifically designed to the deaf and dumb in learning MSL.

Paragon et al. made use of a publicly ASL dataset that consist of more than 2000 images from Barczak et al. (2011) and fed directly to machine learning algorithm for training. The images in the dataset consists of hand cut out with black background. The works continue by applying Convolutional Neural Network algorithm. The design incorporates two bunches of fully connected layers followed by a dropout layer and one final output layer, as well as four groups of two convolutional layers followed by a max-pool layer and a dropout layer.

3 Discussion

The focus of our review in this section is to further discuss a few important aspects that we strongly believe may contribute to this review.

Figure 1 illustrated how the training procedure of hand gesture recognition process is conducted. It starts from the input signals that consists of hand gesture of letter spell, number spell and vocabulary. Next step is, selecting the hardware for capturing hand gesture movement. Follows by choosing the appropriate machine learning algorithm for the study. This section requires with a large amount dataset for training purpose because, poor approximation is caused by not enough training data. In order to achieve good performance, the system requires a proper tune.

3.1 Image Acquisition Devices

Table 1 summarizes previous research on hand gestures and sign languages recognition. This study is divided into three domain of image capturing methods, namely, vision, sensors and hybrid. (Ahmed et al., 2016; Bantupalli & Xie, 2018; Chung et al., 2019; Fasihuddin et al., 2018; He, 2019; Kolivand et al., 2021; Naglot & Kulkarni, 2016; Tamiru et al., 2018) have applied camera-based approach as the image processing technique in recognizing hand gesture and sign language. The researchers implemented varieties of image capturing devices such as still camera, video camera, web cam, Kinect, and leap motion controller in performing image requisition of hand gesture activity. A special laboratory setup needs to be considered because most of the cameras have issue with surrounding.

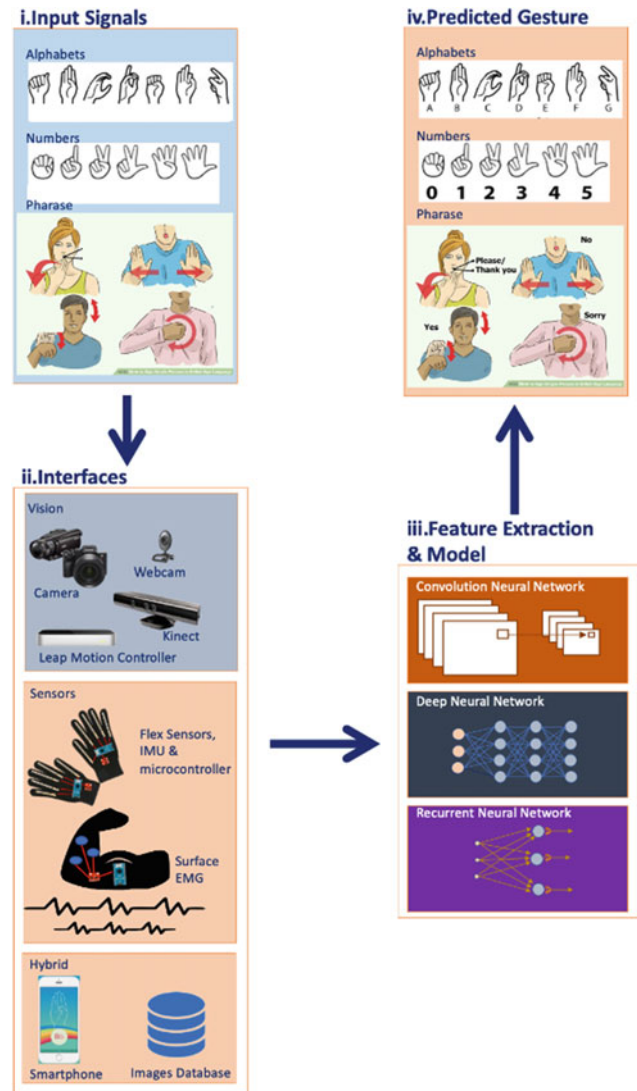


Fig. 1 Block diagram of hand gesture and sign language process

Hand gesture and sign language also can be captured through wearable sensors including flex sensors, inertial measurement unit and surface electromyography (Lee et al., 2020; O'Connor et al., 2017; Wu et al., 2016). Sensors must be connected to microcontroller. The flex sensors records every digits of fingers' adduction and abduction data, the IMU document the palm's position and orientation information and the sEMG reads the electrical activity of wrist's muscles. Normally, these sensors including the microcontroller are wired together into a sensory glove or smart glove to make it portable and ease of use.

Hybrid methods differ with vision-based system and sensor-based system in the sense of capturing devices, ANN engine and image dataset. Hybrid domain deals with method that uses either smartphone (Das et al., 2020; Haron et al., 2019) or cloud-based services such as Firebase or Google

Table 1 Comparison of this study with the related publications

ID	Year	Focus	Domain	Interface	Accuracy (%)
He (2019)	2019	– By applying Faster R-CNN which improved the accuracy of locating hand and 3D CNN is applied to determine the feature extraction and classification of SL	Vision	Camera	99.0
Ahmed et al. (2016)	2016	– Adaptive Boost (ML algorithm): true Boolean value while the subject is performing a particular gesture, – RFRProgress (Random Forest Regression ML algorithm): generates a non-stop result that produce analog data while the subject is performing a gesture allowing the system to determine how much the gesture is completed and the rate at the appropriate frame of the gesture	Vision	Kinect	99.0
Tamiru et al. (2018)	2019	– Histogram, Otsu algorithm, cross-correlation coefficient	Vision	Webcam	95.2
Chung et al. (2019)	2019	– Using skin color detection and morphology to get a good hand's ROI – 2 deep CNN, AlexNet and VGGNet to repeatedly track and recognize hand gesture	Vision	Webcam	95.61
Naglot and Kulkarni (2016)	2016	– Capture sign language using Leap Motion Controller & apply neural network to recognize the signs	Vision	LMC	96.2
Fasihuddin et al. (2018)	2018	– Smart tutoring system using Leap Motion Controller to detect hand tracking	Vision	LMC	Not Specified
Kolivand et al. (2021)	2021	– Introduces a novel hand posture identification system based on a neural network that extracts geometrical features from hands for the American sign language alphabet	Vision	Kinect	96.8
Bantupalli and Xie (2018)	2018	– Inception Network to develop Convolution Neural Network for recognizing spatial features, – Long Short-Term Memory (LSTM) model (one Recurrent Neural Network model) to train on temporal features – For classification—Softmax function—for prediction & Pool layer—output	Vision	Smart-phone Camera	93.0
O'Connor et al. (2017)	2017	– Teensy will map the data from the sensors and match with pre-set data	Sensor	sEMG	Not Specified
Wu et al. (2016)	2016	– K16A large set of well-established features from IMU & sEMG during training phase – Tested 4 classifiers: Decision Tree, Support Vector Machine (LibSVM), Naïve Bayes and Nearest Neighbor. LibSVM outperformed the rest	Sensor	IMU Piezo-resistive	96.2
Lee et al. (2020)	2020	By applying Faster R-CNN which improved the accuracy of locating hand and 3D CNN is applied to solve the feature extraction and classification of sign language	Sensor	IMU sEMG	98.8
Patil and Patil (2017)	2017	– The signal from sEMG will be compared with the pre-captured signal for specific gesture	Sensor	sEMG	Not Specified
Hafit et al. (2019)	2019	– A Malaysian Sign Language mobile learning apps designed for normal people to learn sign languages before communicating with the deaf. Uses Cloud Vision API for gesture detection on images captured by the phone	Hybrid	Smart-phone	87.0
Haron et al. (2019)	2019	– An app that only focuses on teaching the disabilities learning sign language – No input, just output	Hybrid	Smart-phone	Not Specified
Das et al. (2020)	2020	– Using MU Hand Images ASL image dataset created by Barczak, A.L.C. et al. for training and Kinect for validation	Hybrid	Images database	94.3

Collab (Barczak et al., 2011; Das et al., 2020). This configuration is the least cost compared to vision and sensor system.

3.2 Performance Metrics

The overall performance in this review achieved an efficient performance of face recognition. Both sensor and vision-based studies achieved an efficient performance of hand validation (greater than 90%). There are a few factors that contribute to the achievement when analyzing gesture recognition algorithms. Factors including the amount of classes the algorithm can recognize, as well as its tolerance to noise and complex situations, should be taken into account. Not to mention the performance of hardware and software components like as CPUs, graphics cards and compilers also play a vital role in ML (Pisharady & Saerbeck, 2014).

Series of formula and equations applied by to ensure that they will achieve their research objectives and produces high accuracy results. Some researchers applied statistical formula during data preprocessing toward the raw input data as to filter out the unnecessary information. (Lee et al., 2020) for instance, has applied mean (μ) and standard deviation (σ) deviation to its raw input to filter/omit specific data pattern and finding average distribution for a specific time frame. Both equations are illustrated in Eqs. (1) and (2), where N is total number of data and x is sensor data:

$$\sigma = \sqrt{\frac{1}{N} \left(\sum_{i=1}^N (x_i - \mu)^2 \right)} \quad (1)$$

$$\mu = \frac{1}{N} \left(\sum_{i=1}^N x_i \right). \quad (2)$$

A vision-based study conducted by Tamiru et al. (2018) has adapted the cross-correlation coefficient (γ) where Eq. (3) was used to verify the hand gesture recognition. The difference of two time series will be measured with possibility range between -1.0 and $+1.0$. A near to 1 resulting an identical gesture image between captured and trained image.

$$\gamma(x, y) = \frac{\sum_s \sum_t \delta_{I(x+s, y+t)} \delta_T(s, t)}{\sum_s \sum_t \delta_{I(x+s, y+t)}^2 \delta_T(s, t)} \quad (3)$$

where $\delta_{I(x+s, y+t)} = I(x+s, y+t) - I'(x, y)$,

$$\delta T(s, t) = T(s, t) - T', \frac{1}{N}$$

$$s \in \{1, 2, 3, \dots, p\},$$

$$t \in \{1, 2, 3, \dots, q\},$$

$$x \in \{1, 2, 3, \dots, m-p+1\},$$

$$y \in \{1, 2, 3, \dots, n-q+1\},$$

$$I'(x, y) = \frac{1}{pq} \sum_s \sum_t I(x+s, y+t)$$

3.3 Sign Language Dataset

Like spoken languages, sign languages are also unique because it developed by specific groups of people to allow the deaf to interact with each other. Even though, there are countries who share the same spoken language, it does not mean that they share the same sign language. English itself, has three different sign languages: American Sign Language, British and New Zealand Sign language and Australian Sign Language (Sign Language Alphabets from Around the World, <https://www.ai-media.tv/sign-language-alphabets-from-around-the-world>). Nevertheless, on the differences, the aim of having sign language is to ensure that everybody can communicate with others. When we read further and compare pattern or shape, we realized that each letter carries different finger stroke. The differences of sign language pattern and vocabularies is shown in Fig. 2. Meanwhile, Table 2 indicates in detail specific area of each researchers work.

Figure 3 shows American is the prominent language, followed by Malaysian and Arabic. However, in this review, only one article did not mention any specific language. ASL gained its popularity due to huge dataset amount available compared to other languages.

According to Baker (Baker, 2010), fingerspelling is referred to alphabets while sign language will reflect to vocabulary. She also mentioned that the deaf people have to learn fingerspell spell first then follows by vocabulary. For example, when teaching the word, car, ones will do: either *fingerspell C-A-R*; or *sign CAR*.

4 Hand Gesture and Sign Language Prospect in Smart Health Aspect

Industry Revolution 4.0 (Zaidi & Belal, 2019) has tremendously influenced the perspective of hand gesture recognition system. More electronics and electrical devices for instance, smart television, have implemented camera, radar, and radio frequency as their human computer interface unit (Ahmed et al., 2021; Yassen & Jusoh, 2019). Just by waving your hand, you can change the channel, change screen and control the volume.

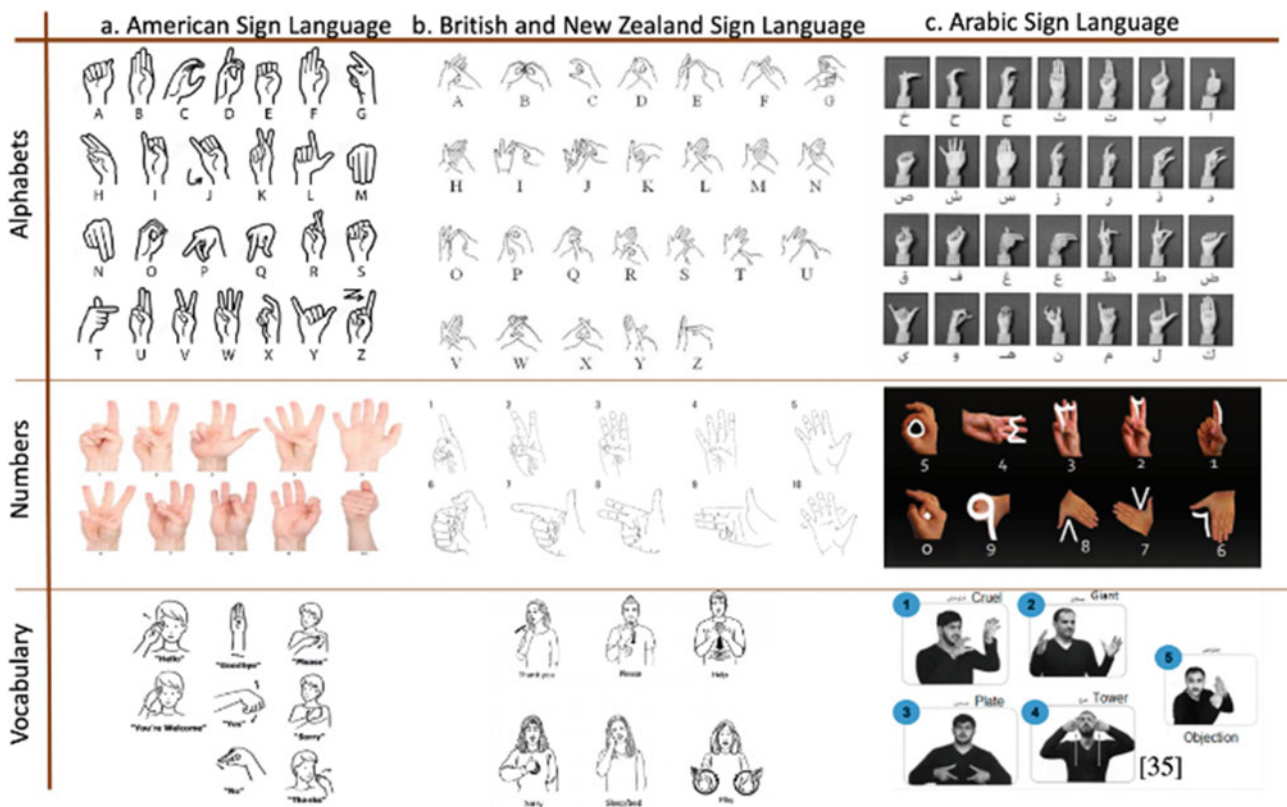


Fig. 2 Samples of sign languages from different countries

Table 2 List of sign languages referred by the authors in this study

ID	Language	Focus
He (2019)	American	Custom
Ahmed et al. (2016)	American	Vocabulary
Tamiru et al. (2018)	American	Direction OE8; Stop, Forward, Left, Right, Backward
Chung et al. (2019)	Unspecified	Custom
Naglot and Kulkarni (2016)	American	Alphabets
Fasihuddin et al. (2018)	Arabic	Alphabets, Numbers, Vocabulary
Kolivand et al. (2021)	American	Alphabets
Bantupalli and Xie (2018)	American	Custom
O'Connor et al. (2017)	American	Alphabets
Wu et al. (2016)	American	Custom
Lee et al. (2020)	American	Vocabulary
Patil and Patil (2017)	American	Custom OE8; Close, Open, Index, Middle, Relax
Hafit et al. (2019)	Malaysian	Alphabets
Haron et al. (2019)	Malaysian	Alphabets
Das et al. (2020)	American	Alphabets, number

In healthcare perspective, quite a number of medical devices make full use of the fourth industrial revolution framework. The usage of wearable sensors (Brezulianu et al., 2019; Jones et al., 2020; Le et al., 2019) shows the ability in

controlling devices from a different geographical location (Fig. 4). Medical doctors can diagnose and perform surgery within his office at one part of the world while, the patient is at the other side of the world (Lee et al., 1999). This will save

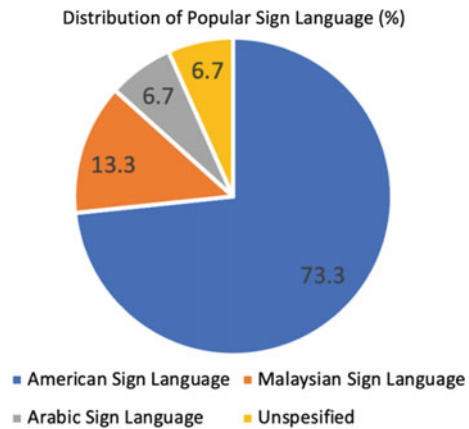


Fig. 3 Number of articles published based on sign language (Ahmed et al., 2016; Bantupalli & Xie, 2018; Chung et al., 2019; Das et al., 2020; Fasihuddin et al., 2018; Hafit et al., 2019; Haron et al., 2019; He, 2019; Kolivand et al., 2021; Lee et al., 2020; Naglot & Kulkarni, 2016; O'Connor et al., 2017; Patil & Patil, 2017; Tamiru et al., 2018; Tripathi et al., 2015; Wu et al., 2016)

cost, time, energy and secure both surgeon's and patient's security. Innovation in health telemetry system (Care & Today, <https://www.link-labs.com/blog/iot-healthcare/>) allows patient's data-gathering. By staying at home, and just wearing sensory glove, doctors can monitor one's vital signs.

5 Conclusion

The findings of this one-of-a-kind study will help to inform theories about translation, identity and well-being, as well as test a novel methodology for doing visual language research.

Parents of deaf children, sign language interpreters and hearing persons who work with Deaf sign language users, as well as deaf people themselves, will benefit from the findings. The ability to communicate with normal people by using the proposed device, provides limitless opportunities for the deaf people to search for jobs to improve their economy.

The revolution of industry era has changed the overview of industry itself. It focuses on internetworking of automation, machine learning and real time big data (What is Industry 4.0—the Industrial Internet of Things (IIoT), <https://www.epicor.com/en-my/resource-center/articles/what-is-industry-4-0/>). Eventually, healthcare industries has also benefited a lot from this revolution. Studies have been carried out in telemedicine application (Pasquale et al., 2018; Sima et al., 2020) on the efficiency and the ability of the health system delivering such a vital task.

The results of this paper can be summarized as the following: the surface electromyography sensors were the most acquisition tools used in the work studied and Artificial Neural Network deep learning approach was the most popular classifier. Vision-based approach is still popular compared to sensor-based and hybrid-based solutions.

In the future work, we are planning to use sensory glove as the input of the gesture and would like to apply different optimization technique so that we can speed up the model run time and achieve high training and validation accuracy.

Acknowledgements This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme for Research Acculturation of Early Career (FRGS-RACER) (RACER/1/2019/TK04/UTHM/5).



Fig. 4 Example of telemedicine operation, the remote arm is mimicking the operator's holding a screwdriver

References

- Ahmed, M., Idrees, M., Abideen, Z. U., Mumtaz, R., & Khalique, S. (2016). *Deaf talk using 3D animated sign language: A sign language interpreter using Microsoft's Kinect v2*. SAI Computing Conference, pp. 330–335.
- Ahmed, S., Kallu, K. D., Ahmed, S., & Cho, S. H. (2021). Hand gestures recognition using radar sensors for human-computer-interaction: A review. *Remote Sensing*, *13*, 527. <https://doi.org/10.3390/rs13030527>
- Alom, M. S., Hasan, M. J., & Wahid, M. F. (2019). Digit recognition in sign language based on convolutional neural network and support vector machine. *International Conference on Sustainable Technologies for Industry*, *4*, 1–5.
- Baker, S. (2010) The importance of fingerspelling for reading research brief. *VL2 Integration of Research and Education*.
- Bantupalli, K., & Xie, Y. (2018). *American sign language recognition using deep learning and computer vision*. IEEE International Conference on Big Data, pp. 4896–4899.
- Barczak, A., Reyes, N., Abastillas, M., Piccio, A., & Susnjak, T. (2011). A new 2D static hand gesture colour image dataset for ASL gestures. *Research Letters in the Information and Mathematical Sciences*, *15*, 12–20.
- Brezulianu, A., Geman, O., Zbancioc, M. D., Hagan, M., Aghion, C., Hemanth, D. J., & Son, L. H. (2019). IoT based heart activity monitoring using inductive sensors. *Sensors*, *19*, 3284. <https://doi.org/10.3390/s19153284>
- Chan, L., Hsieh, C.-H., Chen, Y.-L., Yang, S., Huang, D.-Y., Liang, R.-H., & Chen, B.-Y. (2015). *Cyclops: Wearable and single-piece full-body gesture input devices*. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3001–3009.
- Chung, H.-Y., Chung, Y.-L., Tsai, W.-F. (2019). *An efficient hand gesture recognition system based on deep CNN*. IEEE International Conference on Industrial Technology (ICIT), pp. 853–858.
- Ciotti, S., Battaglia, E., Carbonaro, N., Bicchi, A., Tognetti, A., & Bianchi, M. (2016). A synergy-based optimally designed sensing glove for functional grasp recognition. *Sensors*, 1–17.
- Das, P., Ahmed, T., & Ali, M. F. (2020). *Static hand gesture recognition for American sign language using deep convolutional neural network*. 2020 IEEE Region 10 Symposium (TENSYP), pp. 1762–1765.
- ElBadawy, M. A., Elons, S., Shedeed, H. A., & Tolba, M. F. (2017). *Arabic sign language recognition with 3D convolutional neural networks*. Eighth International Conference on Intelligent Computing and Information Systems, pp. 66–71.
- Fasihuddin, H., Alsolami, S., Alzahrani, S., Alasiri, R., & Sahloli, A. (2018). *Smart tutoring system for Arabic sign language using leap motion controller*. International Conference on Smart Computing and Electronic Enterprise, pp. 1–5.
- Gonçalves, A. R., Gouveia, E. R., Cameirão, M. S., & Bermúdez I Badia, S. (2015). *Automating senior fitness testing through gesture detection with depth sensor*. IET International Conference on Technologies for Active and Assisted Living (TechAAL), pp. 1–6.
- Hafit, H., Xiang, C. W., Mohd Yusof, M., Wahid, N., & Kassim, S. (2019). Malaysian sign language mobile learning application: A recommendation app to communicate with hearing-impaired communities. *International Journal of Electrical and Computer Engineering*, 5512–5518.
- Haron, H., Samad, H., Diah, F. M., & Yusof, H. (2019). E-learning approach using mobile apps: Malaysian sign language for dumb and deaf. *International Journal of Advanced Research in Technology and Innovation*, *1*, 1–7.
- Haroon, N., & Malik, A. N. (2016). Multiple hand gesture recognition using surface EMG signals. *Journal of Biomedical Engineering and Medical Imaging*, *3*, 1–8.
- He, S. (2019). *Research of a sign language translation system based on deep learning*. International Conference on Artificial Intelligence and Advanced Manufacturing, pp. 392–396.
- Hirota, K., & Tagawa, K. (2019). Interaction with virtual object using deformable hand. *IEEE Virtual Reality*, 49–56.
- How IoT Medical Devices Are Changing Health Care Today. <https://www.link-labs.com/blog/iot-healthcare/>
- Jeong, U., & Cho, K.-J. (2016). A novel low-cost, Large curvature bend sensor based on a Bowden-Cable. *Sensors*, *16*, 1–20.
- Ji, Y., Kim, S., & Lee, K. (2017). *Sign language learning system with image sampling and convolutional neural network*. First IEEE International Conference on Robotic Computing, pp. 371–375.
- Jones, S. B. R., Kumar, N., & Paul, S. S. (2020). Health monitoring wearable glove. *International Journal of Engineering Research and Technology*, *13*(12), 4199–4205. ISSN 0974-3154.
- Kajan, S., Pernecký, D., & Hammad, A. (2015) Hand gesture recognition using multilayer perceptron network, trapped charged particles and fundamental physics.
- Kaur, H., & Rani, J. (2016). *A review: study of various techniques of hand gesture recognition*. IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems, pp. 1–5.
- Kolivand, H., Joudaki, S., Sunar, M. S., et al. (2021). A new framework for sign language alphabet hand posture recognition using geometrical features through artificial neural network (part 1). *Neural Computing and Applications*, *33*, 4945–4963.
- Le, T., Tran, T., & Pham, C. (2019). *The Internet-of-Things based hand gestures using wearable sensors for human machine interaction*. 2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp. 1–6. <https://doi.org/10.1109/MAPR.2019.8743542>
- Lee, B. R., Cadeddu, J. A., Stoianovici, D., & Kavoussi, L. R. (1999). Telemedicine and surgical robotics: Urologic applications. *Reviews in Urology*, *1*(2), 104–120.
- Lee, B. G., Chong, T.-W., & Chung, W.-Y. (2020). Sensor fusion of motion-based sign language interpretation with deep learning. *Sensors*, 1–17.
- Lu, Z., Xiang, C., Li, Q., Zhang, X., & Zhou, P. (2014). A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *Human-Machine Systems, IEEE Transactions*, *44*, 293–299.
- Naglot, D., & Kulkarni, M. (2016). *Real time sign language recognition using the leap motion controller*. International Conference on Inventive Computation Technologies, pp. 1–5.
- O'Connor, T. F., Fach, M. E., Miller, R., Root, S. E., Mercier, P. P., & Lipomi, D. J. (2017). The language of glove: Wireless gesture decoder with low-power and stretchable hybrid electronics. *PLoS ONE*, *12*, 1–12.
- Oh, J., Kim, B., Kim, M., Kang, S., Kwon, H., Kim, I., & Song, Y. (2017). Avatar-based sign language interpretations for weather forecast and other TV programs. *SMPTE Motion Imaging Journal*, *126*, 57–62.
- Pasquale, G., & Mastrototaro, L. (2018). Glove-based Systems for medical applications: Review of recent advancements. *Journal of Textile Engineering Fashion Technology*, *4*(3), 253–262. <https://doi.org/10.15406/jteft.2018.04.00153>
- Patil, N. M., & Patil, S. R. (2017). *Review on real-time EMG acquisition and hand gesture recognition system*. International Conference Electronics on Communication and Aerospace Technology, pp. 694–696.
- Pisharady, P. K., & Saerbeck, M. (2014). Gesture recognition performance score: A new metric to evaluate gesture recognition systems. In C. V. Jawahar, & S. Shan (Eds.), *Computer Vision—*

- ACCV 2014 Workshops—Revised Selected Papers (Vol. 9008, pp. 157–173). Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer-Verlag. https://doi.org/10.1007/978-3-319-16628-5_12
- Sign Language Alphabets from Around the World, <https://www.ai-media.tv/sign-language-alphabets-from-around-the-world>
- Sima, V., Gheorghe, I. G., Subić, J., & Nancu, D. (2020). Influences of the industry 4.0 revolution on the human capital development and consumer behavior: A systematic review. *Sustainability*, *12*, 4035. <https://doi.org/10.3390/su12104035>
- Tamiru, H. G., Ren, S., & Duan, H. (2018). *Vision-based hand gesture recognition for mobile service robot control*. 8th International Conference on Manufacturing Science and Engineering, pp. 48–55.
- Tripathi, K., Baranwal, N., & Nandi, G. C. (2015). *Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds*. International Conference on Advances in Computing, Communications and Informatics, pp. 2211–2216.
- What is Industry 4.0—the Industrial Internet of Things (IIoT)?, <https://www.epicor.com/en-my/resource-center/articles/what-is-industry-4-0/>
- Wu, J., Sun, L., & Jafari, R. (2016). A wearable system for recognizing American sign language in real-time using IMU and surface EMG. *IEEE Journal of Biomedical and Health Informatics*, *20*, 1281–1290.
- Yasen, M., & Jusoh, S. (2019). A systematic review on hand gesture recognition techniques, challenges and applications. *Peer Journal of Computer Science*, *5*, e218. <https://doi.org/10.7717/peerj-cs.218>. PMID: 33816871; PMCID: PMC7924500.
- Zaidi, M. F. A., & Belal, H. M. (2019). A preliminary study to understand the SMEs' readiness on IoT in Malaysia. *International Journal of Accounting, Finance and Business (IJAFB)*, *4*(19), 1–12.
- Zhou, S., Fei, F., Zhang, F. F. G., Mai, J. D., Liu, Y., Liou, J. Y. J., & Li, W. J. (2014). 2D human gesture tracking and recognition by the fusion of MEMS inertial and vision sensors. *IEEE Sensors Journal*, *14*, 1160–1170.



DriveSense: Adaptive System for Driving Behaviour Analysis and Ranking

Sankar Behera, Bhavya Bhardwaj, Aurea Rose, Mohammad Hamdaan, and M. Ganesan

Abstract

Abnormal drivers are individuals who drive above the speed limit, change speed suddenly, or change vehicle lateral position incessantly. Monitoring these abnormal driving behaviours in real-time is very crucial to improving driving safety. To improve drivers' awareness of their driving habits, and prevent potential car accidents, the need to consider a fine-grained monitoring approach is a must, which not only detects abnormal driving behaviours but also identifies specific types of abnormal driving behaviours, i.e. weaving, swerving, side slipping, fast U-turn, Turning with a wide radius and sudden braking. All six types of driving behaviours have their unique patterns on inertial acceleration. By studying the driving profile, necessary features to capture the abnormal patterns in driving behaviours are identified, such as acceleration and rotation along X, Y and Z directions, position of Gas Pedal, Light Status and Indicators (turn-light signal). Using these observations, machine-learning models are trained to perform real-time high-accuracy driving

behaviour monitoring. Abnormal driving errors centre on human mindsets; as such a national ranking system 'DriveScore' has been incorporated that provides tools and statistics for self-analysis, and the use of 'Reward and Reinforcement' approach for encouraging good driving behaviour. Two different datasets—Mendeley and Virginia were individually used to determine deviant driving patterns and conclude whether/not a car will crash, models trained achieved an accuracy of 90% and 98%, respectively. This allows us to predict whether the driving is safe or unsafe, and the probability of a crash. This paper presents an intelligent monitoring and ranking system for all kinds of vehicles.

Keywords

Driving behaviour detection • Rash driving • Machine learning • Smart cities • IMU sensor • Gazebo simulation • Autonomous vehicles • DriveSense • DriveScore

S. Behera (✉)

Department of Computer Science and Engineering, Indian Institute of Technology, Jammu, India
e-mail: 2020pct0067@iitjammu.ac.in

B. Bhardwaj · M. Ganesan

Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: bhavya1705@yahoo.com

M. Ganesan

e-mail: m_ganesan1@cb.amrita.edu

A. Rose

Department of Mechanical Engineering, T K M College of Engineering, Kollam, Kerala 91005, India
e-mail: aurearose100@gmail.com

M. Hamdaan

Department of Mechanical Engineering, Loyola-ICAM College of Engineering and Technology, Chennai, India
e-mail: mohammadhamdaan.22meb@licet.ac.in

1 Introduction

In the late 18th century the first automobile and motorcycle were invented. Since then, road safety has always been a challenge to the manufacturers and the government. These initial stage vehicles were less powerful and ran with lower speed and were equipped with less to no safety features. For over two centuries, automobile companies have achieved tremendous success in terms of active aerodynamic designs, speed, fuel efficiency, performance, power and well-equipped safety features. Airbags, Anti-Lock brakes, High strength windshields, Overspeed warnings, Seat belts are common in all vehicles. Moreover, self-driving cars and autonomous vehicles promise safer roads.

Despite the tremendous progress, even today modern vehicles fail to make roads a safer and a better means of commute. According to WHO, 1.3 million people die each

year as a result of road traffic crashes and this number has been progressively increasing. Most countries spend about 3% of their Gross Domestic Product, GDP on road accidents (<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>). Approximately 60% of the world's total vehicles are present in low-to-medium-income countries; and it is in these countries that 93% of the world's road-accident-related fatalities occur (<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>). Injuries associated with road traffic is a major cause of death for children and young adults aged between 5 and 29 years. Figures 1 (<https://www.statista.com/statistics/1023507/india-registered-vehicles-number/>) and 2 (<https://prsindia.org/policy/vital-stats/overview-road-accidents-india>), illustrate an increase in accidents with number of vehicles over time. In India since 2000, the number of motor vehicles registered has increased by 158% (<https://prsindia.org/policy/vital-stats/overview-road-accidents-india>). As of 2019, there were 1.4 billion motor vehicles in use, estimated by the US publisher Ward. With the increase in vehicles, high-speed chases and drunken driving have become the norm. Figure 3 (Yaqoob et al., 2019) illustrates the fundamental cause of accidents that happen due to rash driving (over speeding), drunk driving, bad weather and pedestrian faults. This prompts a need for a prognostic device that studies driving behaviour and provides necessary reinforcement. The result of this device can be used to dictate government policies, insurance claims and driver license testing.

The paper is organized as follows. In Sect. 2, we review some of the most critical challenges in emerging safe driving technologies. Section 3 presents our proposed method, i.e. DriveSense. In Sect. 4, we present the methodology of our work. The results of the proposed scheme on analyzing and detection of driving behaviour

Fig. 1 Registered vehicles India (<https://www.statista.com/statistics/1023507/india-registered-vehicles-number/>)

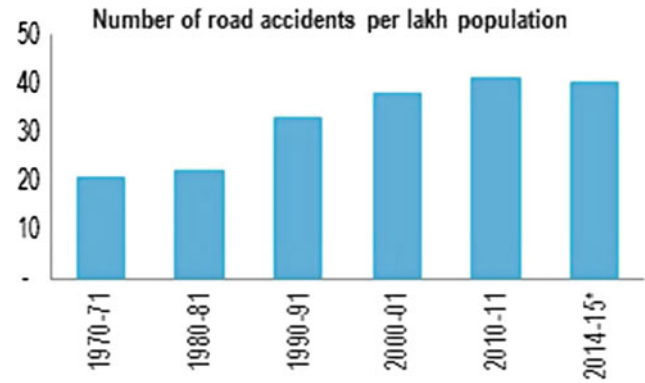
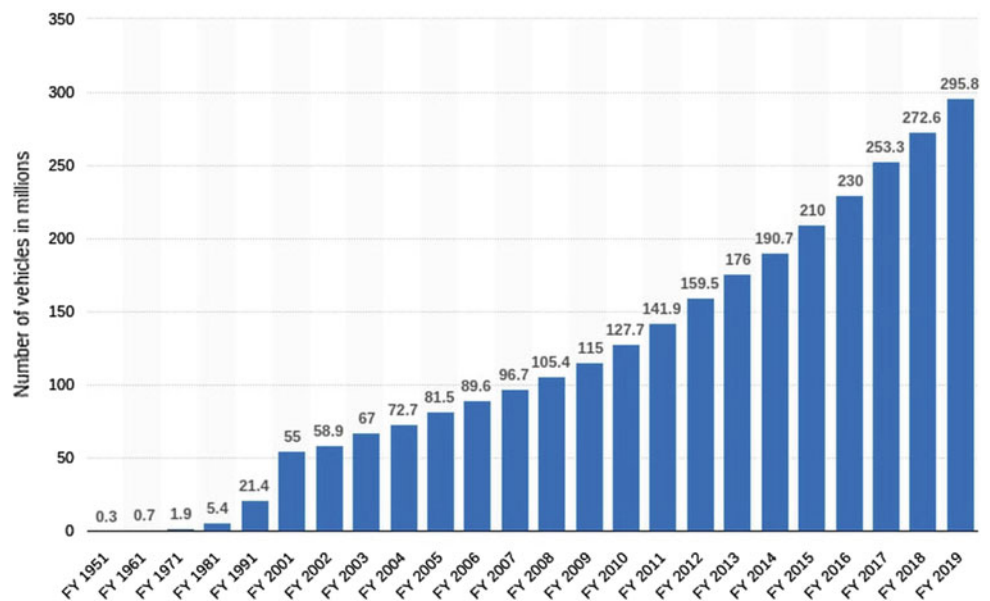


Fig. 2 Accidents (<https://prsindia.org/policy/vital-stats/overview-road-accidents-india>)

described in Sect. 7. In Sect. 8, we provide the simulation results. Section 9 discusses the final result and possible directions for further inspection. Finally, we conclude the paper in Sect. 10.

2 Autonomous Vehicles, Smart Cities and DriveSense

Autonomous Vehicles are vehicles that are capable of driving (in the presence of a pillion) themselves with the help of Computer Vision and AI. Autonomous Vehicles aim to reduce fuel consumption, reduce emissions, accidents and congestion; save time and space; and improve mobility for the people (Yaqoob et al., 2019), But how long would it take to replace all traditional cars? It is estimated that over 1.4 billion vehicles travel the streets and roads of the world today. In a speech at SXSW, 2016 in Austin, Google's car project director Chris Urmson explained that the day when



Fig. 3 Causes of accidents (Yaqoob et al., 2019)

fully autonomous vehicles are widely available, going anywhere those regular cars can, might be as many as 30 years away. Furthermore, autonomous vehicles have their own challenges (Yaqoob et al., 2019) and can be summarized as follows: creating detailed 3D-map of cities or towns by manually driving the test vehicles all over the streets, developing better algorithms to handle bad weather conditions, Radar Interference Management Systems, communicating between Heterogeneous vehicular networks and answering legal questions about how these cars will be insured, and who exactly will be liable—the driver or the manufacturer, in the event of a crash. Another big threat to driverless vehicles is cybersecurity—how to ensure a hack-free system. Till self-driving cars can make city roads safe for transmuting, third-party solutions need to provide the solutions. Planning of smart cities includes developing better transportation systems and networks which can augment the use of more autonomous and self-driving vehicles. However, the safety of the driver is still a major concern until driverless cars are common and easily accessible. Thus systems to improve driver safety remain immensely relevant in sustainable transport and urban planning.

Various methods have been proposed for driver safety. Onken et al. (1994) present an adaptable driving assistant system for monitoring driving behaviour and alerts the driver using haptic displays. Imkamon et al. (2008) consider unsafe driving behaviour to be a subjective quantity and developed a fuzzy inference system to predict the level of safe driving. Saiprasert and Pattara-Atikom (2013) proposed a method

availing use of smartphones to detect whether or not a vehicle has been speeding and records the driving journey data as evidence to report unsafe driving. Chen et al. (2015) proposed an economic system for driving behaviour detection and identification based on smartphones, thus saving installation costs of hardware. Wang et al. (2016) collected indicators for detecting drowsy driving behaviour using Random Forest (RF). RF showed superior preference over ANN (Artificial Neural Networks) and the 20s time window size datasets using the parameter combination of accelerations in lateral and longitudinal directions was concluded to be the most useful information to detect drowsiness. Chen et al. (2017) availed the use of video surveillance systems for developing a machine-learning model to detect dangerous driving behaviours. Haria et al. (2018) proposed a system to prevent car accidents on normal and curved roads using sensors and smart poles as well as a detection system to send an alert message in case the accident occurs. Kashevnik et al. (2019) developed a methodology to detect dangerous driving by monitoring through smartphone sensors and a built-in front-facing camera and further implemented a mobile application, evaluating the system in real-time. Kumar and Suja (2020) developed a warning system to detect, localize and classify obstacles in front of moving vehicles using LIDAR (Light Detection and Ranging) sensor and camera data to improve autonomous vehicle perception. This work will detail the working of DriveSense, an ML-based approach for classifying and studying driving behaviour. DriveSense can further be used to check the performance and safety of smart vehicles.

3 DriveSense

Driving is a complex process that involves a significant amount of human computation. There needs to be perfect coordination between the driver's body (eyes, ears, hands and legs) and mind. Although most drivers take it casually, it is the driver's responsibility to ensure their and public safety. Analyzing driving behaviour can help us assess and understand driver performances, improve traffic safety and promote the development of intelligent, smart and hardy transportation systems to enable many important applications, such as monitoring drivers, vehicles and roads, providing early warning and driving assistance and enhancing driving comfort and energy saving. In this chapter, we study the problem of learning to profile driving behaviour with applications related to transportation safety and the goal of providing a system that encourages good and efficient driving, hence promoting safer society, reduced accidents and limited emissions.

4 Methodology

In this work, we develop a platform, called DriveSense that comprises of five modules—onboard sensor data acquisition, machine-learning models, statistical analysis tools, ranking system and reinforcement on human intelligence (Fig. 4). Machine-Learning models are used for analyzing and detecting various abnormal driving behaviour. This helps in assessing and understanding driver performances, improve traffic and promotes the development of intelligent, smart and hardy transportation systems. Before training the ML models, we collect the sensor data so that is fed to the ML models. Figure 5 demonstrates the architecture of DriveSense. The data acquisition process, dataset study, implementation and design details of the architecture of the proposed system will be explained in the following section.

4.1 System Architecture

In this section, we explain the details of various modules of our proposed platform DriveSense, namely the onboard sensor data acquisition, machine-learning models, statistical analysis tools, ranking system and reinforcement on human intelligence. The onboard sensor data acquisition module is used for collecting raw driving data from the onboard sensors like the Inertial Measurement Unit (IMU) and Onboard diagnostics (OBD). This data is then pre-processed and cleaned. This clean data is then fed to the Machine-Learning Models, where they are used to train and reinforce the model.

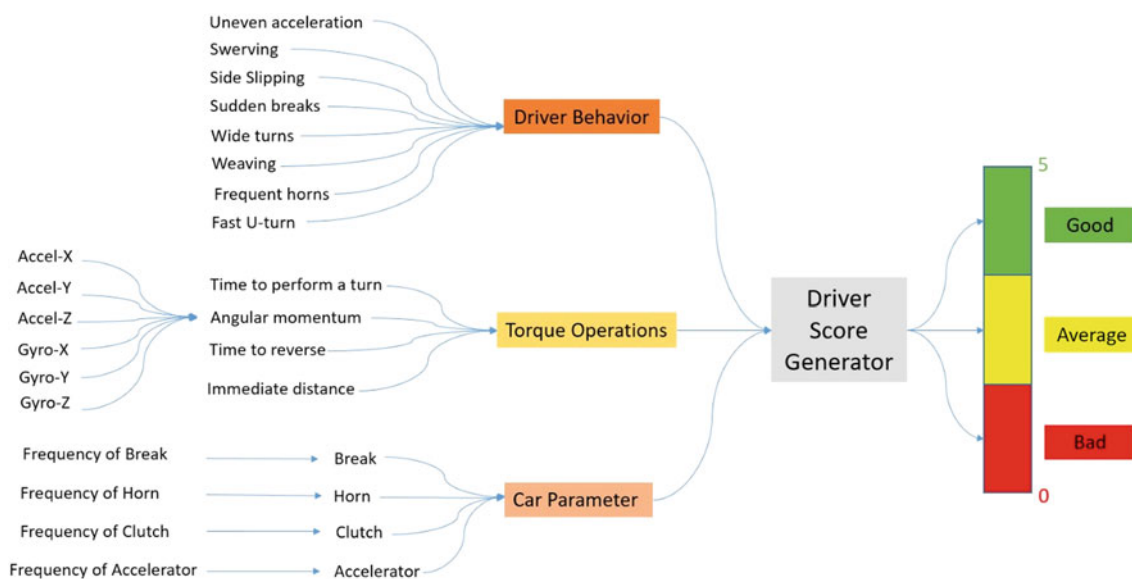


Fig. 4 Methodology

The trained models can predict the specific type of abnormal driving behaviour based on the real-time data. Abnormal behaviours like weaving, swerving, side slipping, fast U-turn, turning with a wide radius and sudden braking based on the real-time driving data. The Self-Analysis Tool comprising of the ranking system and reinforcement on human intelligence helps generate adaptive notifications and warnings for the driver by monitoring the statistics in real-time. These messages are designed to improve driver’s mindset towards driving. The national ranking system generates an adaptive DriveScore, which ranges from 0 to 5. Where towards 0 represents rash driving and toward 5 represents good driving skills of a person. The rate of change is DriveScore depending on the miles of distance covered by the vehicle.

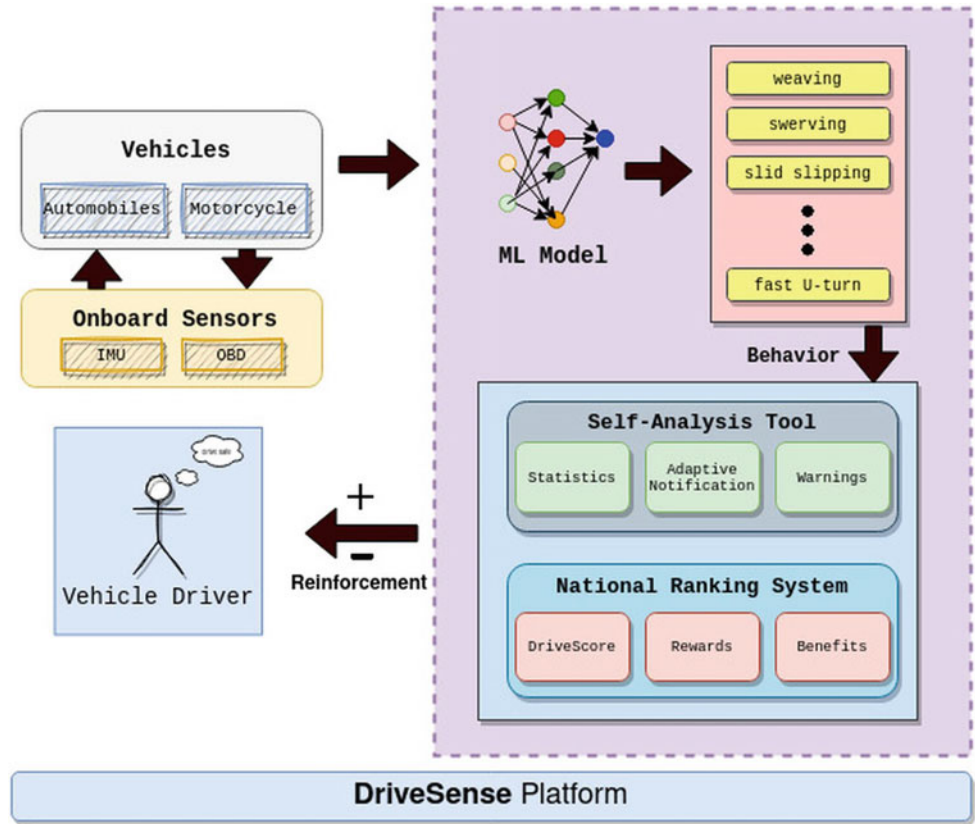
4.2 Machine-Learning Models and Algorithms

In the scope of this work, we will using five different Machine-Learning Algorithms namely:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Boosted Tree Classifier
4. Logistic Regression
5. SVM Classifier.

To better understand the working of these different Machine-Learning models we need to understand how they work.

Fig. 5 System architecture



4.2.1 Decision Tree Classifier

The Decision Tree is the basis of all Tree architectures, it can be understood as a sequence of conditional statements as in Figure 6. It is a commonly used architecture for classification problems. The nodes in a decision tree give the attribute/feature that is being evaluated, the branches represent the outcomes and the leaves give the output class.

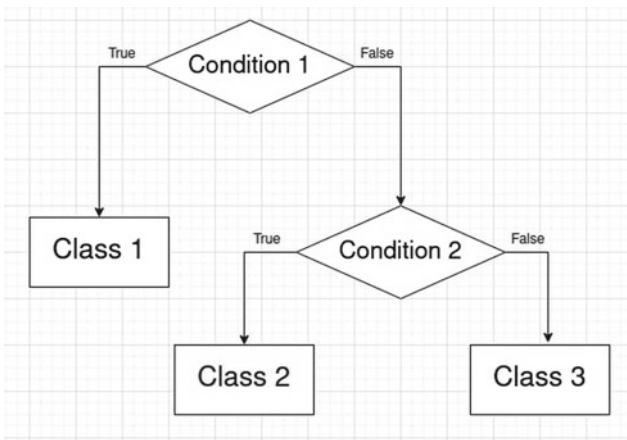


Fig. 6 Decision tree

4.2.2 Random Forest Classifier

The Random Forest Classifier is an improvement on the Decision Tree Architecture. In Random Forest the Dataset is split into small groups with corresponding labels and then using majority votes or weighted output to get the required class. Let’s consider a 4-class problem (<https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1>) with data [X1, X2, X3, X4] and corresponding classes [L1, L2, L3, L4]. A random classifier may split this as [X1, X2, X3], [X2, X3, X4], [X3, X4, X1] to train decision trees. The final output obtained would be the aggregate output of all trees.

4.2.3 Boosted Tree Classifier

The Boosted Tree Classifier is again an improvement on the basic Decision Tree Architecture. It uses boosting, i.e. it uses consecutive trees to improve and reduce the errors of the past trees. Since the initial trees may be poor learners of attributes, consecutive trees can learn and improve performance.

4.2.4 Logistic Regression

The Logistic Regression is a form of regression model that passes the regression output through a logistic function to obtain the binary class output. For multi-class classification, a group of logistic classifiers can be used concurrently.

Mathematically, $y = \sum_0^n \alpha_i * x_i + \beta$ would be the output of the regression model, where α are the weights, β the bias and x the features of the dataset. To use this result for classification purposes, the result y is passed through a logistic function, which gives us the distinctive S shape curve and result y' . The output y' is either 0 or 1, which are used as labels for the classes.

4.2.5 SVM Classifier

The SVM classifier is a simple classification algorithm that generates a hyperplane that is used to distinguish between different classes of data. It tries to obtain the best shape that can be used to segregate the classes as shown in Figure 7 (<https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>).

5 Data Acquisition of Driving Behaviour

In order to analyze and rank driving behaviour in general, we must first gather the data to perform the analysis. The acquired data needs to be relevant and must be potent enough to provide insights on the efficiency of a driver's ability to control a vehicle. Therefore, a plethora of necessary sensors must be placed inside the body of a vehicle to facilitate data acquisition. All existing sensors that could be embedded in a modern car are displayed in Figure 8 (Guerrero-Ibáñez et al., 2018).

5.1 Use of Sensors to Determine Driving Behaviour Traits

In order to generate a rating for the user, continuous monitoring of driving activities is essential. The rating system can also be employed on self-driving cars to rate their efficiency in driving. As cities get smarter, car services are going to be

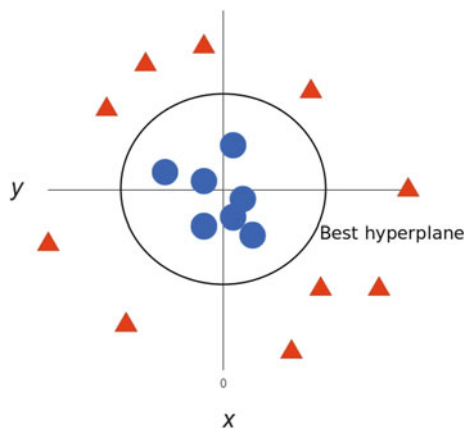


Fig. 7 Classification using support vector machine

more prominent and a rating system ensures trust in the vehicle, be it autonomous or driven by an individual. Data acquired from sensors can be used to assess driving behaviour in real-time. The use of proximity sensors can determine if the vehicle is going to be in collision with another vehicle. Practices like tailgating are unprofessional and increase the chances of a collision, therefore a threshold distance is to be constantly maintained between two vehicles. Crossing the threshold value indicates bad driving but scores will be adjusted based on the traffic around the immediate area. Frequent honking is proof of impatient driving and causes distress to nearby vehicle drivers too. If the driver uses the horn continually for a certain time frame, then the model first sends an alert to the driver. Further repetition will reduce the driving score. Similarly, the inefficient use of brake pad and clutch is also an indicator of driver stress and our model monitors if they are used suddenly or smoothly. The use of seatbelts is of utmost importance to ensure maximum protection in case of an accident. Cars give a warning when seat belts aren't used and if the warning is ignored, it will be negatively reflected in the driving score. It is also important to note that the introduction of Connected Automated Vehicles (CAV) has achieved a significant milestone in the use of onboard sensors to prevent accidents, thereby improving driving safety. With the help of sensors, when a collision is imminent, split-second decisions can be made by the vehicle that facilitates the best possible course of action.

5.2 Detecting Rash Driving Behaviour Using IMUs

The features of the datasets used for validating the ML model majorly correspond to the data coming from IMU sensors. In this section, we describe how rash driving behaviour patterns can be extracted from IMU data.

From (Chen et al., 2015) we were able to identify the waveforms in acceleration associated in each type of rash driving behaviour and it is displayed in Figure 9 (Wang et al., 2016) below.

Figure 9 shows the acceleration and orientation patterns of the six types of abnormal driving behaviours from an accelerometer and an orientation sensor's readings (Chen et al., 2015). To detect rash driving behaviour, our model accesses the accelerometer and gyroscope data from the inbuilt sensors in smartphone devices or the data from the sensors built into the car device and catalogue it. If using the accelerometer and gyroscope inbuilt in phones to get the data it is essential to keep the smartphone fixed at one location in the car to get accurate readings. However, it was concluded in Chen et al. (2015) that the smartphone can be placed anywhere in the car and it doesn't affect the readings

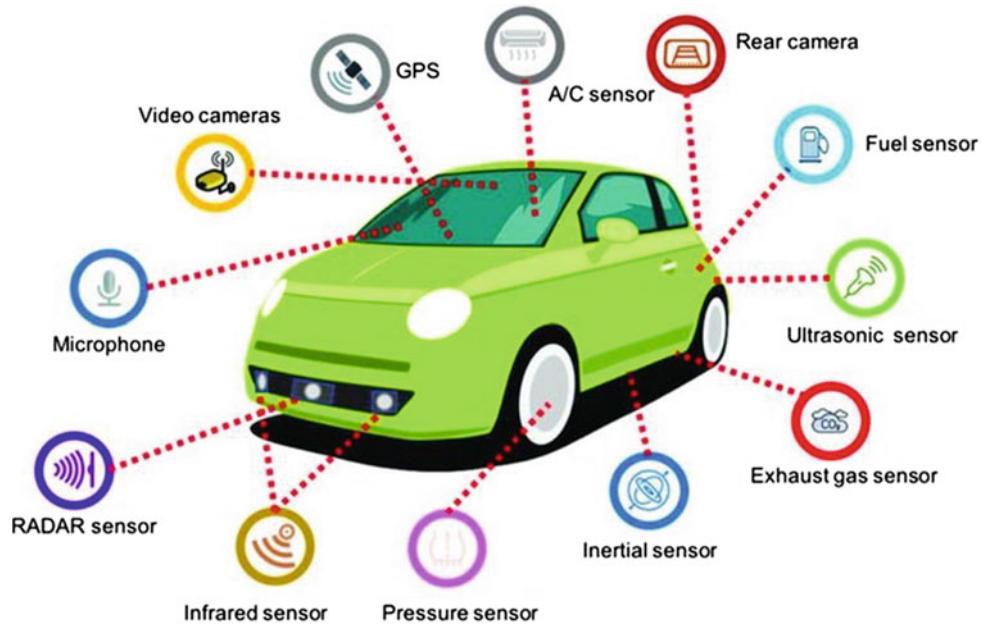


Fig. 8 Sensors (Guerrero-Ibáñez et al., 2018)

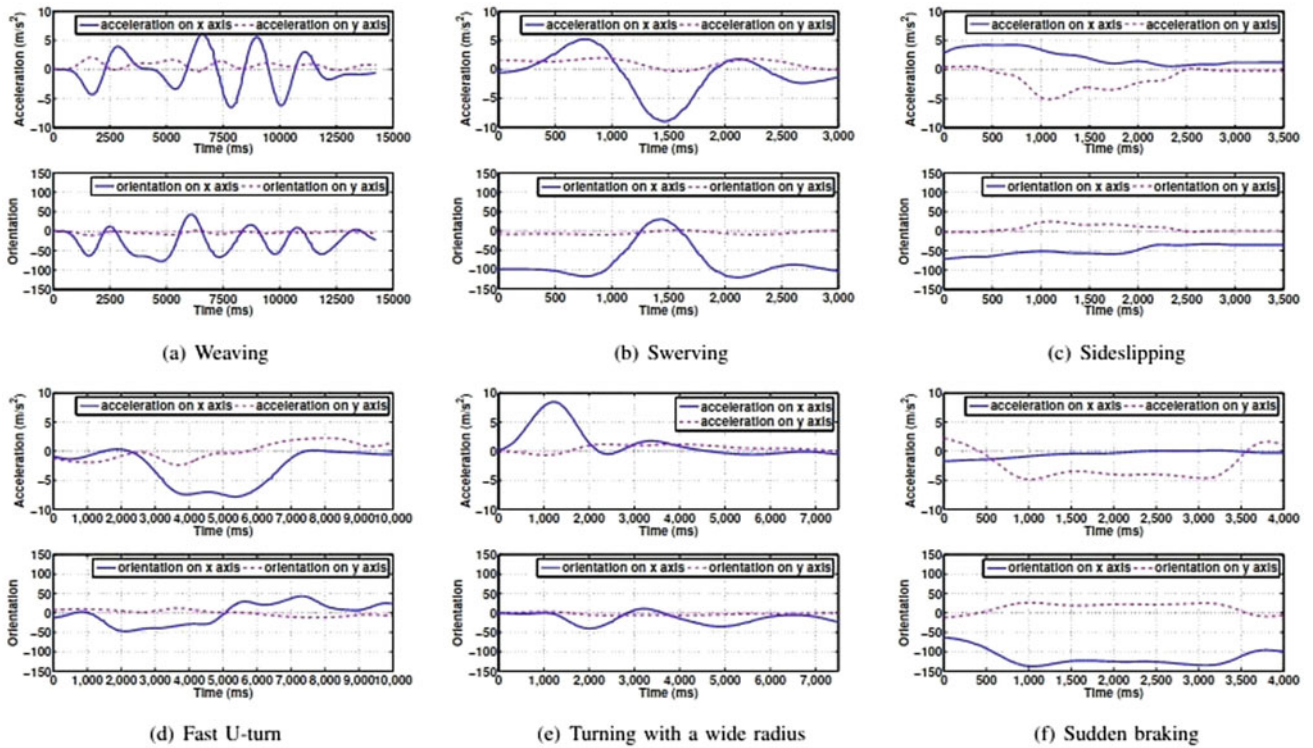


Fig. 9 Rash behaviour detection (Chen et al., 2015)

as long as it is kept static. We use 3-axis accelerometers to measure acceleration along x-axis and y-axis. When a vehicle changes its direction suddenly or if there is sudden increase in acceleration, the system detects the waveform and if it corresponds to any one of the six types of rash

driving behaviour then the manoeuvre is considered as rash driving. With the help of sensors, when collision is imminent split-second decisions can be made by the vehicle and facilitate deceleration and braking. The models used for detection will be discussed in subsequent sections.

6 Dataset Study

To study and analyze the driving behaviour, two different datasets were used for training. Here, they will be referred to as the Virginia Dataset (Custer, 2018) and the Mendeley Dataset (Yuksel & Serafettin, 2020).

The Virginia Dataset has 370264 samples and 76 features. This dataset is used to study the probability of a driver having an accident based on past and present features, allowing necessary remedial actions to be taken. The features can be broadly understood to be Gas Pedal Position, Speed, Yaw Rate, Lateral Acceleration, Longitudinal Acceleration, Distance from Lanes, Radar (Front, Back and Azimuth), Break Positions, Turn Signal Indicator Status and Lights. The data is used for binary classification of Crash or No Crash. The data was split 80–20; 80 for training and 20 for testing. The results of the models trained will be discussed in subsequent sections.

The Mendeley Dataset has 1114 samples and seven columns: Gyroscope X-axis, Gyroscope Y-axis, Gyroscope Z-axis, Accelerometer X-axis, Accelerometer Y-axis, Accelerometer Z-axis and multi-class targets (five). The target classes are 1. Sudden Acceleration, 2. Sudden Deceleration, 3. Sudden Left, 4. Sudden Right and 5. Normal. The dataset provides raw and augmented data. The dataset was split 70–30; 70 used for training and 30 for testing. The results of the models trained will be discussed in subsequent sections (Table 1).

7 Machine-Learning Models

Driver behaviour analysis is a vast field with varying approaches taken. Some approaches use the IMU readings obtained from a smartphone, as in Chen et al. (2015) or

through dedicated hardware present in vehicles as in Imkamon et al. (2008). Using Traffic Cameras to detect behaviour is also becoming common; these techniques rely on simple Machine-Learning algorithms to classify dangerous driving behaviour. The scope of this work will be limited to the analysis and modelling of time series data as mentioned in sections above, and will mainly focus on readings obtained from onboard sensors. The sensor's readings will then be stored and processed to identify discernable driving patterns and classify them as Rash or Not Rash.

7.1 Machine-Learning Models—Virginia Dataset

The models used on the Virginia Dataset were trained using the machine-learning library Turicreate. To study and compare the performance of different machine-learning algorithms, two different methods were used:

1. SVM Classifier
2. Logistic Regression Classifier.

7.1.1 SVM Classifier

Support Vector Machines are one of the commonly used methods in machine learning. The use of SVM for driving behaviour detection is presented by Chen et al. (2015). The SVM creates a hyperplane to separate the binary classes. The results of the SVM trained on the Virginia Data are given in Table 2.

Logistic Regression

Logistic regression is a statistical model that uses the logistic or sigmoid function to model a binary classification problem. The results of the trained model are given in Table 3.

Table 1 Datasets

Property	Mendeley dataset	Virginia dataset
Rows	1114	372,046
Columns	7	76
Features	Gyroscope X-axis Gyroscope Y-axis Gyroscope Z-axis Accelerometer X-axis Accelerometer Y-axis Accelerometer Z-axis	Gas Pedal Speed Yaw rate Lateral acceleration Longitudinal acceleration Distance from lanes Radar (front, back, azimuth) Break (on/off) Turn signal state (indicator) Lights
Classes	Sudden acceleration Sudden deceleration Sudden left Sudden right Normal	No-crash Crash
Train-test split	70–30	80–20

Table 2 Results SVM model

S. No.	Property	Value
1	Number of samples	372,063
2	Number of classes	2
3	Number of feature columns	76
4	Optimization method	L-BFGS
5	Number of coefficients	410,708
6	Total time of training	0.9985 s
7	Iterations	10
8	Training accuracy	0.9986
9	Validation accuracy	0.9986
10	Training loss	0.04978
11	Validation loss	0.049943
12	Training precision	0.99738
13	Validation precision	0.99721
14	Training recall	0.99275
15	Validation recall	0.99255

Table 3 Results logistic classifier model

S. No.	Property	Value
1	Number of samples	372,063
2	Number of classes	2
3	Number of feature columns	76
4	Optimization method	L-BFGS
5	Number of coefficients	410,708
6	Total time of training	10.1148 s
7	Iterations	27
8	Training accuracy	0.9993
9	Validation accuracy	0.9992
10	Training loss	0.02414
11	Validation loss	0.02950
12	Training precision	0.99667
13	Validation precision	0.99580
14	Training recall	0.99849
15	Validation recall	0.998286

Based on the performance of the above two models it can be seen that Logistic Regression performs better, though the difference is very less and largely insignificant. When it comes to time complexity, the SVM performed better with a lesser number of iterations and time taken to reach an optimal solution.

7.2 Machine-Learning Models—Mendeley Dataset

The models used on the Mendeley Dataset were also trained using the machine-learning library Turicreate. To study and compare the performance of different machine-learning algorithms, four different methods were used:

1. Boosted Tree Classifier
2. Random Forest Classifier
3. Decision Tree Classifier
4. Logistic Regression.

7.2.1 Boosted Tree Classifier

Gradient Boosted Trees are one of the most effective machine-learning models for predictive analysis. The result of the model trained is given below.

7.2.2 Random Forest Classifier

Random Forest is an advanced form of tree algorithm which trains multiple trees, and then uses average results for prediction and to prevent overfitting. The results are given below.

7.2.3 Decision Tree

A Decision Tree is a flow of conditions that need to be fulfilled to obtain the necessary class. The results are given below.

7.2.4 Logistic Classifier

Generally, Logistic Classifiers are used for binary classification. Here, a group of Logistic Classifiers are trained together to perform multi-class classification.

A quick study of Tables 4, 5, 6 and 7 shows that Boosted Tree Classifier provides the maximum training accuracy of 90%. It is followed by Random Forest Classifier with 88.5%. The modelling of the Mendeley Data allows us to classify some cases of rash driving. The results from the models are then passed to a grading algorithm that assigns a score to the driver.

8 Simulation

A model of the Toyota Prius was simulated using ROS Noetic and Gazebo 11. The vehicle's throttle (velocity), brake and steering were controlled by publishing to their respective ROS Topics. As per the proposed methodology, DriveSense majorly focuses on extracting meaningful

Table 4 Results boosted tree classifier

S. No.	Property	Value
1	Number of samples	1114
2	Number of classes	5
3	Number of feature columns	6
4	Optimization method	-
5	Number of coefficients	-
6	Total time of training	0.117602 s
7	Iterations	10
8	Training accuracy	0.902299
9	Validation accuracy	0.866248
10	Training loss	0.435027
11	Validation loss	0.486890
12	Training precision	0.88560
13	Validation precision	0.88411
14	Training recall	0.86812
15	Validation recall	0.86376

Table 5 Results random forest classifier

S. No.	Property	Value
1	Number of samples	1114
2	Number of classes	5
3	Number of feature columns	6
4	Optimization method	-
5	Number of coefficients	-
6	Total time of training	0.104365 s
7	Iterations	10
8	Training accuracy	0.885057
9	Validation accuracy	0.795332
10	Training loss	0.716707
11	Validation loss	0.779665
12	Training precision	0.863095
13	Validation precision	0.821640
14	Training recall	0.83392
15	Validation recall	0.78611

information from IMU (accelerometer and gyroscope) and onboard sensor (OBD) data. The placement of various sensors inside the vehicle can be seen in Figure 10.

The URDF model of the Toyota Prius was taken from the GitHub repository: osrf/car demo (<https://github.com/osrf/cardemo>). The vehicle was controlled in a way that mimicked real-world situations such as slow and rapid acceleration, slow and rapid deceleration, sharp and wide turns, fast U-turns, etc. The IMU sensor data can hence be recorded

Table 6 Results decision tree classifier

S. No.	Property	Value
1	Number of samples	1114
2	Number of classes	5
3	Number of feature columns	6
4	Optimization method	-
5	Number of coefficients	-
6	Total time of training	0.020345 s
7	Iterations	1
8	Training accuracy	0.827586
9	Validation accuracy	0.763016
10	Training loss	0.662293
11	Validation loss	0.733968
12	Training precision	0.78247
13	Validation precision	0.73554
14	Training recall	0.76158
15	Validation recall	0.72373

Table 7 Results logistic classifier

S. No.	Property	Value
1	Number of samples	1114
2	Number of classes	5
3	Number of feature columns	6
4	Optimization method	Newton-Raphson
5	Number of coefficients	21
6	Total time of training	0.027327 s
7	Iterations	5
8	Training accuracy	0.551724
9	Validation accuracy	0.4614
10	Training loss	1.37483
11	Validation loss	1.47593
12	Training precision	0.46466
13	Validation precision	0.455067
14	Training recall	0.445282
15	Validation recall	0.41123

and used to train and validate the ML model. However, the recorded data was inherently noisy, requiring the need for preprocessing.

9 Metrics

The following metrics are used for evaluating model performances.

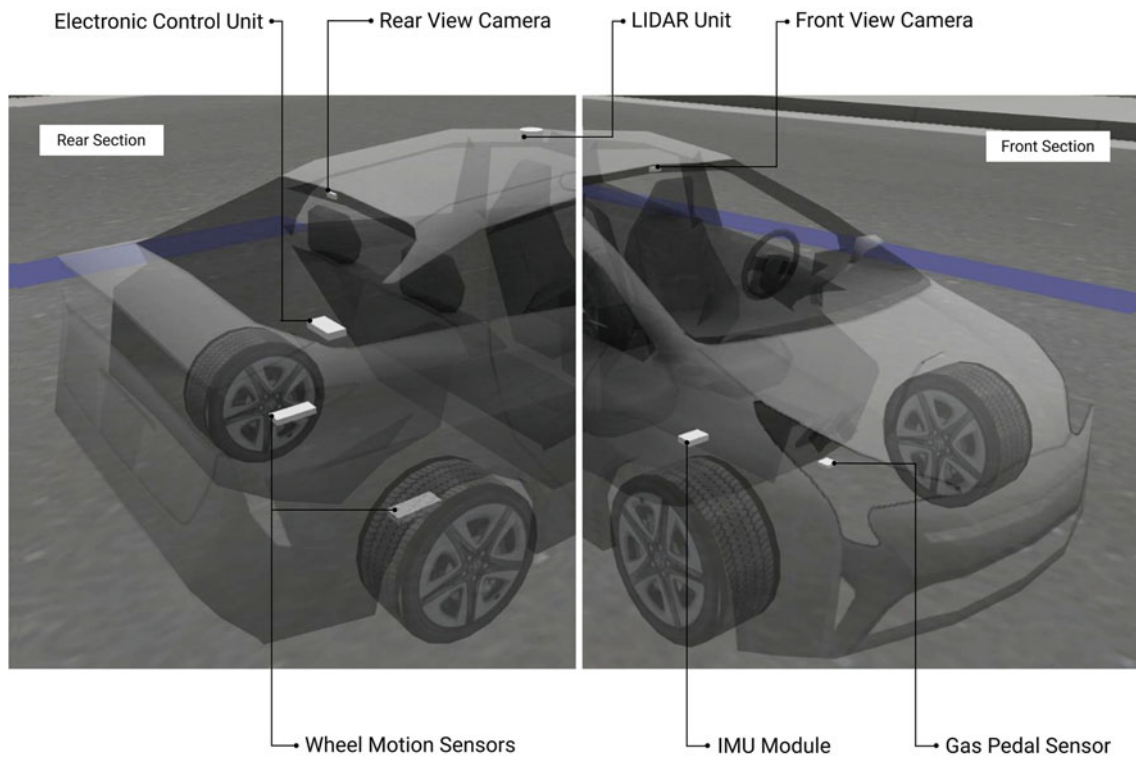


Fig. 10 Sensor placement

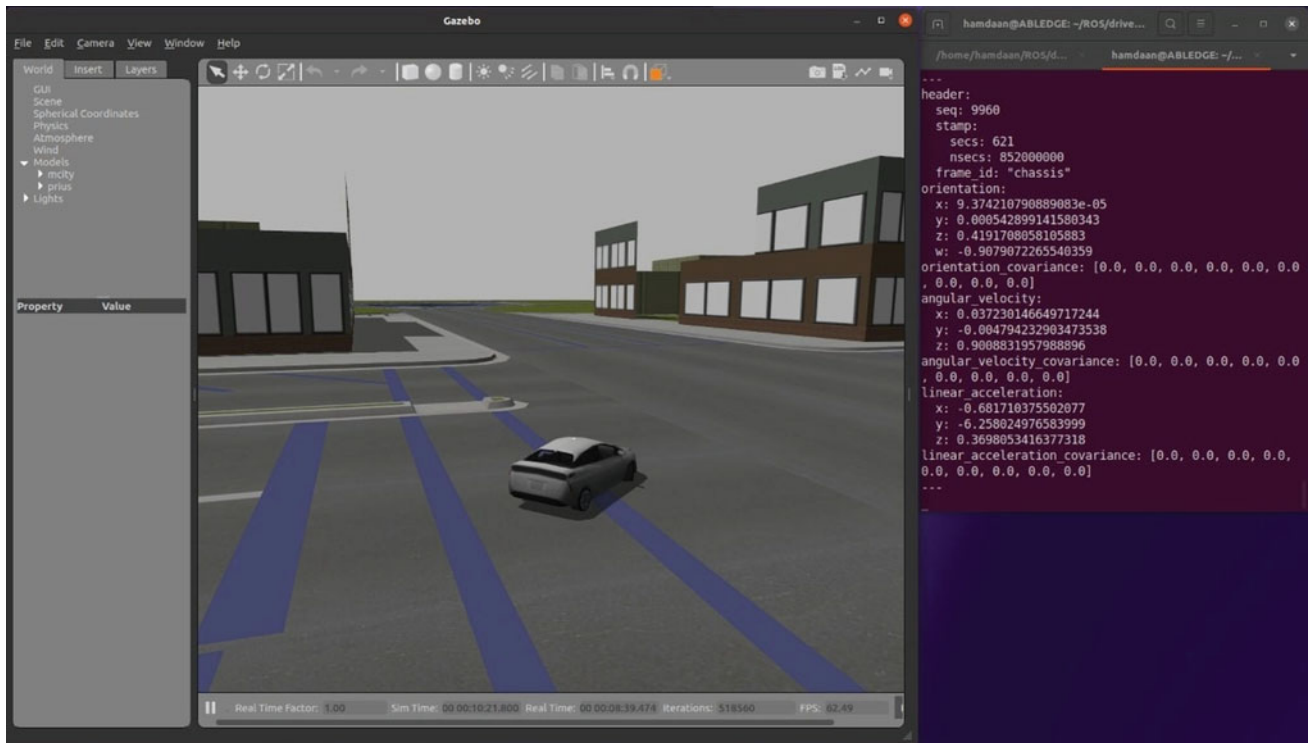


Fig. 11 Gazebo simulation

Table 8 Sensors and applications

Sensor	Application
Central acceleration sensor	Provide signals along longitudinal and lateral axes of the vehicle
Electromagnetic Sensors	In vehicles moving slowly and smoothly towards the obstacles
LIDAR (light detection and ranging sensors)	Detect the edges of roads Identify lane markings
Radar sensors	Predict the position of nearby vehicles
Reverse car parking sensors	Used during parking the car in a reverse car parking system
Ultrasonic sensors	Detect the position of curbs and other vehicles when parking
Crash/impact sensor	Release airbag
Lateral acceleration sensor	Measures the lateral acceleration of the vehicle
Low-g-sensor	For sensing the rollover of the vehicle
IMU sensor	Accelerometer and gyroscope

$$\text{Accuracy} = \frac{\text{Number of Incorrect Classifications}}{\text{Total Number of samples}}$$

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

$$\text{Loss} = \frac{1}{n} \sum_i^n (Y_i - \hat{Y}_i)$$

10 Results

The placement of sensors was discussed in Section 5. Some sensors and their application is mentioned in Table 8.

Sections 7.1 and 7.2 compared and studied the performance of six machine-learning algorithms on two different datasets. Table 1 details the properties of the Datasets and Tables 2, 3, 4, 5, 6 and 7 show the performance of the algorithms. Both SVM and Logistic Regression performed well on the Virginia Dataset. Boosted Tree gave the best results on the Mendeley Dataset, followed by Random Forest classifier. The progression of models across different epochs in given below. Figure 12 compares the performance of the Machine-Learning Models on the Virginia Dataset. From the graph, it is apparent that both machine-learning models perform suitably well and their overall performance across different iterations is similar. Figure 13 shows the performance of Machine-Learning Models on the Mendeley Dataset. Algorithms based on the Tree architecture perform almost similarly across different iterations, with the exception of Boosted Trees, which outperform other algorithms after the 5th iteration(epoch). Compared to other algorithms the performance of the Logistic Classifier is sub-par.

Tables 9 and 10 compiles the results from Section 7.1 and 7.2. Table 9 shows that the Logistic Classifier and Support Vector Machine has almost similar metrics. Of the two the Logistic Classifier gives the highest performance of 0.9993 training accuracy as compared to SVM's 0.9986. Table 10 shows that Boosted Tree Classifier gives the best result at 0.902299 training accuracy followed by Random Forest at

Fig. 12 Validation accuracy—
Virginia dataset

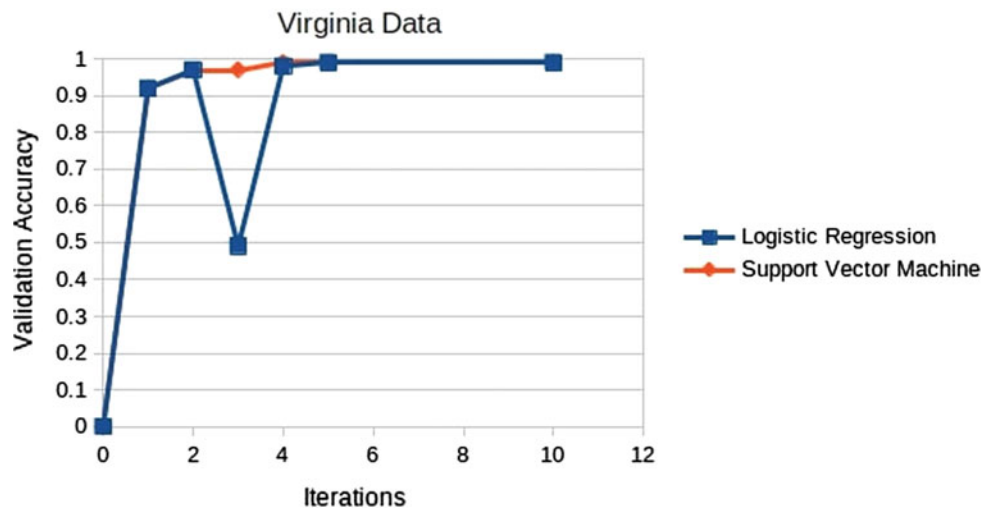


Fig. 13 Validation accuracy—Mendeley dataset

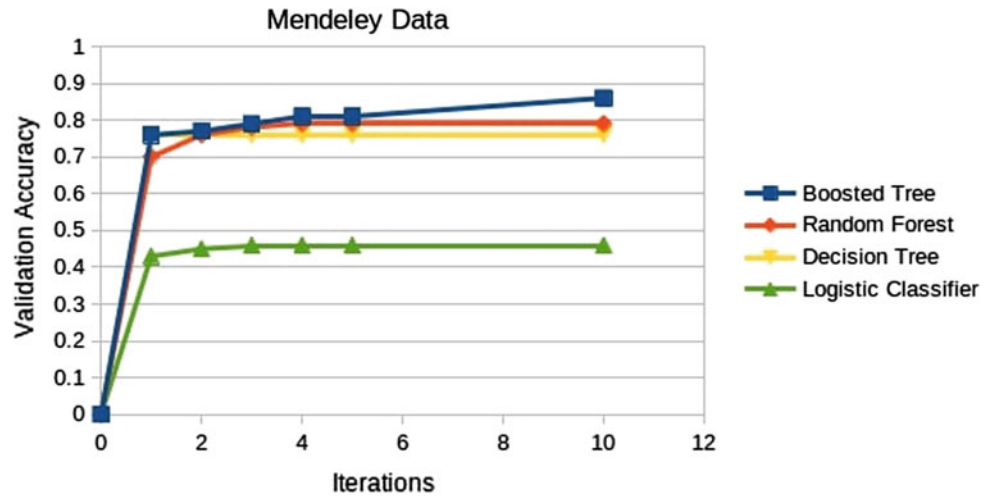


Table 9 Results—Virginia dataset

S. No.	Property	SVM classifier	Logistic classifier
1	Training accuracy	0.9986	0.9993
2	Validation accuracy	0.9986	0.9992
3	Training loss	0.04978	0.02414
4	Validation loss	0.049943	0.02950
5	Training precision	0.99738	0.99667
6	Validation precision	0.99721	0.99580
7	Training recall	0.99275	0.99849
8	Validation recall	0.99255	0.998286

Table 10 Results—Mendeley dataset

S. No.	Property	Decision tree classifier	Boosted tree classifier	Random forest classifier	Logistic classifier
1	Training accuracy	0.827586	0.902299	0.885057	0.551724
2	Validation accuracy	0.763016	0.866248	0.795332	0.4614
3	Training loss	0.662293	0.435027	0.716707	1.37483
4	Validation loss	0.733968	0.486890	0.779665	1.47593
5	Training precision	0.78247	0.88560	0.779665	0.46466
6	Validation precision	0.73554	0.88411	0.821640	0.455067
7	Training recall	0.76158	0.86812	0.83392	0.445282
8	Validation recall	0.72373	0.86376	0.78611	0.41123

0.885057 and Decision Tree at 0.827586. Logistic Classifier gives the least here at 0.551724.

11 Conclusion

In the scope of this paper, the need for driver behaviour profiling was discussed, and its importance in safe and comfortable commute. To implement this, DriveSense an Adaptive, ML-based driving behaviour Monitoring and Ranking System was suggested. DriveSense presents an efficient framework for recognizing specific abnormal driving patterns and providing score. The collected data can reveal trends in driver behaviour, and provide them with the necessary feedback. This can help motivate safer driving and reduce accidents and injury. DriveSense can further be used by insurance companies to study claims and take necessary action. In autonomous vehicles, DriveSense can be used to facilitate smart training and actions. The working of DriveSense including—acquisition of data and modelling said data is presented. The results and observations were tabulated and presented in the above sections. From the results, it can be concluded that SVM and Logistic Classifiers perform well on Binary Classification Problems, whereas the Boosted Trees Classifier performed well on Multi-class problems.

References

- Chen, Z., Wu, C., Huang, Z., Lyu, N., Hu, Z., Zhong, M., Cheng, Y., & Ran, B. (2017). Dangerous driving behaviour detection using video-extracted vehicle trajectory histograms. *Journal of Intelligent Transportation Systems*, 21(5), 409–421.
- Chen, Z., Yu, J., Zhu, Y., Chen, Y., & Li, M. (2015). D3: Ab normal driving behaviours detection and identification using smartphone sensors. In *2015 12th annual IEEE international conference on sensing, communication, and networking (SECON)* (pp. 524–532). IEEE.
- Custer, K. (2018). 100-Car data. <https://doi.org/10.15787/VTT1/CEU6RB>, VTTI, V2
- Dinesh Kumar, A., Karthika, R., & Soman, K. P. (2020). Stereo camera and LI-DAR sensor fusion-based collision warning system for autonomous vehicles. In S. Jain, Sood, M., & Paul, S., (Eds.), *Advances in computational intelligence techniques* (pp. 239–252). Springer Singapore.
- Guerrero-Ibañez, J., Zeadally, S., Contreras-Castillo, J. (2018). Sensor technologies for intelligent transportation systems. *Sensors*, 18(4), 1212. <https://doi.org/10.3390/s18041212>
- Haria, S., Anchaliya, S., Gala, V., & Maru, T. (2018). Car crash prevention and detection system using sensors and smart poles. *Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, 800–804. <https://doi.org/10.1109/ICCONS.2018.8663017>
- Harisankar, V., Sajith, V. V. V., & Soman, K. P. (2020). Unsupervised depth estimation from monocular images for autonomous vehicles. In *2020 Fourth international conference on computing methodologies and communication (ICCMC)*, Erode, India. <https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1>
- <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>
- <https://prsindia.org/policy/vital-stats/overview-road-accidents-india>
- <https://www.azosensors.com/article.aspx?ArticleID=19>
- <https://www.statista.com/statistics/1023507/india-registered-vehicles-number/>
- <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
- Imkamon, T., Saensom, P., Tangamchit, P., & Pongpaibool, P. (2008). Detection of hazardous driving behaviour using fuzzy logic. In *2008 5th international conference on electrical engineering/electronics, computer, telecommunications and information technology* (vol. 2, pp. 657–660). IEEE.
- Jones, W. D. (2002). Building safer cars. *IEEE Spectrum*, 39(1), 82–85.
- Jones, W. D. (2002, January). Building safer cars. *IEEE Spectrum*, 39(1), 82–85. doi: <https://doi.org/10.1109/6.975028>
- Kalra, N., & Divya, B. (2014). Analyzing driver behaviour using smartphone sensors: A survey (vol. 7, pp. 697–702).
- Kashevnik, A., Lashkov, I., & Gurtov, A. (2019). Methodology and mobile application for driver behaviour analysis and accident prevention. *IEEE Transactions on Intelligent Transportation Systems*, 21(6), 2427–2436.
- Kumar, A., & Suja, P. (2020). Steering angle estimation for self-driving car using deep learning. In *Machine learning and metaheuristics algorithms, and applications* (pp. 196–207). Springer Singapore
- Onken, R. (1994). DAISY, an adaptive, knowledge-based driver monitoring and warning system. In *Proceedings of the intelligent vehicles '94 symposium* (pp. 544–549). IEEE.
- osrf/car demo—(<https://github.com/osrf/cardemo>)
- Raviteja, S., & Shanmugasundaram, R. (2019). Advanced driver assistance system (ADAS). In *Proceedings of the 2nd international conference on intelligent computing and control systems, ICICCS 2018* (pp. 737–740).
- Saiprasert, C., & Pattara-Atikom, W. (2013). Smartphone enabled dangerous driving report system. In *2013 46th Hawaii international conference on system sciences* (pp. 1231–1237). IEEE
- Vilaca, A., Cunha, P., & Ferreira, A. L. (2017). Systematic literature review on driving behavior. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)* (pp. 1–8). doi: <https://doi.org/10.1109/ITSC.2017.8317786>
- Wang, M. S., Jeong, N. T., Kim, K. S., Choi, S. B., Yang, S. M., You, S. H., Lee, J. H., & Suh, M. W. (2016). Drowsy behaviour detection based on driving information. *International Journal of Automotive Technology*, 17(1), 165–173.
- Yaqoob, I., Latif, U. K., Ahsan Kazmi, S. M., Imran, M., Guizani, N., & Hong, C. S. (2019). Autonomous driving cars in smart cities: Recent advances, requirements, and challenges. *IEEE Network*, 34(1), 174–181.
- Yuksel, A., S,erafettin, A. (2020). Driving behaviour dataset. *Mendeley Data*, 2. doi: <https://doi.org/10.17632/jj3tw8kj6h.2>



Classification and Tracking of Vehicles Using Videos Captured by Unmanned Aerial Vehicles

Jorge E. Espinosa, Jairo Espinosa, and Sergio A. Velastin

Abstract

Road congestion levels currently seen in the main cities of the world, have generated problems associated with environmental air quality, road safety and excessive trip times. Intelligent Transport Systems (ITS) emerge as an important alternative to assist urban planning by monitoring the different urban road corridors. Traditionally, ITS use fixed camera systems which are sometimes insufficient to monitor and extend the traceability of the vehicles with extended coverage. Video capture mechanisms supported by Unmanned Aerial Vehicles (UAVs) e.g. drones, have greater continuous visibility of road segments, better visibility and allow more precise analysis on the density and behaviour of vehicle flows. This chapter presents detection, classification and tracking analysis for three categories of vehicles: Light (automobiles), Heavy (trucks and buses) and Motorcycles corresponding to 95% of urban traffic in emerging countries. For this purpose, 3 datasets with more than 2.5 million annotated objects are introduced using video sequences captured from UAVs from different views. An analysis of detection ability is carried out by considering traditional methods (prior to deep learning detectors) and using state-of-the-art detectors such as YOLO v3 and Faster R-CNN, obtaining results up to 98.5% AP (Average Precision) in all the analysed categories. Finally, multiple

object tracking techniques such as Deep Sort are investigated, achieving results of up to 82% MOTP (Multiple Object Tracking Precision) in light vehicles.

Keywords

Urban vehicle detection • Urban vehicle tracking • Yolo • Faster R-CNN • Region-based detector • CNN • Deep learning • Multiple object tracking

1 Introduction

As city populations grow, problems associated with vehicle traffic rise. Even with the mobility restrictions imposed by the recent Covid-19 pandemic, traffic jams and environmental pollution demand an improved vehicle infrastructure. Intelligent transport systems (ITS) are an strategic approach to mitigate those issues, implementing services such as traffic security monitoring, transportation planning, control of traffic violations, electronic toll collection, pollution measurement, traffic flow analysis amongst others (Tian et al., 2017). Nevertheless, the full implementation of such services has high costs related to sensors deployment and maintenance. Surveillance based on video cameras is currently a key component of modern ITS, since cameras are relatively low-cost sensors and capture a significant amount of data than can be analysed by means of computer vision techniques (Buch et al., 2011). Tian et al. (2017) describe important issues and aspects that computer vision still has to solve, such as: 24/7 surveillance cycles, pose variation, variety of vehicles, different range of video resolutions, vehicle behaviour understanding and vehicle occlusion, amongst others.

Vehicle occlusion, can present a major problem for detection, classification and even more so for tracking a given vehicle. This difficulty can be overcome by means of the use of unmanned aerial vehicles (UAV), which provide a

J. E. Espinosa (✉)
Politécnico Colombiano Jaime Isaza Cadavid,
Carrera 48 No. 7-151 El Poblado, Medellín, Colombia
e-mail: jeespinosa@elpoli.edu.co

J. Espinosa
Universidad Nacional de Colombia, Sede Medellín,
Carrera 80 No.65-223, Medellín, Colombia
e-mail: jjespino@unal.edu.co

S. A. Velastin
University Carlos III Madrid, Getafe, Spain
e-mail: sergio.velastin@ieec.org

Queen Mary University of London, London, UK

completely new perspective to the problem, allowing to capture a top view of the different class of vehicles and greater geographic coverage of the road network.

This chapter introduces a novel UAV urban vehicle dataset (UTUAV) with more than 2.5 million of annotated objects. A method is also described to detect, classify and track vehicles in three separate categories: Light (automobiles), Heavy (trucks and buses) and Motorcycles, which combined account for 95% of urban traffic in emerging countries. The remaining sections are organized as follows: Sect. 2 describes work related with detection and tracking of vehicles in urban areas. Detectors and a tracking mechanism implemented for this work are also described in that section. Section 3 depicts the metrics and evaluation mechanisms used to assess the performance of the proposed method, both for detection and also for multiple object tracking. Section 4 presents the UTUAV dataset, captured by the authors and consisting of three subsets, describing the amount of annotated objects, categories and tracks associated with each vehicle. The proposed methodology is described in Sect. 5, which integrates the detection and tracking algorithms described in Sect. 2, and also proposes a zoom strategy for vehicles that appear too small, such as the motorcycles category. Section 6 presents baseline results on vehicle detection classification and tracking obtained using the described methodology on the UTUAV dataset so that future researchers can report improved results. Finally, Sect. 7 analyses the results and Sect. 8 presents conclusions and proposes future work in this evolving field.

2 Related Work

Tian et al. (2017) describe an ITS as a general system comprising four layers. The first layer implements sensing through image acquisition. The second layer defines the extraction of dynamic and static vehicle properties. Vehicle behaviour understanding is carried out in the third layer. Finally, ITS services are provided in the last layer.

Most cities that have implemented ITS, for the first layer rely on extensive traffic surveillance networks using and storing data coming from fixed cameras. The second layer involves automation to distinguish different traffic conditions. This automation may not be limited to extracting vehicle attributes, but it may also provide other services such as incident detection, vehicle counting, traffic rules violations, etc. (Buch et al., 2011; Sivaraman & Trivedi, 2010). For that second layer, a large variety of methods have been developed for vehicle detection covering different scenarios such as urban or highway areas, different sensors such as monocular vision or stereo vision, and the problem of real-time operation (Buch et al., 2011). According to Tian et al. (2017), vehicle detection methods come in two

categories, depending on whether they are based on motion or appearance features.

2.1 Vehicle Detection Based on Motion Features

This approach is normally found in systems with fixed cameras. Objects are detected by identifying image areas that move in comparison to their (more stable) background. The key concept is background subtraction which is one of the most studied strategies for motion detection (Tian et al., 2017).

Earlier methods include Gaussian average with pixels updating (Koller et al., 1994) and a temporal median filter that uses the pixel median value in the last n frames to calculate the background. Stauffer and Grimson (1999) used a Gaussian mixture (GMM) which models pixels intensity of foreground and background using Gaussian distributions. The model was improved by Elgammal et al. (2000) who proposed the use of two models for dealing for sudden and stable changes in background pixels. Zivkovic also improved GMM (Zivkovic, 2004) by finding the number of components that generates the Gaussian model in each iteration, taking in to account the incoming image data and discarding the old one. There is also what is called sequential kernel density approximation (Han et al., 2004) that tackles the problems associated to fixed number of Gaussians in the GMM approach (not suitable for complex backgrounds) and the requirement of considerable memory to store the background model of non-parametric kernel density estimation. Seki et al. (2003) used spatial co-occurrences in images due to small movements of objects, which could be very sensitive to shadows. This phenomena is addressed with Eigen-backgrounds proposed by Oliver et al. (1999).

Vehicle detection based on motion features could also be carried out with dynamic saliency maps, using motion and saliency to highlight the important pixels of a video: visual saliency is used to detect vehicles with a single picture, and static saliency maps in succession obtain time varying properties as well as spatial features (Woo et al., 2010). The approach is improved by Lee et al. (2015) proposing a fast dynamic analysis by measuring salience maps variance instead of maximum entropy amongst consecutive frames. This algorithm still faces the problem of shadows casts by the detected object.

2.2 Vehicle Detection Based on Appearance Features

Visual features as shape, texture or colour are also used to locate vehicles in images.

Feature Extraction is the representation by a numerical vector that describes a visual feature obtained from a group

of pixels in a given image. This vector is called a feature descriptor. Visual features used for vehicle detection include instance shape symmetry. Kuehne (1991) proposed three types of symmetry: contour, grey level and rear vehicles horizontal lines, which are used to locate vehicles in traffic scenes. Meanwhile von Seelen et al. (2000) proposed U-symmetry, which is combined with model matching to detect, track and classify vehicles. Nevertheless, vehicles are symmetric in a limited set of points of views, limiting the capture angle of the cameras.

Edges are also used for vehicle detection as in Goerick et al. (1996), where an edge detector is followed by a histogram of rows and a histogram of columns as features that are later classified by an artificial neural network. A Canny edge detector is used in combination with Harris corner (1988) and colour for features are used for vehicle classification by a dynamic Bayesian Network in Cheng et al. (2011). Histogram of oriented gradients (HoG) (Dalal & Triggs, 2005) is a frequently used feature that describes the contours in an image using the directions of its edges. HoG is used for Zhang (2013), for classifying cars and vans. Buch et al. (2009) extends HoG to a three three-dimensional space with 3D Extended Histogram of Oriented Gradients using it to classify pedestrians and vehicles. Since corners have a specific location in image, they are also used as descriptors. The Harris corner detector (1988) extracts eigenvalues from a matrix using first horizontal and vertical gradients of an image. Corner features in combination with others features are used for vehicle detection in Cheng et al. (2011). Colour is another image attribute which is used as a feature, for instance Tsai et al. (1999) map RGB colours to a new colour space using a normalized orthogonal transformation, simplifying the vehicle detection process. Meanwhile, texture is also used as a feature for vehicle hypotheses. Textural features such as contrast and entropy are used as indicators of a region of interest (Haralick et al., 1973). In (Kalinke et al., 1998) texture is used to propose vehicle hypotheses that are optimized by the camera model geometry to then undertake classification by a multilayer perceptron.

More recently, other feature descriptors used for vehicle detection are key points, which correspond to a pixel (or group of them) different to the surround pixels and integrated into a single vector. The Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) is amongst this kind of descriptors, which use a difference of Gaussian method. Key points are found at different scales, to provide invariance to scale. By normalizing and orienting to the greatest gradient variation, the descriptor is also invariant to orientation. Another feature descriptor that is invariant to scale and rotation is Speeded-Up Robust Features (SURF) (Bay et al., 2008), which is faster than SIFT and uses a Hessian matrix to detect interest points and sums of Haar wavelet components to describe them. Using a video taken from an unmanned aerial

vehicle (UAV), Moranduzzo and Melgani (2014) used SIFT to detect key points from cars, which combined with colour and morphology features are then classified with a support-vector machine (SVM). Meanwhile Hsieh et al. (2014) use SURF combined with Histogram of Oriented Gradients (HoG) (Dalal & Triggs, 2005) to classify vehicles with an SVM. Speed for finding key point descriptors is improved by Features from Accelerated Segment Test (FAST) (Rosten & Drummond, 2006) that uses difference in intensity between the centre of a circle with 3 pixels of radius and its perimeter. BRIEF, Binary Robust Independent Elementary Feature (Calonder et al., 2010), describes a region by means of pairwise comparison of intensities. These two descriptors are combined in ORB (Oriented FAST and Rotated BRIEF) (Rublee et al., 2011) where FAST is used as a key point detector and BRIEF as a feature descriptor. Buliali et al. (2017) use ORB for detecting vehicles in satellite images.

These traditional methods for vehicle detection have two main drawbacks: difficulties in dealing with occlusion and cast shadows. This issues and the real-time requirement in many applications boost the use of deep learning algorithms as more accurate alternatives for vehicle detection and classification, as explained in the next section object.

2.3 Vehicle Detection with Deep Learning

Traditional classification techniques often rely on feature extraction algorithms and training classifiers, as separate tasks. However, feature extraction and classifiers can both be trained in a single pipeline. Deep learning algorithms used for object detection/classification avoid the finely tuned step for feature extraction. Instead, they exploit the optimization of a huge number of parameters in a non-linear mathematical model able to detect and classify objects in an image. In 2012, Krizhevsky et al. (2012) designed AlexNet, a deep convolutional neural network that won the ImageNet challenge. AlexNet outperformed the second and third placed methods with a lower classification error of 10.8% lower and 11.7% respectively, both using traditional feature extraction classification techniques. AlexNet's uses convolutional layers which implicitly incorporate feature extraction. Complexity of features are obtained going deeper in the network architecture.

Object detection involves classification of an object and prediction of its position on an image, and the state-of-the-art (Soviany & Ionescu, 2018) in this task is deep learning. Deep learning detectors are divided into two main approaches. Region-based (two-stage) detectors, using a region proposal network (RPN) to initially analyse the complete image and propose candidate objects that are then classified in the second stage. The other approach is known as

Single-shot (one-stage) detectors, which pose detection and classification as a regression problem for bounding boxes with an associated class probability.

In this chapter we show a baseline for the new public dataset, using two of the best deep learning object detectors: Faster-RCNN (region-based), and YOLO v3 (one-stage).

Faster R-CNN (Ren et al., 2015) is a region-based (two-stage) algorithm for object detection that uses convolutional neural networks (CNN) for region proposal and later for classification. The original model uses a well-known CNN model such as VGG (Simonyan & Zisserman, 2015) or ZFNet (Zeiler & Fergus, 2014) to extract a convolutional feature map from the image, then a region proposal network (RPN) defining region proposals which are later reshaped using an RoI (Region of Interest) pooling layer, classifying the image within the proposal region and predicting the offset adjustment of the bounding boxes that encloses the classified object.

YOLO (You Only Look Once) (Redmon et al., 2016) is a single-stage object detection algorithm. YOLO incorporates in its architecture a single convolutional network with many hidden convolutional layers followed by max-pooling layers. The feature map is flattened and computed by fully-connected layers after convolutional layers, and is used to predict bounding boxes parameters and classes for the image detected objects.

Yolo takes the input image and splits it into a grid of $S \times S$ pixels. Each cell of the grid can produce m bounding boxes. For each bounding box the network outputs a class probability and the offset of the bounding box parameters. The bounding box with a class probability above a defined threshold is selected and is used to locate the object within the image.

Even though YOLO is used for real-time detection, it has some limitations. For instance, if multiple objects share the grid cell where their centre is contained, bounding box assignment will fail if the objects share the same aspect ratio. The features extracted by YOLO are coarse features, meaning that small objects might not be easily recognized.

2.4 Vehicle Tracking

Vehicle tracking is often used in traffic analysis, and it involves the labelling of a detected object and the estimation of its trajectory estimation in a video sequence. Whilst detection algorithms are able to discriminate objects in a single image, tracking algorithms follow the trajectory of each object throughout the video, and since there are many objects detected in the scene, a more complex tracking process arises i.e. multiple object tracking (MOT). Tian et al. (2017) classified tracking approaches into 4 categories:

vehicle representation, Kalman filter (1960) tracking, particle filter tracking and dense inferencing architectures.

Vehicle representation uses a unique object representation such as vehicle region, contour, model or feature vector. The similarity of representations between frames is then used to track vehicles. A Kalman filter estimates the tracking of vehicles positions using measurements with noise and inaccuracies using historical information to predict and thus reduce the search space of the object in the image. For vehicle tracking, Kalman filter is used in Rad and Jamzad (2005) showing improved performance than other methods in occluded condition and light changed scenarios. When vehicle flow may exhibit non-linear behaviour, an extended Kalman filter can be used (Lou et al., 2002). A Particle filter uses recursive Bayesian filters to detect and track objects (vehicles) in a video sequence. The particle filter is made up of a set of samples (the particles) and values, or weights, associated with each of those samples. Particles are possible states of the process, which can be represented as points in the state space of the process. The method has advantages over the Kalman filter, using multiple Gaussian distributions. Motion estimation is achieved by the use of dense inferencing architectures with a probabilistic graphical model. The model can work at object level using generic image features or at pixel level.

DeepSORT Algorithm In this work, tracking is carried out by the DeepSORT algorithm, which was proposed by Wojke et al. (2018) to track multiple objects using a deep association metric. It is based on Simple Online and Real-time Tracking (SORT) proposed by Bewley et al. (2016). After each detection, the algorithm estimates the location of objects with a model that uses Kalman filtering (1960), assigns each prediction with the new detections, matches objects between two frames by maximizing intersection over union (IoU) and closeness of appearance descriptors obtained with convolutional neural networks.

The appearance feature vector is extracted from a trained classifier, where all the detected bounding boxes crops are passed obtaining for each one a 128×1 dimensional feature vector later used to evaluate the cosine distance between appearance features vectors. To estimate the next state of each tracked object, an eight dimensional state space is defined $(u, v, \gamma, h, \dot{u}, \dot{v}, \dot{\gamma}, \dot{h})$. The location of the centre of the bounding box is defined by (u, v) , the aspect ratio by γ and height h . A standard Kalman filter is used, the model assumes constant velocity motion and linear observations. The coordinates (u, v, γ, h) are taken as direct observations of the state of an object. To deal with occlusions, for each track k , the number of frames since it was successfully associated with a detection is stored in a_k . The maximum amount of frames that an object can be totally occluded is given by A_{max} . If an a_k exceeds A_{max} the k th track is

considered to have left the scene. When a detection cannot be associated with an existing track is considered to be a new track hypothesis. However, new tracks are treated as tentative for the first three frames. Only tentative tracks that are successfully associated over these three frames are considered valid new track hypotheses, otherwise they are deleted. To assign newly arrived measurements to existing tracks, motion and appearance information are integrated. For motion information, the squared Mahalanobis distance between the distribution of Kalman states of the i th track and the j th bounding box is taken, and those distances outside a 9% confidence interval are excluded.

According to the authors, the Mahalanobis distance is not a suitable metric to track through occlusions. Therefore, they included a second metric related to visual appearance. A vector r_j with norm 1 is used as an appearance descriptor of detection d_j . For each tracked object k , the last $L_k = 100$ descriptors are stored in a gallery $\mathcal{R}_k = \{r_k^{(i)}\}_{i=1}^{L_k}$. The metric associated with appearance $d^{(2)}$ measures the cosine distance between the i th tracked object and the j th new detection.

3 Metrics

3.1 Object Detection Metrics

Metrics used for evaluate performance in detection are well established such as those proposed by the VOC challenge (Everingham et al., 2010) and by Yin et. al. (2010). Average Precision (AP) is the common metric used to evaluate object detection performance, and computes the average precision of the recall value in a range from 0 to 1.

The following definitions are important to understand average precision:

- **Precision:** It measures what proportion of the detected objects are true objects i.e. it is calculated as the amount of true positives (TP) over the sum of true positives (TP) and false positives (FP), Eq. 1.
- **Recall:** It measures the proportion of true objects detected, i.e. it is calculated as the amount of true positives (TP) over the sum of true positives (TP) and false negatives (FN), Eq. 2.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

The intersection over union (IoU) or Jaccard index (1901) measures the overlap of two bounding boxes. It is used to measure how much the predicted bounding box overlaps with the ground truth and is calculated by using the intersection of the ground-truth area and predicted area divided by the union of this two arguments (Eq. 3). When the metric AP@0.5 is referenced, it means that a true positive is only considered when the $\text{IoU} \geq 0.5$. Figure 1 shows examples of IoU scores.

$$\text{IoU} = \frac{A_{\text{GT}} \cap A_{\text{Predicted}}}{A_{\text{GT}} \cup A_{\text{Predicted}}} \quad (3)$$

The inverse relationship between precision and recall is exploited to define the AP metric, precision is set as a function of the recall ($P(r)$), and the metric calculates the area under the curve, calculating the finite sum over every sorted precision-recall pair (see Eq. 4). Values of $\text{AP} \in [0, 1)$ which are the area under the curve of precision vs recall. The higher the AP, the better the detector.

$$\text{AP} = \sum_{k=2}^N P(k)[r(k) - r(k-1)] \quad (4)$$

3.2 Tracking Metrics

Since tracking in this scenario corresponds to Multiple Object Tracking (MOT), the metrics used are:

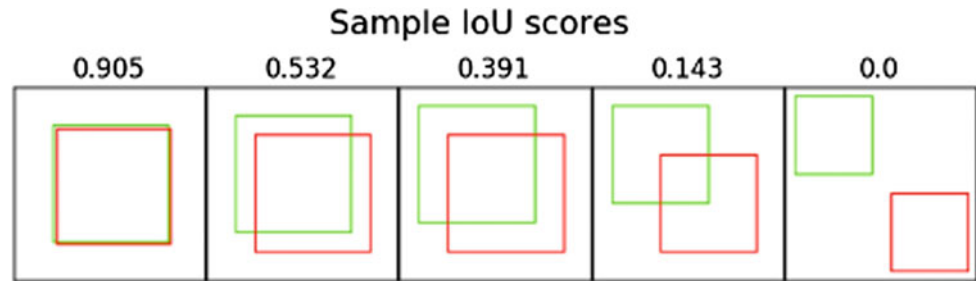
- **Multiple Object Tracking Precision (MOTP):** It computes the error of matched objects estimation over the total number of obtained matches, see Eq. 5. It only measures the precision of objects locations neglecting aspects as consistency of trajectories or objects configurations.

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \quad (5)$$

- **Multiple Object Tracking Accuracy (MOTA):** Which calculates the sum of number of missed objects (m_t) plus number of false positives (fp_t) and the number of mismatches (mme_t), all this sum over the number of ground-truth objects (g_t), all elements calculated in the t th frame. Equation 6 describes the metric calculation.

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (6)$$

Fig. 1 Samples of different IoU indices. Taken from R. Restrepo, IoU scores



4 The UTUAV Urban Traffic Dataset

The literature has many urban vehicle datasets (SCALE & Open Datasets—Scale, 2015), mostly generated using surveillance cameras from ITS installations, and oriented to computer vision tasks such as detection (Dollar et al., 2011; Wen et al., 1511) and tracking (Geiger et al., 2012; Kristan et al., 2020). Nevertheless those datasets generally are captured by fixed or from cameras attached to moving vehicles, in which the major limitation is the viewing angles that generate occlusions amongst urban objects. An Unmanned Aerial Vehicle (UAV) has the potential to capture from a privileged angle (top view) and, depending on flying altitude, it allows to capture large areas for video analysis, covering a large number of objects. In this regard, works as the Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking (Du et al., 2018) is a benchmark with more than 80,000 frames consisting of 100 video scenes selected from 10 h of video, captured in urban areas, with common scenes including squares, toll stations, highways, arterial streets, crossings and T-junctions. Vehicle categories are limited to trucks, cars and buses. Videos are captured with camera movement along the UAV path. There is also the UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking (Wen et al., 2020) that captured more than 140,000 frames of real-world traffic scenes, which includes 8250 annotated vehicles including cars, vans, buses and other categories. From 2018 the Vis-Drone Challenge (Zhu et al., 2020), has created a dataset with 400 video clips formed by 265,228 frames and 10,209 static images. This dataset has been updated yearly by the AISKYEYE team at Lab of Machine Learning and Data Mining of Tianjin University in China. Annotated images include 10 different class categories (pedestrian, vehicles, bicycles, etc.) and reach up to 2.6 million of objects bounding boxes.

Here, we introduce the UTUAV dataset which consists of three different scenes captured in the second largest city of Colombia: Medellín. Road user classes representative of those in emerging countries such as Colombia, have been chosen: motorcycles (MC), light vehicles (LV) and heavy

vehicles (HV) as seen in Fig. 2. Vehicle tracks are indicated by a unique identifier assigned to an annotated object throughout its appearance in a scene (Fig. 3). The dataset is available for download at.¹

Specifications for annotations include: the minimum object size (both width or height) is set to 20 pixels. Occluded objects are annotated just in their visible area. For tracking, the identifiers of fully or partially occluded objects are preserved. Annotations were done using the Viper annotation tool developed by Mariano et al. (2002).

The following is the description of each dataset.

4.1 UTUAV-A Dataset

The first dataset corresponds to an extension of Espinosa et al. (1808) which originally only annotated motorbikes in 10,000 frames with a resized resolution of 640×364 pixels. The images were taken from an unmanned air vehicle (UAV), elevated 4.5 m from the ground. The UAV was a DJI Phantom 4[®] drone, with an HD camera under windy conditions, which affected the image stabilizer capabilities. The UAV is kept at the same position and small movement of the camera is noted. The extension includes the annotation of light and heavy vehicles (Fig. 3).

Table 1 presents the main dataset characteristics, including the number of annotated vehicles, mean area of the vehicle bounding box, total occluded vehicles, the mean duration of total occlusions measured in frames and the mean displacement in pixels of objects when are occluded. Note that due to the limited elevation, and the capture angle of the sequence, occlusions appear frequently, and object sizes change significantly.

4.2 UTUAV-B Dataset

Exploiting the top view angle that a high elevation UAV can reach, the second dataset is composed of 6,500 labelled images with a resolution of 3840×2160 (4 k) pixels, the

¹ <http://videodatasets.org/UTUAV/>

Fig. 2 Detected objects with classes

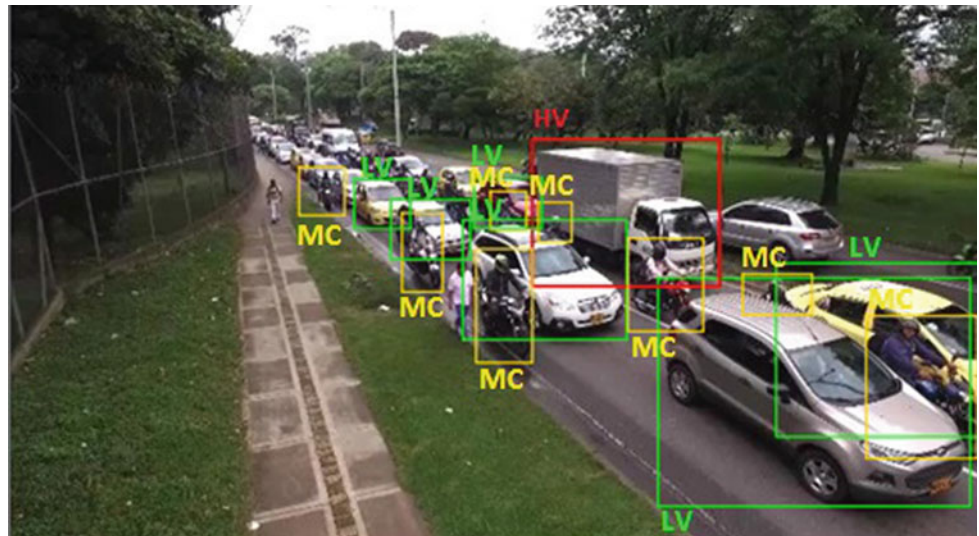


Fig. 3 Detected objects with classes and identifiers

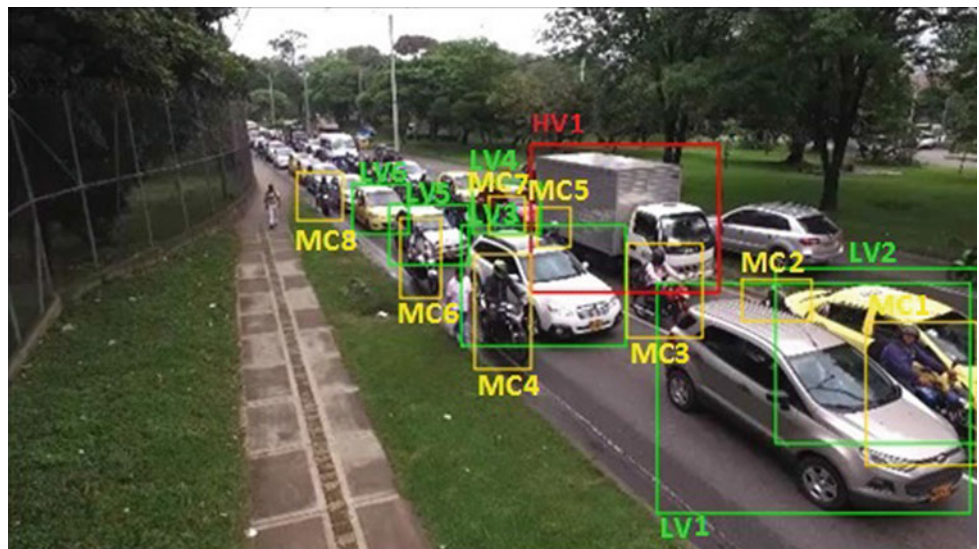


Table 1 Characteristics of the UTUAV-A dataset

Vehicle	Motorcycle	Light vehicle	Heavy vehicle
Number of annotations	56,970	44,415	7717
Annotated Objects	318	159	16
Mean size (pixels)	1763	4546	4771
Totally occluded objects	7	6	3
Mean occlusions duration (frames)	8.1	2.9	269.3
Mean occlusions duration (displacement)	31.2	6.3	369.4

UAV used is a DJI Inspire 2[®] drone with a zenmuse X5s gimbal camera. The entire sequence is captured with a top view elevated up to 100 m from the ground, turbulence or camera movement is hardly perceived. The annotation

bounding box colour codes are red for light vehicles, blue for motorbikes and green for heavy vehicles (Fig. 4). A summary of characteristics of the dataset is described in Table 2, with the same parameters used in UTUAV-A.



Fig. 4 Sample image from the UTUAV-B dataset

Table 2 Characteristics of the UTUAV-B dataset

Vehicle	Motorcycle	Light vehicle	Heavy vehicle
Number of annotations	70,064	331,508	18,864
Annotated objects	128	282	13
Mean size (pixels)	992	3318	5882
Totally occluded objects	80	84	4
Mean occlusions duration (frames)	19.6	18.6	34.0
Mean occlusions duration (displacement)	108.4	130.7	197.5

4.3 UTUAV-C Dataset

Finally, the third dataset has a sequence of 10,560 labelled frames with a resolution of 3840×2160 (4 k) pixels. This video sequence was also captured from an UAV DJI Inspire 2[®] drone with a zenmuse X5s gimbal camera, elevated at 100 m from the ground. This dataset also uses a top view angle and shares the same colour code for annotations as per UAVUT-B (Fig. 5).

The summary of characteristics of the UAVUT-C is given in Table 3 for the same parameters as UAVUT-A and UATUV-B.

5 Methodology for Urban Vehicles Detection and Tracking

Recent investigations for motorbike (Espinosa et al., 2018, 2019) and pedestrian (Velastin et al., 2020) detection and tracking suggested algorithms best suited for this task. The methodology proposed here integrates the use of detection and tracking algorithms based on the best results obtained evaluating the presented dataset, particularly on high resolution (4 K) datasets B and C.

The methodology starts by extracting single frames images from video, then a binary classifier for each urban



Fig. 5 Sample image from the UTUAV-C dataset

Table 3 Characteristics of the UTUAV-C dataset

Vehicle	Motorcycle	Light vehicle	Heavy vehicle
Number of annotations	463,009	1,477,287	130,142
Annotated objects	456	997	86
Mean size (pixels)	467	1722	4275
Totally occluded objects	211	265	31
Mean occlusions duration (frames)	89.9	86.8	110.8
Mean occlusions duration (displacement)	226.9	210.1	260.3

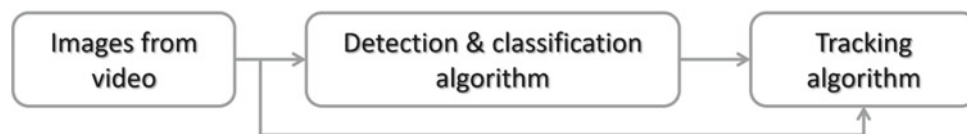


Fig. 6 Methodology for urban traffic analysis

traffic vehicles class (light vehicles, motorcycles and heavy vehicles), and finally a tracker algorithm is applied to the detection results obtained in conjunction with the analysis of the video sequence to reinforce the tracking algorithm (Fig. 6).

5.1 Light Vehicles

For light vehicles detection and tracking, initially single frames of the video sequence are extracted and analyzed with a Faster R-CNN (Ren et al., 2015) binary classifier. The architecture generates light vehicle hypothesis using the

region proposal network (RPN), a convolutional neural network (CNN) extracts features that are used to verify those hypothesis, generating bounding boxes and classifications scores. A threshold of 0.2 is used to take a box as a detected vehicle.

The tracking process starts from the detection results obtained from Faster R-CNN, and is carried out by DeepSORT (Wojke et al., 2018) algorithm. Object tracks are followed by using the Kalman Filter, whose initial state is defined by the bounding boxes parameters and their respective velocities. An appearance feature vector is also used to resolve uncertainties and occlusions when Kalman filter predictions are not reliable. The pipeline for light vehicles is shown in Fig. 7.

This pipeline was also implemented using YOLOv3 (Redmon & Farhadi, 2018) as an object detector, considering its almost real-time performance, but leading to less accurate results than a region-based detector such as Faster-RCNN (Ren et al., 2015).

For this pipeline Faster R-CNN is implemented from the source code in² using Tensorflow 1.15.0 GPU compatible on Python 3.6. DeepSORT is implemented using the nanonets³ version.

5.2 Motorcycles

Motorcycle detection and tracking shares almost identical aspects of the light vehicles strategy. Nevertheless, given the relatively small size of motorbike objects in relation with the frame image size, a zoomed-in strategy is used to improve detection accuracy. After single images are extracted from the video sequence, a region of interest (ROI) is established which is zoomed-in and fed to the Faster R-CNN (Ren et al., 2015) binary model to perform motorcycle classification. The processed area is reduced after the zoom-in process, the bounding boxes and classification scores are defined related now to this subarea. A threshold of 0.2 is also used to take a detected bounding box as a motorcycle. The use of Faster-RCNN as a classifier for this vehicle category is derived from the studies made on (Espinosa-Oviedo et al., 2019) where the authors highlight the improved performance of Faster R-CNN over YOLO on detecting small objects: “This can be explained again by the absence of RPN in YOLO V3, which fails to detect objects that appear too close or too small”.

The tracking process for motorcycles also uses the DeepSORT (Wojke et al., 2018) algorithm. Tracking is implemented in the zoomed-in sub areas, to preserve the

visual features obtained in the detection process which are crucial for cosine distance evaluation for object association. The pipeline for motorcycle detection and tracking is shown in Fig. 8.

Note that only 44.4–32.7% of the full image area is processed since the zoomed-in range starts from 150 to 175%. For this pipeline the same version for Faster R-CNN (Ren et al., 2015) and DeepSORT (Wojke et al., 2018) is used as for light vehicles.

5.3 Heavy Vehicles

YOLOv3 is the detector used for heavy vehicles detection. Heavy vehicles are bigger and less frequent in scenes than motorcycles and light vehicles, and their detection does not need the use of a CNN specialized to obtain precise results. The process also starts from the single frame image extracted from the video sequence, which is fed to a YOLOv3 (Redmon & Farhadi, 2018) binary classifier, the image is split in grid regions, a CNN model produces a fixed number of bounding boxes along with a class score per grid region.

The version of YOLOv3 uses the *darknet* implementation, found in.⁴ To increase accuracy, the configuration file was changed for width and height from 608 to 700 pixels. DeepSORT uses the same sources as light and motorcycle pipelines for detection and tracking of urban vehicles. The pipeline for heavy vehicles detection and tracking is shown in Fig. 9.

6 Experiments

The processes described here are evaluated by means of the following experiments, which initially provide detection results and then the final tracking results. The computer server used has 2 CPU E5-2683 v4 @ 2.10 GHz, 64 GB of RAM, a GPU Titan XP with 12 GB of memory and the algorithms run on the Ubuntu 16.04.6 operating system.

6.1 Detection and Classification

YOLOv3 on dataset A: For first experimentation, dataset A was evaluated using YOLO v3 with weights (network parameters) already trained on COCO dataset (Lin et al., 2014). The model achieves an average precision (AP@0.5) of 0.578 on light vehicles, 0.521 for heavy vehicles and 0.333 for motorcycles. This results forces to train the

² <https://github.com/AlessioTonioni/tf-objdetector>

³ https://github.com/abhyanrika/nanonets/_object_tracking/

⁴ <https://github.com/pjreddie/darknet/>

Fig. 7 Pipeline of the method for light vehicle detection and tracking

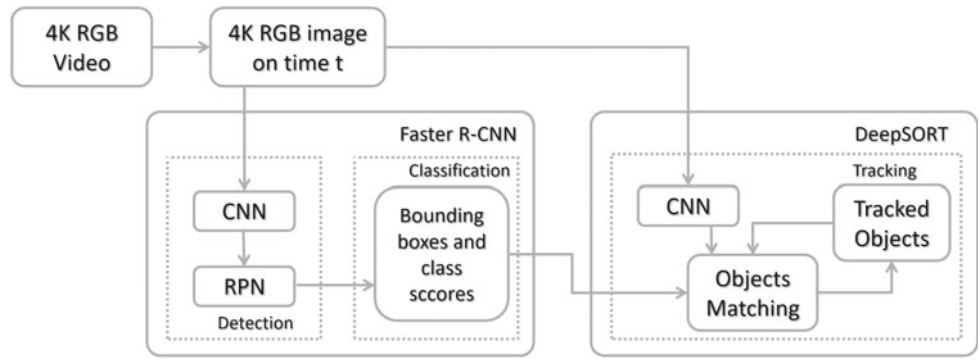


Fig. 8 Pipeline of the method for motorcycle detection and tracking

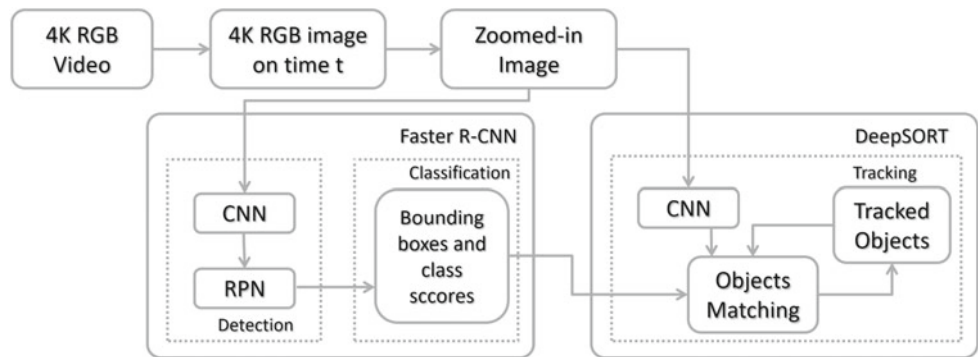
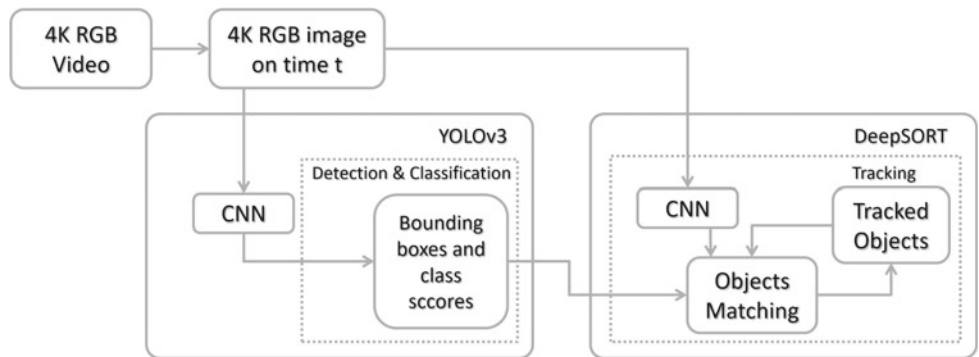


Fig. 9 Pipeline of the method for heavy vehicle detection and tracking



classifiers from scratch, since the results described are not sufficiently accurate.

YOLOv3 trained on dataset B: Dataset B was used for train YOLOv3, configuring three binary classifiers, one for each class (light vehicle, heavy vehicle and motorcycle) and evaluated on datasets B and C. The training and testing ratio of the dataset corresponds to 80/20, that is 5,200 images for training and 1,200 for test. The results for light and heavy vehicles classes were outstanding reaching a AP@0.5 of 0.999 evaluating dataset B. Meanwhile the binary classifier configured and trained for motorcycles resulted in a poor AP@0.5 of 1%, generating many false positives, miss classifying pedestrian as motorcycles. This can be explained because YOLOv3 reduces all input images to a fixed size of

608 × 608 pixels as its input layer size. Experiments increasing the input size up to 68%, and reducing the number of filters in each YOLO v3 convolutional layer did not improve results and the AP@0.5 remained less than 1% for motorcycle detection.

The binary classifier trained in dataset B is used to evaluate dataset C. The generalization achieved on this dataset does not produce good results as in dataset B. Light vehicles achieve only an AP@0.5 of 0.107, and even worse in heavy vehicles with an AP@0.5 of 0.043. Table 4 summarizes the obtained results.

YOLOv3 trained on datasets B and C: To improve AP and reduce overfitting, YOLOv3 was trained from scratch on datasets B and C, working just on light (LV) and heavy

Table 4 YOLOv3 trained on dataset B performance using average precision (AP@0.5) metric

Classifier	Dataset B	Dataset C	1.15 × Dataset B	1.68 × Dataset B
LV	0.999	0.107	–	–
HV	0.999	0.043	–	–
Motorcycle	<0.01	–	<0.01	<0.01

Table 5 Performance of YOLOv3 trained on datasets B and C using (AP@0.5) metric

Classifier	Dataset B	Dataset C	Total
LV	0.895	0.421	0.552
HV	0.975	0.960	0.962

vehicles (HV). The training and testing ratio of the combined datasets B and C corresponds to 80/20, that is 13,648 images for training and 3,412 for test. Results are shown in Table 5 for an AP@0.5.

The frame rate of all classifiers is approximately 11.48 frames per second, achieving real time (29.09 f/s) if the image resolution is reduced to 2 k. The classifier used for Heavy vehicles (HV) achieved good performance in datasets B and C, nevertheless the binary classifier for light vehicles failed to perform well on dataset C. In the other hand, YOLOv3 was used also to implement a binary motorcycle detector, which resulted in a disappointing AP@0.5 of less than 1%.

Faster R-CNN trained on datasets B and C: To improve accuracy for motorcycles and light vehicles detectors, the Faster R-CNN (Ren et al., 2015) architecture was used, with ResNet-101 (He et al., 2016) as the CNN-based network.

The training and testing ratio was also 80/20, that is 13,648 images for training and 3412 for test, as the former experimentation. On dataset B, using the same training and testing ratio, the binary motorcycle classifier achieves an AP@0.5 of 0.734 and 0.303 on dataset C. A Light vehicles binary classifier was trained only on dataset C achieving an AP@0.5 of 0.729.

To improve results for motorcycle detection, dataset B and C were zoomed to 150% and 175% respectively, and used for retrain the Faster R-CNN motorcycles binary classifier. Final results achieve an AP@0.5 0.984 for dataset B and 0.945 on dataset C (see Table 6).

Video results for detection of motorcycles can be found in.⁵ For light vehicles detection in,⁶ and detection on heavy vehicles in.⁷

6.2 Vehicle Tracking

As stated earlier, DeepSORT (Wojke et al., 2018) is the algorithm selected for tracking. We evaluate the results using the metrics described on Sect. 3.2. Deep association metric was retrained using a training set regions of interest (RoI) extracted from datasets B and C. In this way an appearance descriptor for motorcycles, light and heavy vehicles was created allowing to extract descriptive features from the three classes.

Table 7 shows tracking results using detections from YOLOv3 trained on dataset B, AP@0.5 indicates the detection quality. LV and HV stands for light and heavy vehicles respectively.

Table 8 shows the tracking results obtained using detections from YOLOv3 algorithm trained on datasets B and C. Again AP@0.5 indicates the detection quality. Finally, Table 9 shows the tracking results obtained using the detections from Faster R-CNN trained on motorcycles for datasets B and C. Note the change in average precision (AP) from the detector to the tracker, a visual inspection shows that motorcycle detection specially on Dataset C appears to produce intermittent detections, causing imprecise tracks for the DeepSORT algorithm.

The motorcycles tracking results can be visualized in this link⁸, tracking results on light vehicles in,⁹ and tracking results of heavy vehicles in.¹⁰

7 Discussion

Current state-of-art demands representative dataset with enough examples to train machine learning algorithms which provide significant generalization. The use of UAVs for

⁵ <http://videodatasets.org/UTUAV/motorcycle-D.htm>

⁶ <http://videodatasets.org/UTUAV/LV-Det.htm>

⁷ <https://videodatasets.org/UTUAV/HV-Det.htm>

⁸ <http://videodatasets.org/UTUAV/motorcycle-T.htm>

⁹ <http://videodatasets.org/UTUAV/LV-Track.htm>

¹⁰ <http://videodatasets.org/UTUAV/HV-Track.htm>

Table 6 Faster R-CNN trained on datasets B and C performance using average precision (AP) metric

Classifier	Dataset B	Dataset C
LV	–	0.729
Motorcycle	0.734	0.303
Zoomed-in motorcycle	0.984	0.945

Table 7 DeepSORT results for YOLOv3 trained on dataset B

Tested on	AP@0.5	MOTA	MOTP
LV on dataset B	0.999	0.991	0.821
LV on dataset C	0.107	0.052	0.665
HV on dataset B	0.999	0.996	0.796
HV on dataset C	0.043	0.036	0.705

Table 8 DeepSORT results for YOLOv3 trained on datasets B and C

Tested on	AP@0.5	AP tracker	MOTA	MOTP
LV on dataset B	0.895	0.812	79.1	64.1
LV on dataset C	0.421	0.467	42.6	60.9
HV on dataset B	0.975	0.949	94.0	63.2
HV on dataset C	0.960	0.922	92.3	64.0

Table 9 DeepSORT results for Faster R-CNN trained on dataset B and C

Tested on	AP@0.5	AP tracker	MOTA	MOTP
Motorcycle on dataset B	0.734	0.500	0.723	0.519
Motorcycle on dataset C	0.303	0.033	0.646	0.094

obtaining top-view sequences avoids occlusion, one of the most frequent problems of ITS surveillance video dataset, and provides a wider range of visual capture, improving detection and tracking results and providing new tools for traffic analysis. Therefore, the dataset results presented in this chapter will allow researchers in this field to establish and improve upon the state-of-the-art.

Vehicle detection algorithms previous to deep learning irruption require significant intervention to determine parameters related with appearance features or those based on spatio-temporal information (e.g. background subtraction). Deep learning allows the implementation of a straightforward strategy. However, it requires many examples to produce accurate results. There is also consideration related to the ratio of detected objects in comparison with the total size of the image, very small objects will generate failures in detection. The dataset provides enough examples and can be used also for future improvement of detection and tracking algorithms.

Detection results obtained using high resolution images show a remarkable accuracy AP@0.5 of almost 90% working with YOLOv3 detector, nevertheless when the object appears to be too small in proportion to the high resolution image it was necessary to appeal to the Faster-RCNN detector and it was even necessary to perform

the detection analysis in sub-selected areas, as was the case with motorcycles. As is expected for tracking by detection algorithms, DeepSORT best results are obtained with best detection scores.

8 Conclusion and Future Work

This chapter presented a methodology for vehicle detection and tracking in urban scenarios. The model is trained and validated on the new UTUAV Urban Traffic Dataset which has more than 2.5 million annotated objects divided in three categories: light vehicles, heavy vehicles and motorcycles. Each object has a unique identifier making the dataset suitable for training and evaluate Multiple Object tracking (MOT) algorithms. The main advantage of datasets B and C is the capture angle which allows to reduce object occlusions and preserve the size of the objects during their time of life on the video-sequence.

State-of-the-art detectors were implemented in conjunction with tracking as a pipeline methodology that integrates the detection and tracking process. YOLOv3 (Redmon et al., 2016) was more successful for the detection of bigger objects obtaining performance close to real time. To deal with small objects Faster R-CNN (Ren et al., 2015) exploits

the use of the region proposal network generating more accurate results, which were improved using an image zoomed-in approach.

DeepSORT (Wojke et al., 2018) was used as a tracking algorithm which strongly relies in the quality of detections to produce good results in tracking.

Future work will exploit the use spatio-temporal along with visual appearances information of videos to improve detection algorithms. In the other hand, tracking algorithm as DeepSORT can implement strategies to deal with longer occlusions. The UTUAV vehicle dataset captures vehicles that sometimes stop under a big tree and their image is lost for a long time. Then it is necessary to keep the memory buffer for objects that exhibit such behaviour.

Acknowledgements This work was partially supported by COL-CIENCIAS project: Reduccion de Emisiones Vehiculares Mediante el Modelado y Gestion Optima de Trafico en Areas Metropolitanas—Caso Medellin—Area Metropolitana del Valle de Aburra, codigo 111874558167, CT 049-2017. Universidad Nacional de Colombia. Proyecto HERMES 25374. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research. Specially thanks to Eng. MSc. Jose Arroyo on whose Master's thesis this work was largely based.

References

- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110, 346–359.
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. In *Proceedings—International conference on image processing, ICIP* (pp. 3464–3468).
- Buch, N., Velastin, S. A., & Orwell, J. (2011). A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12, 920–939.
- Buch, N., Orwell, J., & Velastin, S. A. (2009). 3D extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes. In *Proceedings of the British machine vision conference on BMVC 2009*.
- Buliali, J. L., Faticah, C., Herumurti, D., Fenomena, D., Widyastuti, H., & Wallace, M. (2017). Vehicle detection on images from satellite using oriented fast and rotated brief. *Journal of Engineering and Applied Sciences*, 12, 4500–4503.
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary robust independent elementary features. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6314 LNCS, 778–792.
- Cheng, H.-Y., Weng, C.-C., & Chen, Y.-Y. (2011). Vehicle detection in aerial surveillance using dynamic bayesian networks. *IEEE Transactions on Image Processing*, 21, 2152–2159.
- Dalal, N., Triggs, W. (2004). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition CVPR05* (vol. 1, pp. 886–893).
- Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 743–761.
- Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., & Tian, Q. (2018). The unmanned aerial vehicle benchmark: Object detection and tracking. *CoRR*, abs/1804.00518.
- Espinosa, J. E., Velastin, S. A., & Branch, J. W. (2018). Motorcycle detection and classification in urban scenarios using a model based on faster R-CNN. [arXiv:1808.02299](https://arxiv.org/abs/1808.02299) [cs], August 2018.
- Espinosa-Oviedo, J. E., Velastin, S. A., & Branch-Bedoya, J. W. (2019). EspiNet V2: a region based deep learning model for detecting motorcycles in urban scenarios. *Dyna*, 86, 317–326.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 303–338.
- Elgammal, A., Harwood, D., & Davis, L. (2000). Non-parametric model for background subtraction. In *European conference on computer vision* (pp. 751–767). Springer.
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*.
- Goerick, C., Noll, D., & Werner, M. (1996). Artificial neural networks in real-time car detection and tracking applications. *Pattern Recognition Letters*, 17, 335–343.
- Han, B., Comaniciu, D., & Davis, L. (2004). Sequential kernel density approximation through mode propagation: Applications to background modeling. *ACCV*, 32, 16.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. In *IEEE transactions on systems, man, and cybernetics* (vol. SMC-3, pp. 610–621).
- Harris, C., Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey vision conference 1988* (pp. 23.1–23.6).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 770–778).
- Hsieh, J. W., Chen, L. C., & Chen, D. Y. (2014). Symmetrical SURF and Its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, 15, 6–20.
- Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin De La Société Vaudoise Des Sciences Naturelles*, 37, 547–579.
- Kalinke, T., & Tzomakas, C., & Seelen, W. V. (1998). A texture-based object detection and an adaptive model-based classification. *IEEE Intelligent Vehicles Symposium*, 341–346.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 35–45.
- Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., & Russell, S., I. I. (1994). Towards robust automatic traffic scene analysis in real-time. In *Proceedings of 12th international conference on pattern recognition* (pp. 3776–3781).
- Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kamarainen, J.-K., Zajc, L. C., Danelljan, M., Lukezic, A., Drbohlav, O., He, L., Zhang, Y., Yan, S., Yang, J., & Fernandez, G., et. al. (2020). *The eighth visual object tracking VOT2020 challenge results*. Springer.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances In Neural Information Processing Systems*, 1–9.
- Kuehne, A. (1991). Symmetry-based recognition of vehicle rears. *Pattern Recognition Letters*, 12, 249–258.
- Lee, G., & Mallipeddi, R. (2015). A genetic algorithm-based moving object detection for real-time traffic surveillance. *Signal Processing Letters*, 22, 1619–1622.
- Lin, T.-Y., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L.

- (2014). Microsoft COCO: Common objects in context. *CoRR*, abs/1405.0312.
- Lou, J., Yang, H., Hu, W., & Tan, T. (2002). Visual vehicle tracking using an improved EKF*. In *Proceedings on ACCV2002: The 5th Asian conference on computer vision* (pp. 23–25).
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 91–110.
- Mariano, V. Y., Min, J., Park, J.-H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., & Drayer, T. (2002). Performance evaluation of object detection algorithms. In *Proceedings on 2002 international conference on pattern recognition* (pp. 965–969).
- Moranduzzo, T., & Melgani, F. (2014). Automatic car counting method for unmanned aerial vehicle images. *IEEE Transactions on Geoscience and Remote Sensing*, 52, 1635–1647.
- Oliver, N., Rosario, B., & Pentland, A. (1999). A Bayesian computer vision system for modeling human interactions. *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1542, 255–272.
- Rad, R., & Jamzad, M. (2005). Real time classification and tracking of multiple vehicles in highways. *Pattern Recognition Letters*, 26, 1597–1607.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *CoRR*, abs/1804.02767.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 779–788).
- Ren, S., & He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*.
- Restrepo, R., *IoU scores*.
- Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3951 LNCS, 430–443.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE international conference on computer vision* (pp. 2564–2571).
- SCALE. (2015). *Open datasets—Scale*.
- Seki, M., Wada, T., Fujiwara, H., & Sumi, K. (2003). Background subtraction based on cooccurrence of image variations. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (vol. 2).
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations, ICLR 2015—Conference track proceedings* (pp. 1–14).
- Sivaraman, S., & Trivedi, M. M. (2010). A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 11, 267–276.
- Soviany, P., & Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In *Proceedings—2018 20th international symposium on symbolic and numeric algorithms for scientific computing, SYNASC 2018* (pp. 209–214).
- Stauffer, C., & Grimson, W. E. L. (2003). Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition* (pp. 246–252).
- Tian, B., Morris, B. T., Tang, M., Liu, Y., Yao, Y., Gou, C., Shen, D., & Tang, S. (2017). Hierarchical and networked vehicle surveillance in ITS: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 18, 25–48.
- Tsai, L. W., Hsieh, J. W., & Fan, K. C. (2007). IEEE transactions on image processing. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (vol. 16, pp. 850–864).
- Velastin, S. A., Fernández, R., Espinosa, J. E., & Bay, A. (2020). Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera. *Sensors*, 20, 6251.
- von Seelen, W., Curio, C., Gayko, J., Handmann, U., & Kalinke, T. (2000). Scene analysis and organization of behavior in driver assistance systems. In *Proceedings 2000 international conference on image processing* (pp. 524–527).
- Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.-C., Qi, H., Lim, J., Yang, M. -H., & Lyu, S. (2015). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv preprint arXiv:1511.04136*
- Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.-C., Qi, H., Lim, J., Yang, M.-H., & Lyu, S. (2020). *UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking*.
- Wojke, N., Bewley, A., & Paulus, D. (2018). Simple online and realtime tracking with a deep association metric. In *Proceedings—International Conference on Image Processing, ICIP* (pp. 3645–3649).
- Woo, J. W., Lee, W., & Lee, M. (2010). A traffic surveillance system using dynamic saliency map and SVM boosting. *International Journal of Control, Automation and Systems*, 8, 948–956.
- Yin, F., Makris, D., Velastin, S. A., & Orwell, J. (2010). Quantitative evaluation of different aspects of motion trackers under various challenges. *British Machine Vision Association*, 5, 1–11.
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (vol. 8689 LNCS, pp. 818–833).
- Zhang, B. (2013). Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Transactions on Intelligent Transportation Systems*, 14, 322–332.
- Zhu, P., Wen, L., Du, D., Bian, X., Hu, Q., & Ling, H. (2020). *Vision meets drones: Past, present and future*. *arXiv.org*.
- Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. *Proceedings—International Conference on Pattern Recognition*, 2, 28–31.



Tracking Everyone and Everything in Smart Cities with an ANN Driven Smart Antenna

Herman Kunsei, Paul R. P. Hoole, K. Pirapaharan, and S. R. H. Hoole

Abstract

Smart cities require the application of human brain-like intelligence systems for sensing, monitoring, decision making and action. In this chapter this scenario is addressed in a two-fold manner. First, the function and requirements placed upon telecommunications systems are described and the future directions and challenges are discussed specifically in relation to the sixth generation (6G) wireless systems which are both different from the wireless systems up to 5G and are expected to play a critical role in smart cities. Secondly, the chapter presents a novel artificial intelligence (AI) driven smart antenna that is both computationally efficient and low on memory use, with the ability to track machine to machine, M2M, and machine to human, M2H, communications in the smart city. Smart cities depend on smart Internet of Things (IoT) management to ensure that infrastructure is monitored and corrected if there is a failure as in the case of emergency systems. This chapter discusses the application of AI in the physical layer in addressing connectivity issues of IoT in the wide area network and cellular systems landscape. The chapter address the connectivity of wireless systems of low power consumption, enhanced coverage, low latency communications, small bursts of

data and the number of devices (running into hundreds of thousands). Central to the IoT management system is a novel artificial neural network (ANN) driven antenna that may rapidly be set for continuous rotating communication with several machines or smartphones, as well as focus in specific areas of intense or critical activity. Furthermore, the ANN driven array smart antenna will be coded to track moving targets.

Keywords

Artificial intelligence • Machine learning • 6G • Wireless networks • Internet of Things • Smart cities • Artificial neural network • Smart antennas

1 Introduction

For 21 centuries, humans have maintained their cities for their survival with very little real-time feedback from the cities. The smart cities of today and tomorrow will be more connected in terms of human to everything and everything to everything. The objective of the massive connectivity is to enable the smart city to adapt and offer the best possible service to human settlers. This implies that the buildings will be smart, homes will become smart, means of obtaining health care will become timely and smart, vehicles will become greener and smarter, learning will become smarter, the way people live will be enhanced and comfortable (Pelton & Singh, 2019). Collectively, the smarter the services and infrastructure of a city becomes, drives the city to become a smart city. The services and infrastructure become intelligent through the sharing of information, learning from the data collected and making informed decisions. These functions and others can only be efficient if the communication systems are intelligent.

This chapter focuses on the use of intelligent communication systems to enhance and enable the performance and

H. Kunsei (✉) · P. R. P. Hoole
Department of Electrical and Communications Engineering,
Papua New Guinea University of Technology, Lae,
Papua New Guinea
e-mail: herman.kunsei@pnguot.ac.pg

P. R. P. Hoole
e-mail: paul.hoole@pnguot.ac.pg

P. R. P. Hoole
Wessex Institute of Technology, Southampton, UK

K. Pirapaharan
University of Jaffna, Jaffna, Sri Lanka

S. R. H. Hoole
Formerly of Department of Electrical and Computer Engineering,
Michigan State University, Houghton, USA

operations in a smart city. A smart city is a sustainable and digitally intelligent city that employs ICT and other digitally driven transducers and sensors to meet the needs of city dwellers, both present and future, improving the quality of life and the efficiency of the businesses, industries and homes that make up the city (Mahmood & Smart, 2018; Pelton & Singh, 2019; Sun et al., 2018). As such telecommunication systems play a central role in the makeup of the smart city. More specifically, intelligently adaptive sensors and transducers at which the wireless or wired communication systems terminate, play a critical role in smart cities (Hoole, 2020). In this chapter, we shall give a specific example of a new, memory efficient and fast array antenna system that may be used with both sensors and transducers at the moving of node points of a smart city. The nodes may be stationary, a house for instance, or mobile, such as a transport vehicle. The AI enabled smart antenna reported herein may be used for both stationary and mobile machine-to-machine (M2M) or device-to-device (D2D) wireless communication systems (Marzuki, et al., 2021; Pirapaharan et al., 2021; Senthilkumar, et al., 2021).

Such intelligent, fast, and low-on-digital-memory-use antennas become the critical eyes and ears, as well as the transducer driving muscle termination points of a smart city. These antenna terminal points are integral parts of smart buildings, smart mobility, smart energy, smart water, smart waste, smart management and smart digital layers of a smart city. These various activities of a smart city increase since 60% of the world population is expected to dwell in cities, with a decreasing percentage in rural areas. In 2050 the percentage of city dwellers is expected to increase to 75%. And all these cities spread over the globe are expected to occupy only 5% of the surface of the earth. With a high density of people their various demands and activities necessitate an integrated city structure that uses reliable broadband communication networks and an efficient ecosystem for the wide variety and fast operating digitally driven systems that are all interconnected by the Internet or the application of IoTs (Cicirelli et al., 2019; Mahmood & Smart, 2018).

The AI array antenna and associated wireless communication system form the fundamental building block that connects all layers of the IoT devices including M2M communications and data processing, for a wide range of interconnected applications. These applications include: (a) the logistics business analytics, predictive maintenance and high level factory automation of industry, (b) remote monitoring and user consumption control of utilities including gas, electricity and water, (c) traffic monitoring, improving road safety, driverless transportation and public safety of transportation, (d) wearables for games and leisure industries, and fitness and health monitoring, (e) residential, industrial and public security, remote surveillance, remote tracking, alarms, and protection of public infrastructure from

vandalism, damage and theft, and (f) clean environment, green energy, smart homes and smart buildings (Pelton & Singh, 2019).

The chapter shall first survey the intelligent communications systems that are available for use in smart cities. The next section will specifically focus on wireless systems that may be used for both long distance and short distance communication, namely the 5G and future 6G wireless systems. Some keys research challenges in effective communications system are discussed. In the final section of the chapter, we shall present a novel, fast and low memory smart antenna system that uses AI techniques to generate narrow, rapidly steerable beams for IoE communications in smart cities. We discuss three applications of smart cities that can benefit from the ANN-enabled smart antenna.

2 Smart City and Intelligent Communications

The Smart city Internet of Things (IoT) system requires highly reliable optical communication systems or wireless systems for remote M2M communications, monitoring and control. The advantages of a wireless system compared to a wired system are that its set up costs are relatively low and changes at existing sites are more easily done, whereas the artificially intelligent 5G smart antenna reported in this chapter is of immense help in overcoming the two important disadvantages of wireless communications systems. There is potential for interference between the wireless signals in free space and signals of sensors and other networks. Moreover, the directive and rotatable beams of the advanced beamformers presented herein help overcome delays in communicating alarm and emergency data, as well as tracking and communicating with moving targets such as vehicles. Furthermore, intelligent antennas help in network virtualization, where there is a need to operate different networks with different characteristics and functions. With multiple network nodes with steerable, narrow beam, adaptive antennas, it is possible to optimize virtual network routes, as well as to optimize the virtual network capacity allocation based on communications traffic. In Fig. 1 is shown the four major components of a smart city. These are the home, industry, commercial/business and city ecosystems (Pelton & Singh, 2019). Each has its own specialized functions and requirements in which people centered, integrated and context specific design of the city is required. The brain that enables the effective function of a smart city is its communication system that includes the 5G wireless, WIFI, LTE, WiMAX, Femtocell and optical fiber communications systems. Of these the two most important systems are the 5G (and future 6G) wireless and wired optical communications systems (Sun et al., 2018).

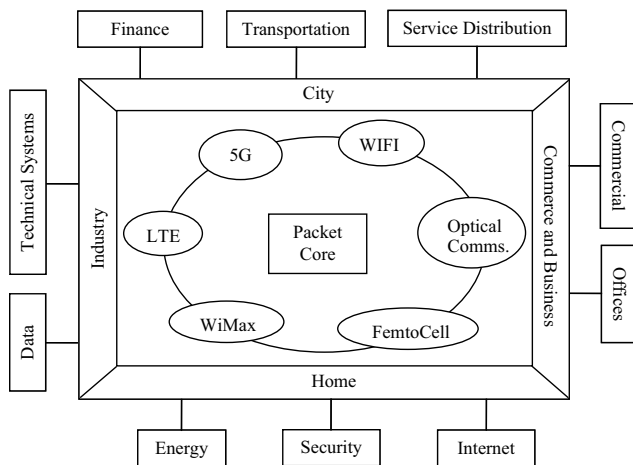


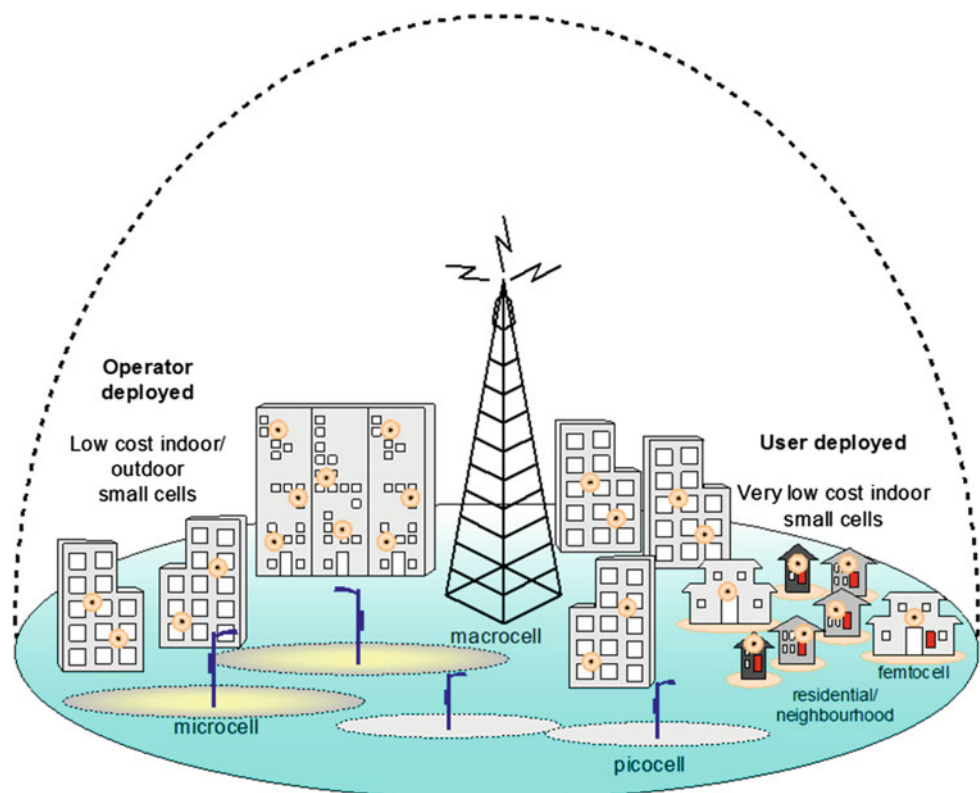
Fig. 1 Smart city integrated by the communications, command and control networks

In Fig. 1 is shown how the communications system is central to the smart city in all its various domains, namely, the city, homes, industries, and business and commerce. All four domains include the following technologies: Internet of Things, M2M and H2H communications, smart electric grids, driverless transport, virtual technologies, artificial intelligence and automation; besides cyber security, tele-education, tele-health services and telemetry architecture. In every single one of these wireless systems smart

antennas will play a key role in wideband as well as narrow band operation of the smart city. Therefore, in whatever aspects of the planning of the city, due consideration needs to be given to the full exploitation of the developing technology of intelligent array antennas discussed in this chapter, since this forms the node points of the communications network that integrates all the different technologies. Hence intelligent wireless communications systems are specifically expected to play a crucial role in smart city planning, which includes, meeting the needs of the citizens, and enhancing competitiveness, sustainability, supportive infrastructure and service industry, security and artificial intelligence technologies.

The deployment of dense small cells overlaid in the existing HetNets (Fig. 2), has been identified as a feasible strategy to expand the network coverage and address the explosive growth of mobile data traffic. Relying solely on macro-cell base stations is no longer an effective strategy, as the data volumes with associated uneven traffic distributions have increased tremendously in recent times. Moreover, deploying additional macro-cells would not be a viable solution, due to their high installation cost and the lack of suitable sites for deployment. Thus, modified networks are required in which smaller cells can be deployed within the existing macro-cell area to form HetNets (Marzuki et al., 2021).

Fig. 2 An example of HetNet deployment (Marzuki et al., 2021)



By reducing the cell size, the use of the limited, available bandwidth from the already scarce spectrum resources can be optimized by adopting frequency reuse. Moreover, as these cells offer local traffic off-loading to a smaller number of users, larger portions of resources can be accommodated for their associated users. The shorter distance between these low-powered base stations and user devices prolongs user battery life, thus increasing both the energy efficiency and signal-to-interference-plus-noise ratio (SINR) due to the low loss path. Although small cells (i.e., microcells, picocells and femtocells), can facilitate high bandwidth and wireless user ubiquity, densely deployed networks introduce a new set of challenges such as backhaul connectivity, resource allocation and energy management issues. Moreover, small cell densification also implies high inter-cell interference (ICI) among these cells due to their close proximity and arbitrary deployment (Marzuki et al., 2021).

Small cell backhaul solutions can be either wired or wireless, depending on the required network coverage, installation complexity, and associated cost (Pirapaharan et al., 2021). Thus, providing high-capacity connectivity between small cells and core network may require extensive planning before an optimal solution can be achieved. Wired connectivity is straightforward and is commonly used in small cell deployments. However, when these small cells are deployed outdoors, the backhaul connectivity becomes more complex. This raises a concern, as 75% of outdoor small cells may be backhauled by using wireless connectivity in the future. Therefore, efficient backhaul solutions utilizing wireless links must be designed to address both 5G spectral and power requirements. In the 4G small cell deployment, user traffic demand is backhauled to the network operator via broadband gateways such as DSL cables. The absence of dedicated wired backhaul links can diminish the network real-time Quality of Services (QoS) (Marzuki et al., 2021).

Indirectly, this will cause the sub-optimal small cell access point placement issue, considering that a high cost is involved in connecting these small cells to an existing infrastructure. Thus, the main motivation in deploying small cells in cell edge regions, and in areas with high user density, will not be realized.

Several studies have focused on determining the optimal small cell deployment while considering different performance metrics. To realize the random and flexible deployment of these small cells, optimal cell placements in an existing infrastructure must be accomplished. A theoretical framework for small cell placement is designed to maximize the network spectral efficiency and mitigate co-tier interference. An optimal solution of backhaul design for small cell networks is developed by taking the network requirement constraints, such as coverage and capacity, into consideration. Small cell placement strategies are used to achieve a

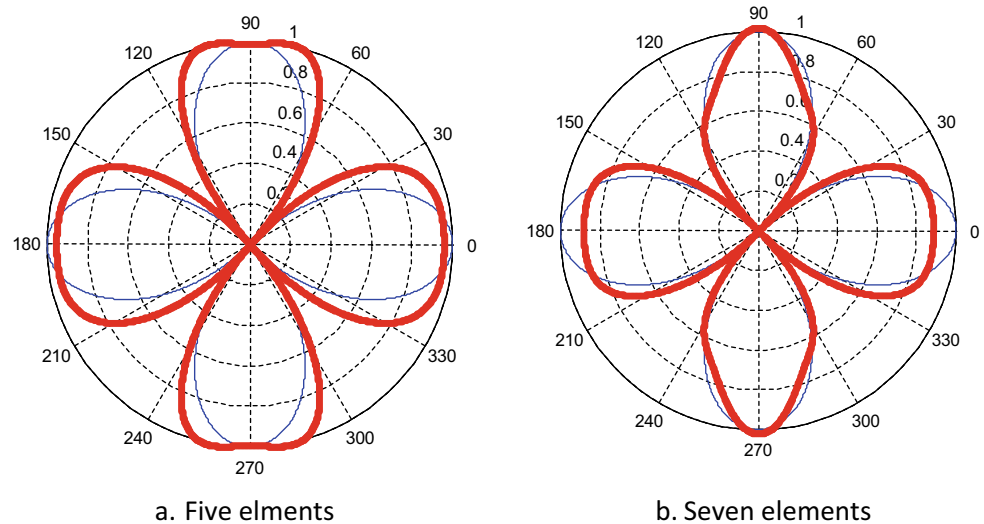
fault-tolerance network with the use of small cell self-healing features (Marzuki et al., 2021).

Despite the potential of small cells in delivering high data rates to the ever-increasing number of users, another research challenge remains, particularly with respect to resource allocation to users in small cells. Small cells, specifically femtocells, are deployed by the end user with minimal intervention from network operators. Short distances between small cells cause overlapping coverage areas and hence inefficient resource management and associated interference problems. In such scenarios, inefficient resource management will result in severe interference problems. A cluster-based resource allocation scheme has been developed for femtocell-assisted macro-cellular networks. This scheme is developed to mitigate both cross- and co-tier interference in the network. Other solutions proposed are a dynamic resource allocation approach and fractional frequency reuse.

It is anticipated that small cell densification will play a critical and growing role in future wireless communications, by enabling the network service providers to deliver high data rates through increased frequency reuse, enlarged coverage areas, and greater spectral and energy efficiencies. For densification, the wireless antenna beams need to be narrowly directed in specified directions and be able to track in the case of mobile M2M communications. Such an antenna is shown in Fig. 3, where the antenna beams, for instance, may be directed along long narrow roadways communicating with moving vehicles. It is seen that there is maximum radiation over a wider area than what is required by the desired beam (Senthilkumar et al., 2021). However, this is rectified by increasing the number of elements from five to seven, thus getting greater accuracy at the cost of computational time and the cost of adding extra elements. As we have expected, with an increased number of elements, adaptive array beamforming is much closer to the desired beam. However, the amplitudes in the 0° and 180° directions are better for the five element array antenna than for the seven element array. This is due to the characteristics of the desired beam selected. In order to have the comparison of accuracy of weights optimized using the SNWOM method with the weights optimized using the traditional LMS method, the weights are calculated for five element and seven element array smart antennas using LMS optimization. The radiation patterns for five and seven elements optimized from LMS methods are shown in Fig. 3a and Fig. 3b, respectively (Senthilkumar et al., 2021).

The communications systems that are widely employed in smart cities include the following six technologies: (i) The IEEE 802.15.4 which defines the physical layer (PHY) and medium access control (MAC) layer specifications for low rate wireless personal area network, (ii) WiFi where the IEEE 802.11 standard providing high data rate

Fig. 3 Comparison of radiation patterns between optimized beam and desired beam obtained by SNWOM (Senthilkumar et al., 2021)



transmission (1 Gbps) for a limited number of interconnected devices over a short distance, at 2.4, 5 or 60 GHz at bandwidths ranging from 20 to 160 MHz, (iii) Bluetooth providing a low-cost wireless communications personal network over distances less than 100 m, at 2.4 GHz, (iv) RFID for short range connectivity for identification purposes, (v) Low Power Wide Area Network (LPWAN), connecting devices over a wide area, long range data communication, up to distances of 10 km. Some of the well-known systems are LPWAN, Ingenu, LoRa, NWave, Platanas, SigFox, and weightless, (vi) Cellular systems: These were originally used for H2H, human to human communications, and now extended toward M2M communications. These are the most powerful wireless systems, allowing for wide coverage, easy deployment, access to dedicated spectrum and high security level. The recent 5G systems, making use of intelligent, steerable beam antenna reported in this chapter, allow for narrow band operation, power saving modes, multihop and group based communications, enhanced coverage and ultra-reliable, low latency communications. The system handles a massive number of devices, including mobile devices, as well as small bursts of data (Sun et al., 2018).

3 5G/6G Systems and Other Communications Systems in Smart City with Research Challenges

3.1 5G/6G Systems for Smart Cities

The 5G network was designed with promise of providing eMBB (enhanced Mobile Broadband), URLLC (Ultra Reliable Low Latency Communications) and mMTC (massive Machine Type Communications) (ITU-R &

Detailed specifications of the terrestrial radio interfaces of International Mobile Telecommunications-, 2020 2021). eMBB data rates of 0.1 Gbps are promised with very low latency, decreasing by a factor of 5 of <1 ms and enabling communications from machine-to-machine and IoT. The 6G system is offering even more throughput, almost zero latency and connects anything and everything (Zhao et al., 2020). For that to happen, the new system specification is required to provide uDB (ultra-fast device broadband), smMTC (smart massive machine type connection) and suRLLC (super ultra-Reliable Low Latency Communications) as depicted in Fig. 4. Connectivity is anticipated to reach peak data rates of 1000 Gbps for data intensity applications such VAR. Every other device in the network can experience a data rate of 1 Gbps which enables services and applications in smart cities to be realized. Device-to-device (D2D), Internet of Everything (IoE), vehicle-to-vehicle (V2V), and vehicle-to-everything (V2X) will be added to the massive machine type communications to offer a network for machine and people. With inclusion of AI in the devices and machines, the second major service will see smart massive machine type connection with a connection density of 10^7 devices/km². To have a connection always reliably to support smart cities, the network must be super ultra-reliable with very low latency and delay jitter. 6G promised to offer latency of 0.1 ms and delay jitter of 10^{-3} ms.

Research has commenced in the area of 6G communications systems with interesting new technologies that can enrich the concept of smart cities. Various comprehensive surveys in Zhao, et al. (2020); Zhang et al., 2020; Imoize et al., 2021; Akyildiz et al., 2020) have envisioned the following technologies as shown in Fig. 5. The following technologies will make 6G the enabling communications system for smart cities; artificial intelligence (AI),

Fig. 4 The evolution from 5G to 6G communications systems

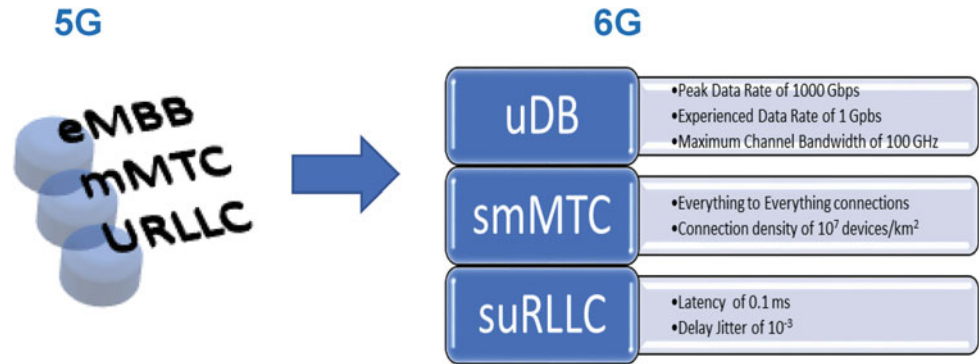
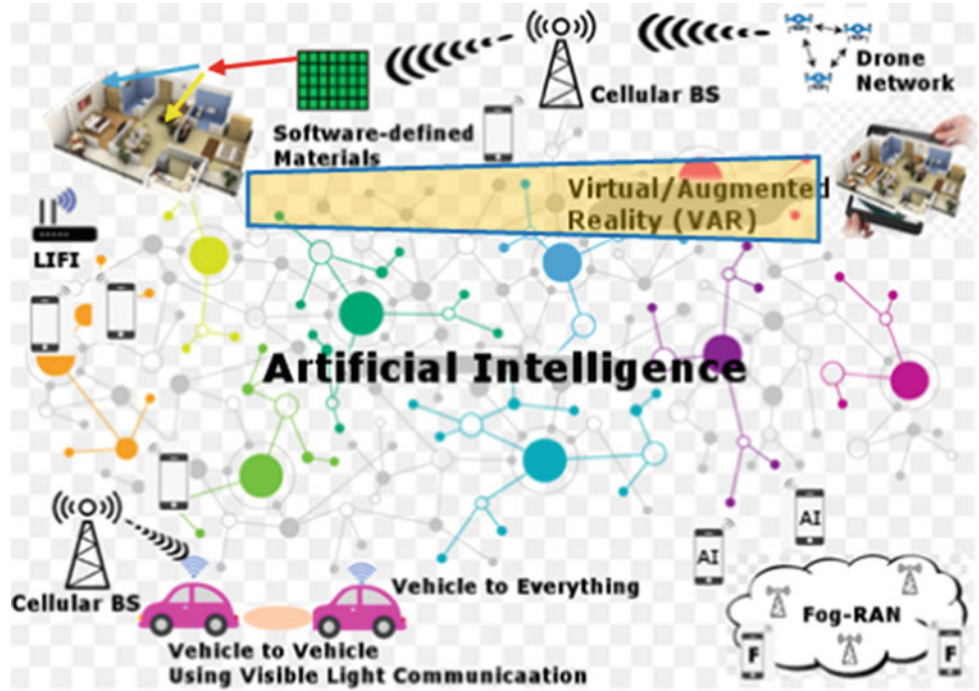


Fig. 5 What's possible with 6G?



AI – AI Enable Device; F – Fog User Equipment; Unlabeled – Smart Device

software-defined materials, virtual and augmented reality, ubiquitous network, index modulation and device-to-device communications.

Artificial Intelligence. In the middle of the communications network is the application of pervasive or collective AI to provide efficient management of the communications network. With the vision to connect everything connectable to the network, resource management for the massive heterogeneous connections will be too much for current technology. Also, the big data available will overpower the current supercomputers; however, AI tools such as machine learning (ML) are expected to analyze the data, learn from it and make better choices in the management of the network (Imoize, et al., 2021; Zhao, et al., 2020). Therefore, AI will be an integral part of 6G.

Software-Defined Materials. As the use of the macro-cell in current mobile network is incapable of providing the high availability for smart city connectivity, complementary technology is needed. The application of an intelligent reflecting surface (IRS) on infrastructure by software-defined materials, is a good candidate to improve network connectivity by creating smart propagation paths for signal transmission (Liu et al., 2021). The IRS consists of an array of IRS units that can be individually controlled by a controller to change the signal characteristics as it reflects off the surface in the propagation direction as shown in Fig. 6 where a base station is communicating with a mobile user via the IRS. With the high user-density and large volume of data generated in a smart city, it is vital that the controller in this system be based on some form of AI (Liaskos et al., 2019). The software-defined materials contain meta-atoms

that allow the currents on the surface of the materials to be manipulated. As the electromagnetic wave touches the surface as it propagates, the meta-atoms can modify the direction, power, frequency spectrum, polarity and phase of the wave (Liaskos et al., 2019). This feature allows the IRS to extend the coverage range as well as capacity of the communications link.

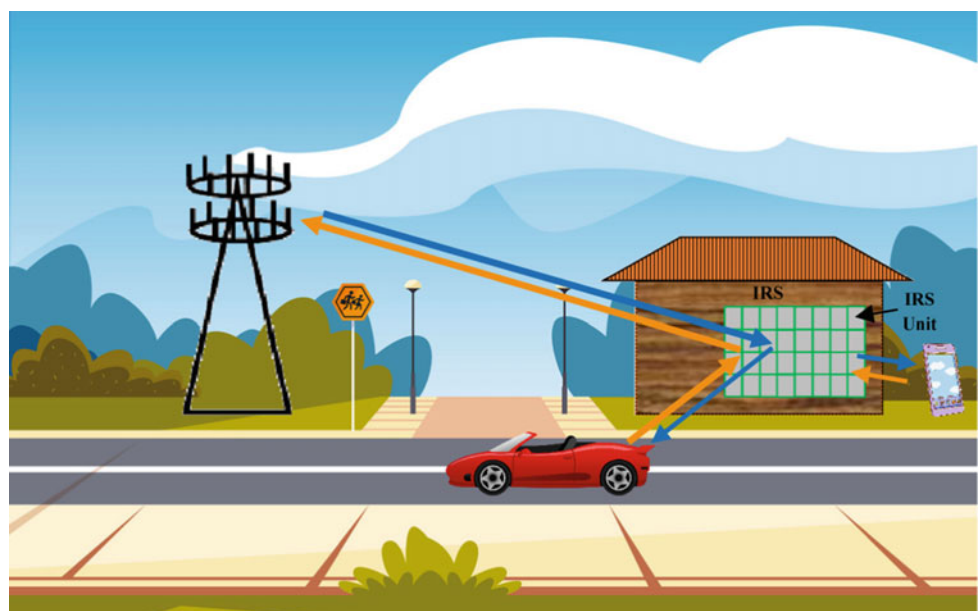
Virtual/augmented reality (VAR). The promise by the 6G network to have continuous connectivity with latency as low as 0.1 ms, provides the platform to have virtual and augmented reality (VAR) in smart cities. The application of VAR can be realized in tourism, navigation, education and disaster response to make the life of people easier and safer (Yagol et al., 2018). A tourist can use virtual reality to evaluate tour packages on offer to make informed choice on available services even before taking the tours. In comparison, in this present era, more tourists make their choices on tourist feedback which may not always be true or accurate. Thus, using virtual reality, tourists can be certain about the quality of service expected, that their security is assured and have instilled in themselves a peace of mind. Having a navigation system ensure that people, including tourists, can get the most out of their travels in the city, is most beneficial. Augmented reality can improve traveling within the smart city with the information about traffic, cab stops and even availability to help people make informed decision regarding their travel. Also, the methods in offering education to its people will improve with the use of VAR in the classroom. Finally, the use of VAR in offering first responders in an emergency makes rescue efforts efficient and offers peace of mind to people living in the smart city that their lives can be

and are protected. The 6G system in smart cities empowers the city to offer a safe and peaceful environment for the people.

Ubiquitous network. In current mobile networks, we experience busy traffic, variable speed and frequent drops in network connectivity (Xiao, 2018). However, smart cities will thrive on a communications network that is always on. Therefore, new network access and connectivity methods are required to support reliable communications in smart cities. A ubiquitous network where any device which can make a network connection can become a connection node in the network to extend coverage and establish a link connection. By doing so, there will always be a path between two communicating devices within the network. These paths can be in space, in air, on the ground and even under the sea. In Fig. 5 we see the drone network in the air which can be used during an emergency when the normal network is disrupted. This can be coordinated from space using satellites. A redundant path to the emergency network can be through submarine cables if the terrestrial links are destroyed. Connectivity is established across any available means. Also, for traffic management, different media can be used to balance traffic and make the network efficient (Zhao et al., 2020). Thus, the ubiquitous network is another equally important technology to enhance communications in smart cities that is made possible with D2D communication.

Device-to-Device Communications. To implement VAR, the ubiquitous network, and Internet of Everything in smart cities would to some degree need the services offered by D2D communications. Also, AI through ML will be an

Fig. 6 Communicating through the intelligent reflecting surface (IRS)



integral part of the smart city communications network. D2D services will require mobile edge computing, network slicing and non-orthogonal multiple access cognitive networking to be effective (Zhang et al., 2020). Mobile edge computing will require some form of AI to manage the resources to be shared among all the devices that needed connectivity within the network edge supporting the smart cities (Imoize et al., 2021; Shahraki, 2021). The smart phones of today will become sensor nodes, speedometers, medical devices, educational tools, search and rescue first respondents' tools and much more. The application of AI will be implemented in the network core and edge to support network slicing. The dynamics environment in smart cities with every user device having heterogeneous latency and processing demands establishes the need for ML to perform adaptive network slicing to satisfy these demands (Nassar & Yilmaz, 2020; Singh et al., 2020; Zhou et al., 2020).

3.2 Other Communications Networks

In the effort for increased speed in the backhaul, fiber networks, microwave networks, and satellite communications (SATCOM) networks will also be utilized to fully implement ubiquitous networks to support smart cities' applications and services.

Satellite networks offers a wider footprint and coverage area than any mobile base station. Smart antennas offer increased channel capacity and coverage for SATCOM on-the-move (STOM) (Luo & Gao, 2017). Thus, for smart city application, STOM would replace the macro-cells at the outer edges for unreached enabling connectivity to be established in space to ground. However, IoT in smart cities cannot tolerate the high latency and delay when using satellite communications. The situation is further complicated with high mobility. The movement changes the propagation environment thus disrupting the communications link and reducing link reliability. Also, shadowing (Hornillo-Mellado et al., 2020) is an issue with satellite communications in high rise areas but with the use of IRS, the effects of shadowing can be contained. SATCOM is also a transmission medium in implementing full ubiquitous networks from smart cities in conjunction with fiber and terrestrial communications.

Terrestrial backhaul networks have been the backbone of the communications networks for years. Despite the low speed and minimum coverage area, the network speed is sufficient for IoT connectivity in smart cities. Alternatively, terrestrial links can be used to complement the wider covering non-terrestrial link as a redundant path or load balancing in the network to improve latency and reliability (Wang et al., 2020). Therefore, terrestrial communications

networks will also form part of the communications networks for smart cities mainly for IoT application.

3.3 Research Challenges for Smart Cities

There is much to do in preparing the communications networks to support the known and unknown requirements for smart cities. The known challenges that pose a challenge for smart cities include cybersecurity, handling of the big data to be collected (Akyildiz et al., 2020; Georgescu & Popescu, 2016) and developing effective green energy systems for mobile devices and sensor.

Cybersecurity. The threat to cybersecurity in smart cities is amplified with the availability of network connectivity and people conducting business on the go especially on public transport. Reference (Rubin, 2011) presented some intriguing work by researchers on the possibility of hacking into vehicles, two-way radios and private accounts. It was shown that with the appropriate equipment and knowledge it is possible to take control of a vehicle through one of the many communications systems available for the comfort of the passengers. In smart cities where IoE is an application, the availability of capturing private information will be much easier. Therefore, more awareness and policy formation to safeguard personal information and people in smart cities is much needed. Further to cyber security is the need to handle data securely.

Big Data. The handling of the extensive data generated by every connected thing in smart cities also offers some challenges. The first is the safe handling of personal data to ensure that the privacy of people is protected. The choice of storage will determine the legality of the location and speed in which this data is accessed. If the data is stored further away from the user, there will be delays in the network which thus reduces the overall performance of the network (Georgescu & Popescu, 2016) as in the case of cyber-physical systems such as autonomous vehicles and unmanned aerial vehicles (UAV) (Akyildiz et al., 2020) and analysis ...

Power Consumption. With the use of pervasive AI in the core and AI enabled devices on the edge the power requirement for mobile devices will be a challenge in smart cities. However, by clustering the AI enabled smart devices the computing burden can be shared with edge devices on the cooperated network and thus power consumption is minimized (Zhang et al., 2020). The mechanisms and techniques on how to achieve the load sharing is unknown and is an open research question.

4 A Fast and Light ANN Enabled Antenna for Internet of Everything for a Smart City

High mobility would be the major complementary and enabling concept with smart cities (Obaidat & Nicopolitidis, 2016). Prior to 5G networks, mobility was mostly by humans. Beyond 5G with the implementation of vehicle to everything, V2X, machine-to-machine, M2M, and device-to-device communications, there is a need for a new radio interface. The new radio interface must be able to support the varying need of all applications in implementing their services. A vital requirement for the new radio interface would be sensing the direction of arrival (DoA) and adapting the antenna parameters to track the mobile equipment. Since the mobile equipment is always on the move, real-time tracking in three dimensions is important (Obaidat & Nicopolitidis, 2016). In addition to application need, there are the security threats and variabilities to embrace with open medium and massive data exchanges in networks beyond 5G (Silva & Flauzino, 2016). These two requirements can be implemented in the physical layer with a new radio. This section discusses the artificial neural network (ANN) antenna for smart cities, beginning with an overview of ANN antennas, and their application in smart physical safety and security, smart education, smart energy, smart transportation and smart sensing of human activity. This section concludes with a discussion of the ANN driven antenna with which to support the applications mentioned earlier.

The antenna systems for 5G networks and beyond have to be smart array antennas with multiple input multiple output (MIMO) and it is predicted that the number of smart antenna would rise by 74% by the year 2026 (Markets & R.a., 2021). The key requirement remains that a new radio interface would be necessary to provide mobility for many new and enhanced applications envisioned to support smart cities through IoT (Cicirelli et al., 2019; Markets & R.a., 2021; Sun et al., 2018). An effective mobility function is dependent on a reliable and secure communications link. A reliable MIMO communications link is a secured, high availability

link with less interference. In Reference (Wang et al., 2012) the authors demonstrated the reliability of a MIMO link against malicious attracts or interference and link failures with space–time block coding. The reliability of the MIMO link is improved with the increased number of elements (Senthilkumar et al., 2021; Wang et al., 2012). However, this also increased computation and power consumption. This approach is not applicable to mobile devices. Therefore, we propose that fast and simple MIMO antenna with tracking be employed.

4.1 ANN Antenna Model

The smart antenna and an ANN enabled smart antenna are shown in Fig. 7. In Fig. 7a we show the analytical smart antenna based on an array factor of a 2×2 MIMO antenna system. As shown in Fig. 8a and Fig. 8b the smart antenna can form nulls in the direction of the interfering signal and maximum beam in a desired look angle respectively. The weights, w_1 and w_2 , were calculated numerically and used to plot the array factor. However, in Fig. 7b the weights are determined using an ANN based beamformer in Senthilkumar et al., (2021). The operation of the ANN enabled smart antenna is discussed in the next section with three cases; case 1 involves beamforming, case 1 demonstrates null forming and case 3 demonstrates the tracking ability of ANN enabled smart antenna.

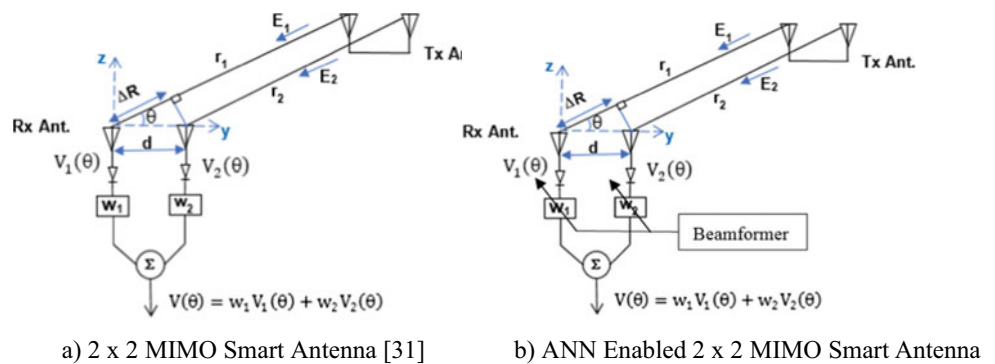
The output of the summer is determined to be

$$V(\theta) = w_1 e^{j\delta_1} V_0 e^{-jk r_1} + w_2 e^{j\delta_2} V_0 e^{-jk(r_1 kd \cos \theta)} \quad (1)$$

where w_1 and w_2 are the weights, δ_1 and δ_2 are the phase and θ being the angle of arrival of the incident wave. The analytical model determines the weights using numerical methods in MATLAB. The case illustrated in Fig. 8 is plot of the normalized array factor from (1) to be:

$$AF_N = AF/2 = \sin(\gamma/2) \quad (2)$$

Fig. 7 a Smart antenna model and b ANN enabled smart antenna



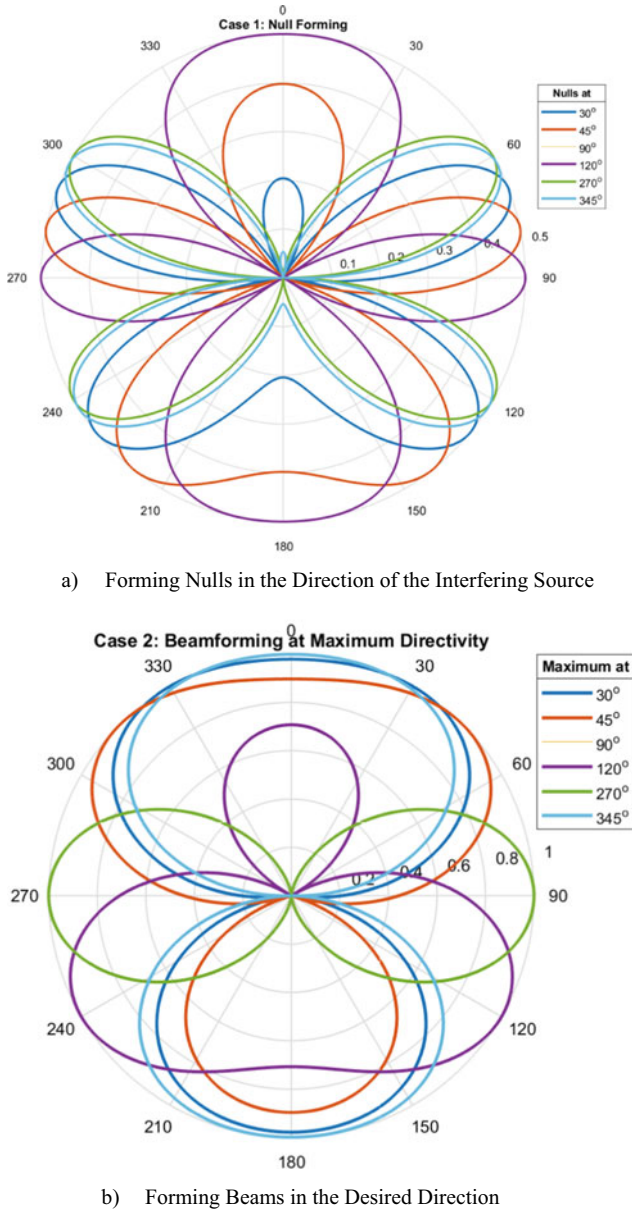


Fig. 8 Analytical smart antenna null and beam forming ability (Kunsei et al., 2021)

with $\gamma = kd(\cos\theta - \cos\theta_1)$ and θ_1 being the direction of interference and

$$AF_N = AF/2 = \cos(\gamma/2) \quad (3)$$

with $\gamma = kd(\cos\theta - \cos\theta_m)$ and θ_m as direction of maximum beam.

The single perceptron ANN use in Fig. 7b is configured with initial weights of -0.001 for w_1 and w_2 with a bias of 0.1 . The learning rate was taken to be 0.0025 . The training data is both -0.001 for both inputs. The performance of the ANN model is discussed in Sects. 4.2 and 4.3 in implementing the tracking algorithm and compared with the analytical model.

4.2 Localization with ANN Enabled Antenna

A step prior to beamforming is to determine the look angle for the antenna to use in the beamformer to form either a null or a beam. It is established that smart antennas can be used for localization based on the angle of arrival (AOA) (Giorgetti et al., 2009). The process usually works off some known reference. In this work we are working off the radiation pattern within the coverage area of the base station.

The concept is illustrated in.

Figure 9, where a beam from the tower is covering some area. The normalized power is maximum with power level of 1 at the tip edge of the radiation pattern. Any movement away from that point would see a drop in the signal power as indicated in the longitudinal centerline. The spine on the pattern indicates that the power level is the same as the spherical distance from the base of the tower would be if there is no interference. Thus, the location of the mobile transceiver can be determined from the signal strength in the beam pair. As the mobile transceiver moves, the power level also changes. Figure 10 shows the change in the power level and the tracking implemented in the analytical beamformer (Kunsei et al., 2021) compared with the SNWOM (Senthilkumar et al., 2021). As shown in Fig. 10, the analytical beamformer agrees with the ANN beamformer despite the very small movement.

The benefit of the proposed tracking methods would reduce the overhead needed to find the beam for user communication, reduce latency and reduce straddling loss (Asplund et al., 2020). If the beam is wide, there will be less handoff taking place in the mobile network. In smart cities where a reliable communications link for device-to-device, machine-to-machine or humans to everything is required, tracking becomes vital.

Other localization techniques include the timing, signal strength, signal pattern matching, directionality, and using Infra-Red (IR) (Bulusu et al., 2000). Most of these methods would require additional instruments to act as reference points for comparison. These reference points would need wired connectivity to operate. Thus, as the network grows the system does not scale well. Directionality methods have

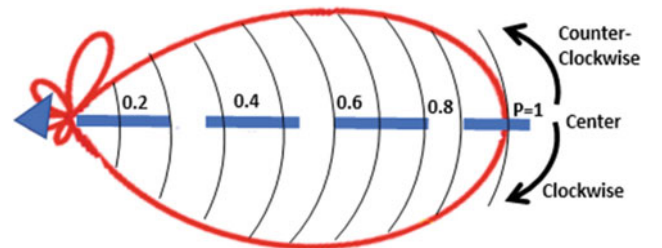


Fig. 9 Beam coverage area showing normalized power levels for tracking

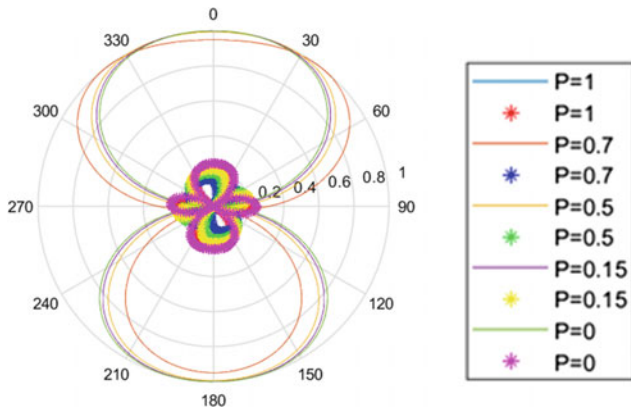


Fig. 10 Tracking the power level using analytical beamformer and ANN beamformer, SNWOM with Sigmoid as the activation function

drawbacks in requiring a complex array and incompatible scalability issues dealing with a large number of nodes as receivers increase. Thus, a less complex and fast method of tracking is necessary for the smart city applications.

4.3 Beamforming with SNWOM

The ANN enabled smart antenna with 2 elements in Fig. 8 with the configuration described in Sect. 4.1 is used to demonstrate beamforming at angle of departure of 60° and 150° . The desired function is plotted in blue while the ANN enabled smart antenna's is the red plot. It was shown in Reference (Senthilkumar et al., 2021) that the shape of the radiation pattern would improve with more elements with a non-uniform linear array. Therefore, the radiation pattern

does not agree with the desired pattern, however the ability to form a beam in the desired direction is demonstrated to confirm the concept. In this case, the activation function used is the sigmoid function (Fig. 11).

In Fig. 12, we compare the two activation functions; the sigmoid and the squash activation functions. We note that the squash activation function shows a better magnitude response compared to the sigmoid function. Thus, it does confirm the hypothesis that there must be a better activation function for certain applications. However, the difference between the desired output and either activation function is quite high. This may be improved by adding more elements and changing the geometry of the array antenna as shown in Fig. 13.

4.4 Tracking with ANN Enabled Smart Antenna

The challenge with tracking the mobile transceiver is usually a two-fold approach. The first approach is to determine the DoA using either the Multiple Signal Classification (MUSIC) Technique or the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) (Macharia et al., 2019). Though these two techniques are electrical implemented, they are still computationally intensive for the limited power in the mobile devices. Once the DoA is established, it becomes the look angle for the beamformer algorithm to enhance the desired signal coming from either a known or unknown direction and to null any interfering signals (Liu & Weiss, 2010). We propose a method to manage the beam pair between the base station and the mobile transceiver by sensing the received signal strength in

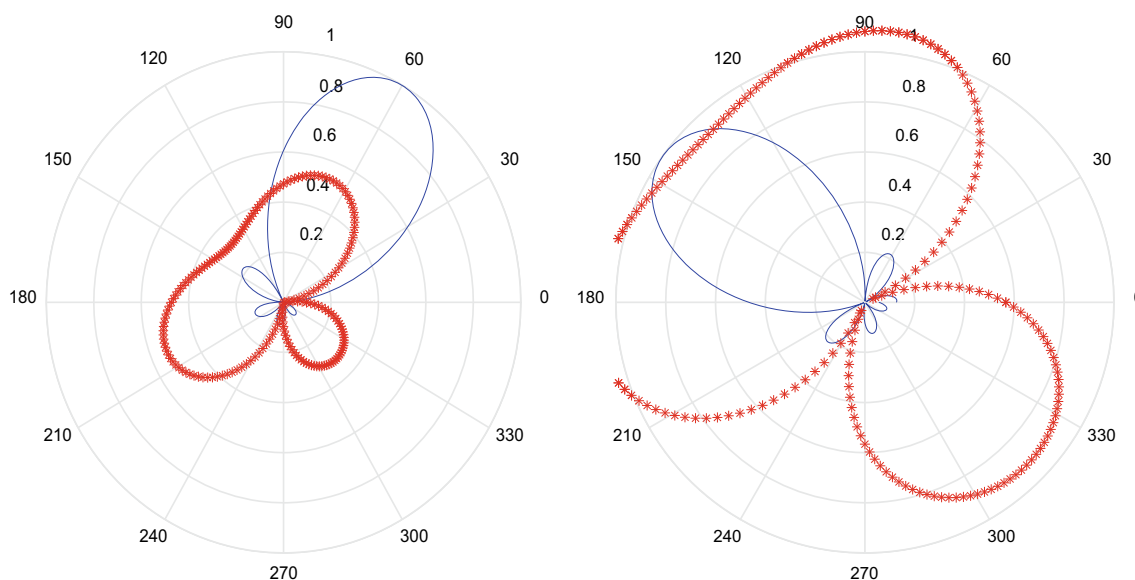


Fig. 11 Beamforming with SNWOM in the look angles of 60° and 150° with the bias constant with the sigmoid activation function

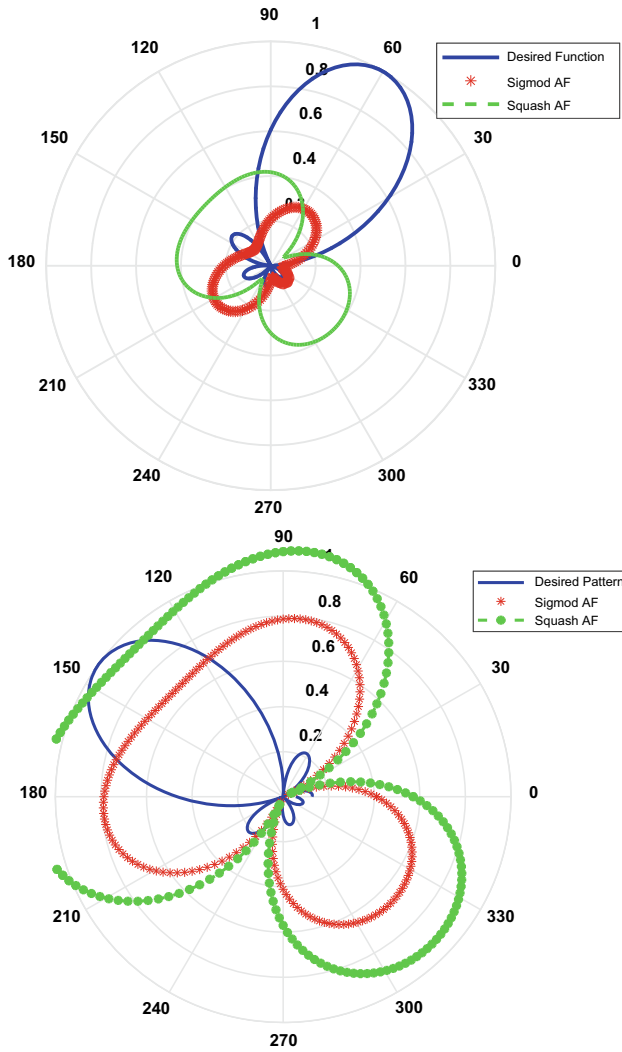


Fig. 12 Beamforming with SNWOM in look angles of 60° and 150° comparing two activation functions with the bias constant

comparison to the radiation pattern of the transmitting base station (Kunsei et al., 2021) as discussed in Sect. 4.2.

Tracking of mobile users can be achieved in several ways. However, the existing methods are computational intensive (Caylar et al., 2006). Thus, in this section we develop a simple, yet accurate and less computational need by using the antenna radiation. However, since the coverage space is extremely large, ANN is employed to determine exact angle of arrival.

Figure 14 illustrates the radiation pattern of sectorial antenna. Figure 14a shows the tilted beam toward the ground to provide coverage. Figure 14b is the approximate antenna pattern aligned on the ground on the x - y axis. The reference is taken from the x axis; thus, the maximum is about 90° . To track user using the radiation pattern, the power level is normalized by the maximum power. Thus, the normalized power of 1 is directed to 90° .

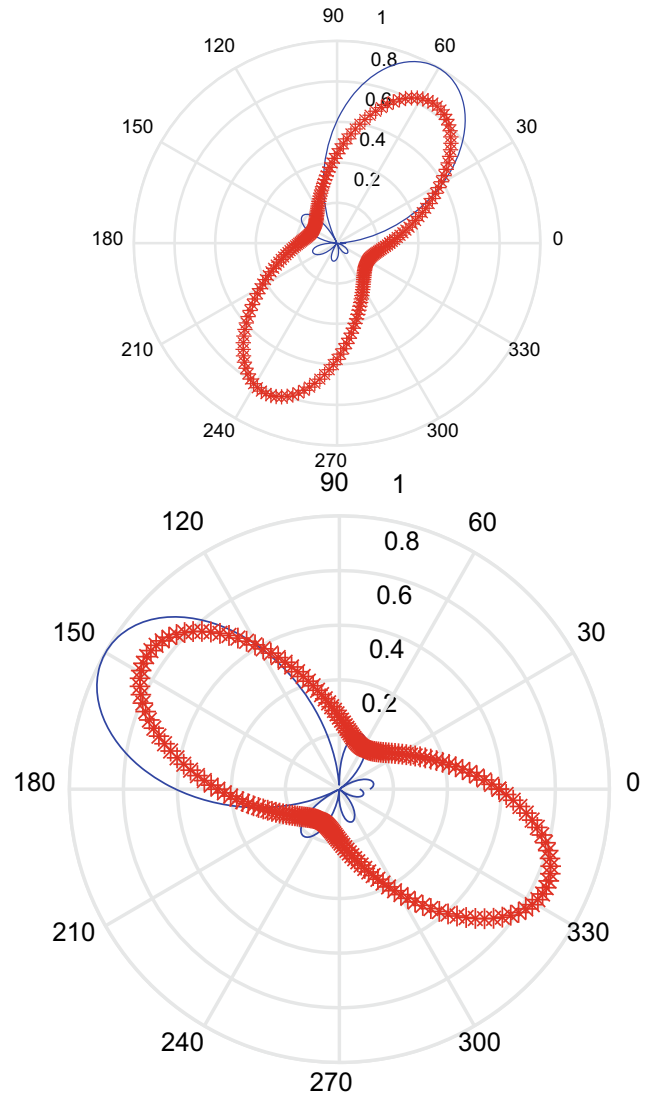


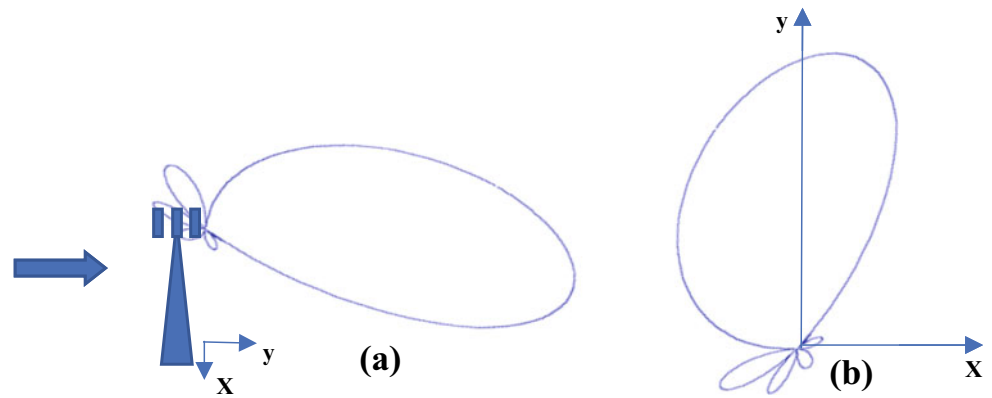
Fig. 13 Reproducing with 4 elements for look angles of 60° and 150°

The signal strength will determine the radial distance from the origin or pole. In terms of antenna pattern measurements, the power along a radial distance will be the same level. Thus, for fixed radial distance, theta angle for the each of the quadrant will have the same magnitude with different direction as indicated in Table 1.

The power level in Table 1 is determined from Kunsei et al. (2021).

$$p = \cos(kd(\cos\theta - \cos\theta_m)/2) \quad (4)$$

where theta, θ , is the angle of arrival and can be taken as new angle based on the power level. The power level for each of the quadrant with the same radial distance is the same as

Fig. 14 Radiation Pattern in the Propagation Environment**Table 1** Comparison of directional of arrival from propagation environment with radiation pattern

x	y	Theta from mobile unit position	Power level from Theta of mobile unit position	Theta from power level	Difference
2	4	63.4	0.8406	41.3	22.1
2	-4	-63.4	0.8406	41.3	22.1
-2	4	-63.4	0.8406	41.3	22.1
-2	-4	63.4	0.8406	41.3	22.1
4	6	56.3	0.7858	50.8	5.5
100	200	63.4	0.8046	41.3	22.1
-4	6	-56.3	0.7585	50.8	5.5
-100	200	-63.4	0.8046	41.3	22.1
10,000	10,000	45	0.8839	13.25	31.75
-10,000	10,000	-45	0.8839	13.25	31.75

shown in the Table 1. We assume that sectorial antenna is 180° as shown in Fig. 14.

The new angle based on the power level can be calculated from (4) as:

$$\theta_n = \cos^{-1} \left[\frac{(2 * \cos^{-1}(p))}{(kd + \cos \theta_m)} \right] \quad (5)$$

Equation (5) was used to calculate the new angle is recorded in Table 1. If the coordinates and the antenna pattern are aligned the beamforming angle determined from the geographical x, y coordinates should equal the new angle determined from the normalized power level of the radiation pattern of the antenna. It is evident that there is a mismatch in the theta values as shown in the last column in Table 1. Thus, a correction variable is needed in addition to the shifting variable, β , to indicate the direction of the mobile unit.

For 4G system the normalized power level of 1 is equal to -75 dBm. The maximum range is more than 10 km when the operating frequency is 1800 MHz and lower with the transmit power set to 20 W, transmit and receive antenna gains of 1 as shown in Fig. 15. The choice of operating frequency will determine the range of the antenna pattern.

Therefore, the coverage area at 1800 MHz will be at least 10 km per sectorial antenna. The coverage space above about 20×20 km area from the pole is a huge area to determine all possible mobile station location and the direction of movement. However, it does show that the resolution of the tracking angle would be high and easier to determine.

If we use the contour of field strength of -79 dBm as an illustration, then at the boresight at $x = 0$, the normalized power would be 1 at about 23 km from the antenna as indicated by an A in Fig. 16. As the mobile user moves away from the maximum beam the power level will be less than 1 as illustrated in Table 1. In this case, the angle of maximum strength is 90° . At 10 km, calculations in Table 1 shows that the angle is 45° with power level of 0.8839. Using the power level relationship in (4) we determine the new theta angle the beamformer to direct the main lobe. Figure 16 shows the radiation pattern with the positional coordinates from Table 1 when x and y are 10 km. The power level has reduced by 1 dBm which implies that the normalized power level is 0.99. Therefore, the new angle of the mobile unit in clockwise direction at position B is:

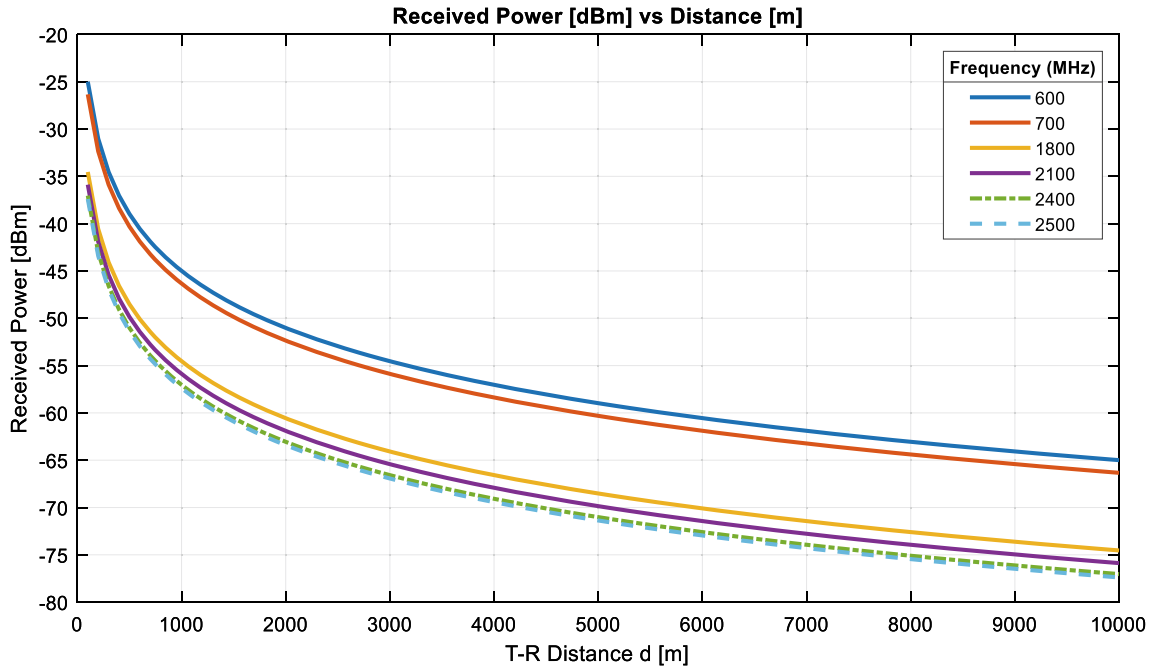


Fig. 15 Comparing received power against distance for a 4G base station operating at 600 MHz, 700 MHz, 1.8 GHz, 2.1 GHz, 2.4 GHz, and 2.5 GHz with unity antenna gains and 20 Watts transmit power

$$\theta_{\text{newcw}} = \theta_{\text{newfromposition}} - \theta_{\text{newfrompower}} \quad (6)$$

when the mobile unit is moving in the counter clockwise direction, the new angle at position C in Fig. 16 is.

$$\theta_{\text{newccw}} = \theta_{\text{max}} + \theta_{\text{newfromposition}} - \theta_{\text{newfrompower}} \quad (7)$$

The reference point is taken to be $x = 0$. Thus for a clockwise movement of the mobile station, the new angle should be less than the previous maximum angle, $\theta_{\text{newcw}} < \theta_{\text{max}}$, as shown in (6). Following on the counter clockwise direction, the new angle should be more than previous maximum angle, $\theta_{\text{newccw}} > \theta_{\text{max}}$ as shown in (7).

4.5 Application of ANN Smart Antennas in Smart Cities

4.5.1 Smart Energy with ANN Enabled Smart Antennas

Smart cities will see integration of renewable energy in smart homes and industries. In developing countries where the smart city concept is extended to rural areas, renewable sources may be the only possibilities in the smart village setting (Kempener et al., 2015). In this setting the monitoring and control of hybrid sources of renewable energy become vital for the sustainability of the energy system. By making the renewable energy system smart, the system is sustained with technologies such as by demand response

(DR), smart inverters, renewable distribution automation (DA), virtual power plants and microgrids (Kempener et al., 2015). These smart technologies enable the renewable energy system to be flexible, responsive and intelligent. However, this is only possible with a reliable and intelligent wireless communications system.

For example, in a community depending on the micro-hydro renewable system, the regulator of the governor for the generator can be controlled to match the load on demand response. The smart antenna with tracking can establish a beam pair with the controller circuit and with the smart meters at the home. Whenever the load changes, the controller can adjust the speed on the generator to produce the required power to match the load.

Another example involves the monitoring and control of the generation source in a hybrid of hydro-solar renewable system as in Reference (Urame & Hoole, 2020). The system is equipped with a smart controller for monitoring the charge on the storage capacity between two renewable sources as illustrated in Fig. 17. To ensure that the system can be monitored remotely a smart antenna is installed in the controller to communicate with any mobile device via the mobile network and controlled anywhere in the world. In addition, environmental sensors installed in the vicinity of the renewable energy system installation can be monitored by the same smart antenna with the dedicated beam pair for remote sensing and monitoring of the environment.

Fig. 16 Field strength of a directional antenna at 1800 MHz with 1 V at transmitting voltage

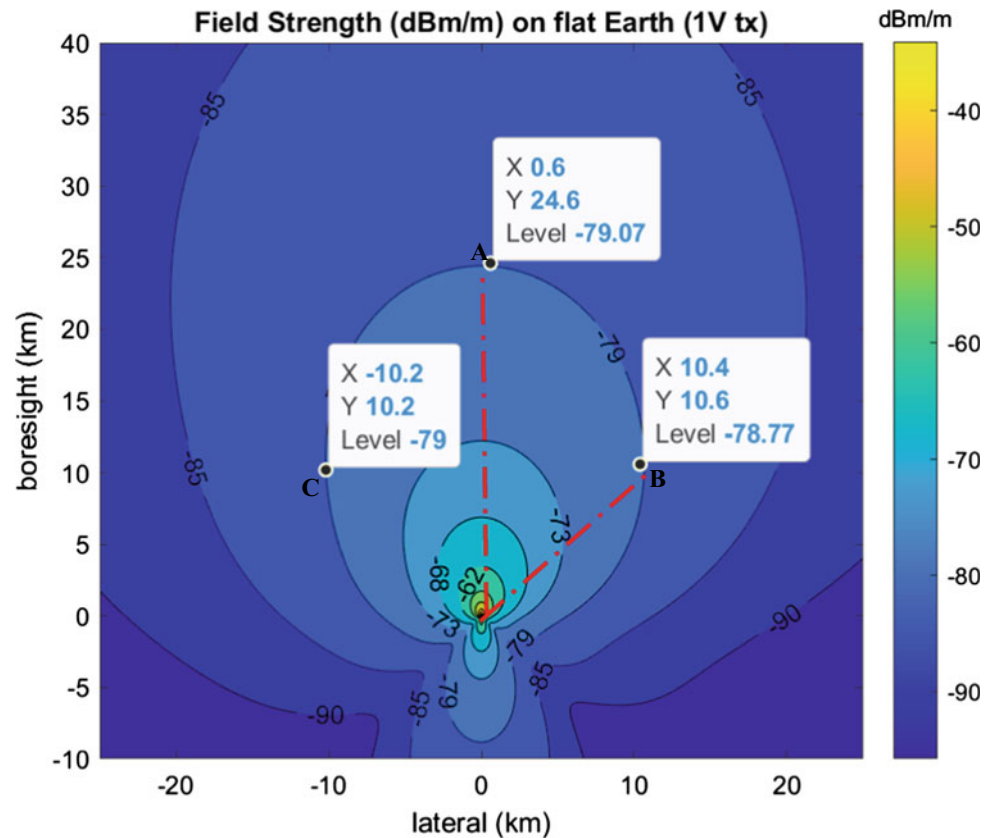
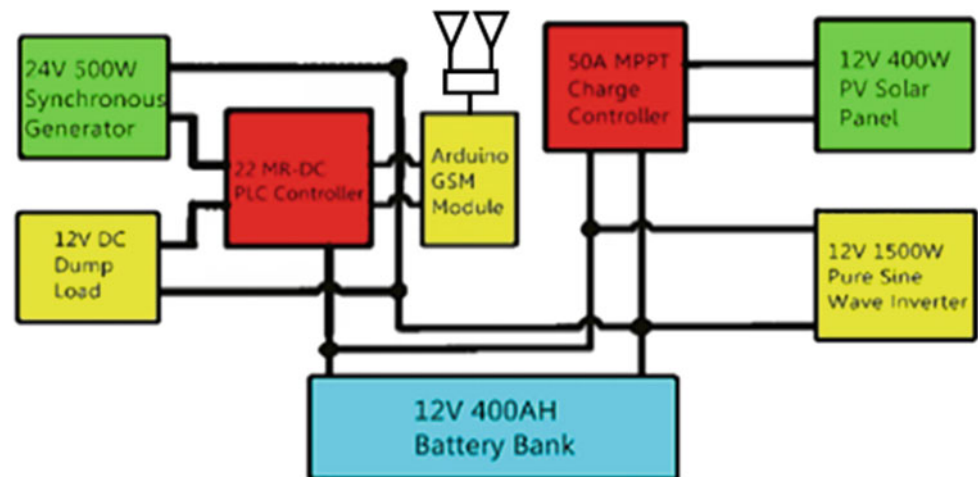


Fig. 17 Smart antenna enabled controller for monitoring and control



In the smart grid in the urban areas, the smart antenna can be used to monitor the health of the components in a substation or distribution network. In the substation, the health of the circuit breakers is the difference between a long downtime and short downtime, or the difference between a less reliable or reliable power system. In current wireless systems, conventional antenna systems are used to connect all circuit breakers to the control room. The data exchange can be large and informative. However, with the use of a

smart antenna, the system can do more than just exchange information between nodes. Some processing could be performed on the data in the antenna before alerts or records could be updated at the end of the data processing. A circuit breaker time in operation either in the mechanical or electrical system can be measured from the uptime and downtime of the component as the beam link is dedicated between the transmitting and receiving antennas. An intermittent break in uptime may indicate the health of the circuit breaker

and an alert can be generated by the monitoring system for maintenance to be scheduled, thus reducing downtime and improving system reliability (Mathur et al., 2018).

4.5.2 Smart Transportation with ANN Enabled Smart Antennas

Transportation systems involve high mobility with varying speeds. Implementing a smart transportation system will depend entirely on a wireless communications system for connectivity. The reliability of the wireless connection depends on the antenna system for the selected application. In vehicle to vehicle (V2V) or car to car (C2C) communications, omnidirectional antennas are applicable because of the need to have coverage all around the vehicle or car for short distances. However, for applications such as traffic and fleet monitoring and control, integrated public transportation, and real-time services for drivers and passengers require an antenna system with the ability to track the mobile unit, perform distance coverage and mitigate multipath interference.

Figure 18 shows a typical smart transportation service and application with the relevant communications requirements. In Fig. 15a the concept of providing real-time messages for drivers is illustrated with the sending of a warning message alerting nearby or approaching drivers of the accident ahead in the traffic to safeguard them and their passengers. The use of an ANN beamforming antenna would maintain the beam pair between the vehicles or with base stations. A warning message can be delivered in real time and with low latency. Also, the link can be so secured that no hoax or bogus messages should be delivered through the system.

Furthermore, to implementing V2V communication is as shown in Fig. 18b, which illustrates a 2 element (blue dots) ANN smart antenna. It is highly applicable when a narrow or variable sized beam width and a highly directed beam is required to establish a reliable communication link all around the vehicle. In a multipath environment where communications links are established based only on signal strength, the reliability of the link is much lower because of the continuous change in the received signal strength as the distance between the vehicles changes due to acceleration and deceleration.

Another application in Fig. 18b can be the implementation of multiple-input-multiple-output (MIMO) in V2V communications. MIMO antenna systems are renowned for exploiting multipath propagation in improving signal quality and capacity through diversity gain. However, the computational and power requirements may be an issue with compact vehicles. Alternatively, an ANN smart antenna with beamforming at millimeter wave frequencies can offer a reliable link with high capacity with stable performance as shown in Fig. 19.

By maintaining the distance between the antenna elements, the performance of the antenna is maintained when the power levels of the radiation pattern are used to estimate the location of the mobile user. The shape of the beam can be improved with more elements with nonlinear array configuration.

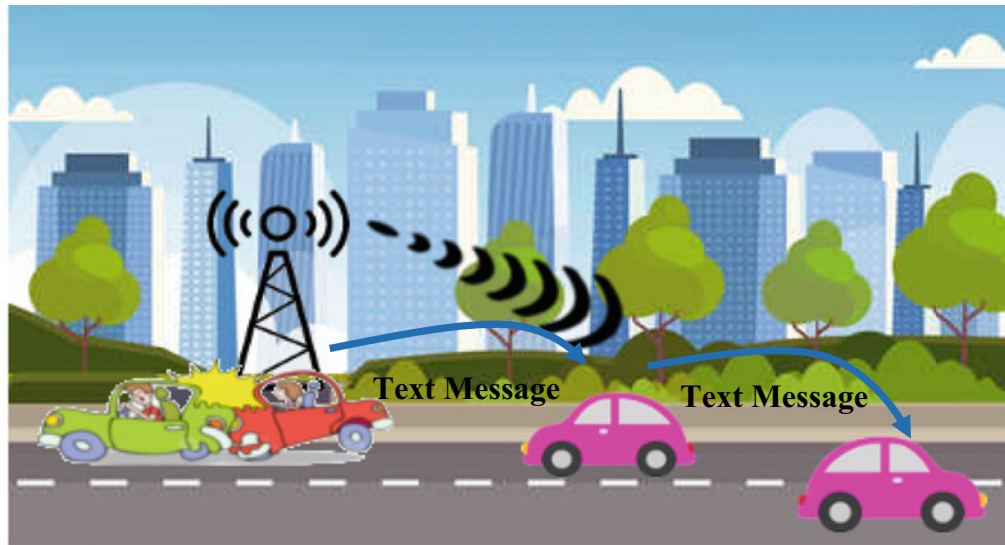
4.5.3 Sensing Human Activity with ANN Enabled Smart Antennas

The final case that may benefit from an ANN smart antenna is sensing human activity. The objective of knowing human mobility is to gain insights that can be used to implement efficiently managed human mobility in smart cities (Semanjski & Gautama, 2016). The process of acquiring the need to form the insights is usually manual, tedious and time consuming. Also, the records may not be complete and correct due to human incapability to remember every detail of their movement unless recorded on paper. With the widespread use of smart mobile phones with adequate mobile applications to track and store human activity, sensing human activity is much more exciting and accurate. Capturing the GPS data, information about the movement within a specified time frame can be obtained. Also, real-time location of the mobile device can be obtained through GPS tracking or tracking by an ANN enabled smart antenna.

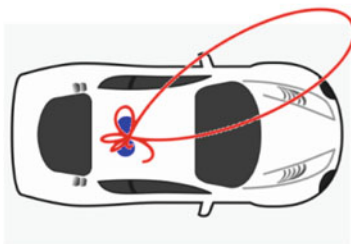
Location-enabled devices have been used in tracking vehicles but have relied on the global navigation satellite system (GNSS). A special electronic device is designed to communicate with the satellites to determine their location with very high precision. To track human mobility, one must carry the tracking device if a mobile phone is not enabled to communicate with the GNSS. However, with an ANN enabled smart antenna, the mobile radio system is all that is needed to ensure that tracking within a coverage area is achieved with high precision as discussed in Sect. 4.2. The tracking of a human in the case of an emergency is possible with this smart antenna. Also, the tracking of individuals in sporting areas, parks, forests, schools and neighborhoods is now possible with an ANN enabled smart antenna.

An added benefit of using an ANN smart antenna in human sensing with location-enabled devices is in implementing active or interactive tracking (Semanjski & Gautama, 2016). With the use of machine learning in the specialized mobile app (Developer, 2019) the smart antenna can be further tuned for other functions thus adding more features is a simple software enabled without the need for new hardware.

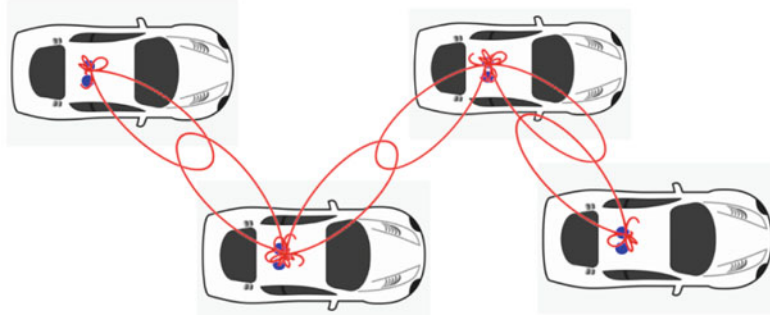
The challenge of sensing and tracking human activity is illustrated in Fig. 20 where two instances of human activity has been predicted in the space of $50 \times 50 \text{ m}^2$ area. At the center of the area we have base stations, as illustrated by the red dot. The green cells in the illustration indicates the



a) A Warning Message System



b) Smart Antenna:
Vehicle-to-Vehicle



c) Vehicle to Vehicle using a Smart Antenna with multiple beams.

Fig. 18 Typical services and application for smart transportation. **a** Real-time messaging of traffic incident, **b** Antenna requirement for vehicle-to-vehicle (V2V) or vehicle to everything (V2X) communication, **c** V2V communication using a smart antenna on a road

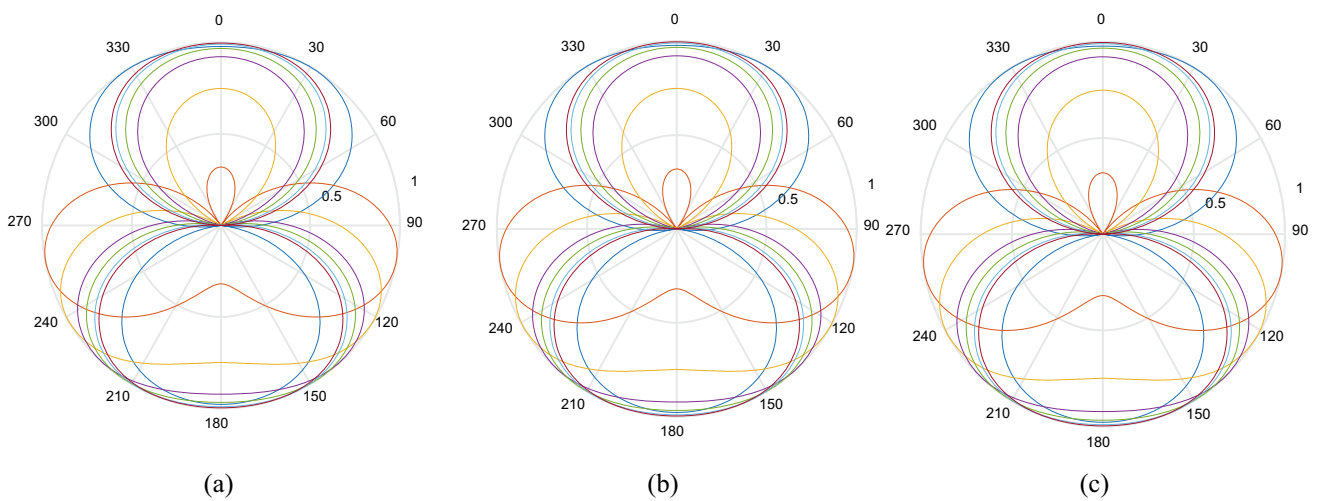


Fig. 19 Stability plots for case 5 for three different frequencies, **a** 6 GHz, **b** 30 GHz and **c** 60 GHz in Kunsei et al. (2021)

position of small cells which may be mobile devices that is being moved about by the movement of the human owners. It is anticipated that these small cells would change position more frequently. The challenge of sensing and tracking becomes more problematic when power efficiency is also a concern. A simple and energy efficient tracking method with the aid of ANN as discussed in Sect. 4.4 offers some hope for a truly mobile yet connected network in smart city for 5G networks and beyond.

A more robust algorithm would also predict the location of small cells and decide where to participate as a small cell as the location changes. As shown in the Fig. 20 some cells make be too close to the base station that it disqualifies to act as a small cell. However, the same mobile may again qualify should it move to another location. By using less but high quality set of data, the ANN smart antenna can perform the sensing with improved accuracy with less power which is a concern in mobile devices.

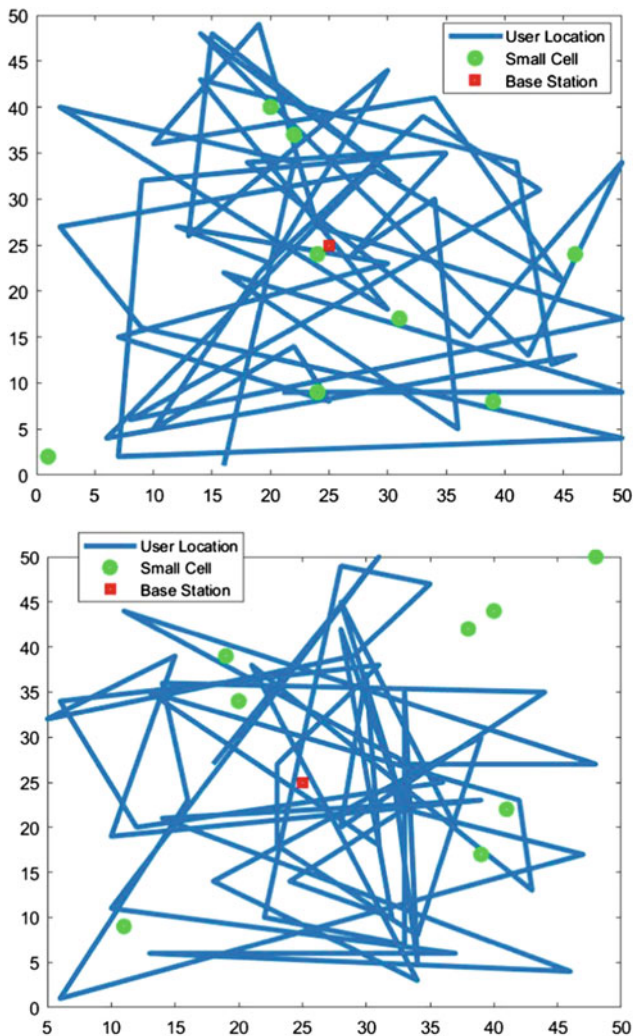


Fig. 20 Human movement in a 50×50 m area

5 Summary

This chapter addressed a vital component in the quest to make cities smart. Smart cities rely on data from everything and anything that offers some service to make the city safer, conducive for living and adaptable to enhance the livelihood of its citizens to learn from past experience, make informed decisions today and serve the citizens better tomorrow. It was argued that for a smart city to progress forward, it must not only have a reliable communications system but also an intelligent communications system. An optical fiber communications system is highly suited for backhaul networks with their high bandwidth and throughput. Therefore, wireless communications would be the widely deployed systems of choice in smart cities. Wireless communications systems include terrestrial microwave, Wi-Fi, RFID, Bluetooth, 5G and 6G networks.

Terrestrial microwave links are easy deploy on a short-term basis but are not enough for high intensity bandwidth applications such as gaming and VAR. Thus, it is not widely used in smart cities. For short range communication, Wi-Fi, RFID and Bluetooth are employed in small cell communications. These are used in service organizations that require payment via smart wallet or some other means while requiring network connectivity.

5G and 6G network characteristics offer super enhanced mobile broadband speeds that are required to support the new applications of smart cities such as virtual/augmented reality, V2V, D2D and M2M. In addition to speed, the 6G network offers extremely low latency to support critical applications such VAR and an emergency response team. Finally, the massive number of devices requiring connectivity in smart cities demands that any network should be reliable and intelligent.

The massive connectivity requires network management so that the established protocols and standards are not enough to handle the new and numerous connections. Thus, the core of the network must be intelligent. The specific form of AI to be applied in any level of the intelligent communications network depends on the function or service required. Furthermore, the edge of the network can be made intelligent by having edge devices that are AI enabled. However, some edge devices may not be intelligent as expected due to hardware limitations. An approach at the edge to make communication intelligent is the development and employment of AI enabled smart antennas.

We demonstrated in Sect. 4, a fast and low resource demanding ANN antenna that is capable of localizing and tracking the desired user with a narrow beamform. The application of the ANN enabled smart antenna is discussed in the provisioning of a smart energy system with monitoring and alerting in a digital distribution substation. The

second application involves the support of an intelligent system with V2V communications. Finally, the vital function of sensing human activity is given the opportunity to use the tracking features to ANN enabled smart antennas to sense data for the planning and maintenance of public facilities. In all three applications, we note the relief to current manual techniques and burdens on humans when the ANN enabled smart antenna is used for monitoring, collecting data and analyzing. While the ANN enabled antenna offers some promising solutions to support the functions and operation of smart cities, challenges still exist.

In smart cities where everything is connected to something or everything, cyber security is a concern. The concern can lower the confidence and trust of the user in using the services offered by the smart city. Thus, the risk of doing too much for nothing will certainly hinder some progress in moving the development of smart cities. Another challenge dealing with intelligent communications system is the storing and handling of the massive data collected from all the sensors throughout the network. The location of the data can slow down system performance if the data is stored on servers too far from the user.

Addressing these challenges and others will enable reliable wireless communication in small cells for smart cities application using ANN enabled smart antennas.

References

- Akyildiz, I. F., Kak, A., & Nie, S. (2020). 6G and beyond: The future of wireless communications systems. *IEEE Access*, 8, 36.
- Asplund, H., et al. (2020) *Advanced antenna systems for 5G network deployments: bridging the gap between theory and practice*. London: Academic Press. 740.
- Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communication*, 7(5), 28–34.
- Caylar, S., Leblebicioglu, K., & Dural, G. (2006) A new neural network approach to the target tracking problem with smart structure. *Radio Science*, 41 (RS5004).
- Cicirelli, F., et al. (2019) The internet of things for smart urban ecosystems. In F. Cicirelli et al., (Ed.), *The internet of things for smart urban ecosystems*. Springer International Publishing.
- Developer, A. (2019) Detect when users start or end an activity. 27 Dec. 2019 [cited 2021 27 May 2021]; Available from <https://developer.android.com/guide/topics/location/transitions>.
- Georgescu, M., & Popescu, D. (2016) The importance of internet of things security for smart cities. In I. N. D. Silva & R. A. Flauzino (Ed.), *Smart cities technologies*, ExLi4EvA.
- Giorgetti, G., et al. (2009). Single-anchor indoor localization using a switched-beam antenna. *IEEE Communications Letters*, 13(1), 3.
- Hoole, P. R. P. (2020) Smart antennas and electromagnetic signal processing in advanced wireless technology: with artificial intelligence applications. In P. R. P. Hoole (Ed.), *Smart antennas and electromagnetic signal processing in advanced wireless technology: With artificial intelligence applications*. Rivers Publishers.
- Hornillo-Mellado, S., Martín-Clemente, R., Baena-Lecuyer, V. (2020) Prediction of satellite shadowing in smart cities with application to IoT. *Sensors*, 20.
- Imoize, A. L., et al. (2021) 6G enabled smart infrastructure for sustainable society: opportunities, challenges, and research roadmap. *Sensors*, 21.
- ITU-R (2020) Detailed specifications of the terrestrial radio interfaces of International Mobile Telecommunications-2020 (IMT-2020). In *M Series*, r. Mobile, amateur and related satellite services. (Ed.), International Telecommunication Union: Geneva.
- Kempener, R., Komor, P., & Hoke, A. (2015) Smart grids and renewables a cost-benefit analysis guide for developing countries. *International Renewable Energy Agency*. p. 44.
- Kunsei, H., Pirapaharan, K., & Hoole, P. R. P. (2021). A new fast, memory efficient wireless electromagnetic beamformer antenna with fast tracking for 5/6g systems. *Progress in Electromagnetics Research C*, 110, 253–265.
- Liascos, C., et al. (2019) An interpretable neural network for configuring programmable wireless environments. In *2019 IEEE 20th international workshop on signal processing advances in wireless communications (SPAWC)*. IEEE: Cannes, France.
- Liu, W., & Weiss, S. (2010) Wideband beamforming concepts and techniques. In X.S. Shen & Y. Pan (Eds.), *Wireless communications and mobile computing*. United Kingdom: Wiley.
- Liu, Y., et al. (2021) Reconfigurable intelligent surfaces: Principles and opportunities. *IEEE Communications Surveys & Tutorials*
- Luo, Q., & Gao, S. (2017) Smart antennas for satellite communications on the move. In: *2017 international workshop on antenna technology: small antennas, innovative structures, and applications (iWAT)*, IEEE: Athens, Greece.
- Macharia, R., Kibet, P., & PeterKihato. (2019). An artificial neural network approach to DOA estimation and switched beamforming in rectangular array based smart antennas. *Progress in Electromagnetics Research C*, 93, 79–92.
- Mahmood, Z. (2018) Smart cities: Development and governance frameworks. In Z. Mahmood (Ed.), *Smart cities: Development and governance frameworks*.
- Markets, R.A. (2021) 5G Smart antenna market by type (switched multi-beam antenna and adaptive array antenna), technology (SIMO, MISO, and MIMO), use case, application, and region 2021–2026, Research and Markets.
- Marzuki, A. S. W., et al. (2021) Emerging technologies for 5g/6g wireless networks. In P. R. P. Hoole (Ed.), *Smart antennas and electromagnetic signal processing in advanced wireless technology—With artificial intelligence application and coding* (p. 337–360).
- Mathur, A. L., Chauhan, D., & Singh, Y. (2018). A review on substation monitoring and control technologies and problems. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 2(3), 6.
- Nassar, A., & Yilmaz, Y. (2020) Deep reinforcement learning for adaptive network slicing in 5g for intelligent vehicular systems and smart cities. ArXiv abs/2010.09916, p. 13.
- Obaidat, M. S., & Nicopolitidis, P. (2016) *Smart cities and homes: Key enabling technologies*. In M. S. Obaidat & P. Nicopolitidis (Ed.), Elsevier Inc.
- Pelton, J. N., & Singh, I. B. (2019). *Smart cities of today and tomorrow: better technology, infrastructure and security* (1st ed.). Copernicus.
- Pirapaharan, K., Hoole, P. R. P. & Hoole, S. R. H. (2021) Advanced wireless systems: A comprehensive survey. In P. R. P. Hoole (Ed.), *Smart antennas and electromagnetic signal processing in advanced wireless technology—With artificial intelligence application and coding*. River Publishers.
- Rubin, A. (2011) All your devices can be hacked. In *TEDxMidAntic*. TED.com.

- Semanjski, I., & Gautama, S. (2016) Sensing human activity for smart cities' mobility management. In I.N.D. Silva and R.A. Flauzino (Eds.), *Smart cities technologies*. ExLi4EvA.
- Senthilkumar, K.S., et al. (2021) Real- and complex-valued artificial intelligence weight optimization algorithms for smart antennas in 5/6G wireless systems: Linear and nonlinear arrays. In P. R. P. Hoole (Ed.), *Smart antennas and electromagnetic signal processing in advanced wireless technology: With artificial intelligence applications* (p. 300). River Publishers
- Shahraki, A., et al. (2021) *A comprehensive survey on 6G networks: Applications, core services, enabling technologies, and future challenges*. submitted to IEEE Internet of Things
- Silva, I. N. D. & Flauzino, R. A. (2016) *Smart Cities Technologies*. In I. N. D. Silva & R. A. Flauzino (Ed.), ExLi4EvA.
- Singh, S.K., et al. (2020) machine learning-based network sub-slicing framework in a sustainable 5g environment. *Sustainability*, **12**.
- Sun, H., Wang, C., & Ahmad, B. I. (2018) *From internet of things to smart cities: Enabling technologies*. Chapman and Hall/CRC. 430.
- Urame, C., & Hoole, P. R. (2020). Design and implementation of hybrid pico-hydro—Photovoltaic (PV) solar power plant in Massy-Gahuku LLG. *European Journal of Electrical Engineering*, **22**(6), 395–403.
- Wang, X., et al. (2012). Enhanced security and reliability with MIMO communications for smart grid. *Security and Communication Networks*, **8**, 2723–2729.
- Wang, A., et al. (2020) A review on non-terrestrial wireless technologies for Smart City Internet of Things. *International Journal of Distributed Sensor Networks*, **16**(6).
- Xiao, K. (2018) Congestion control and resource allocation in emerging wireless networks. In *Electrical and Computer Engineering*. Auburn University.
- Yagol, P., et al. (2018). New trends in using augmented reality apps for smart city contexts. *ISPRS International Journal of Geo-Information*, **7**, 23.
- Zhang, S., et al. (2020). Envisioning device-to-device communications in 6G. *IEEE Network*, **34**(3), 6.
- Zhao, Y., et al. (2020) *A comprehensive survey of 6G wireless communications*. arXiv e-prints, p. 34.
- Zhou, F., et al. (2020). Automatic network slicing for IoT in smart city. *IEEE Wireless Communications*, **27**(6), 108–115.

Wavelet-Based Saliency and Ensemble Classifier for Pedestrian Detection in Infrared Images

R. Newlin Shebiah  and S. Arivazhagan

Abstract

Night vision systems designed for intelligent transportation system ensuring pedestrian safety are the need of the hour. The pedestrian detection system should be real time and precise during day and night-time and should alert the driver with an alarm or automatically apply brake to avoid collision. In this chapter, a robust approach using wavelet-based saliency for region of interest detection and local denary signature for pedestrian representation in thermal infrared imagery is suggested. Unlike the state-of-the-art approaches that rely on the intensity thresholding for pedestrian detection, this framework depends on wavelet-based saliency map generated by both approximation and detailed sub-bands for prospective regions where pedestrians are more likely to be found. There may be other regions that have similar thermal characteristics of the pedestrian in infrared image, raising false alarm. Thus, an effective representation with ten intensity levels termed as local denary pattern, and the descriptor based on edge orientation histogram precisely describes the pedestrian characteristics. Local denary pattern is a robust and discriminative representation as it accomplishes a better tolerance against greyscale variation. Finally, Logitboost ensemble classifier is used to classify pedestrians from background. The efficacy in detecting pedestrians is demonstrated by experimental findings on a publically available OSU Thermal Pedestrian Database.

Keywords

Wavelet transform • Saliency representation • Pedestrian detection • Infrared images

R. N. Shebiah (✉) · S. Arivazhagan
Centre for Image Processing and Pattern Recognition, Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, 626-005, India
e-mail: newlinshebiah@mepcoeng.ac.in

1 Introduction

The modern developments of smart vehicles and the demand for road safety have ensued incorporating systems for enhanced driver assistance. Some advanced driver assistance systems in existence include adaptive cruise control, adaptive lighting control, detect the presence of an object within a blind spot, lane-keeping assistance systems, night vision systems, etc. Night vision systems designed for pedestrian safety have to be real time, precise during day and night-time and should alert the driver with an alarm or automatically apply brake to avoid collision. The existing data acquisition system for pedestrian detection includes passive sensor (Alonso et al., 2007; Bertozzi et al., 2006; Dalal & Triggs, 2005; Jungling & Arens, 2009) with visible or infrared cameras, active sensors (Gate & Nashashibi, 2008; Gidel et al, 2008), and hybrid sensors (Fayad & Cherfaoui, 2007; Frolov & Leon, 2010). Among these available sensors, passive sensors are acceptable as it impersonates human perception and can be used to classify the obstacles. Visible cameras fail to detect objects in the dark without an illuminating source, whereas an infrared sensor perceives the heat signature emitted by the object and creates an electronic image in shades of grey. Numerous challenges existing in the field of pedestrian detection make this an evergreen area of research. The challenges are as follows (a) the motion pattern of the pedestrian is unpredictable (b) the appearance of the pedestrian varies with the view point, direction, clothing, temperature, etc. (c) objects other than the pedestrian appearing bright in the IR images may cause false positives (d) lack of fine discriminative details of the image makes the pedestrian detection difficult.

Thermal cameras are independent of illumination, and different objects are represented in varying brightness level. Thermal photographs of pedestrians have a high perceptibility at night because the pedestrian's temperature is higher than the background ambient temperature. Pedestrians can be distinguished from the background by using an appropriate

threshold because pedestrians have greater intensity values (Suard et al., 2006; Sun et al., 2004). Selecting appropriate threshold value is a big challenge as pedestrian intensities vary with respect to temperature and distance. While binarizing the image with static threshold, if the threshold selected is too small, it results in unwanted background image classified as pedestrian. On the other hand, when the threshold is selected too high, the pedestrian may be fragmented. The dynamic threshold value is calculated using the image's statistical properties and its percentage to the area beneath the histogram. Region growing algorithm is used to isolate pedestrian from background (Malley et al., 2010); here, the region merges when two region of interest in close proximity. State-of-the-art techniques focused on extracting region of interest by thresholding assuming the pixel intensities follow Gaussian distribution (Bertozzi et al., 2007) and histogram analysis (Xu et al., 2005).

Pedestrian detection is a binary object classification problem in which learning-based methods are used to classify candidate objects as pedestrian or non-pedestrian. Mining the region of interest, extracting discriminant features, and feature classification are all part of the pedestrian detection architecture. The classification accuracy relies more on the discriminating features selected for classifying pedestrian from background details rather than depending on the classifiers choice. Based on the prior model developed with the distinct features of pedestrians and non-pedestrians, the classifier categorizes the candidate region into pedestrian or non-pedestrian.

Fang et al. (2004) employed horizontal and vertical projection to pick possible targets and then used multidimensional histograms, inertia, and contrast-based characteristics to classify the possibilities. Liu et al. (2013) used gradient and threshold-based picture segmentation to identify the candidate pedestrians and pyramid entropy weighted histograms of oriented gradients to represent them. The most commonly employed features for pedestrian recognition are grey-level image features (Sun et al., 2004), Gabor wavelets (Apatéan et al., 2008; Sun et al., 2006), Haar wavelets (Andreone et al., 2005; Papageorgiou et al., 1998), and histograms of oriented gradients (HOGs) (Hurney et al., 2015). Zhang et al. used an AdaBoost cascade classifier and a cascade of SVM classifiers to extract local features like edgelets and HOG and classify them (Zhang et al., 2007). To collect contour cues, Riaz et al. (2013) employed CENsus Transform hISTogram (CENTRIST) features and support vector machines (SVMs). Shape characteristics (), such as compactness from the object's skeleton, can also be used as input to SVM. Context saliency maps (CSMs) (Davis & Sharma, 2004, 2005, 2007), CSM template matching (Davis & Keck, 2005), form and appearance-based detection (), spatiotemporal texture vectors (Latecki et al., 2005), and boosting frameworks (Calafut, 2015) are all utilized to detect pedestrians. The advantage of

learning-based method for pedestrian detection can detect the pedestrian without a background image, and thus, background subtraction is evicted.

Convolutional neural network (CNN) is used by Kim et al. for night-time pedestrian identification utilizing visible images (Kim et al., 2017). For multispectral pedestrian identification, Liu et al. (2016) and Wagner et al. (2016) used fusion architectures to combine visual and thermal channel information in CNN. Cai et al. (2016) employed a saliency map to generate candidates and a deep belief network as a classifier for vehicle recognition at night. For generating candidates and verifying pedestrian detection in thermal pictures, John et al. (2015) employed fuzzy C-means clustering and CNN. Biswas and Milanfar (2017) experimented effective strategies with multidimensional template using local steering kernel for detecting pedestrians in infrared images. Hou et al. (Zhang et al., 2018) combine pixel-level fusion methods and CNN architectures for accomplishing the pedestrian detection task.

The highlights of this chapter include:

1. Human can recognize pedestrians at faster rate even at night-time due to the strong visual significance. Meanwhile, pedestrians are frequently brighter than their surroundings in far infrared photos, indicating their saliency. This study is motivated by the human attention process, and a saliency model based on a wavelet feature map is employed to generate a suitable pedestrian region based on this logic.
2. To separate pedestrians from other background objects, pedestrian classification model with local denary signature that has ten levels of image representation is used to classify the ROI region as pedestrian or other objects.

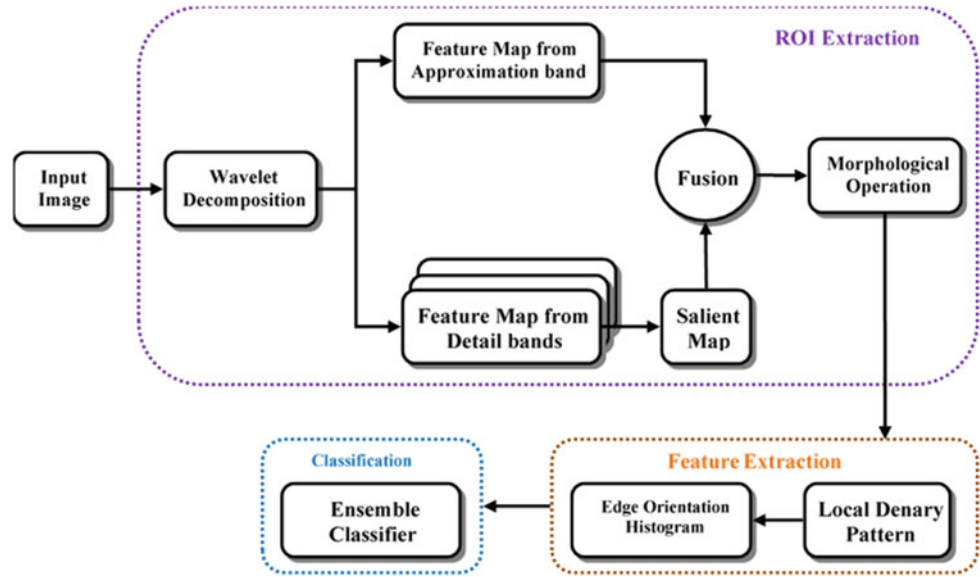
The remainder of this chapter is organized as follows: Sect. 2 describes methodology for pedestrian detection in far infrared images. Section 3 presents experimental results and related discussion on extracting potential pedestrian region and classification using the proposed LDEP features. Finally, Sect. 4 gives the concluding remarks of the proposed method.

2 Proposed Methodology

In this section, the methodology for detecting pedestrians in infrared images using saliency-based region of interest extraction and local denary description (LDeP) is explained. Figure 1 depicts the block diagram of the recommended approach for infrared image pedestrian detection.

From the input image, the region of interest was detected by multi-resolution-based saliency map generated by fusing feature map from approximation band and detailed sub-bands

Fig. 1 Block diagram of proposed method



followed by thresholding. Artifacts due to illumination and shadows, non-pedestrian may exist in the extracted region. Here, a pre-learned model is formed using local denary pattern followed by edge orientation histogram (EOH) from the database having 5578 pedestrian thermal images and 3375 background thermal images. With this model, the detected region of interest was categorized as pedestrian or non-pedestrian using ensemble-based classifier.

2.1 Wavelet Decomposition and Feature Map Generation

Wavelet transform has the advantage of providing multi-scale spatial and frequency analysis of the image. When the input image is processed with multi-scale filter banks, it results in approximate and detailed sub-bands. As the pedestrian in thermal images appears at different scales, wavelet transform with Daubechies wavelets (Db4) is chosen to examine the thermal image to highlight the salient regions.

The multi-scale wavelet decomposition is defined as follows.

$$\begin{cases} [A(k), H(k), V(k), D(k)] = WT(I(x, y)) & k = 1 \\ [A(k), H(k), V(k), D(k)] = WT(A(k-1)) & k > 1 \end{cases} \quad (1)$$

where $I(x, y)$ represents the input image, k is the number of decomposition level, and $WT(.)$ represents wavelet decomposition. $A(k)$ is the approximation output representing low frequency information, and $H(k)$, $V(k)$, and $D(k)$ denotes the wavelet coefficients of horizontal, vertical, and diagonal detailed information at k th decomposition level.

Feature Map Generation Using Detailed Sub-band: Feature maps are generated by reconstructing the decomposed high-frequency sub-bands ignoring approximation sub-band. The feature maps are constructed as follows.

$$f_{kd}(x, y) = IWT_k(H(k), V(k), D(k)) \quad (2)$$

Global Saliency Map Construction: The richer the information of interest, the higher the frequency information of the saliency map. The greatest value across the feature map is used to determine the relevance of the pixels from the created feature maps. Let $p(i, j, n)$ be the feature maps generated by the detailed sub-band, n represents the number of feature maps, and (i, j) indicates the coefficient location.

$$SFM(i, j) = \max(p(i, j, 1 : n)) \quad (3)$$

$SFM(i, j)$ contains all of the feature maps' statistical relationships and can highlight critical information that local contrast cannot discover.

Feature Map Generation using Approximation Sub-band: Here, feature maps are generated by reconstructing the decomposed approximation sub-band by ignoring detailed sub-bands. The feature maps are constructed as follows.

$$f_{ka}(x, y) = IWT_k(A(k)) \quad (4)$$

Feature Fusion: The feature map generated from the approximation (f_{ka}) and detailed band ($SFM(i, j)$) is thresholded to get the binary representation (B_a for approximation band and B_d for detailed band). *Morphological Operations:* Morphological procedures allow changing the geometric content of an image in a systematic way keeping the geometric assets stable. Opening, closure,

dilation, and erosion are the fundamental morphological operations. The shape and size of the structuring element used to process the image determine the amount of pixels added or removed from the object in the image. In the morphological dilation and erosion processes, the state of any given pixel in the output image is specified by the pixel and its neighbours in the input image.

2.2 Local Denary Pattern

Thermal image representations lag the fine scale textures. Therefore, spatial structures can be used as the discriminating features. Local denary pattern with directional representation is efficient in capturing local structural properties of the pedestrian. Consider the smallest whole unit, i.e. 3×3 pixels region having eight directions surrounding the pixel. The nine intensity values of the 3×3 pixels region are represented as: $\{B_0, B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8\}$. The local denary signature can be extracted by.

$$D(\text{abs}(B_{x \pm dx, y \pm dy} - B_{x,y})) \quad (5)$$

where $(x \pm dx, y \pm dy)$ is the neighbouring pixel, and p is $\text{abs}(B_{x \pm dx, y \pm dy} - B_{x,y})$.

The image with eight-bit representation has the maximum value of 255, and the maximum absolute difference will be 255. The difference range of (0–255) is fragmented into ten parts and assigned ten representation levels as shown in Table 1.

As the local denary signature has totally ten levels of representation, the combination of all eight neighbouring pixels results in 10^8 possible units in total. It is splitted into ten representative units corresponding to each level, i.e. the first representation named as L_0 is assigned with binary value 1 for $D(p) = 0$ and 0 for other values of $D(p)$, similar pattern is assigned to representations ($L_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8$ and L_9).

Table 1 Relationship between p and $D(p)$

p	$D(p)$
$P = 0$	9
$P \geq 1$ and $p \leq 28$	8
$P \geq 28$ and $p \leq 56$	7
$P \geq 56$ and $p \leq 84$	6
$P \geq 84$ and $p \leq 112$	5
$P \geq 112$ and $p \leq 140$	4
$P \geq 140$ and $p \leq 168$	3
$P \geq 168$ and $p \leq 196$	2
$P \geq 196$ and $p \leq 224$	1
$P \geq 224$	0

$$L_j = \begin{cases} 1 & D(p) = j \\ 0 & D(p) \neq j \end{cases} \quad (6)$$

The eight bits generated from each representation are put together clockwise and converted to a base 2 number. The local denary pattern formed as a result is less sensitive to greyscale variation. Example of local denary pattern computation is shown in Fig. 2.

Local Denary Pattern Computation—Example

The edge orientation histogram describes the local orientation and is useful for pedestrian categorization because pedestrian has sharp edges in the legs or trunk. The edge orientation histogram of all the signatures is concatenate to serve as the feature vector.

2.3 Overlapping Pedestrian Detection

As a result of salient region extraction, it is observed that in some cases more than one blob merges and generates erroneous result. The merging may occur in horizontal/vertical direction, and there is a need to split the merged pedestrian blobs. Here, vertical and horizontal projection profiles are used to identify the overlapping pedestrians for the candidate blobs having width to height ratio greater than 0.76 (Davis & Keck, 2005).

For an image $p(x,y)$ with m_p rows and n_p columns, vertical projection profile (Vp) (Li et al., 2012) and horizontal projection profile (Hp) are given by

$$H_p = \sum_{x=1}^{(m_p)} p(x, y) \quad (7)$$

$$V_p = \sum_{y=1}^{n_p} p(x, y) \quad (8)$$

By analysing the crest and trough of the horizontal/vertical profile, it is evident that the overlapping pedestrians can be easily separated. The profile exhibits trough at boundaries, and the location of these minima points marks the boundaries. Similarly, vertical projection profile is used to perform segmentation of overlapping pedestrian in horizontal direction as the valleys are created corresponding to the point of merging. These boundaries can be identified with the help of these minima points.

The proposed foreground extraction algorithm is as follows:

- Compute the width/height ratio of the candidate ROI as $P_{wh} = l_w/l_h$, where l_w and l_h represent the width and height of the candidate ROI.
- Obtain the intensity vertical projection curves and horizontal projection for $P_{wh} > 0.76$.

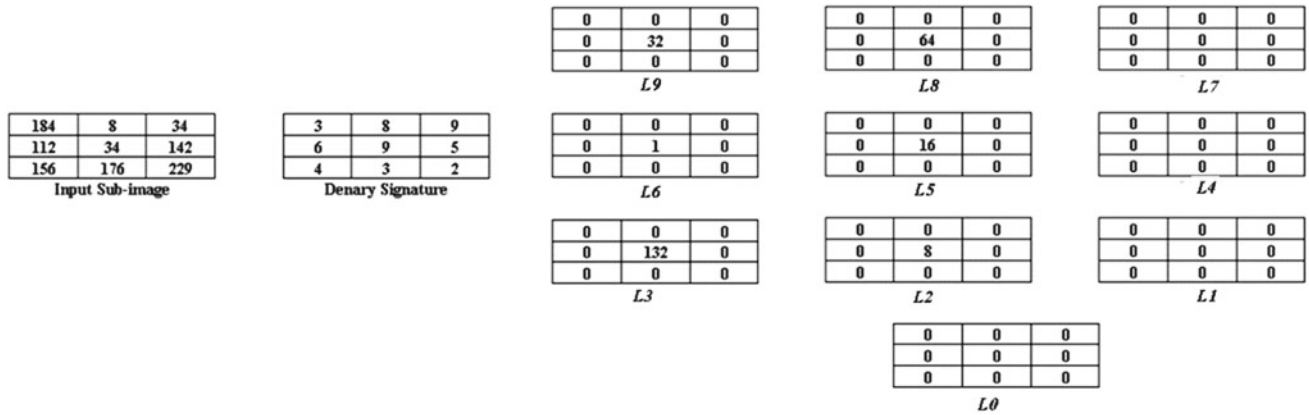


Fig. 2 Example for local denary pattern computation

- (c) Detect the trough locations from the projection curves.
- (d) The trough location decides the point of separation between the two pedestrians. Separate the overlapping pedestrian in to two pedestrians with the trough location.

2.4 LogitBoost Classifier

Boosting is a popular strategy for training binary and multi-class classification classifiers. Multiclass LogitBoost (Li et al., 2012), one of the prominent boosting versions influenced by a statistical approach, includes loss, the function model and the optimization technique.

For N-class classification ($N \geq 2$), consider training set with K number of samples $\{x_i, y_i\}_{i=1}^k$ where x_i is a feature value and $y_i \in \{1, \dots, K\}$ represents class label. A prediction function $F(x) \in \mathbb{R}^K$ is learned from the training set.

For the test data, the prediction is by $\hat{y} = \arg \max_k F_k, k = 1, \dots, K$. By minimizing a goal function using training data, the function F is obtained:

$$\text{Loss} = \sum_{i=1}^N L(y_i, F_i) \quad (9)$$

$L(y, F)$ term is a loss for a single training example (x, y) .

LogitBoost adopts the logistic loss, which is an implicit function of y and F :

$$L(y, F) = - \sum_{k=1}^K r_k \log(p_k) \quad (10)$$

2.5 Performance Measures

The confusion matrix is used to display how well a model made its predictions. Both precision and recall can be interpreted from the confusion matrix. Here, the task is considered as a binary classification problem where the classifier needs to classify the region of interest into background or human. If the model is to predict the human as background, then it is false positive. A false negative works the opposite way; the model predicts background as human. True positive is predicting human as human, and true negative is predicting background as background.

Precision

The ratio of true positives to the total of true positives and false positives is known as precision.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall

Instead of focusing on the number of false positives predicted by the model, recall considers the number of false negatives that were included in the mix.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (12)$$

When a false negative is anticipated, the recall rate is punished. The yin and yang of assessing the confusion matrix are precision and recall.

3 Results and Discussion

In this section, the robustness of the proposed pedestrian detection algorithm evaluated on the benchmarked OSU (Ohio State University) Thermal Pedestrian Database (Davis & Keck, 2005) is discussed.

The experimental validation of the proposed algorithm is on OSU (Ohio State University) Thermal Pedestrian Database which serves as a benchmark dataset with infrared images. The dataset was acquired from the pathway of Ohio State University campus with the resolution of 360×240 . The dataset has images captured under varying environmental conditions like rainy cloudy and normal conditions.

3.1 Wavelet Transform-Based Saliency Region Construction—Results

Figure 3 shows the decomposition of the input image into approximation and detailed bands. Here, k indicates the level of decomposition. $k = 1$ indicates first level of decomposition resulting in four sub-bands (1 approximation and 3 detailed). With $k = 3$ resulting in 10 bands (1 approximation and 9 detailed).

Figure 4 shows the feature maps of the global saliency by multi-level reconstruction. With the level of decomposition is increased, the more detailed components are visualized clearly.

Sample results showing candidate regions obtained by wavelet-based saliency analysis are shown in Fig. 5. Figure 5a, a1, and a2 depicts sample input images from classes 1, 2, 4, and 7, respectively, obtained under various environmental conditions such as fair cloudy and light rain. Figure 5b, b1, and b2 presents feature map generated by the approximation component, and it is observed that it retains the low frequency components and smoothen the image and reduce background clutter. The feature map is binarized by thresholding and portrayed in Fig. 5c, c1, and c2. The sample results of the saliency map generated from the

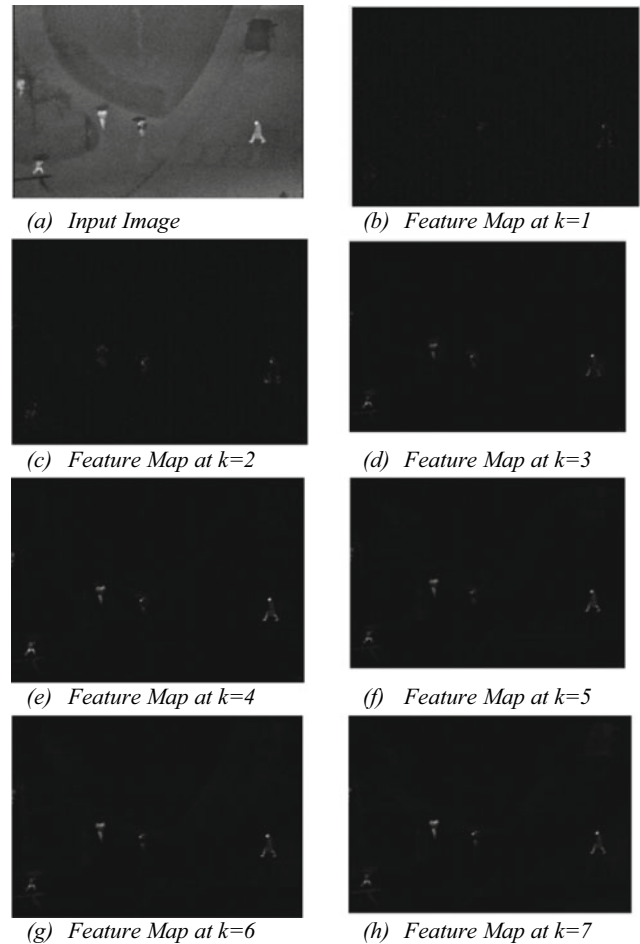
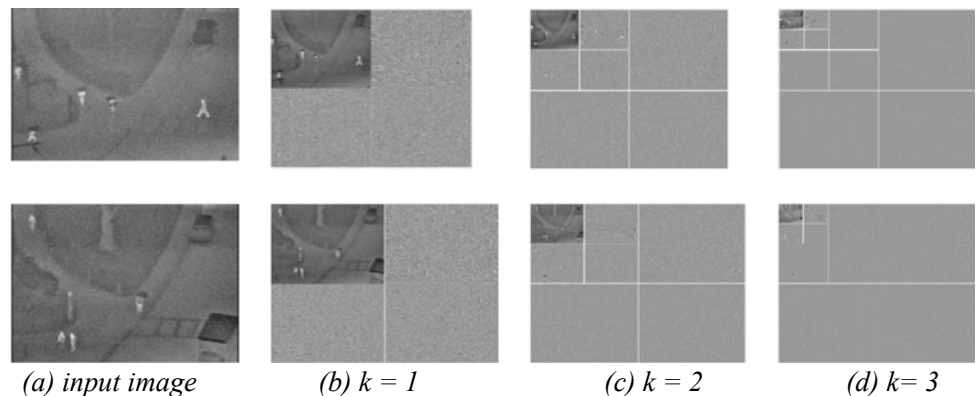


Fig. 4 Feature maps generated from detail sub-bands

detailed sub-bands is shown in Fig. 5d, d1, and d2. The saliency map generated by detailed sub-band indicates the distinction between pedestrian and the surrounding background. Thus, the targets under the various complex backgrounds with varying environmental conditions can be detected accurately and firmly. The feature map is binarized by thresholding and portrayed in Fig. 5e, e1, and e2.

Fig. 3 First three levels of the wavelet decomposition



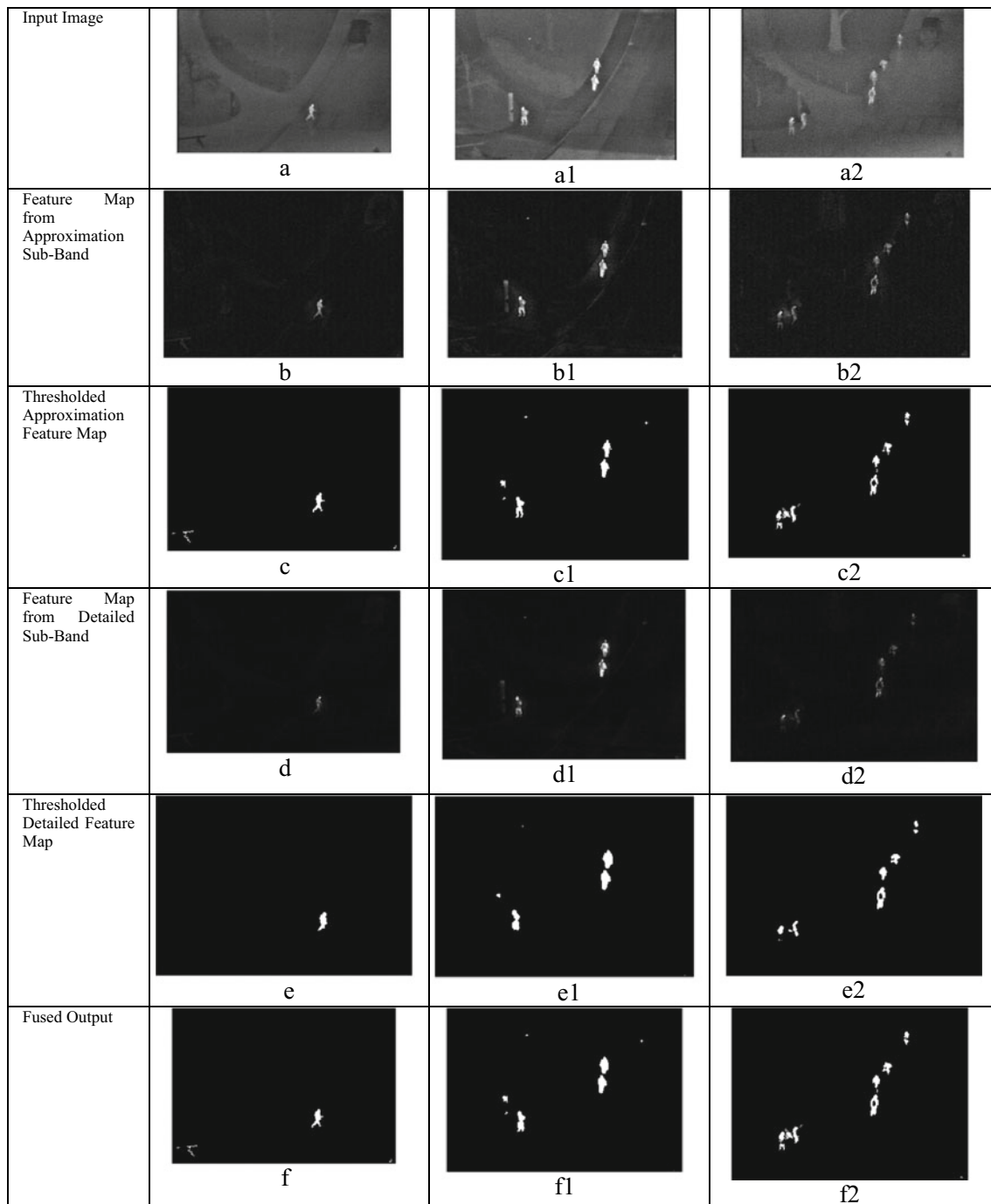


Fig. 5 Salient regions detected from OSU thermal pedestrian database

The fused output in Fig. 5f, f1, and f2 suggests that the pedestrian's shape and structure information are better maintained. Even if some scattered interferences remain, the pseudo-target can be largely eliminated during the subsequent target identification. Furthermore, because of the low repetition and lower number of false alarms, it is conducive to the discrimination process, and the computing speed may be substantially increased.

3.2 Pedestrian and Non-Pedestrian Classification Results

The salient blobs extracted have pedestrian and non-pedestrian objects having higher saliency characteristics as shown in Fig. 5. Hence, a strong classifier built upon candid discriminative features is required to distinguish pedestrian from other background objects. Here, an effective

human descriptor with local denary pattern with edge orientation histogram is used with ensemble-based classifier. For getting the prior knowledge, 5578 pedestrian images and 3375 images without pedestrians were used. Some of the sample pedestrian and background objects are shown in Fig. 6. These images are used to build a model for pedestrian/background classifier. Classification accuracy of 95.05% is reported using LogitBoost ensemble classifier with local denary pattern and edge orientation histogram.

Table 2 illustrates the pedestrian classification results based on features such as grey-level co-occurrence matrix (GLCM), histogram of gradients (HOG), edge orientation histogram (EOH), CENTRIST, local binary pattern (LBH), local ternary pattern (LTP), local denary pattern (LDEP), LBP + EOH, LTP + EOH, LBP + HOG, LTP + HOG, and LDEP + EOH. Among the used features, LDEP + EOH reports higher classification rate and used in further processing.

Figure 7 plots the horizontal and vertical projection profile curve of sample thermal image (Fig. 8).

The obtained prior model with LDEP + EOH used in online classification of salient regions detected from OSU

Thermal Pedestrian Database. Figure 8a, a1, a2, a3, a4, and a5 is the input image, and it can be seen that the pole, the car and the tree are also recognized in the salient region extraction phase due to heat radiation, resulting in a false alarm. Figure 8b, b1, b2, b3, b4, and b5 displays the detected results. From Fig. 8b, it is observed that three significant regions are detected and classified as pedestrians and marked in green colour, Fig. 8b1 has three pedestrians labelled in green colour and an overlapping pedestrian clearly separated and labelled correctly. In Fig. 8b2, a pole detected as salient region is classified as background object and marked in red colour. Similarly, a pole, a tree, and a car are tagged in red colour, whereas the detected pedestrians are highlighted in green colour in Fig. 8b3. Figure 8b5 shows the three pedestrians highlighted in green colour and a pavement tagged in red colour generating false alarm and classifying background information as pedestrian.

Figure 9 shows the confusion matrix for classifying human and background objects.

The recall and precision of the pedestrian detection method are tabulated in Table 3 to assess its performance. In

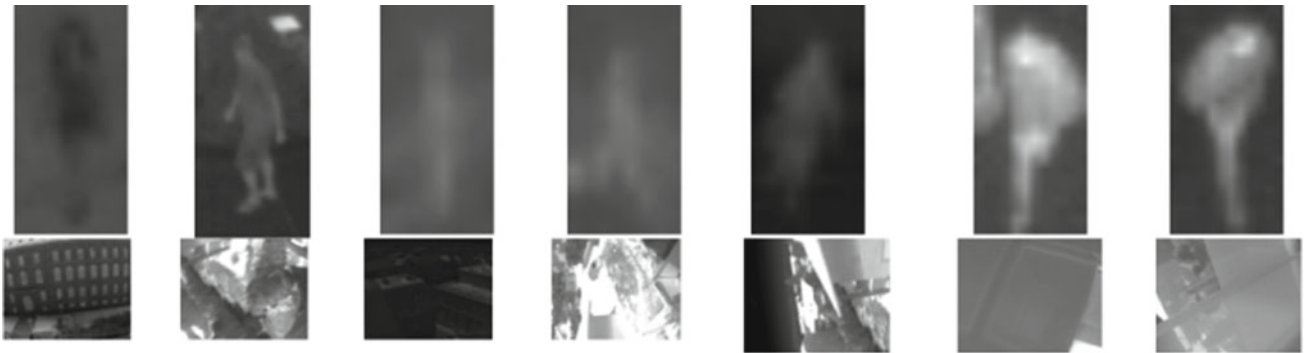


Fig. 6 Sample thermal pedestrian and background images used for training the model

Table 2 Pedestrian versus background classification

	Features used	Recall	Precision	Accuracy
1	GLCM	87.912	89.544	88.728
2	HOG	94.505	92.470	93.499
3	EOH	93.681	90.210	91.880
4	Centrist	90.109	86.772	88.352
5	LBP + EOH	82.417	83.798	83.434
6	LTP + EOH	87.362	89.325	88.587
7	LDEP + HOG	92.582	92.837	92.806
8	LTP + HOG	90.109	87.002	88.486
9	LBP + HOG	85.439	84.741	85.213
10	LDEP + EOH	95.879	93.315	95.052

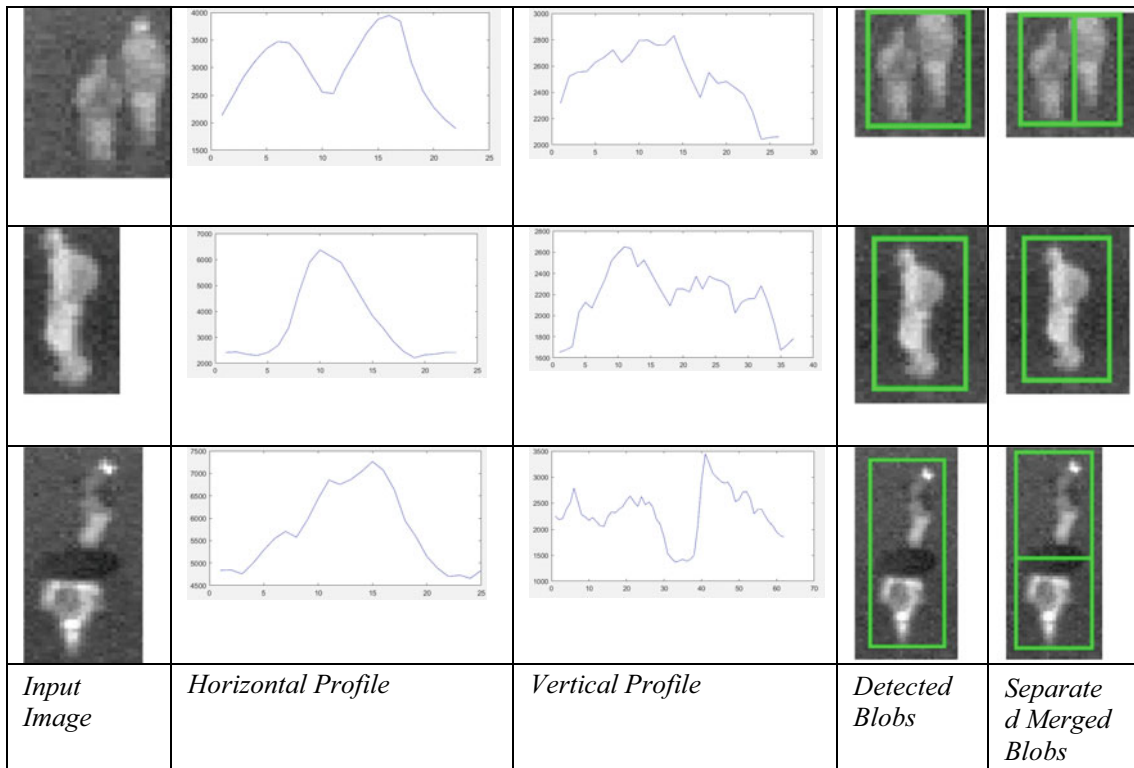
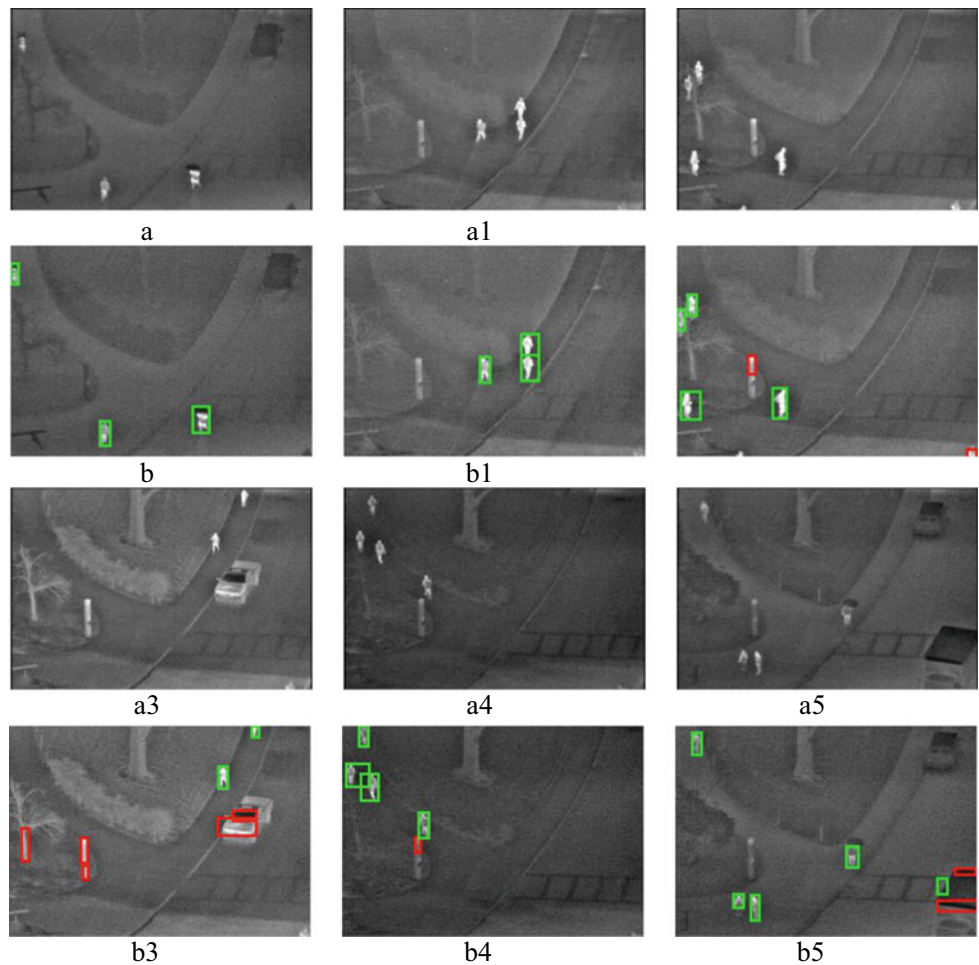


Fig. 7 Horizontal and vertical profile of an infrared image

Fig.8 Results showing pedestrian and background classification



	Human	Background
Human	98.91	1.09
Background	0	100

(a): Confusion Matrix - Class 1 Image Frames

	Human	Background
Human	100	0
Background	25	75

(c): Confusion Matrix - Class 3 Image Frames

	Human	Background
Human	99.01	0.99
Background	0	100

(e): Confusion Matrix - Class 5 Image Frames

	Human	Background
Human	100	0
Background	0	100

(g): Confusion Matrix - Class 7 Image Frames

	Human	Background
Human	100	0
Background	10	90

(i): Confusion Matrix - Class 9 Image Frames

	Human	Background
Human	98.04	1.96
Background	0	100

(b): Confusion Matrix - Class 2 Image Frames

	Human	Background
Human	98.18	1.82
Background	5.26	94.74

(d): Confusion Matrix - Class 4 Image Frames

	Human	Background
Human	98.99	1.01
Background	0	100

(f): Confusion Matrix - Class 6 Image Frames

	Human	Background
Human	98.04	1.96
Background	0	100

(h): Confusion Matrix - Class 8 Image Frames

	Human	Background
Human	98.95	1.05
Background	5.88	94.12

(j): Confusion Matrix - Class 10 Image Frames

Fig. 9 Confusion matrix for human background classification

all classes, the LDEP + EOH feature has a higher recall rate and precision rate than the other features, indicating that the proposed technique surpasses the others.

Table 4 presents the comparisons between proposed method and other popular pedestrian detection methods and summarizes the comparison results. The Li et al. (2012) method combines the HOG features with geometric characteristics and attained the 86.00% recall rate and 95.00% precision rate. Davis et al. (2004) method secured the 94.60% recall rate and 99.40% precision rate by performing initial screening with generalized person template derived from contour saliency maps to quickly detect person regions while ignoring most of the background. The hypothesized person regions are then validated with an AdaBoost ensemble classifier. Zelin et al. (Li et al., 2011) method earned the 97.80% recall rate and 99.60% precision rate by mid-level feature descriptor for pedestrian detection in thermal domain, which combines pixel-level steering Kernel regression weights matrix (SKRWM) with their corresponding covariances. Soundrapandiyan et al. (Rajkumar, 2017) used LoG filter with kurtosis for background suppression and detected pedestrians using L-moment-based local thresholding and obtained the 99.09% recall rate and 99.50% precision rate. Further, it can be observed that the

proposed method with LDEP + EOH-based classifier has the best detection performance from Table 4 as it represents human visual system for region of interest extraction and effective structural descriptor to define pedestrian.

4 Conclusion

In this chapter, a reliable method for detecting pedestrians in infrared image is presented. By merging the information from the feature maps generated by the approximate and detailed wavelet sub-bands, the background suppression module processes the image and extracts the salient region. The sparsity basis is wavelets, and the conspicuous blobs are well represented by thresholding, reducing the strain on post processing. A dictionary of LDEP features is used to effectively categorize the observed blobs as pedestrian or other background objects. It is observed that pedestrian in IR images lacks fine details and it is effectively represented by LDEP which is a structural descriptor. The experimental results proved that the proposed method with LDEP + EOH exhibits high detection rate by comparing it with other different feature extraction methods and other existing pedestrian detection methods.

Table 3 Classification accuracy of different features

Class	No. of images	GLCM		Centrisit		LBP + EOH		LTP + EOH		HOG		LBP + HOG		LTP + HOG		LDEP + EOH	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
1	31	90.11	87.23	90.11	87.23	93.10	98.78	93.02	91.38	100.00	98.91	92.05	98.78	92.05	98.78	100.00	98.91
2	28	95.00	92.23	95.00	92.23	100.00	92.78	93.02	91.38	100.00	92.78	93.75	92.78	93.75	92.78	100.00	98.04
3	23	83.17	92.31	79.21	80.81	80.77	48.55	93.02	91.38	94.95	78.99	94.95	78.99	94.95	93.07	100.00	100.00
4	18	84.40	97.87	84.40	97.87	84.30	98.08	93.02	91.38	92.73	98.08	92.73	93.58	92.73	93.58	99.08	98.18
5	23	87.13	88.00	87.13	88.00	90.65	96.04	93.02	91.38	97.98	96.04	91.51	96.04	91.51	96.04	99.01	100.00
6	18	84.54	90.11	84.54	90.11	92.86	91.00	93.02	91.38	92.86	91.00	88.35	91.00	93.81	91.00	100.00	98.98
7	22	94.68	88.12	94.68	88.12	97.67	100.00	93.02	91.38	97.67	100.00	97.67	100.00	97.67	100.00	100.00	100.00
8	24	90.91	91.84	90.91	91.84	82.46	97.92	93.02	91.38	90.38	97.92	90.38	97.92	94.95	97.92	100.00	98.02
9	73	94.74	90.00	94.74	90.00	89.13	100.00	93.02	91.38	94.74	100.00	90.00	100.00	94.74	100.00	98.94	100.00
10	24	92.78	88.24	82.47	86.96	81.31	87.88	93.02	91.38	89.69	87.88	89.69	87.88	89.69	87.88	99.94	98.96
Average		89.75	90.59	88.32	89.32	89.23	91.10	93.02	91.38	95.10	94.16	92.11	93.70	93.58	95.10	99.40	99.11

Table 4 Comparison of proposed method with state-of-the-art methods

Class	No. of images	Wei Li et al. method (Li et al., 2011)		Davis et al. method (Davis & Sharma, 2004)		Zelin Li et al. method (Rajkumar, 2018)		Soundrapandiyan et al. method [45]		Proposed Method	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
1	31	86.00	98.00	97.00	100.00	98.00	100.00	100.00	100.00	100.00	98.91
2	28	95.00	97.00	94.00	100.00	100.00	100.00	100.00	100.00	100.00	98.04
3	23	69.00	84.00	100.00	99.00	91.00	100.00	99.01	98.02	97.03	100.00
4	18	100.00	92.00	98.00	99.00	99.00	100.00	100.00	100.00	99.08	98.18
5	23	83.00	94.00	89.00	100.00	98.00	100.00	98.02	98.02	99.01	100.00
6	18	91.00	98.00	96.00	100.00	100.00	100.00	98.97	100.00	100.00	98.98
7	22	68.00	94.00	98.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
8	24	83.00	100.00	76.00	99.00	100.00	97.00	96.96	98.99	100.00	98.02
9	73	96.00	91.00	100.00	100.00	99.00	100.00	100.00	100.00	98.94	100.00
10	24	79.00	100.00	98.00	97.00	93.00	99.00	97.93	100.00	99.94	98.96
Average		85.00	94.80	94.60	99.40	97.80	99.60	99.09	99.50	99.40	99.11

References

- Alonso, I., Llorca, D., Sotelo, M., Bergasa, L., Del Toro, P., Ocana, M., & Garrido, M. (2007). Combination of feature extraction methods for SVM pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 8, 292–307.
- Andreone, L., Bellotti, F., de Gloria, A., & Lauletta, R. (2005). SVM-based pedestrian recognition on near-infrared images. In: *2005 international symposium on image and signal processing and analysis*, pp. 274–278.
- Apatean, A., Rogozan, A., & Benschair, A. (2008). Objects recognition in visible and infrared images from the road scene. In: *2008 IEEE international conference on automation, quality and testing, robotics*, pp. 327–332.
- Bertozzi, M., Broggi, A., Felisa, M., & Vezioni, G. (2006). Low-level pedestrian detection by means of visible and far infra-red tetra-vision. In: *2006 IEEE intelligent vehicles symposium*, pp. 231–236.
- Bertozzi, M., Broggi, A., Gomez, C., Fedriga, R., Vezioni, G., & Del Rose, M. (2007). Pedestrian detection in far infrared images based on the use of probabilistic templates. In: *2007 IEEE intelligent vehicles symposium*, pp. 327–332.
- Biswas, S. K., & Milanfar, P. (2017). Linear support tensor machine with LSK channels: pedestrian detection in thermal infrared images. *IEEE Transactions on Image Processing*, 22, 4229–4242.
- Cai, Y., Sun, X., Wang, H., Chen, L., & Jiang, H. (2016). Night-time vehicle detection algorithm based on visual saliency and deep learning. *Journal of Sensors*, pp. 1–7.
- Calafut, M. Multiple-object tracking in the infrared. Final Project (EE368) of Stanford University, Stanford University: Stanford, CA, USA, pp. 1–6.
- Dai, C., Zheng, Y., & Li, X. (2007). Pedestrian detection and tracking in infrared imagery using shape and appearance. *Computer Vision and Image Understanding*, 106, 288–299.
- Dai, C., Zheng, Y., & Li, X. (2005a). Layered representation for pedestrian detection and tracking in infrared imagery. In: *2005a IEEE computer society conference on computer vision and pattern recognition—workshops*, pp. 1–8.
- Dai, C., Zheng, Y., & Li, X. (2005b). Layered representation for pedestrian detection and tracking in infrared imagery. *IEEE computer society conference on computer vision and pattern recognition—workshops*, pp. 1–8.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In: *2005 IEEE conference on computer vision and pattern recognition*, pp. 886–893.
- Davis, J.W., & Keck, M.A. (2005). A two-stage template approach to person detection in thermal imagery. In: *2005 IEEE workshop on applications of computer vision*, pp. 364–369.
- Davis, J. W., & Keck, M. A. (2005). A two-stage approach to person detection in thermal imagery. In: *2005 proceedings workshop on applications of computer vision*.
- Davis, J. W., & Sharma, V. (2004). Robust detection of people in thermal imagery. In: *2004 international conference on pattern recognition*, pp. 713–716.
- Davis, J. W., Sharma, V. (2005). Fusion-based background-subtraction using contour saliency. In: *2005 IEEE computer society conference on computer vision and pattern recognition—workshops*, pp. 1–9.
- Davis, J. W., & Sharma, V. (2007). Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding*, 106, 162–182.
- Fang, Y., Yamada, K., Ninomiya, Y., Horn, B. K., & Masaki, I. (2004). A shape-independent method for pedestrian detection with far-infrared images. *IEEE Transactions on Vehicular Technology*, 53(6), 1679–1697.
- Fang, Y., Yamada, K., Ninomiya, Y., Horn, B., & Masaki, I. (2004). Comparison between infrared-image-based and visible-image-based approaches for pedestrian detection. *IEEE Transactions on Vehicles Symp*, 6, 1679–1697.
- Fayad, F., & Cherfaoui, V. (2007). Tracking objects using a laser scanner in driving situation based on modeling target shape. In: *2007 IEEE Intelligent Vehicles Symposium*, pp. 44–49.
- Frolov, V., & Leon, F. (2010). Pedestrian detection based on maximally stable extremal regions. In: *2010 IEEE intelligent vehicles symposium*, pp. 910–914.
- Gate, G., & Nashashibi, F. (2008). Using targets appearance to improve pedestrian classification with a laser scanner. In: *2008 IEEE intelligent vehicles symposium*, pp. 571–576.
- Gidel, S., Checchin, P., Blanc, C., Chateau, T., & Trassoudaine, L. (2008). Pedestrian detection method using a multilayer laserscanner: Application in urban environment. In: *2008 IEEE/RSJ international conference on intelligent robots and systems*, pp. 173–178.

- Hurney, P., Waldron, P., Morgan, F., Jones, E., & Glavin, M. (2015). Night-time pedestrian classification with histograms of oriented gradients-local binary patterns vectors. *IET Intelligent Transport Systems*, 9, 75–85.
- John, V., Mita, S., Liu, Z., & Qi, B. (2015). Pedestrian detection in thermal images using adaptive fuzzy C-means clustering and convolutional neural networks. In: *2015 international conference on machine vision applications (MVA)*, pp. 246–249.
- Jungling, K., & Arens, M. (2009). Feature based person detection beyond the visible spectrum. In: *Computer vision and pattern recognition workshop*, pp. 30–37.
- Kim, J. H., Hong, H. G., & Park, K. R. (2017). Convolutional neural network-based human detection in night-time images using visible light camera sensors, 17, pp. 1065.
- Latecki, L. J., Miezianko, R., & Pokrajac, D. (2005). Tracking motion objects in infrared videos. In: *IEEE international conference on advanced video and signal based surveillance*, pp. 99–104.
- Li, Z., Wu, Q., Zhang, J., & Geers, G. (2011). SKRWM based descriptor for pedestrian detection in thermal images. In: *Proceedings of IEEE 13th international workshop on multimedia signal processing*, pp. 1–6. China
- Li, W., Zheng, D., Zhao, T., & Yang, M. (2012). An effective approach to pedestrian detection in thermal imagery. In: *Proceedings of eighth international conference on natural computation*, pp. 325–329. China
- Liu, Q., Zhuang, J., & Ma, J. (2013). Robust and fast pedestrian detection method for far infrared automotive driving assistance systems. *Infrared Physics & Technology*, 60, 288–299.
- Liu, J., Zhang, S., Wang, S., & Metaxas, D. N. (2016). Multispectral deep neural networks for pedestrian detection. *Computer Vision and Pattern Recognition*, arXiv:1611.02644.
- Malley, R. O., Jones, E., & Glavin, M. (2010). Detection of pedestrians in far infrared automotive night vision using region-growing and clothing distortion compensation. *Infrared Physics & Technology*, 6, 439–449.
- Papageorgiou, C., Evgeniou, T., & Poggio, T. (1998). *A trainable pedestrian detection system* (pp. 241–246). IEEE Intelligent Vehicles Symposium.
- Rajkumar, & Mouli, C. An approach to adaptive pedestrian detection and classification in infrared images based on human visual mechanism and support vector machine. *Arabian Journal of Science and Engineering*. <https://doi.org/10.1007/s13369-017-2642-8>
- Riaz, I., Piao, J., & Shin, H. Human detection by using centrist features for thermal images. *IADIS International Journal on Computer Science and Information Systems*, 8, 1–11
- Suard, F., Rakotomamonjy, A., Bensrhair, A., & Broggi, A. (2006). Pedestrian detection using infrared images and histograms of oriented gradients. In: *2006 IEEE intelligent vehicles symposium*, pp. 206–212.
- Sun, Z., Bebis, G., & Miller, R. (2006). Monocular pre-crash vehicle detection: Features and classifiers. *IEEE Transactions on Image Processing*, 15, 2019–2034.
- Sun, H., Hua, C., & Luo, Y. (2004). A multi-stage classifier based algorithm of pedestrian detection in night with a near infrared camera in a moving car. In: *2004 international conference on image and graphics*, pp. 120–123.
- Wagner, J., Fischer, V., Herman, M. & Behnke, S. (2016). Multispectral pedestrian detection using deep fusion convolutional neural networks. In: *2016 computational intelligence and machine learning (ESANN)*, pp. 509–514.
- Xu, F., Liu, X., & Fujimura, K. (2005). Pedestrian detection and tracking with night vision. *IEEE Transactions on Intelligent Transportation Systems*, 6, 63–71.
- Zhang, W., Wang, Z., Liu, X., Sun, H., Zhou, J., Liu, Y., & Gong, W. (2018). Deep learning-based real-time fine-grained pedestrian recognition using stream processing. *IET Intelligent Transport Systems*, 12(7), 602–609.
- Zhang, L., Wu, B., & Nevatia, R. (2007). Pedestrian detection in infrared images based on local shape features. In: *2007 IEEE conference on computer vision and pattern recognition Minneapolis*, pp. 75–85.



A Survey of Emerging Applications of Machine Learning in the Diagnosis and Management of Sleep Hygiene and Health in the Elderly Population

B. Banu Rekha and A. Kandaswamy

Abstract

Sleep hygiene is essential for maintaining a healthy lifestyle. Insomnia is the inability to sleep or to stay asleep at night. While several factors such as medication, stress, anxiety and work cycle may affect sleep patterns, it is a fact that aging also becomes a factor of disturbed sleep. Apart from facing the societal, physical and psychological challenges that arrive with aging, seniors also have to face the challenges posed by sleep inadequacy. Hence, timely assessment and self-management of sleep are essential. This chapter introduces the etiology of sleep and sleep disorders to introduce the readers to the rhythmic nature of sleep and the complexity involved in its testing. The chapter further describes the significant sleep disorders found in aging population and their diagnostic techniques. The chapter then presents a survey of the significant and emerging applications of machine learning algorithms to objectively assess the sleep through signals acquired from conventional equipments and sophisticated equipments with reduced complexity. Further, the recent technological advances in non-contact and unobtrusive monitoring of sleep are presented. The chapter finally proposes the need for smart bedding solutions and controlled environment for inducing sleep hygiene.

Keywords

Sleep disorders • Elderly population • Screening • Self management • Machine learning

1 Introduction

Mental health and sleep are twined together. Deprivation of sleep affects the state of mental well-being and in turn, and an unhealthy mental state altogether affects sleep. The American Academy of Sleep Medicine (AASM) and the Sleep Research Society (SRS) have recommended that ‘adults aged 18 to 60 years should sleep seven or more hours per night on a regular basis for ideal sleep health’. According to the National Sleep Foundation, 7 to 8 h of sleep is suggested for persons aged 65 years or older (Hirshkowitz et al., 2015). The World Economic Forum has published the findings of a worldwide study by an app called sleep cycle (<https://www.weforum.org>). New Zealand topped in this study with an average of 7.5 h of sleep clocked by each person. The top five countries included Finland, the Netherlands, Australia and UK. South Korea and Japan emerged as the countries with lowest sleep time per person. The important finding of this study was that loss of sleep is linked to a loss in GDP of the nation. Hence, measures should be taken for improvement of sleep quality and quantity. According to a 2019 Philips Global Sleep Survey study, 62% of adults reported unsatisfactory sleep across the globe (Chattu et al., 2018). Although being the primary factors in drowsiness led accidents, sleep deprivation on a long term leads to mental, emotional and physical woes (Stranges et al.). Clinical studies conducted on mental health issues view sleep deprivation as a co-occurring condition, while studies conducted on sleep deprivation view mental health issue as at relatively less risk. Hence, the interrelation between sleep and mental health issues looms large.

Maintaining sleep hygiene is essential for elderly people who might be battling age-related illness and sore health. Stranges et al. conducted a study on prevalence of sleep disorders on 40,000 participants across Asia and Africa with age 50 years and older (FitzGerald et al., 2013). The study provided bivariate and multivariate analyses on the data. This study observed that ‘sleep problems might represent an

B. B. Rekha (✉)
Department of Biomedical Engineering,
PSG College of Technology, Coimbatore, India
e-mail: bbr.bme@psgtech.ac.in

A. Kandaswamy
Center for Industrial Research and Development,
PSG College of Technology, Coimbatore, India

unrecognized public health issue in many developing countries, especially among older adults, given the ongoing aging of the population at the global level.’ Their findings also report that sleep problems were prevalent among the disabled persons that also lead to depression invariant of gender. It is also observed that aged population suffers from a lack of age-specific insurance policies. This lack of policies poses a challenge for the aging population who choose to pursue their careers (<https://www.philips.co.in>). Hence, it is suggested to have close functioning between professional health bodies, geriatric medicine and policymakers to draft age-specific welfare policies including sleep-related ailments.

In an observational study conducted on more than 500 elderly people who reported to a geriatric clinic in India with a mean average age of 66 years, a significant 32% of the study population was suffering from some form of insomnia (Indrajeet et al., 2014). While this study was conducted in a regional population, similar studies from regions across India support the growing cause of concern about insomnia in the elderly population. Table 1 presents the commonly occurring co-morbidities of sleep deprivation in the elderly population.

In the past few decades, sleep disorder has emerged as a discipline by itself, and immense research work has culminated in the state-of-the-art technologies for effective and efficient diagnosis, intervention and therapeutic approaches toward mitigation of sleep deprivation. Among the rising concerns of sleep related issues due to the Covid-19 pandemic. Considering these factors, this chapter is built to provide a comprehensive view of sleep etiology, consequences of sleep deprivation and quality of life concerns in the aged population due to sleep. Further this chapter provides pointers to the technological advancements and machine learning-based solutions used to address the issues of sleep disorders.

2 Etiology of Sleep

Historically, sleep was considered to be a state where the brain and body shut off and take rest. But, with the invention of instruments to record brain activity like

electroencephalogram (EEG), it was discovered that sleep is not a state. Interestingly, sleep was a process that consisted of majorly two phases: rapid eye movement (REM) and non-REM (Colten et al., 2006). The occurrence of these phases appeared to be in a cyclic manner. Experiments have shown that REM sleep phase is experienced during 30% of the total sleep time, even though it appears in a disjointed manner. The remaining 70% of the sleep time is experienced as NREM phase. Like REM, non-REM sleep also occurs in a disjointed manner. However, there are distinct changes in the NREM sleep phase that leads to labeling of the phase from stage 1 to stage 4. During a healthy sleep process, the shallow NREM sleep occurs first, and it gradually progresses through the stages. The REM sleep follows with short-term duration but gradually increases with the progress in sleep.

Figure 1 represents a typical sleep cycle. The vertical axis represents the stages of sleep such as awake, REM and NREM stages 1, 2, 3 and 4. The horizontal axis represents the number of hours of sleep progress. As seen from Fig. 1, the duration of REM sleep, also known as the deep sleep, is initially less but expands as time grows by. Toward the end of sleep time, it gets reduced. This cycle of sleep is estimated to repeat itself every 90 min roughly and nearly five times during an entire night time. As NREM sleep progresses, brain becomes immune to external stimuli. REM and NREM cycles alternate throughout the duration of sleep to provide a good quality of sleep. A good quality sleep can help in boosting the immune system of a person and also help in regulating the various hormones. The process of sleep is regulated by two bodily systems: circadian rhythm and sleep/wake homeostasis. If the REM/NREM sequence is altered temporally due to physiological or psychological disorders, the sleep of the person is said to be disrupted. The peculiarity of certain kind of sleep disorders is that the affected persons are unaware of their condition or give less importance to the fact of not getting enough sleep. Long-term disruption of sleep can lead to cardiovascular morbidity and even sudden death (Hoddes et al., 1972).

Table 1 Common co-morbidities of insomnia in the elderly population

Nature of co-morbidity	Co-morbidity
Physical	Chronic pain, risk of chronic obstructive pulmonary disease, diabetes and cardiovascular diseases
Cognitive	Higher likelihood of developing dementia, Alzheimer’s disease
Psychological	Depression, irritation, anxiety, mood swings
Social	Reduced work performance, reduced mobility

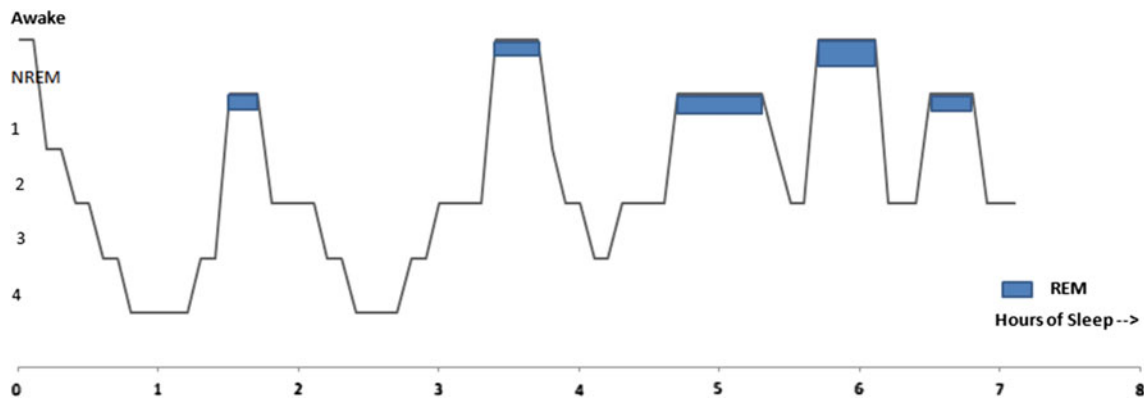


Fig. 1 Typical stages of a nocturnal sleep cycle (representational purpose)

3 Types of Sleep Disorders

The International Classification of Sleep Disorders (ICSD) is a standard reference text used for the clinical diagnosis of sleep disorders. This classification has been released as three versions: ICSD-1, ICSD-2 and ICSD-3. According to the ICSD-3 classification released in 2014, the following are the major categories of sleep disorders.

- Insomnia
- Sleep-related breathing disorders
- Central disorders of hypersomnolence
- Circadian rhythm sleep–wake disorders
- Parasomnias
- Sleep-related movement disorders.

3.1 Insomnia

Insomnia primarily describes about the quality of nocturnal sleep. The inability of a person to initiate or maintain sleep that leads to insufficient sleep duration is a characteristic of insomnia. Insomnia leads to a disgruntled daytime life, and a fair diagnosis can be made using patient-reported outcomes and sleep surveys. Even though the time spent on bed may be considerably normal, insomnia patients experience a non-refreshed sleep. Proper diagnosis of insomnia depends on identifying the individual sleep disorder that is one of the following categories: adjustment sleep disorder (acute insomnia), psychophysiological insomnia, paradoxical insomnia (formerly sleep state misperception), idiopathic insomnia, insomnia due to mental disorder, inadequate sleep hygiene, behavioral insomnia of childhood, insomnia due to drug or substance, insomnia due to medical condition, insomnia not due to a substance or known physiological condition and physiological (organic) insomnia.

Insomnias are of two types: primary and secondary. Primary insomnia is a disorder on its own and is not co-occurring with any other disorder of the body. Secondary insomnias occur as a co-morbid condition that is triggered due to a medical or psychiatric illness or may result from another type of sleep disorder.

3.2 Sleep-Related Breathing Disorders (SRBD)

Sleep-related breathing disorders are the common type of sleep disorders found in the modern society. SRBD represents the incidence of abnormal breathing events during sleep. While it does not deal specifically with nocturnal sleep, these disorders may occur even during daytime sleep. Sleep apnea is the clinical term that measures the number of times breath cessation happens during sleep and is also concerned about the duration of cessation (Max & Amir, 2017). SRBD is grouped into four categories such as obstructive sleep apnea (OSA), central sleep apnea, sleep-related hypoventilation and sleep-related hypoxemia disorder. While OSA is largely reported as an outcome of obesity and lifestyle disorders, central sleep apnea occurs due to the improper brain signals controlling the muscles of breathing. OSA primarily occurs due to obstruction in the airway and results in regular awakenings during the sleep. Hypoventilation and hypoxemia occur due to the shortness of ventilation. This leads to buildup of carbon-di-oxide and loss of oxygen in the body. The duration of apnea is critical as long durations of hypoxemia could be fatal (Fig. 2).

3.3 Central Disorders of Hypersomnolence

Hypersomnia refers to the condition wherein the subject experiences daytime sleepiness irrespective of the nocturnal sleep and circadian rhythm being undisturbed. Unintended

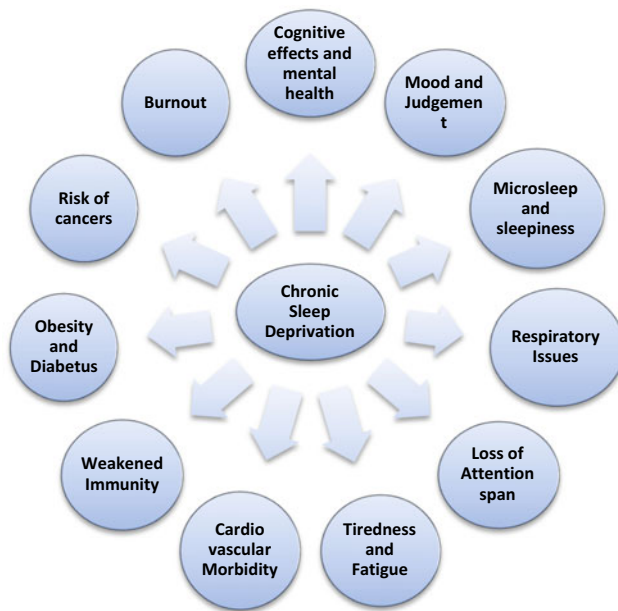


Fig. 2 Manifestations of chronic sleep deprivation

bouts of sleep during the daytime make the subject incapable of performing regular activities and lead to poor quality of life. Technically, hypersomnia is defined as sleep duration greater than 9 h in a day. Hypersomnia is categorized into the following: narcolepsy, recurrent hypersomnia, idiopathic hypersomnia, behavioral-induced insufficient sleep syndrome and hypersomnia due to a medical condition or drug abuse. Cataplexy is a condition in which a person may experience sudden loss of voluntary muscle tone due to strong emotions. Though appearing in brief, cataplexy can co-occur with narcolepsy.

3.4 Circadian Rhythm Sleep–Wake Disorders

Disruption in circadian rhythm can manifest as a misalignment between the pattern of sleep as accepted by the societal norm and one that the subject with this type of disorder experiences. Even though the REM and NREM cycling behaves normal after falling asleep, the time of sleep does not meet the need or expectations of the subject. This may lead to the apprehension of excessive sleepiness as reported by patients affected with this circadian rhythm disorders. The circadian rhythm sleep disorders are reported as belonging to one of the categories: delayed sleep phase type, advanced sleep phase type, irregular sleep–wake type, free-running (none trained) type, jet lag type, shift work type, circadian rhythm sleep disorders due to medical condition and sleep disorder due to drug or substance. Sleep disorders in the aged people are more likely to be an outcome of disturbed rhythm.

3.5 Parasomnias

Parasomnias are not actual sleep disorders but physical or experiential events that occur abnormally during sleep. The persons may be experiencing abnormal dreams, behaviors, emotions and perceptions. It may be affecting the functioning of autonomous nervous system. Hence, the central nervous system gets activated. Literature reports that in majority of parasomnia cases, it occurs along with another type of sleep disorder. Incidence of multiple type of parasomnias occurring in a single patient has also been reported. The types of parasomnia disorders are disorders of arousal (from non-REM sleep), confusional arousals, sleepwalking, sleep terrors, parasomnias usually associated with REM sleep, REM sleep behavior disorder, recurrent isolated sleep paralysis, nightmare disorder, sleep-related dissociative disorders, sleep enuresis, sleep-related groaning (catathrenia), exploding head syndrome, sleep-related hallucinations, sleep-related eating disorder, parasomnia due to a drug or due to a medical condition. Arousal disorders are related with NREM sleep while many of the other types of parasomnias occur during the REM sleep stage.

3.6 Sleep-Related Movement Disorders

Sleep-related movement disorders exhibit symptoms such as an urge to move the limbs that is irresistible. These movements are generally stereo-typed and follow a simple pattern. Although accompanied with uncomfortable or painful symptoms, the movement itself brings a relief to the subject. Restless legs syndrome, sleep-related bruxism, rhythmic movement disorder and periodic limb movement disorder are the common types of movement disorders experienced during sleep.

4 Aging-Related Sleep Disorders

The technological advancements in the field of medicine have considerably increased the lifespan of humans. This furthers the ability of elderly people to pursue careers and professional goals. Persons aged 65 or elder are considered to be the older workers in the workforce. According to a study by the World Bank, the population of older workers across the world is nearly 700 million in 2019. The age dependency ratio of old aged workforce is around 55% (<https://data.worldbank.org>). United Nations estimate that the population of 60+ years would double by 2050 and triple by 2100 (<https://www.sleepfoundation.org>). Considering that sleep problems among the elderly population are considered as a growing problem, screening for sleep disorders in the elderly has become essential.

The common type of sleep problems identified in the aging population is the multiple sleep–wake cycles, reduced total sleep time, reduced time of REM sleep and slow wave sleep (Suzuki et al., 2017). These are primarily attributed to the drop in homeostatic sleep pressure and reduced circadian signals. Few other sleep disorders found to be prevalent in the aging population are the restless leg syndrome (RLS), periodic limb movement disorder (PLMD) and parasomnias (Shinno, 2010). A significant proportion of sleep problems in the elderly are due to phase changes (getting to sleep earlier than normal or getting to sleep later than normal). In a study conducted on 182 adults who were aged 65 or higher, responses from three questionnaires, namely Athens Insomnia Scale (AIS), ESS and the ISI, were collected. The respondents were from a certain city of Poland and were educationally active. It was found that the prevalence of insomnia was higher, although there was no indication of excessive daytime sleepiness (Cybulski et al., 2019). This study observed that there is a need for improving sleep education programs among the population. Sleep-related data was collected from 14,001 elderly residents in Japan (Etsuji et al., 2009) through a cohort study with duration of 7 years. The major finding of the study was that among persons with poor sleep quality, a high risk of cardiovascular mortality was associated with longer duration of sleep. A meta-analysis carried out on sleep and dementia disorder found that sleep-related breathing disorders was linked to increased risk of dementia and insomnia increased the risk of Alzheimer's disease (Shi et al., 2018). Sleep deprivation, on a shorter term, has been found to be having a significant impact on reasoning ability, attention and speed of processing information and reduced attention spans (Dzierzewski et al., 2018). Research studies generally agree that irrespective of age, chronic sleep deprivation leads to obesity and hypertension.

A multidimensional sleep and mortality study was undertaken as a cohort on 8,668 adults aged 65–99 years (Meredith et al., 2019). Based on the data collected through self-reported sleep measures over a period of nearly 15 years, risk assessment for all-cause and cardiovascular morbidity was performed. The random survival forest (RSF) method was employed for the risk assessment. RSF is an ensemble-based statistical machine learning model, and it has been reported as a better model for survival analysis on high dimensional data. RSF is highly immune to outliers and overcomes the problem of overfitting that occurs with high dimensional data. An effort to predict sleep disorder from an asthma cohort of 1 million samples of adults was pursued. ML (KNN, SVM and RF) and deep learning techniques (RNN, LSTM, GRU and CNN) were employed on the disease histories of the asthma patients collected through Taiwan National Health Insurance Research Database (NHIRD) (Dinh-Van et al., 2020). CNN gave 95.1% accuracy in

predicting sleep disorder among the asthma patients. In a study carried out on osteoporotic fractures in men sleep dataset, a multivariate approach was employed through multivariable Cox models, survival trees and RSF on the sleep data. In this cohort study involving 2887 men over a period of 11 years, rhythmicity and continuity (derived from actigraphy measurements) emerged as the robust sleep predictors (Wallace et al., 2018). This study proposed an improvised sleep screening system for predicting mortality risk. The long-term effect of idiopathic rapid eye movement sleep behavior disorder (iRBD) on males diagnosed at age 50 years or older was studied (Schenck et al., 2013). After 13 years, it was found that nearly 81% of them developed either parkinsonism or dementia. This study presented the need for early neurological interventions in order to halt or delay PD or cognitive impairments.

4.1 Trends in Analysis of Sleep Disorders

This section presents the ways of assessing sleep quality through subjective and objective methods. Subjective methods are simple but are efficient in flagging certain events, while objective methods are based on rigorous scientific testing and data collected through sleep studies. This section also presents the emerging trends of applications of machine learning techniques in the analysis of sleep disorders.

4.1.1 Subjective Assessment of Sleep: Survey Questionnaire

Sleep was traditionally assessed through the level of motor movements, breathing pattern, mental response to stimulus and state of eyes (open/close). When these physical and physiological indicators show a deviation from a pattern, it is flagged off as a disrupted sleep. In spite of these patterns being intact, there is a possibility of an inherent sleep disorders. Hence, early sleep studies were conducted based on the symptoms expressed orally by the patients along with the complications diagnosed at the time of investigation. It was widely assumed that sleep disturbances are related to psychiatric conditions until the formulation of Pittsburgh sleep quality index (PSQI) happened (Buysse et al., 1989). This self-reported study contained a set of nineteen questions. The weighted scores for each question were then summed up to arrive at the index. A global PSQI score was proposed to differentiate good and bad quality of sleep with a threshold score of 5. The Stanford sleepiness scale (SSS) was used to assess the level of ‘alertness’ of people suffering from insomnia (MacLean et al., 1992) through a 7-point Likert scale. The descriptors ranged from ‘feeling active, vital alert, wide awake’ assigned with score 1 to that of ‘no longer fighting sleep, sleep onset, having dream like thoughts’

assigned to score 7. The Karolinska sleepiness scale (KSS) is a similar questionnaire-based tool aimed to evaluate the behavioral sleepiness of subjects during daytime (Kosuke et al., 2006). The Epworth sleepiness scale (ESS) is an assessment scale in which the subjects are asked to evaluate their likeliness to doze off during eight day today activities such as watching television or sitting inactive in a public park (Johns, 1991). The responses are collected as one among four options such as ‘never, slight chance, moderate chance and high chance’. Insomnia severity index (ISI) questionnaire subjectively assessed the severity of insomnia within a time window such as past 2 weeks and the effect of insomnia on the quality of life of the respondents (Morin et al., 2011). Further, the questionnaire tries to seek whether the respondents are worried or depressed due to their insomnia condition. Functional outcomes of sleep questionnaire (FOSQ) is a self-reported questionnaire used to assess the impact of excessive daytime sleepiness on the quality of life of adults. FOSQ also appraises the effectiveness of treatment methodologies used to mitigate daytime sleepiness (Izci et al., 2004). OSA Berlin questionnaire identifies high risk subjects to obstructive sleep apnea (Tan et al., 2017). Using a combination of questions on the snoring habits and the blood pressure, the subjective assessment is made on middle aged and older subjects. A concise and simple questionnaire was formulated by Chung et al. (Chung et al., 2008), known as STOP-BANG questionnaire. STOP-BANG targeted at finding sleep apnea conditions in surgical patients. The questionnaire collects information on the following aspects: snoring, tiredness during the day, existence of observed apnea, high blood pressure, body mass index, age, neck circumference and gender. The neck circumference has been found to be one of the crucial factors in diagnosing sleep apnea conditions.

The questionnaires such as PSQI, SSS, KSS, ESS, ISI, FOSQ, OSA Berlin, STOP-BANG and ISI are primarily tools for subjective assessment of sleep quantity and quality. Multiple sleep latency test (MSLT) performs objective assessment of sleep quality (Arand & Bonnet, 2019). Instead of self-reported responses, a sleep specialist measures the extent of daytime under controlled conditions. A measure of sleep latency and the number of naps consisting of REM stage are used to flag positive or negative subjects. Restless legs syndrome diagnostic index (RLS-DI) consists of ten items (Benes & Kohnen, 2009). The level of questions is framed in such a way to primarily understand whether the patient’s inability to sleep is due to an urge to move lower limbs and/or they feel any painful sensation in their legs. On affirming these conditions, the questionnaire estimates the essential criteria to diagnose RLS disorder. Narcolepsy is confirmed through the polysomnography test along with the results of MSLT. Patient-reported outcomes measurement information system (PROMIS) is a set of questions used to

evaluate the physical and mental health of people. It also can be used to estimate the level of social relationships of them. An improvised PROMIS physical function geriatric rehabilitation (PROMIS-PF-GR) template was used to study the physical health of geriatric patients (Jacobson et al., 2020). The study reported that the improvised questionnaire was effective and adept at evaluating geriatric health.

4.1.2 Objective Assessment of Sleep: Polysomnography—The Gold Standard Test

Polysomnography (PSG) test has laid a strong foundation for the diagnosis of sleep disorder and had unfolded new insights into sleep research (Jafari & Mohsenin, 2010). Polysomnography is a systematic method of acquiring sleep-related physiological parameters of a patient. It is generally a nocturnal test done under the supervision of sleep specialists and technicians. Designated sleep laboratories are equipped with polysomnography machine that can record inputs up to 40 channels. The signals from EEG, EOG, ECG, EMG, SpO₂, heart rate, abdominal effort, thorax effort, limb and chin movement, nasal airflow and snoring give a complete picture of the pathological processes that determine sleep. The sleep signals are divided into epochs of 30 s each and are manually scored. Till today, newer and advanced polysomnography equipments are built that are crucial in establishing sleep laboratories.

Polysomnograms enabled creation of standard data for the sleep research community. Even though polysomnography is the ‘gold’ standard for all forms of sleep studies, it is an expensive and time-consuming procedure. Various alternative methods were developed to screen sleep disorders and also to bring down the complexity of the testing procedure. Efforts for identification of sleep stages with the help of computers were made by researchers. Automated sleep staging algorithms appeared to reduce the cost of screening, reduce the complexity involved in a laboratory sleep test and also require less man power. The objective of sleep stage detection is to differentiate the REM and non-REM sleep. EEG alone cannot discriminate REM/NREM since the spatial and morphological characteristics are similar in both states. Hence, ECG, EOG and EMG signals are concurrently considered with EEG to perform REM/NREM discrimination (Fiorillo et al., 2019). Since acquiring EEG signals is a complex task, researches are being carried out to identify alternative indicators of sleep stages through secondary signals such as EMG, ECG and sPO₂ in lieu of EEG.

4.1.3 Automated and Intelligent Machine Learning Systems for Sleep Assessment

The primary task of polysomnography is the sleep staging, i.e., detection of REM, NREM and its subtypes. It is estimated that a PSG technologist may take up to 1.5 to 2 h

duration to manually score and verify one overnight PSG recording. Artificial intelligence (AI) algorithms have the potential to (i) improve the accuracy of event detection (ii) improve the accuracy of classifying diseases and disorders and (iii) predict the prognosis and treatment outcomes. The automated algorithms use the power spectrum of the EEG signal to distinguish between sleep and wake stages. The sleep epochs are typically classified as REM and NREM stages using the power spectrum of chin EMG, presence of K-complexes, spindles and the duration of delta waves. The rules laid by Rechtschaffen and Kales are considered for the delineation of these events. Table 2 presents the emerging applications of ML in automatic analysis of polysomnographic data.

U-Sleep has been developed as a pre-trained ready-to-use deep neural network (Perslev et al., 2021) that has set a milestone in the automated sleep staging process. It considers a continuous EEG signal from any one channel and an EOG signal from either right or left eye. The EEG is considered as a primary indicator of sleep/wake discrimination. High frequency and low amplitude variations represent wake state, and low frequency and high amplitude variations in EEG represent NREM state. The rhythm of the EEG signal in REM is similar to wake rhythm. Hence, REM stage is detected by observing the rapid changes occurring in the EOG signal along with EEG rhythm. The signals are

converted into a high frequency representation of 128 Hz, and individual segments are classified using ML techniques. Figure 3 pictorially represents the process involved in the U-sleep algorithm. An ELM algorithm was used to classify sleep stages from real-time polysomnographic recordings, and it was able to achieve an accuracy of 76.9% (Sun et al., 2017). A customized ML model named Somnivre (Allocca et al., 2019) was developed and applied on a combined datasets of human as well as animals. The dataset also represented the sleep data of old-aged people with mild cognitive impairment, but it was not found to have a significant impact on sleep staging. Modified and joint versions of CNN such as multi-task CNN and CNN+LSTM were used for automated stage classification and visualization (Krauss et al., 2021; Phan et al., 2019). An improvised method of deep kernel learning, viz. a DL model with a Gaussian process, was used to effectively analyze K-complexes which occur frequently during the sleep (Lechat et al., 2020). Apart from its clinical significance, K-complexes are also known to point toward influence of aging on sleep. Figure 4 represents the four stages involved in the work of extracting K-complexes. Extraction of KCs also aids in exploring daytime sleepiness and health impacts of impaired sleep.

Sleep breathing disorders have been estimated to affect roughly 30% of the population. The condition worsens with increases in age (McMillan & Morrell, 2016). Sleep apnea

Table 2 Trends in application of ML methods for the analysis of polysomnogram (PSG)

Type of ML/DL algorithm	Nature of analysis	Data used in the analysis	Performance of the ML algorithms
U-Sleep based on CNN (Perslev et al., 2021)	Label sleep stages based on EEG and EOG signals	PSG recordings from 16 various clinical studies	F1 score: 0.79
Extreme learning machine (ELM) (Sun et al., 2017)	Classify five sleep stages using EEG signals	EEG recordings of 2000 patients from PSG	Accuracy: 76.9%
Deep Kernel learning (DKL)_deep learning with a Gaussian process (Lechat et al., 2020)	K-complex frequency and morphology	Synthetic KCs derived from montreal archive of sleep study	F1 score: 0.78
Multi-task CNN (Phan et al., 2019)	Automatic sleep stage classification	Sleep-EDF expanded (Sleep-EDF) and montreal archive of sleep studies (MASS), unimodal	Accuracy: 82.3% & 83.6%
CNN + LSTM (Krauss et al., 2021)	Automatic Sleep Stage Classification	EEG recordings of 68 participants from PSG, at FAU	Increasing layers of NN increases chance of separation
Somnivre algorithm (Allocca et al., 2019)	Automated wake-sleep stage classification	UMH cohort data, UOH cohort and animal data (data of older adults with mild cognitive impairment)	F1 score range: 0.84–0.90

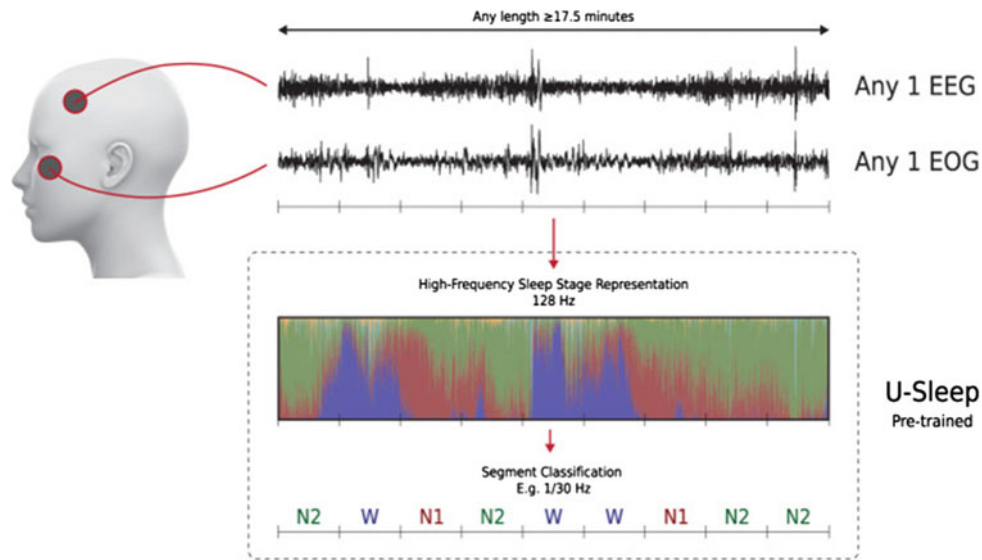


Fig. 3 Schematic of U-sleep HF sleep staging process (Perslev et al., 2021). A single-lead EEG and a single-channel EOG is utilized to extract the features of sleep stages

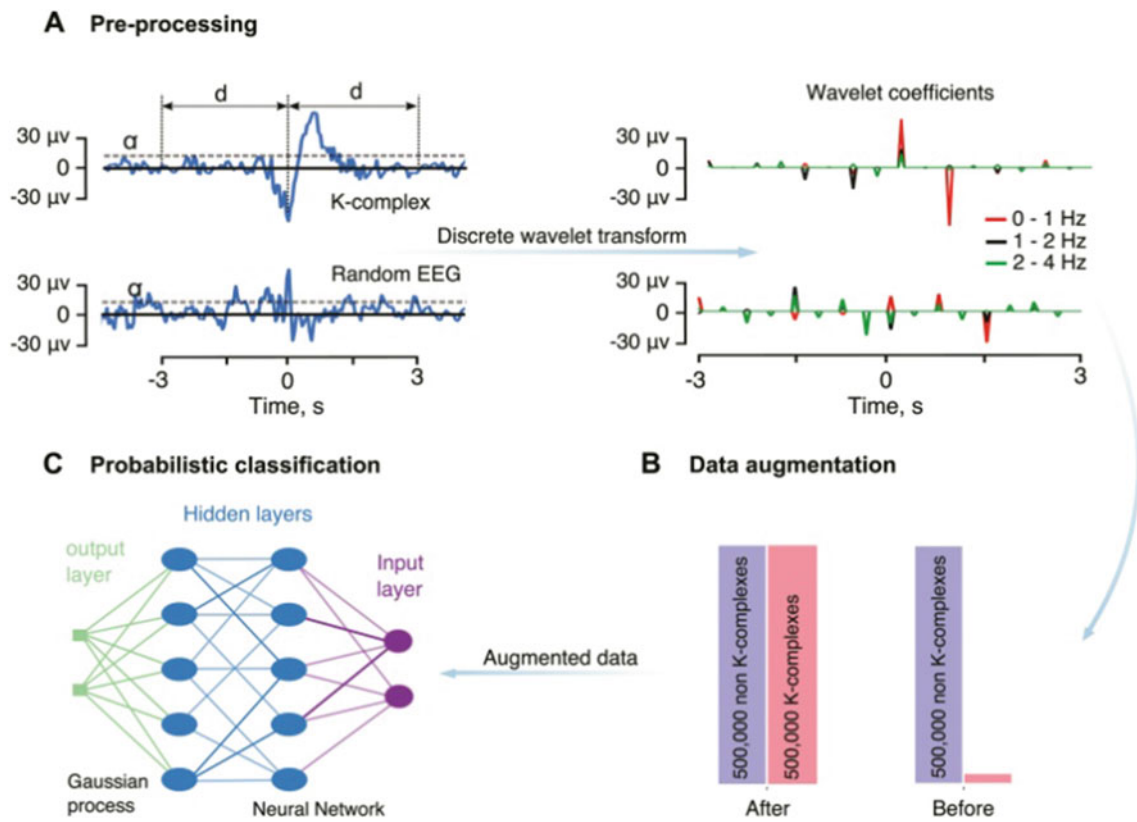


Fig. 4 Image representing the process of detecting KC complexes (Lechat et al., 2020): a DWT-based preprocessing, synthetic balancing of dataset and use of probabilistic classifier

manifests itself in primarily three forms: OSA, central apnea and a combination of both. OSA is caused by collapsing of soft tissue inside the throat that restricts the upper airway.

Snoring is caused due to the vibration of tissues in the backside of the throat while inhaling and exhaling air. Heavy and consistent snoring may be an indication of sleep

breathing disorder. The treatment approaches are continuous positive airway pressure (CPAP) therapy, oral appliance therapy and surgery. Screening for OSA based on the heart rate variability (HRV) has been a compelling research field since the heart rate significantly varies due to respiratory interruptions by virtue of the sympathetic control system of the human body. Along with HRV, combination of several signals such as SpO₂, photoplethysmography (PPG), respiratory effort signals, snoring sounds and body mass index (BMI) has been used as inputs to identify sleep apnea syndrome. A survey of literature on intelligent detection of sleep apnea leads to the application of ML techniques such as ANN, KNN, SVM, J48 decision tree, ensemble classifiers, CART, gradient boost method (GBM) and eXtreme gradient boosting (XGBoost) (Álvarez et al., 2020; André et al., 2019; Banu Rekha & Kandaswamy, 2015; Banu Rekha et al., 2018). The features of HRV such as time domain, frequency domain and statistical features are extracted and are provided as inputs to these algorithms. Emergent of deep learning algorithms reduces the efforts of manually extracting features and instead identifies features unknown hitherto. Table 3 presents the emerging trends of DL applications in the detection of sleep apnea.

Convolutional neural network (CNN) is being widely used for the extraction of features from one-dimensional heart rate variability signals (Chang et al., 2020; Dey et al., 2017; Muhammed et al., 2017; Sinam & Swanirbhar, 2019). The episodes of OSA and its relation with plethysmography signals were explored using ML techniques (Wang et al., 2019). A multi-layer feed forward network (MLFFNN) outperformed KNN, RBFNN, PNN and ensemble classifiers by achieving an overall accuracy of 97.07% in detection of OSA episodes.

4.1.4 Non-Contact and Unobtrusive Methods of Sleep Assessment in the Elderly

The polysomnography and other home-based sleep test equipments exhibit superior performance in the diagnosis of sleep disorders. These equipments rely on the connection of electrodes to the surface of the body in order to continuously gather sleep data. Hence, a need exists for developing non-contact and unobtrusive methods for screening of sleep or extended clinical monitoring. The primary requirements from a non-contact or unobtrusive sleep assessment system for the elderly are:

- Detection of sleep posture
- Detection of sleep/wake cycle
- Duration of sleep
- Distinguish between REM/NREM stages
- Limb movement disorders
- Respiratory and apneic events.

In sleep studies, wrist actigraphy has been widely acknowledged as a reliable technique of measuring circadian rhythms and sleep cycles. Actigraphy utilizes tiny motion sensors to measure the movement of a part of body. Since sleep stages are known to have an effect on the limb movements, sleep patterns could be effectively identified (Desta et al., 2020). The signals are typically sampled at 10 Hz and are processed in such a way to get the intensity level of movement, the duration and the frequency of movements. It is estimated that a minimum length of 1 min is required to accept actigraphy recording as informative. Standardized algorithms such as Cole–Kripke algorithm and University of California San Diego (UCSD) algorithm are applied to the data to differentiate sleep–wake stages.

Table 3 Emerging use of deep learning methods in the assessment of sleep apnea

Machine Learning/Deep learning method employed	Parameters measured	Data used in the analysis	Performance of the ML algorithms
Convolutional neural network (CNN) (Dey et al., 2017)	RR interval from ECG signal	PhysioNet ECG-Apnea Database	Acc: 98.91% Spec:97.82% Sen:99.20%
1D CNN (Muhammed et al., 2017)	Automated feature extraction from ECG signal	PhysioNet ECG-Apnea Database	Acc: 97.1% Spec: 100% Sen 95.7%
CNN using a pre-trained (AlexNet) model (Chang et al., 2020)	2D scalogram images of single-lead ECG	PhysioNet ECG-Apnea Database	86.22% with 90%
Modified LeNet-5 CNN (Sinam & Swanirbhar, 2019)	Automatic feature extraction from ECG signal	PhysioNet ECG-Apnea Database UCD dataset	Acc: 87.6% Spec: 90.3% Sen: 83.1% AUC: 0.95
MLFFNN (Wang et al., 2019)	Pulse plethysmography signal	PhysioNet ECG-Apnea Database	Acc: 97.07%

A study was carried out on older adults suffering from insomnia to assess the sensitivity of actigraph as against sleep EEG recording during insomnia treatment phase. The older adults were able to assess their sleep subjectively by themselves using the simple actigraph, and this leads to betterment of treatment outcomes (Brooks et al., 1993). In a study to differentiate perceived fatigue and natural fatigue in the backdrop of poor sleep, actigraphy measures for a period of 7 days were considered from Baltimore Longitudinal Study of Aging (BLSA) (Alfonso et al., 2020). The sleep measures derived from the actigraphy were: 24 h total sleep time, total sleep time at night, wake after sleep onset, average length of wake bouts, nap time and sleep efficiency. Multivariable logistic regression models were employed to study the effect of poor sleep, and it was found that while poor sleep might have an influence the perception of fatigue, it may not get exhibited in a conventional way of tiredness and less energy levels. In order to validate the effectiveness of actigraphy used in sleep studies of elderly people, a comparison was performed between single-channel EEG, actigraphy and the self-reported sleep diary of nearly 300 participants (Chou et al., 2020). The factors such as total sleep time, delay in sleep onset, episodes of wake after onset of sleep and sleep efficiency were considered. Intraclass correlation coefficients (ICC) were used to observe the similarity of the signal groups. The ICC is a descriptive statistical measure and is described through Eqs. (1) and (2).

Considering that an estimator N_{jk} of a class of variables is modeled as a random effect model given by:

$$N_{jk} = \mu + \omega_k + \theta_{jk} \quad (1)$$

where N_{jk} is the j th variable in the k th group, and ω_k represents the randomness observed by all the variables in group k . θ_{jk} is a unobserved noise component. The variance of ω_k is represented by the term, σ_ω . The variance of θ_{jk} is represented by the term, σ_θ . The ICC is then defined as:

$$\text{Intra Class Correlation(ICC)} = \frac{\sigma_\omega^2}{\sigma_\omega^2 + \sigma_\theta^2} \quad (2)$$

It was found that there was a high agreement between the EEG and actigraphy signals of normal elderly people, but there was a decline in the agreement on elderly people with mild cognitive impairments and those with risk of Alzheimer's disease.

A typical unobtrusive setup that has potential for sleep monitoring in the elderly is shown in Fig. 5. The mattress or inflatable bed is fixed with sensor array for monitoring body movements and acoustic sensor for recording snoring sounds and breathing sounds. A bedside unit is used to record the signals and also send to the server or cloud for remote monitoring. On the bedside unit, algorithms are implemented to extract features such as frequency domain, time domain, statistical and nonlinear features. Machine learning and deep

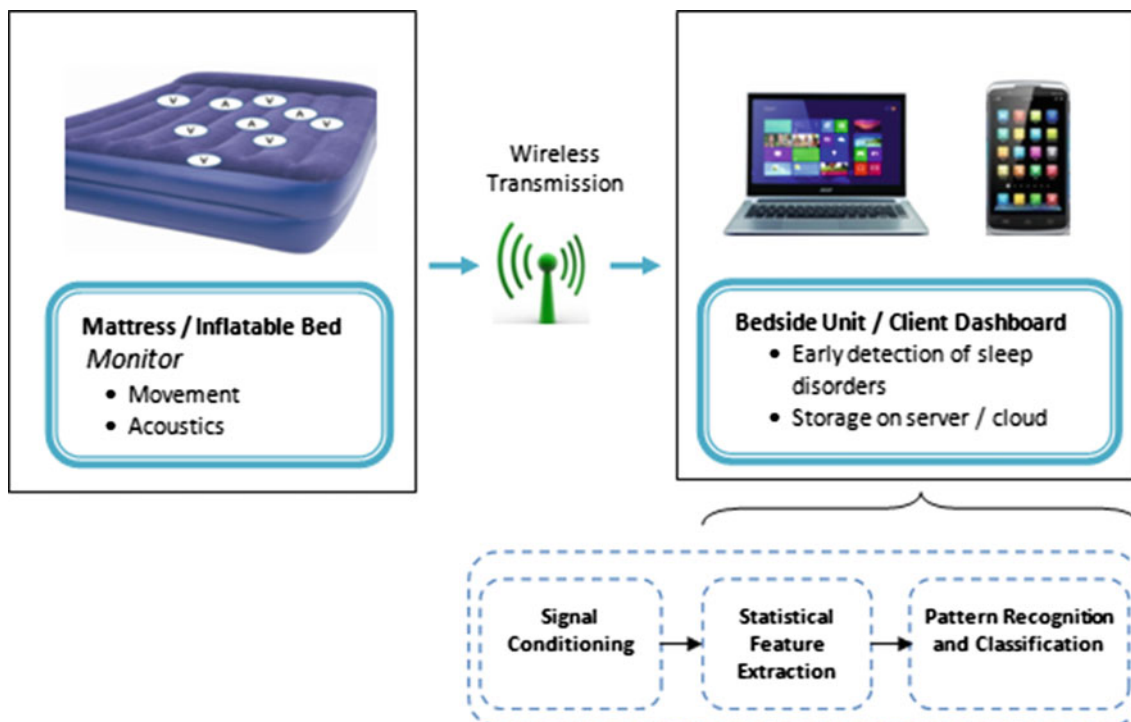


Fig. 5 Proposed setup for unobtrusive sleep monitoring in the elderly

learning algorithms would be employed to detect the corresponding sleep patterns and disturbances.

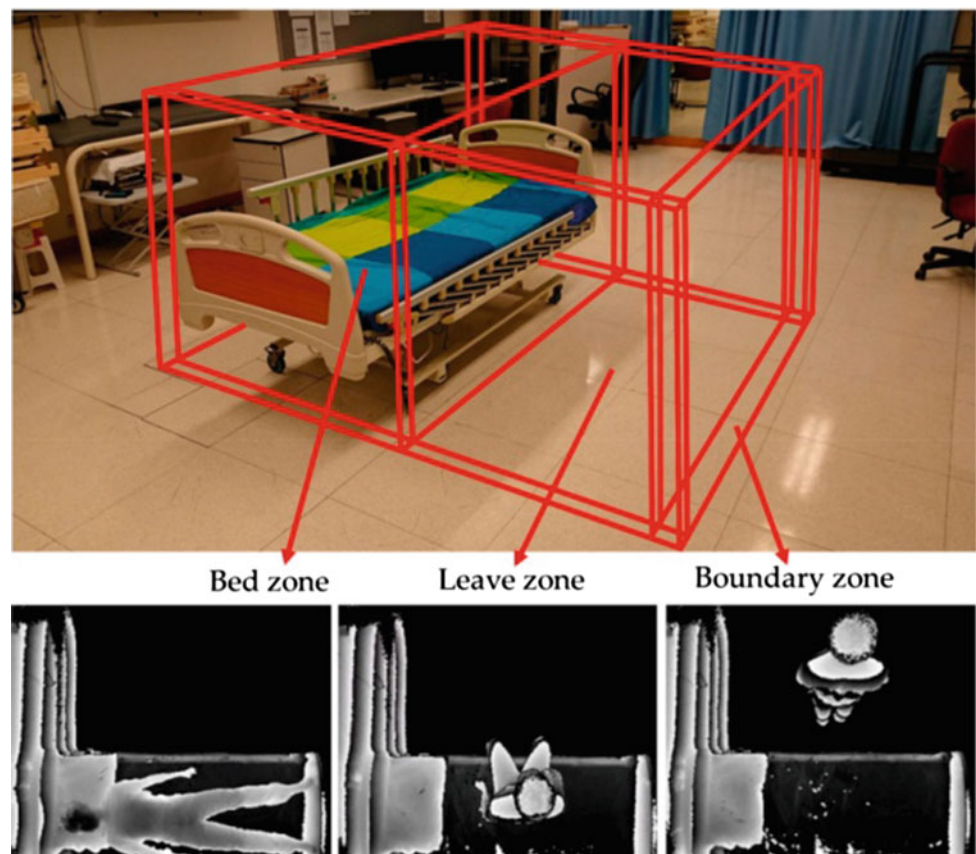
According to Canada's Technology and Aging Network AGE-WELL, a mattress with a network of hundreds of sensors beneath has a huge potential of prevention of health hazards in the elderly, particularly those associated with bed sores and pressure ulcers (<https://hospitalnews.com>). Bed sores and pressure ulcers occur due to the pressure exerted by the bony parts of the body on the skin. These conditions occur due to prolonged lying in a particular position, sitting on a wheelchair or even wearing some casts on the body. The pressure sensors have been found to be effective in analyzing the sleep-wake patterns of the elderly. Along with tiny motion and vibration sensors, the movements and respiratory patterns during sleep could be effectively mapped out (Waltisberg et al., 2014). The development of a smart bed with embedded sensors was able to classify behavioral sleep patterns such as wakefulness, REM, NREM and no bed occupancy (Laurino et al., 2020). The sleep data was segmented into 30 s epochs, and the extracted features were used to train a decision tree with bootstrap aggregation. The activity data was trained on an ANN. The work achieved detection of sleep position, motion, respiratory and heart activity. Ultra wide band (UWB) radars are increasingly finding applications in health care due to its high-resolution

imaging capability and low signal-to-noise ratio. In a particular application for sleep research, UWB tags along with a matrix of embedded pressure sensors were used to assess the sleep patterns (Ni et al., 2015). This application was targeted at the elder care giving community who could benefit from the power of machine learning algorithms. Rhythmic sleep disturbances also lead to wandering at night and increase the risk of fall and injury. eNightlog (Cheung et al., 2021) is a remote monitoring system based on two sensors: infra-red 3D time-of-flight sensor and a impulse-radio ultra-wideband sensor. The former was used to capture the physical movements, while the latter was used to measure the respiration rate and sleep quality. Figure 6 illustrates the division of images into various zones for monitoring movements at night. Decision tree algorithm was used to estimate the entry and exit of zones.

5 Conclusions and Future Scope

This chapter presented a survey on the applications of ML algorithms on generic sleep health and hygiene with an emphasis on the elderly community. Sleep hygiene is essential for the healthy living of humans and is critical for the mental and physical health of elders. The etiology and

Fig. 6 Process of monitoring in the eNightlog system (Cheung et al., 2021)



the types of sleep disorders were discussed at the beginning of the chapter. The changes to the sleep architecture by virtue of aging have been highlighted in this discussion.

The primary methods of sleep assessment such as subjective and objective have been dealt with in this chapter. The subjective assessment is largely based on collecting responses through questionnaires and sleep diaries, and its nature is self-reported. While normal elderly people can self-report their perspectives of sleep, those with mild or severe cognitive impairments may not be able to express themselves. Hence, objective methods of assessment are widely used. Polysomnogram being the gold standard and consisting of a complex array of signals, machine learning techniques are finding increased applications in differentiating the sleep stages as wake/sleep and REM/NREM. While the large extent of computation involved in processing polysomnographic data could be reduced through deep learning methods, it is imperative that simpler methods with less number of sensors connected at sleep time could reduce computational complexity and also develop sleep monitoring devices for clinical and at-home use. In lieu of EEG signals, biological signals such as heart rate variability (derived from ECG), respiration rate, oxygen saturation, EOG and EMG have been considered to differentiate sleep stages with a reasonable amount of success. In studies on diagnosing obstructive sleep apnea, heart rate variability and respiration rate have been found to be the primary parameters followed by oxygen saturation. The typical features considered are the time domain, frequency domain and statistical features. Segmentation of the sleep signal is essential before the analysis is performed. Typical epochs are 30 s or 1 min long. Preprocessing of the signal is performed to eliminate the noises and to identify and remove portions that are affected due to movement or sensor failure. While traditional machine learning algorithms such as KNN, SVM, ANN, decision tree and regression models require the features to be extracted priori, emerging deep learning algorithms such as CNN, PNN, MLFNN are being used by researchers to automatically extract features of interest. But commercial sleep monitoring equipments using DL approaches are yet to be popularized, but it may happen in the near future. Apart from these, wearable electronics are finding applications in the field of sleep study to unobtrusively monitor sleep. Large number of sleep studies has been carried out on well known and publicly available databases such as Physionet, University College Dublin Sleep Apnea Database (UCD), National Sleep Research Resource, Sleep Research Society and Montreal Archive of Sleep Studies (MASS). It is an interesting trend to observe that upcoming researches are being conducted on patients presenting themselves with sleep problems and in cohort studies such as Baltimore Longitudinal Study of Aging (BLSA). This points to the rise of sleep and sleep-induced disorder and at the same time the

elevated levels of awareness of population to seek assistance. Continuous education programs are required to sensitize the elderly population and their caregivers about these monitoring devices that can monitor their sleep habits and thereby aid in early detection of health issues due to insufficient sleep.

One of the challenges faced in the field of sleep study is the unavailability of large number of standardized datasets. This can be attributed to the various types of sleep study modes such as clinical studies, at home tests and wearable sleep tracking devices used for sleep monitoring. With the widespread usage of smart phones as tracking devices, it has become necessary to standardize the measurements in order to ensure reliability and interoperability. IEEE open mHealth is an effort to standardize data interoperability for patient generated health data (<https://www.openmhealth.org/>). In the ongoing efforts of the community-driven approach, schemas are drafted for describing sleep metrics such as total sleep time, sleep duration and sleep episode along with many other health measurements.

In view of sleep being multi-modal in nature and its longer duration of occurrence, data-driven technologies are revolutionizing the field of sleep medicine. In future, applications of wearable electronics with flexible designs would enhance the application of sleep monitoring. While pressure mats are being developed and even commercialized, they have to be used in large scale through tie-ups with hospitals and elderly homes. The relation between sleep and biological markers is being explored, but they are yet to be standardized to be able to predict or diagnose a sleep disorder. With the onus set on preventive health care, monitoring physical activities, circadian rhythm and the sleep quality go a long way in the early identification of sleep disorders and could lead to timely interventions for better sleep outcomes. Across the globe, several countries are facing an uprise in the elder population coupled with an increased lifespan. Sleep disorders, if unnoticed, would create financial and emotional burden to a country. Hence, this chapter emphasizes that a smart city with smart elders should have technically enriched solutions based on ML techniques for their well-being and longevity.

References

- Alfani, A. J., Schrack, J. A., Urbanek, J. K., Wanigatunga, A. A., Wanigatunga, S. K., Zipunnikov, V., Ferrucci, L., Simonsick, E. M., & Spira, A. P. (2020). Associations of actigraphic sleep parameters with fatigability in older adults. *The Journals of Gerontology: Series A*, 75(9), e95–e102. <https://doi.org/10.1093/geron/glaa137>
- Allocca, G., Ma, S., Martelli, D., Cerri, M., Del Vecchio, F., Bastianini, S., Zoccoli, G., Amici, R., Morairty, S. R., Aulsebrook, A. E., & Blackburn, S. (2019). Validation of ‘Somnivore’, a machine

- learning algorithm for automated scoring and analysis of polysomnography data. *Frontiers in Neuroscience*, 13, 207. <https://doi.org/10.3389/fnins.2019.00207>
- Álvarez, D., Cerezo-Hernández, A., Crespo, A., Gutiérrez-Tobal, G. C., Vaquerizo-Villar, F., Barroso-García, V., Moreno, F., Arroyo, C. A., Ruiz, T., Hornero, R., & Del Campo, F. (2020). A machine learning-based test for adult sleep apnoea screening at home using oximetry and airflow. *Scientific Reports*, 10, 5332. <https://doi.org/10.1038/s41598-020-62223-4>
- André, P., Nuno, P., Bruno, M. C., Silva, K. B., & Nuno, G. (2019). Towards an accurate sleep apnea detection based on ECG signal: The quintessential of a wise feature selection. *Applied Soft Computing*, 83, 105568.
- Arand, D. L., & Bonnet, M. H. (2019). The multiple sleep latency test. *Handbook of Clinical Neurology*, 160, 393–403.
- Banu Rekha, B., Kandaswamy, A., Ramanathan, R. M. P. L. (2018). Ensemble classification approach for screening of obstructive sleep apnoea using ECG. *International Journal of Biomedical Engineering and Technology*, 27(1/2), 139–150.
- Banu Rekha, B., & Kandaswamy, A. (2015). A complexity reduction approach for screening of obstructive sleep apnea from single lead ECG. *Journal of Medical Imaging and Health Informatics*, 5(8), 1668–1673.
- Benes, H., & Kohnen, R. (2009). Validation of an algorithm for the diagnosis of restless legs syndrome: The restless legs syndrome-diagnostic index (RLS-DI). *Sleep Medicine*, 10(5), 515–523.
- Brooks, J. O., Friedman, L., Bliwise, D. L., & Yesavage, J. A. (1993). Use of the wrist actigraph to study insomnia in older adults. *Sleep*, 16(2), 151–155.
- Bysses, D. J., Reynolds, C. F., Monk, T. H., Berman, S. R., & Kupfer, D. J. (1989). The pittsburgh sleep quality index: A new instrument for psychiatric practice and research. *Psychiatry Research*, 28(1), 193–213.
- Chang, H. Y., Yeh, C. Y., Lee, C. T., & Lin, C. C. (2020). A sleep apnea detection system based on a one-dimensional deep convolution neural network model using single-lead electrocardiogram. *Sensors (basel, Switzerland)*, 20(15), 4157. <https://doi.org/10.3390/s20154157>
- Chattu, V. K., Manzar, M. D., Kumary, S., Burman, D., Spence, D. W., & Pandi-Perumal, S. R. (2018). The global problem of insufficient sleep and its serious public health implications. *Healthcare*, 7(1), 1.
- Cheung, J. C., Tam, E. W., Mak, A. H., Chan, T. T., Lai, W. P., & Zheng, Y. P. (2021). Night-time monitoring system (eNightLog) for elderly wandering behavior. *Sensors (basel, Switzerland)*, 21(3), 704. <https://doi.org/10.3390/s21030704>
- Chou, C. A., Toedebusch, C. D., Redrick, T., Freund, D., McLeland, J. S., Morris, J. C., Holtzman, D. M., & Lucey, B. P. (2020). Comparison of single-channel EEG, actigraphy, and sleep diary in cognitively normal and mildly impaired older adults. *Sleep Advances* 24;1(1), zpaa006. <https://doi.org/10.1093/sleepadvances/zpaa006>. PMID:33644758; PMCID:PMC7898727.
- Chung, F., Yegneswaran, B., Liao, P., Chung, S. A., Vairavanathan, S., Islam, S., Khajehdehi, A., & Shapiro, C. M. (2008). STOP questionnaire: A tool to screen patients for obstructive sleep apnea. *Anesthesiology*, 108(5), 812–821.
- Colten, H. R., & Altevogt, B. M. (Eds.). (2006). *Sleep disorders and sleep deprivation: An unmet public health problem*. National Academies Press.
- Cybulski, M., Cybulski, L., Krajewska-Kulak, E., et al. (2019). Sleep disorders among educationally active elderly people in Białystok, Poland: A cross-sectional study. *BMC Geriatrics*, 19, 225.
- Desta, F., Michael, E. A., Mingming, S., John, M. V., Sarah, K., & Kim, E. I. (2020). Actigraphy-based assessment of sleep parameters. *Annals of Work Exposures and Health*, 64(4), 350–367.
- Dey, D., Chaudhuri, S., & Munshi, S. (2017). Obstructive sleep apnoea detection using convolutional neural network based deep learning framework. *Biomedical Engineering Letters*, 8(1), 95–100. <https://doi.org/10.1007/s13534-017-0055-y>
- Dinh-Van, P., Nan-Ping, Y., Ching-Yen, K., & Chien-Lung, C. (2020). Deep learning approaches for sleep disorder prediction in an asthma cohort. *Journal of Asthma*. <https://doi.org/10.1080/02770903.2020.1742352>
- Dzierzewski, J. M., Dautovich, N., & Ravyts, S. (2018). Sleep and cognition in older adults. *Sleep Medicine Clinics*, 13(1), 93–106.
- Etsuji, S., Takashi, Y., Kazumune, U., Soshi, T., Masumi, S., Toshiki, O., Kazuko, I., & Hiroyuki, D. (2009). Sleep duration, sleep quality and cardiovascular disease mortality among the elderly: A population-based cohort study. *Preventive Medicine*, 49(2), 135–141.
- Fiorillo, L., Puiatti, A., Papandrea, M., Ratti, P. L., Favaro, P., Roth, C., Bargiotas, P., Bassetti, C. L., & Faraci, F. D. (2019). Automated sleep scoring: A review of the latest approaches. *Sleep Medicine Reviews*, 48, 101204.
- FitzGerald, D., Keane, R. A., Reid, A., & O'Neill, D. (2013). Ageing, cognitive disorders and professional practice. *Age and Ageing*, 42(5), 608–614.
- Hirshkowitz, M., Whiton, K., Albert, S. M., Alessi, C., Bruni, O., DonCarlos, L., Hazen, N., Herman, J., Katz, E. S., Kheirandish-Gozal, L., & Neubauer, D. N. (2015). National sleep foundation's sleep time duration recommendations: Methodology and results summary. *Sleep Health*, 1(1), 40–43.
- Hoddes, E., Dement, W., & Zarcone, V. (1972). The development and use of the Stanford sleepiness scale (SSS). *Psychophysiology*, 9, 150.
- Hospital News. <https://hospitalnews.com>
- Indrajeet, S. G., Sankha, S. C., Amit, R. S., & Dharam, P. S. (2014). Insomnia in the elderly—A hospital-based study from North India. *Journal of Clinical Gerontology and Geriatrics*, 5(4), 117–121.
- Izci, B., Firat, H., Ardiç, S., Köktürk, O., Gelir, E., & Altinörs, M. (2004). Adaptation of functional outcomes of sleep questionnaire (FOSQ) to Turkish population. *Tüberküloz Ve Toraks*, 52(3), 224–230.
- Jacobson, R. P., Kang, D., & Houck, J. (2020). Can patient-reported outcomes measurement information system (PROMIS) measures accurately enhance understanding of acceptable symptoms and functioning in primary care? *Journal of Patient-Reported Outcomes*, 4, 39.
- Jafari, B., & Mohsenin, V. (2010). Polysomnography. *Clinics in Chest Medicine*, 31(2), 287–297.
- Johns, M. W. (1991). A new method for measuring daytime sleepiness: The Epworth sleepiness scale. *Sleep*, 14(6), 540–545.
- Kosuke, K., Masaya, T., Torbjörn, Å., Akinori, N., Yasumasa, O., Takashi, H., & Kenji, F. (2006). Validation of the Karolinska sleepiness scale against performance and EEG variables. *Clinical Neurophysiology*, 117(7), 1574–1581.
- Krauss, P., Metzner, C., Joshi, N., Schulze, H., Traxdorf, M., Maier, A., & Schilling, A. (2021). Analysis and visualization of sleep stages based on deep neural networks. *Neurobiology of Sleep and Circadian Rhythms*, 10, 100064. <https://doi.org/10.1016/j.nbscr.2021.100064>
- Laurino, M., Arcarisi, L., Carbonaro, N., Gemignani, A., Menicucci, D., & Tognetti, A. (2020). A smart bed for non-obtrusive sleep analysis in real world context. *IEEE Access*, 8, 45664–45673.
- Lechat, B., Hansen, K., Catchside, P., & Zajamsek, B. (2020). Beyond K-complex binary scoring during sleep: Probabilistic classification using deep learning. *Sleep*, 43(10), zsa077. <https://doi.org/10.1093/sleep/zsa077>
- MacLean, A. W., Fekken, G. C., Saskin, P., & Knowles, J. B. (1992). Psychometric evaluation of the stanford sleepiness scale. *Journal of Sleep Research*, 1(1), 35–39.

- Max, H. & Amir, S. (2017). Evaluating sleepiness. In K. Meir, R. Thomas & C. D. William (Eds.), *Principles and practice of sleep medicine* (pp. 1651–1658).
- McMillan, A., & Morrell, M. J. (2016). Sleep disordered breathing at the extremes of age: The elderly. *Breathe (sheffield, England)*, 12(1), 50–60.
- Meredith, L. W., Daniel, J. B., Susan, R., Katie, L. S., Kristine, E., Yue, L., Sonia, A., & Martica, H. H. (2019). Multidimensional sleep and mortality in older adults: A machine-learning comparison with other risk factors. *The Journals of Gerontology: Series A*, 74(12), 1903–1909.
- Morin, C. M., Belleville, G., Bélanger, L., & Ivers, H. (2011). The insomnia severity index: Psychometric indicators to detect insomnia cases and evaluate treatment response. *Sleep*, 34(5), 601–608.
- Muhammed, K. U., Mehmet, R. B., Cahit, B., & Kemal, P. (2017). Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques. *Neural Computing and Applications*, 28(10), 2931–2945.
- Ni, H., Wu, S., Abdulrazak, B., Zhang, D., Ma, X., & Zhou, X. (2015). Non-intrusive sleep pattern recognition with ubiquitous sensing in elderly assistive environment. *Frontiers of Computer Science*, 9, 966–979.
- Open mHealth. <https://www.openmhealth.org/>
- Perslev, M., Darkner, S., Kempfner, L., Nikolic, M., Jennum, P. J., & Igel, C. (2021). U-Sleep: Resilient high-frequency sleep staging. *NPJ Digital Medicine*, 4(1), 72. <https://doi.org/10.1038/s41746-021-00440-5>
- Phan, H., Andreotti, F., Cooray, N., Chen, O. Y., & De Vos, M. (2019). Joint classification and prediction CNN framework for automatic sleep stage classification. *IEEE Transactions on Bio-Medical Engineering*, 66(5), 1285–1296. <https://doi.org/10.1109/TBME.2018.2872652>
- Philips. <https://www.philips.co.in>
- Schenck, C. H., Boeve, B. F., & Mahowald, M. W. (2013). Delayed emergence of a parkinsonian disorder or dementia in 81% of older men initially diagnosed with idiopathic rapid eye movement sleep behavior disorder: A 16-year update on a previously reported series. *Sleep Medicine*, 14(8), 744–748.
- Shi, L., Chen, S. J., Ma, M. Y., Bao, Y. P., Han, Y., Wang, Y. M., Shi, J., Vitiello, M. V., & Lu, L. (2018). Sleep disturbances increase the risk of dementia: A systematic review and meta-analysis. *Sleep Medicine Reviews*, 40, 4–16.
- Shinno, H. (2010). Sleep disturbances in the elderly: Pathology, symptoms and treatment. *Seishin Shinkeigaku Zasshi*, 112(8), 709–719.
- Sinam, A. S., & Swanirbhar, M. (2019). A novel approach OSA detection using single-lead ECG scalogram based on deep neural network. *Journal of Mechanics in Medicine and Biology*, 19(4). <https://doi.org/10.1142/S021951941950026X>
- Sleep Foundation. <https://www.sleepfoundation.org>
- Stranges, S., Tigbe, W., Gómez-Olivé, F. X., Thorogood, M., Kandala, N. B. Sleep problems: An emerging global epidemic? Findings from the INDEPTH WHO-SAGE study among more than 40,000 older adults from 8 countries across Africa and Asia. *Sleep*, 35(8), 1173–1181.
- Sun, H., Jia, J., Goparaju, B., Huang, G. B., Sourina, O., Bianchi, M. T., & Westover, M. B. (2017). Large-scale automated sleep staging. *Sleep*, 40(10), zsx139. <https://doi.org/10.1093/sleep/zsx139>
- Suzuki, K., Miyamoto, M., & Hirata, K. (2017). Sleep disorders in the elderly: Diagnosis and management. *Journal of General and Family Medicine*, 18(2), 61–71.
- Tan, A., Yin, J. D., Tan, L. W., van Dam, R. M., Cheung, Y. Y., & Lee, C. H. (2017). Using the Berlin questionnaire to predict obstructive sleep apnea in the general population. *Journal of Clinical Sleep Medicine*, 13(3), 427–432.
- Wallace, M. L., Stone, K., Smagula, S. F., Hall, M. H., Simsek, B., Kado, D. M., Redline, S., Vo, T. N., & Buysse, D. J. (2018). Osteoporotic fractures in men (MrOS) study research group: Which sleep health characteristics predict all-cause mortality in older men? An application of flexible multivariable approaches. *Sleep*, 41(1), zsx189.
- Waltisberg, D., Arnrich, B., & Tröster, G. (2014). Sleep quality monitoring with the smart bed. In A. Holzinger, M. Ziefle, & C. Röcker (Eds.), *Pervasive health. Human-Computer interaction series*. Springer.
- Wang, T., Lu, C., Shen, G., & Hong, F. (2019). Sleep apnea detection from a single-lead ECG signal with automatic feature-extraction through a modified LeNet-5 convolutional neural network. *PeerJ*, 7, e7731. <https://doi.org/10.7717/peerj.7731>
- World Bank Open Data. <https://data.worldbank.org>
- World Economic Forum. <https://www.weforum.org>



Smart City Traffic Patterns Prediction Using Machine Learning

David Opeoluwa Oyewola, Emmanuel Gbenga Dada,
and Muhammed Besiru Jibrin

Abstract

Traffic affects every citizen's life in many ways by how long it takes for him or her to travel from home to office, the air condition he or she inhales, the strain generated by traffic jams, sleep, and workouts induced by time spent in traffic. Since motorists cannot see the entire traffic system, the urban traffic system must be anticipated in order to sensitize residents about their mobility choices and the subsequent impact on the environment, as well as to implement smart transport system. The paper used five machine learning models: Bagging (BAG), K-Nearest Neighbors (KNN), Multivariate Adaptive Regression Spline (MARS), Bayesian Generalized Linear Model (BGLM), and Generalized Linear Model (GLM) to predict traffic pattern in a smart city. The dataset consists of 48,120 rows and 4 columns from which the weekday, year, month, date, and time were extracted. Analysis results show that increase in the number of junctions of the city can alleviate problem being faced on the road by commuters. The Root Mean Square Error (RMSE) of BAG, KNN, MARS, BGLM, GLM are 13.09, 9.23, 23.34, 8.7, and 8.6 respectively. Experimental results demonstrated that GLM attained minimal prediction error compared to other machine learning models such as BAG, KNN, MARS, and BGLM used in this study.

Keywords

Vehicular traffic prediction • Smart cities • Bagging (BAG) • Multivariate adaptive regression spline (MARS) • Bayesian generalized linear model (BGLM) • Generalized linear model (GLM) models

1 Introduction

Among the most difficult problems affecting cities is designing and developing an effective transportation system. As the population of the city grows, so do the city's governmental and non-governmental transport networks, and even minor faults and accidents can have a negative impact on a system that is already stretched to its limits. Traffic jam and delays on transportation systems can cost a city losses in terms economy and have a negative impact on inhabitants' standard of living as they spend a lot of time traveling from one location to another. Traffic gridlock has a number of negative consequences, including lost time, environmental degradation, and safety hazards, as well as a negative influence on economic growth and a deterioration of individuals' relationship with their municipal authorities. It is required that cities address fundamental concerns such as the amount of time drivers spend looking for parking spaces, enhancing highway safety, guaranteeing that public transportation is properly routed, and allowing citizens to travel by many modes of transportation.

The idea of smart cities became a reality thanks to advancement in Information Technology (IT). The use of Advanced Traveler (ATS) and Advanced Traffic Management Systems (ATMS) to effectively managed, control and manage traffic flows is a key element of the smart or intelligent cities of the future. The ATMS/ATS aims to improve the overall performance of the traffic system, for example, to reduce emissions, noise, and travel times. Various types of transport models are routinely employed in the estimation of

D. O. Oyewola · M. B. Jibrin
Department of Mathematics and Computer Science, Federal
University Kashere, Kashere, Nigeria

E. G. Dada (✉)
Department of Mathematical Sciences, University of Maiduguri,
Maiduguri, Nigeria
e-mail: gbengadada@unimaid.edu.ng

traffic status. However, the models cannot cover all aspects of the real system and the models must be supplemented with observed traffic status data, for example, traffic numbers and speed/travel measures, in order to have a proper depiction of reality.

Many Internet of Things (IoT) sensors are put across so many sites in a smart city to gather data on traffic, drainage, passenger movement, and so on, and the revelations derived from these data are used to better manage resources, assets, and the like. Many researchers have employed Machine Learning substantially on data collected by IoT sensors in a smart city. The growth of the Internet of Things (Iwendi et al., 2020), as well as big data analytics (Reddy, Reddy, et al., 2020) and machine learning, has made the concept of smart city a possibility.

The goal of a smart city is to deliver good services to its citizens by using modern technology and data analytics on data collected by sensors (Numan et al., 2020). Technology-driven infrastructure, community development programs, smart transportation systems, the use of technology to minimize crimes and burglary (Kumar et al., 2017; Raghavan et al., 2017; Reddy et al., 2019), giving safety to residents, and other factors all contribute to a city's smartness. Machine learning techniques may be used to extract hidden information from data, comprehend patterns in data, and classify or predict data (Gadekallu et al., 2020; Reddy, Rm, et al., 2020; Vinayakumar et al., 2019). Machine learning reduces variability and bias during the model's training and testing phases. As a result, the proposed system solves the limitations associated with the existing techniques used for prediction of traffic pattern. A recursive cloud data service, Artificial Intelligence (AI), and networking means of transportation, populaces, infrastructure, and logistic partners support the public transport system in a smart city. The autonomous waste pickup's sensors monitor all movements in the area of the environment and proactively respond to any possible threats, ensuring citizen safety, and averting accidents before they occur. AI techniques, data analytics, and communication technologies have been effectively used to connect people, roads, and vehicles in order to solve a variety of congested roads problems (An & Wu, 2020).

According to Bhattacharya et al. (2020), astronomical population increase in cities has resulted in an increasing number of individuals having to stay long hours in traffic. For smart city approaches, transportation has become one of the hottest research areas. As a result, the need to develop effective machine learning models that have the capacity to estimate traffic flow in a city. To address this problem, some scholars have proposed different approaches such as machine learning, computer vision, deep learning, and neural networks techniques.

After carefully reviewing some relevant publications, it was discovered that the existing literature does not provide any work that uses BAG, GLM, BGLM, MARS, and KNN techniques to solve the problem of traffic congestion prediction in smart city. Furthermore, the prediction accuracy of some of the existing work is relatively low while in some cases the authors did not use performance metrics to evaluate the performance of their proposed system. Also, some authors did not compare their work with high performing machine learning or deep learning models. Considering these shortcomings, the novelty of this work centers on the use of machine learning for traffic pattern prediction. The summary of our contribution is as follows:

- (i) This paper proposed five (5) machine learning techniques for traffic pattern prediction in smart city.
- (ii) The Bagging (BAG), K-Nearest Neighbors (KNN), Multivariate Adaptive Regression Spline (MARS), Bayesian Generalized Linear Model (BGLM), and Generalized Linear Model (GLM) models for solving the problem of traffic congestion prediction in smart city was presented in this paper.
- (iii) Minimal Root Mean Square Error (RMSE), Mean Average Square Error (MASE), and Mean Square Error (MASE) for GLM model was attained. The proposed models were evaluated using different performance metrics.
- (iv) A recent review of state-of-the-art proposals for traffic patterns prediction in smart city is presented.

The remainder of this paper is organized as follows: The related works in the field of power consumption prediction in smart cities are presented in Sect. 2. The proposed approach used in this study was described in Sect. 3. Section 4 contains the results presentation and analysis, while Sect. 5 contains the paper's conclusion.

2 Related Works

This section presents a brief discussion on recent researches done in traffic prediction in smart cities. It has been explained in section one that ML algorithms have gained wide acceptance among data scientists and researchers for power traffic pattern prediction. This is because of the efficacy of these algorithms in handling prediction problems. These ML models were selected for use in this paper because they are highly non-linear and have capacity to learn from past data. Ozbayoglu et al., (2016) applied regression tree to predict short-term and long-term traffic flow in work zones.

Wu et al. (2007) employed Support Vector Machine (SVM) and a Markov Random Field (MRF) algorithm to design and execute a system that has the capacity to detect parking spaces. The MRF is an implement, a set of vertices with bidirectional edges, that is used to designate a group of random variables with Markov characteristics. Initially, data from parking lot cameras is acquired, and the input images are preprocessed with the isolation of its most important attributes. The SVM is afterward employed to categories parking spaces as either available or unavailable for parking. Finally, MRF resolves any potential SVM classification inconsistencies.

Ng et al. (2019) applied SVM to detect road surface deformities. Yang et al. (2020) utilized the fusion of k-Means and DBN to enhance the traffic network setup. Nallaperuma et al. (2019) employed Deep Neural Networks (DNN) to decide life-threatening road sections that are responsible for volatile traffic situations in cities. The DNN model depicts the relationship patterns of the affected road area and the neighboring road areas that affect movement of vehicles in the specified road section. Verma and Badade (2019) applied machine learning for traffic prediction. The authors used LoRa and a traffic preplanned algorithm. The shortcoming of the work is that they do not present any results or proofs of the effectiveness of their proposed approach.

Williams et al. (2006) proposed five Machine Learning (ML) algorithms for traffic classification predicated on IP (Internet Protocol). The authors demonstrated the usefulness of work accomplished by computer system as an important measure for traffic categorization. Hinsbergen et al., (2007) applied statistical technique for predicting traffic. The drawback of their work is that the proposed system is not designed to solve the problem of traffic congestion.

The authors in Jia et al. (2016) presented a deep learning model that used local data manually fed into the system. Moreover, the traffic prediction done by the system was carried out using deep belief network. Morales et al. (2017) employed machine learning algorithm to examine, cleanse, transform, and model traffic data with the aim of learning valuable information, and making informed decision. The system is then used for traffic predictions. The downside of the proposed work is that the model wastes much energy in inspecting and processing of data.

Zheng et al. (2006) presented a binary neuronal network algorithm for predicting the vehicular traffic. The system predicts traffic flow over a short period of time, to envisage the manner in which the predicted value is likely to influence decision-making or behaviors. The proposed model needs to utilize a Global Positioning System (GPS) for providing information about locations, road traffic conditions, things, predispositions of road users, and so on. The system saves and subsequently transmits the information to users.

3 Methodology

Smart city traffic patterns dataset is collected from the Kaggle (Gosh, 2018) spans from November 2015 to June 2017 as shown in Table 1. In this study traffic information for that period is taken to examine the system performance. The dataset consists of 48, 120 rows with only four columns such as DateTime, Vehicles, Junction, and ID which are insufficient to obtain an accurate result from the dataset. The fundamental issue of limited data is the fact that variation grows with less data (Hoffmann et al., 2019). In order to have sufficient data for out input, we extracted weekday, year, month, day, and time from the DateTime column as shown in Table 2 which displayed the first 10 rows. Our approach involves the use of machine learning techniques such as: Bagging (BAG), K-Nearest Neighbor (KNN), Multivariate Adaptive Regression Spline (MARS), Bayesian Generalized Linear Model (BGLM), and Generalized Linear Model (GLM). The R programming language version 3.6.0 installation and all the necessary packages such as ggplot2, tseries, xgboost, lattice, caret, C50, kernlab, mlbench, randomforest, caretensemble, mass, klar, nnet, mars, and rocr were used for the simulations and generation of the results and graphs presented in the work. The computer system used for the simulations has Window 8.1 64-bits with 6 GB RAM and Intel Pentium dual CPU processor.

3.1 Machine Learning Algorithms

3.1.1 Bagging (BAG)

In bagging model only one prediction may be made from the decisions of different learners. In the case of classification, voting is simply the sum of those decisions. Individual models are created by a variety of methods of bagging and boosting. Models in bagging are given the same weights, whereas more successful models in boosting are given more weighting since an executive may base alternative outcomes on a variety of expert information based on their prior estimations. Individual decision trees are brought together by requiring them to vote on each test. If one class receives more votes than other classes because there are more votes, projections based on a larger number of votes are more reliable (Oyewola et al., 2019). The following are the definitions of bagging algorithms:

Bagging algorithms are defined as follows:

Step 1: Construct a bootstrap sample $(x_1, y_1), \dots, (x_m, y_m)$ by random drawing m times with replacement from the data $(x_1, y_1), \dots, (x_m, y_m)$.

Step 2: Compute the bootstrapped estimator $b(\cdot)$ by the plug-in principle: $b(\cdot) = e_m((x_1, y_1), \dots, (x_m, y_m))(\cdot)$.

Table 1 Description of the dataset

Terms	Meaning	Data
DateTime	A digital record of occurrence of a particular event	Date and time
Vehicles	Different types of transportation plying the road such as car, lorry, or cart	Number of vehicles at different intervals
Junction	A point where two or more things are joined	1–4 junction
ID	ID of each vehicle DateTime	

Table 2 Comprehensive dataset of smart city traffic pattern

Date	Vehicles	Junction	Weekday	Year	Month	Day	Time
01–11–15	15	1	1	2015	11	1	00
01–11–15	13	1	1	2015	11	1	01
01–11–15	10	1	1	2015	11	1	02
01–11–15	7	1	1	2015	11	1	03
01–11–15	9	1	1	2015	11	1	04
01–11–15	6	1	1	2015	11	1	05
01–11–15	9	1	1	2015	11	1	06
01–11–15	8	1	1	2015	11	1	07
01–11–15	11	1	1	2015	11	1	08

Where the function e_m defines the estimator as a function of the data.

Step 3: Repeat steps 1 and 2 N times, where N is often chosen as 50 or 100, yielding $b(\cdot)(k = 1, \dots, N)$.

The Bagged estimator is

$$b_{bag}(\cdot) = N^{-1} \sum_{k=1}^N b^k(\cdot) \quad (1)$$

3.1.2 K-Nearest Neighbor (KNN)

A lazy approach is the K-Nearest Neighbor (KNN) classification, who merely keeps trainings, because there is no obvious process of training. It learns via analogy, meaning that a given test tuple is compared with similar training tuples. These tuples should be the closest to the unidentified tuple. A Euclidean distance measures the closest neighbor and the unknown tuple is chosen from its closest neighbors as the most common class. The rate of k can be experimentally determined (Adeniyi et al., 2016). The mathematics equation of KNN is:

$$R(x) = \{\hat{x} | D(x, \hat{x}) \leq d_k\} \quad (2)$$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{k[y]}{k} \quad (3)$$

$$g(x) = \begin{cases} 1, & k[y = 1] \geq k[y = -1], \\ -1, & k[y = -1] \geq k[y = 1] \end{cases} \quad (4)$$

where $R(x)$ is the predicting region, closest to the training point x , d_k is the k th order of $D(x, \hat{x})$, $D(x, \hat{x})$ is the distance metric, $k[y]$ is the number of samples in region, labelled as y , $p(y|x)$ is the posterior probability of the observation point x , $g(x)$ is the decision that maximizes the posterior probability.

The basic KNN algorithm is:

1. The value of K used in the KNN algorithm in this work is 3. This enables us to achieve desired accuracy.
2. Calculate the distance of the test sample with all the samples of the training dataset.
3. Sort the distance and define the closest neighbors by minimum K -th distance.
4. Assemble the closest neighborhood categories.
5. Make use of simple majority of the closest neighborhood category as a new data item prediction value.

3.1.3 Multivariate Adaptive Regression Spline (MARS)

Multivariate Adaptive Regression Spline (MAR) is a versatile regression method and non-parametric approach that incorporates piecewise linear regression function referred as basic function (bf). In order to estimate performance of

MAR, it uses basic functions (bf) for capturing the hidden non-linear relations between independent input variables (Samadi et al., 2020). Bf is therefore the main component in the generation of a MAR model.

The mathematics equations of MARS:

$$y = c_o + \sum_{m=1}^M c_m bf_m(x) \quad (5)$$

$$bf_m(x) = \max(0, x - t) \quad (6)$$

Or

$$bf_m(x) = \max(0, t - x) \quad (7)$$

where y is the output of the MARS model, c_o is a constant value, c_m is the coefficient, x is the input variable, t is the threshold, and bf_m is the basis function.

3.1.4 Bayesian Generalized Linear Model (BGLM)

The Bayes Generalized Linear Model (BGL) assumes preliminary or prior distribution based on preliminary or prior information and subsequent distribution is achieved through the integration of sample information with such prior information. In general, information collected from the post or posterior distribution is closer to true information, since it brings together sample data and expert views (Shi et al., 2019).

The prior distribution is:

$$\pi(\beta, \theta) \propto \exp\left(-\sum_{i=1}^n n_i \theta_i^{-1} - \sum_{j=1}^p \frac{(\beta_j - u_{\beta_j})^2}{2\sigma_{\beta_j}^2}\right) \prod_{i=1}^n \frac{\theta_i^{-1-m_i}}{\sqrt{\sigma_{\beta_j}^2}} \quad (8)$$

The posterior distribution is:

$$\begin{aligned} \pi(\beta|\theta, Y) \propto L(Y|\beta, \theta)\pi(\beta, \theta) \propto \prod_{i=1}^n \frac{\theta_i^{\exp\sum_{j=1}^p x_{ij}\theta_j}}{\tau(\theta_i \exp(\sum_{j=1}^p x_{ij}\beta_j))} \\ y_i^{\theta_i \exp(\sum_{j=1}^p x_{ij}\beta_j)^{-1}} \exp(-\theta_i y_i) \exp\left(-\sum_{i=1}^n n_i \theta_i^{-1} - \sum_{j=1}^p \frac{(\beta_j - u_{\beta_j})^2}{2\sigma_{\beta_j}^2}\right) \prod_{i=1}^n \frac{\theta_i^{-1-m_i}}{\sqrt{\sigma_{\beta_j}^2}} \end{aligned} \quad (9)$$

where β, θ are independent and $\beta_j (j = 1, 2, \dots, p)$ follows normal distribution $N(u_{\beta_j}, \sigma_{\beta_j}^2)$.

3.1.5 Generalized Linear Model (GLM)

Generalized Linear Model (GLM) is a package that adapts linear and similar models with a penalized maximum probability. The regularization path is calculated on a grid of values for the lambda regulation parameter for the lasso or elastic net penalty (Friedman et al., 2010). The approach is

very quick and can make the input matrix \times sparse. It adapts to model regressions of linear, logistic and transnational, fish and Cox. It can also fit linear regression with multiple responses, linear generalized patterns for individual families and lasso models. GLMNet was developed by Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay and Noah Simon and Junyang Qian (Simon et al., 2011).

The mathematical equation of GLM is:

$$f(y_i|\theta_i) = \exp\{\theta_i y_i - \vartheta(\theta_i) + c(y_i)\} \quad (10)$$

$$\rho_i = x_i' \beta \quad (11)$$

$$\theta_i = g(\rho_i) \quad (12)$$

where $x_i' = (x_{i1}, x_{i2}, \dots, x_{ip})$ is $1 \times p$ vector matrix of covariates X , $\beta = (\beta_1, \beta_2, \dots, \beta_p)'$ is a p vector of regression coefficient, $\vartheta(\cdot), c(\cdot)$ are known function and $g(\cdot)$ is a monotonic differentiation function.

3.2 Performance Evaluation

The accuracy test of the smart city dataset is evaluated using:

A. Mean Absolute Error (MAE)

Consider a set of the actual closing price C_p and the predicted values \widehat{C}_p . MAE are given as follows:

$$\frac{1}{n} \sum_{n=1}^n |C_p - \widehat{C}_p| \quad (13)$$

B. Root Mean Square Error (RMSE)

RMSE is given as:

$$\sqrt{\frac{1}{n} \sum_{n=1}^n (C_p - \widehat{C}_p)^2} \quad (14)$$

C. Mean Square Error (MSE)

MSE is given as

$$\frac{1}{n} \sum_{n=1}^n (C_p - \widehat{C}_p)^2 \quad (15)$$

D. Mean Absolute Scaled Error (MASE)

MASE is given as

$$\frac{1}{n} \sum_{n=1}^n \frac{|C_p - \widehat{C}_p|}{\frac{1}{n-m} \sum_{n=m+1}^n |C_p - \widehat{C}_p|} \quad (16)$$

where m is the seasonal period of the closing price and n is the trading days.

3.3 Proposed Traffic Pattern Prediction System

The parameters settings which consist of values used for the different variables in the different models in this paper are presented in Table 3.

Figure 1 depicts the block diagram of the proposed traffic patten prediction in a smart city.

The proposed Smart city traffic pattern was borne out of the problem associated with urban city. Smart cities use big data for decision-making and a revolution in the Internet of a Thing (IoT) allows such data to be collection and transmission of such data. Transportation is a fundamental element influencing metropolitan areas and a crucial use case for smart cities. Smart transportation technologies, such as GPS Bus, traffic cameras, taxis, and geospatial technology, which is frequently employed in a traffic control center to monitor and coordinate a wide network of sensors, are discovering creative methods to reduce traffic congestion and urban mobility. Smart city traffic pattern incorporates all the big data obtained from GPS Bus, traffic camera, GPS Taxi, and geospatial technology. The traffic pattern system consists of data visualization, data cleaning to remove unwanted and other irrelevant values. However, the predicting modeling is the ware house of machine learning such as KNN, BAG, MARS, BGLM, and GLM used in this research. The accuracy of the machine learning is evaluated in the performance evaluation section.

Table 3 Parameter setting for BAG, KNN, MARS, BGLM, and GLM models

MODEL	RMSE
BAG	nbag = 100, comb = NULL, coob = TRUE
KNN	K = 3
MARS	Penalty = 2, degree = 1, pmethod = "forward", nk = 200
BGLM	Family = Gaussian, prior.mean = 0, prior.scale = NULL, prior.df = 1, maxit = 50
GLM	Family = binomial, optim.method = "Nelder-Mead", emplik.method = "Owen", optim.hessian = FALSE

4 Results and Discussion

In order to accomplish sustained urban growth, Smart City is described as a multifaceted strategy integrating multisector and multinational players, structured into new technologies to address social, economic, and environmental issues (Ruhlandt, 2018). The conditions of road traffic have to be collected in order to manage and control traffic flows. The traffic condition can be characterized on a given section of the road with speed, flow, and density. In this study, traffic management is of paramount importance in a smart city. Computational experiments were conducted with the data obtained from Kaggle.

Figure 2 shows the exponential increases in the number of vehicles between 2015 and 2017. The number of vehicles plying the road is at its peak in 2016. This shows that there will be increase in the traffic congestion during this period.

Figure 3 displays the increase in the number of junctions between 2015 and 2016. There is exponentially increase in the number of junctions every year. By comparing Figs. 2 and 3, we observed that as the number of vehicles increases, government also increases the number of junctions which leads to the reduction of traffic between 2016 and 2017. Congestion in traffic is characterized in transportation by slower speeds, longer travel durations, and an increase in vehicle queuing.

Figure 4 shows the longer trip times experience on the road and vehicular queues from 1 November, 2015 to 31 December 2015. This shows that the vehicular queuing increases with time.

In 2016–2017, the number of vehicles increases to 54,000 against 5700 obtained in 2015 which leads to increase in vehicular movement in (Morning, Afternoon, Evening, and Noon) time but as the number of junctions increases in the city, it resulted into decrease in vehicular queuing in 2017 to 48,000 as also shown in Figs. 5 and 6. The study reveals that increase in the number of junctions on a high way may lead to reduction in vehicular movement on the road.

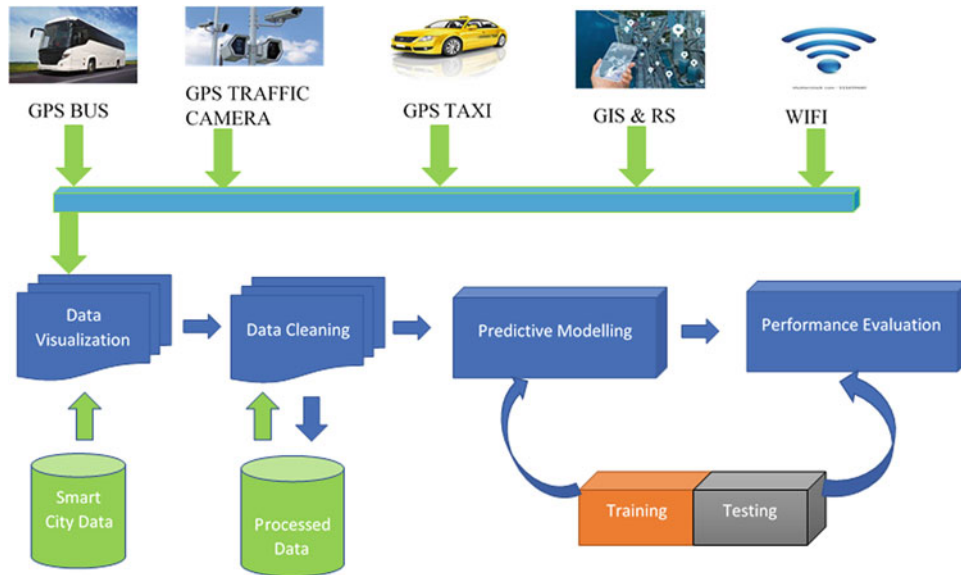


Fig. 1 Block diagram of smart city traffic pattern

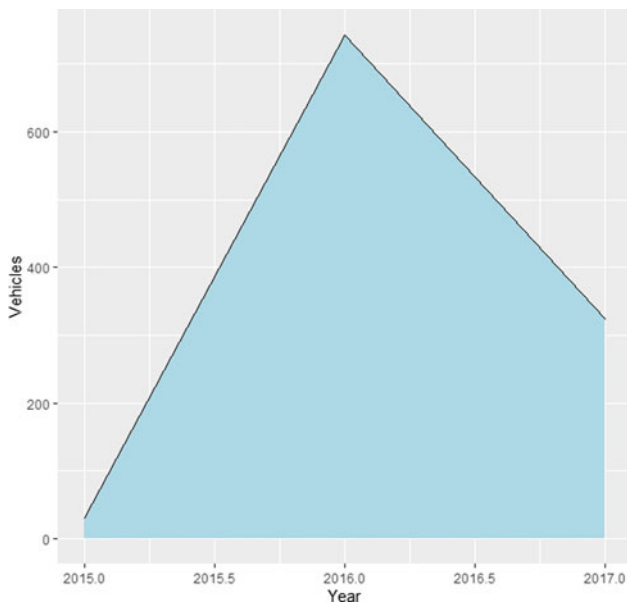


Fig. 2 Number of vehicles versus year

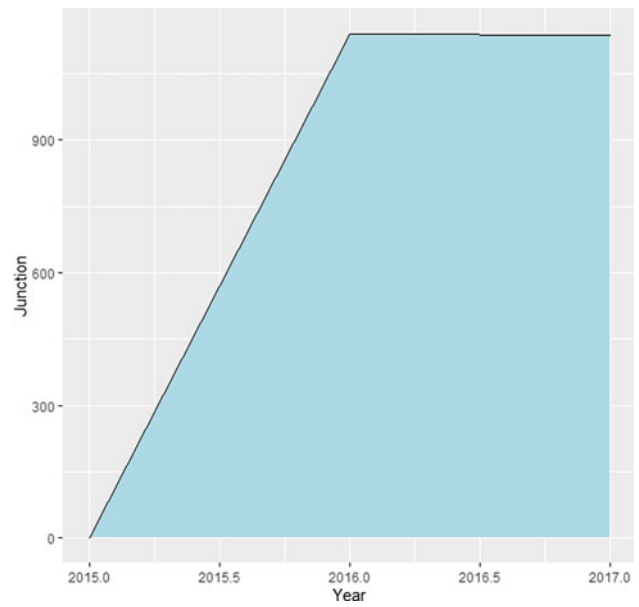


Fig. 3 Number of junctions versus year

Depicted in Fig. 7 are the output results of Actual (Vehicles) that are plying the roads in the city and the predicted values of the number of vehicles in 2017 using BAG, KNN, MARS, BGLM, and GLM models. Figures 8, 9, and 10 are containing the vehicular movements during times of the day such as morning, noon, afternoon, and evening for the year 2015, 2016, and 2017 respectively. There are more vehicular movement in the morning than afternoon, noon and evening for 2015. However, for 2016 and 2017, more vehicular movements are recorded in the afternoon than during any other time of the day.

In this research, we consider five machine learning algorithms to be specific: Bagging (BAG), K-Nearest Neighbor (KNN), Multivariate Adaptive Regression Spline (MARS), Bayesian Generalized Linear Model (BGLM), and Generalized Linear Model (GLM). Figure 7 shows the machine learning algorithms utilized in this paper. Figure 6 shows the actual value of the vehicles and the predicted value of the machine learning such as BAG, KNN, MARS, BGLM, and GLM. Traffic congestion varies within the day event may cause surges in traffic at unexpected times. In Figs. 8, 9 and 10, we considered Morning, Afternoon,

Fig. 4 Vehicles versus time in 2015

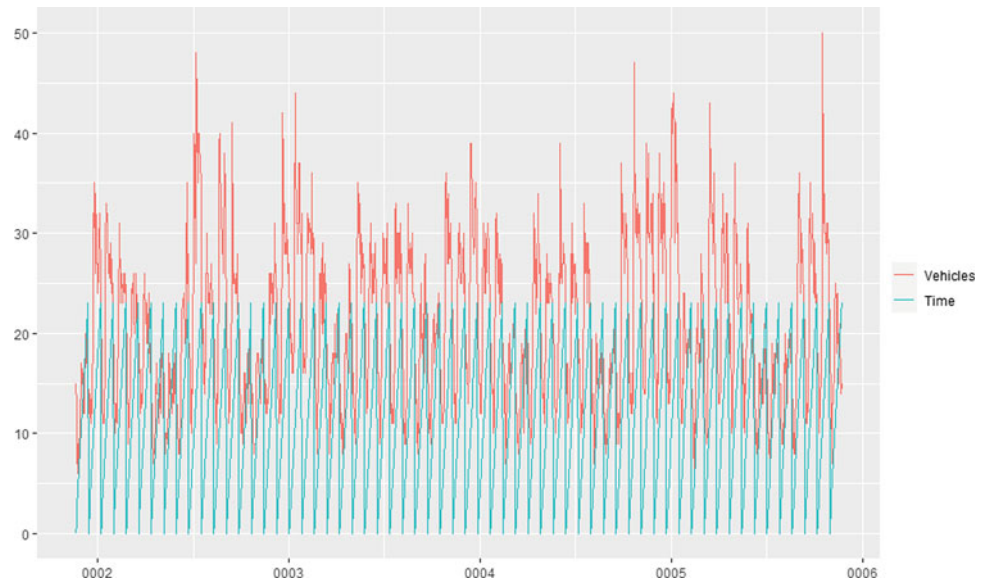
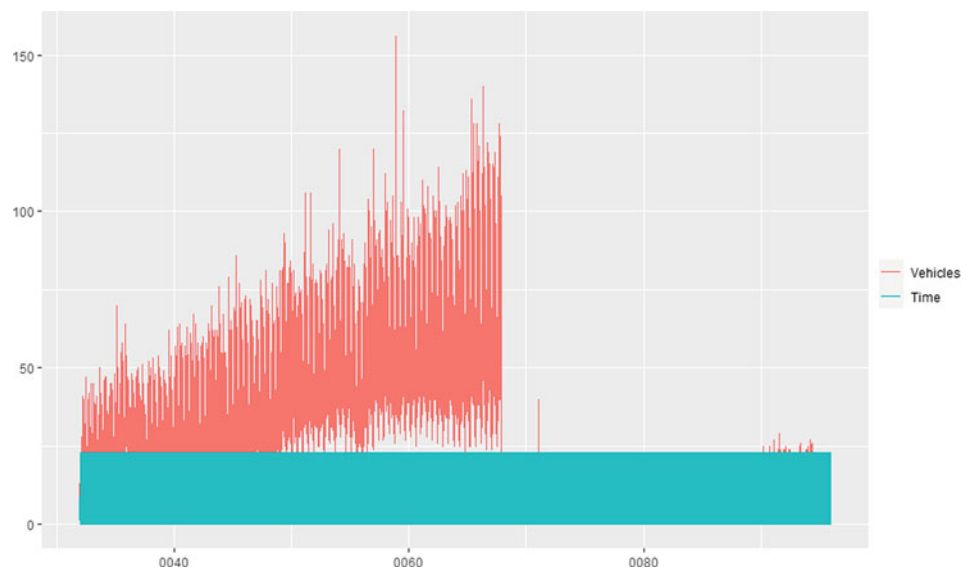


Fig. 5 Vehicles versus time in 2016



Evening, and Noon. The study revealed that in 2015 there is traffic surge in the Morning followed by Afternoon as shown in Fig. 8. In 2017 the traffic congestion increased exponentially which results in traffic surge in Afternoon than in the Noon as shown in Fig. 9. Depicted in Fig. 10 is the vehicular movement in 2017 which shows that Afternoon has the highest number of vehicle plying the road followed by Noon. The performance of each of the machine learning is shown in Table 4 using Root Mean Square Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Scaled Error (MASE). GLM performs better than three the remaining four algorithms such as BAG, KNN, MARS, and BGLM having the least accuracy of 8.6802, 75.3468, 1.0498

using RMSE, MSE, MASE. From Table 3, we can deduce that GLM is a better predictor compared to the other models used in this work. Therefore, it is a promising algorithm for predicting vehicular movements.

5 Conclusion

As a new type of urban development strategy based on information technology, smart city building is committed to reforming the conventional urban organizing and management model to enable efficient data and resources distribution and solve urbanization problems. This research studied

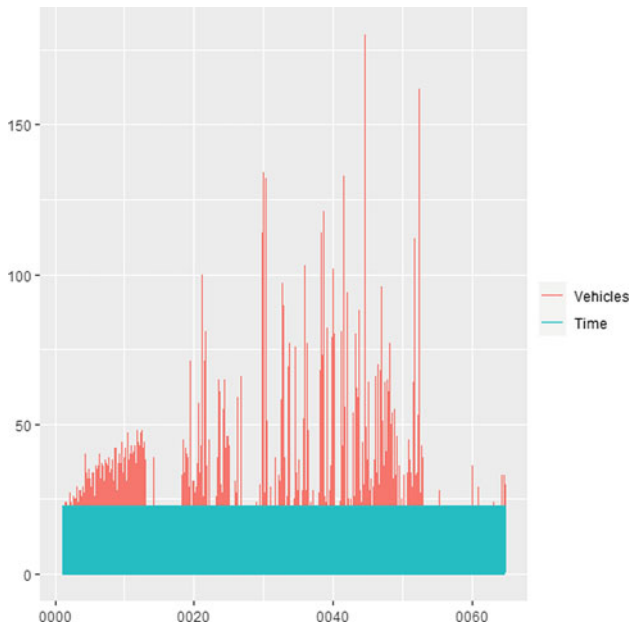


Fig. 6 Vehicles versus time in 2017

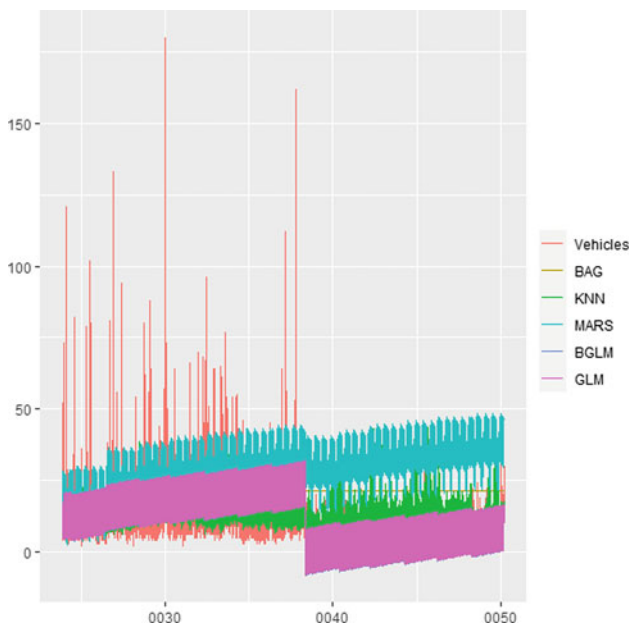


Fig. 7 Output results of actual (Vehicles) and predicted (BAG, KNN, MARS, BGLM, and GLM)

the transport congestion of the smart city. To address this gap in literature, this paper uses data from Kaggle database with 48,120 records from 2015 to 2017 to check if intelligent city-buildings such as road junction can mitigate congestion in cities. Results show that the building of smart cities such as road junctions greatly decreases the congestion level of

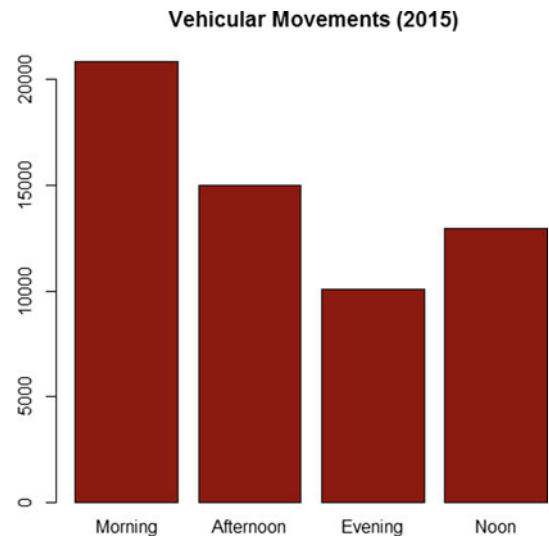


Fig. 8 Vehicular movements in 2015 during the day (Morning, Afternoon, Evening, Noon)

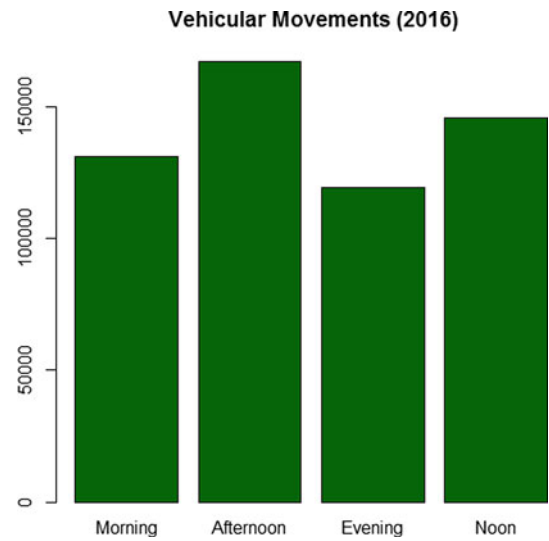


Fig. 9 Vehicular movements in 2016 during the day (Morning, Afternoon, Evening, Noon)

urban traffic and increases urban efficiency. The mechanism check shows that the development of smart cities and information technology is a way to handle road issues but also to reduce the congestion of urban roads by urban overall creativity. In order to promote the development of smart cities and to alleviate congestions in city traffic, this research paper has significant practical implications. The limitations of the work include the inability of the dataset to indicate the number of buses, taxis, and traffic cameras on each junction which will enable us to determine the rate of vehicular queuing on each junction of the road.

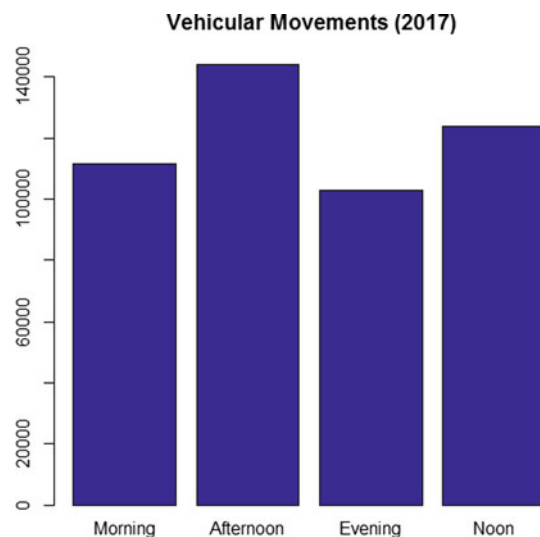


Fig. 10 Vehicular movements in 2017 during the day (Morning, Afternoon, Evening, Noon)

Table 4 Performance metrics of smart city traffic patterns

MODEL	RMSE	MSE	MASE
BAG	13.09740	171.5419	1.958938
KNN	9.236216	85.30769	1.093255
MARS	23.34668	545.0675	3.775596
BGLM	8.703736	75.75501	1.054976
GLM	8.680255	75.34683	1.049848

References

- Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-nearest neighbor (KNN) classification method. *Applied Computing and Informatics*, 12(1), 90–108.
- An, C., & Wu, C. (2020). Traffic big data assisted V2X communications toward smart transportation. *Wireless Networks*, 26(3), 1601–1610.
- Bhattacharya, S., Somayaji, S. R. K., Gadekallu, T. R., Alazab, M., Maddikunta, P. K. R. (2020) A review on deep learning for future smart cities. *Internet Technology Letters*, e187.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1.
- Gadekallu, T. R., Khare, N., Bhattacharya, S., Singh, S., Reddy Maddikunta, P. K., Ra, I. H., & Alazab, M. (2020). Early detection of diabetic retinopathy using PCA-firefly based deep learning model. *Electronics*, 9(2), 274.
- Gosh, U. (2018). Smart city traffic pattern. <https://www.kaggle.com/utathya/smart-city-traffic-patterns>
- Hinsbergen, C., Lint, J. W. C., & Sanders, F. (2007). Short term traffic prediction models. In *14th world congress on intelligent transport systems*.
- Hoffmann, J., Bar-Sinai, Y., Lee, L. M., Andrejevic, J., Mishra, S., Rubinstein, S. M., & Rycroft, C. H. (2019). Machine learning in a data-limited regime: Augmenting experiments with synthetic data uncovers order in crumpled sheets. *Science Advances*, 5(4), eaau6792.
- Iwendi, C., Maddikunta, P. K. R., Gadekallu, T. R., Lakshmana, K., Bashir, A. K., & Piran, M. J. (2020). A metaheuristic optimization approach for energy efficiency in the IoT networks. *Software: Practice and Experience*
- Jia, Y., Wu, J., & Du, Y. (2016). Traffic speed prediction using deep learning method. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)* (pp. 1217–1222). IEEE.
- Kumar, M. E., Reddy, G. T., Sudheer, K., Reddy, M. P. K., Kaluri, R., Rajput, D. S., & Lakshmana, K. (2017). November. Vehicle theft identification and intimation using gsm and IoT. In *IOP conference series: Materials science and engineering* (Vol. 263, No. 4, p. 042062). IOP Publishing.
- Morales, F., Ruiz, M., Gifre, L., Contreras, L. M., Lopez, V., & Velasco, L. (2017). Virtual network topology adaptability based on data analytics for traffic prediction. *IEEE/OSA J Opt Commun Networking*, 9(1), A35–A45
- Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., De Silva, D., Alahakoon, D., & Pothuhera, D. (2019). Online incremental machine learning platform for big data-driven smart traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 20(12), 4679–4690.
- Ng, J. R., Wong, J. S., Goh, V. T., Yap, W. J., Yap, T. T. V., Ng, H. (2019). Identification of road surface conditions using IOT sensors and machine learning. In *Computational science and technology* (pp. 259–268). Springer.
- Numan, M., Subhan, F., Khan, W. Z., Hakak, S., Haider, S., Reddy, G. T., Jolfaei, A., & Alazab, M. (2020). A systematic review on clone node detection in static wireless sensor networks. *IEEE Access*, 8, 65450–65461.

- Oyewola, D. O., Dada, E. G., Omotehinwa, O. T., & Ibrahim, I. A. (2019). Comparative analysis of linear, non linear and ensemble machine learning algorithms for credit worthiness of consumers. *Computational Intelligence and Wireless Sensor Networks*, 1(1), 1–11.
- Ozbayoglu, M., Kucukayan, G., & Dogdu, E. (2016). A real-time autonomous highway accident detection model based on big data processing and computational intelligence. In *2016 IEEE international conference on big data (Big Data)* (pp. 1807–1813). IEEE.
- Raghavan, R., Singh, J. K., Reddy, T. G., Sudheer, K., Venkatesh, P., & Olabiyisi, S. O. (2017). A case study: Home environment monitoring system using internet of things. *International Journal of Mechanical Engineering of Technology*, 8(11), 173–180.
- Reddy, G. T., Kaluri, R., Reddy, P. K., Lakshmana, K., Koppu, S., Rajput, D. S. (2019). A novel approach for home surveillance system using IoT adaptive security. In *Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM)*. Amity University Rajasthan.
- Reddy, T., Rm, S. P., Parimala, M., Chowdhary, C. L., Hakak, S., & Khan, W.Z. (2020). A deep neural networks based model for uninterrupted marine environment monitoring. *Computer Communications*, 157, 64–75.
- Reddy, G. T., Reddy, M. P. K., Lakshmana, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8, 54776–54788.
- Ruhlandt, R. W. S. (2018). The governance of smart cities: A systematic literature review. *Cities*, 81, 1–23.
- Samadi, M., Afshar, M. H., Jabbari, E., & Sarkardeh, H. (2020). Application of multivariate adaptive regression splines and classification and regression trees to estimate wave-induced scour depth around pile groups. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 44(1), 447–459.
- Shi, G., Lim, C. Y., & Maiti, T. (2019). Bayesian model selection for generalized linear models using non-local priors. *Computational Statistics and Data Analysis*, 133, 285–296.
- Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5), 1.
- Verma, S., & Badade, S. (2019). Traffic prediction using machine learning. In *Proceeding of national conference on machine learning*.
- Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7, 41525–41550.
- Williams, N., Zander, S., & Armitage, G. J. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *Computer Communication ACM SIGCOMM*, 36(5), 1–15.
- Wu, Q., Huang, C., Wang, S. Y., Chiu, W. C., & Chen, T. (2007). Robust parking space detection considering inter-space correlation. In *2007 IEEE international conference on multimedia and expo* (pp. 659–662). IEEE.
- Yang, J., Han, Y., Wang, Y., Jiang, B., Lv, Z., & Song, H. (2020). Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city. *Future Generation Computer Systems*, 108, 976–986.
- Zheng, W., Lee, D. G., Asce, M., & Shi, Q. (2006). Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of Transportation Engineering*, 132(2), 1–8.



Emergency Department Management Using Regression Models

S. Kezia, A. Hepzibah Christinal, D. Abraham Chandy,
and M. James Graham Steward

Abstract

COVID-19, a global pandemic has been ravaging the world. The Emergency departments are flooded because of this global pandemic. To provide a good service in the Emergency Departments (ED) in hospitals as a part of smart healthcare, tools that analyze, program, plan or prioritize is required to use the available resources (staff and treatment equipment) in a fine possible way. In this paper, the various queuing methods that are implemented to tackle the patient flow in EDs as well as outpatient departments in the already existing systems are surveyed, and a method is suggested. The previous papers have taken queuing theory into account. Here three different regression techniques namely Linear regression, polynomial regression, and support vector regression are considered for the prediction of the patient flow in Emergency Departments. The hazardous COVID-19 pandemic and its impact on the mounting crisis in the EDs is also discussed. The challenges and suggestive methods are also discussed here.

Keywords

Emergency departments (EDs) • Queuing models • COVID-19 • Regression models

1 Introduction

The COVID-19 pandemic has ravaged the world with a massive 148 billion reported cases across the world and over 3.13 million deaths (Source: Wikipedia). With countless people critically serious, the emergency wards across the world are flooded with patients. With numerous test samples to be tested each day, the testing centers are flooded with samples more than they could hold. On top of this, in India, many states' health infrastructures have been overwhelmed by the pandemic's violent second wave, with hospitals unable to provide enough oxygen to meet the rising case numbers.

As predicted, the emergency departments (ED) across the world are overcrowded. Due to the unpredictability in patient arrival rates, sickness severity, and at large, the health resources needed for treatment, an overcrowded ED is a highly probabilistic environment (material and human). Because of the high demand for medical resources, a technique of classifying patients according to their urgency has been devised.

Also, the Arrival-to-Provider Time (APT) which is also known as door-to-doc time and the Length of Stay (LoS) are Key Performance Indicators (KPIs) of a healthcare center (Fig. 1).

In this paper, an attempt has been made to survey the various methods of dealing patients in an emergency ward using the queuing theory. Then the prediction of the patient flow has been done using regression models and their root mean squared errors are compared.

S. Kezia

Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India

e-mail: kezias20@karunya.edu.in

A. H. Christinal (✉)

Department of Mathematics, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India

e-mail: christyhep@gmail.com

D. A. Chandy

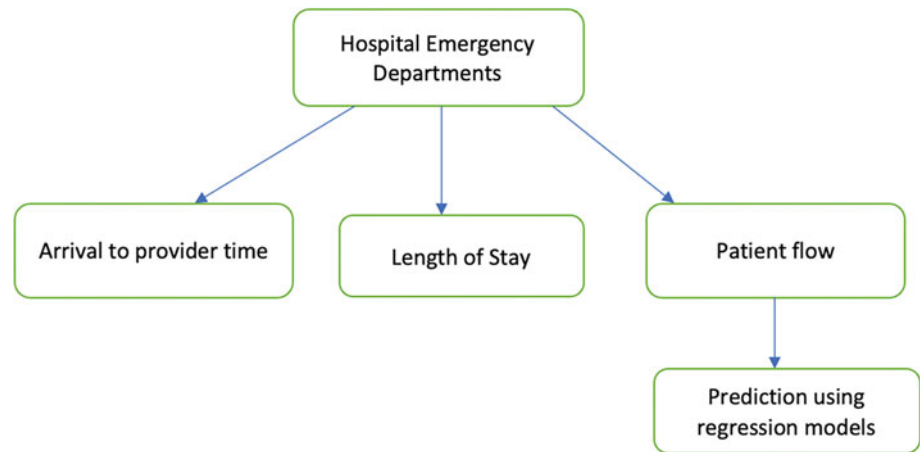
Department of Electronics and Communication, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India

e-mail: abrahamchandy@karunya.edu

M. J. G. Steward

PSG Institute of Management, Coimbatore, Tamil Nadu, India

Fig. 1 Flow of key performance indicators of an emergency department



2 Literature Survey

The various contributions and implementations of queueing theory in the area of healthcare management problems are reviewed in Lakshmi and Iyer (2013). The papers reviewed are from a wide-ranging span of 1952–2011. With the help of queueing models, this analysis suggests a method of classification for healthcare topic areas.

2.1 Types of Queuing Model Used

A priority queue is a type of queue in which each element has a priority assigned to it and is served according to that priority. If elements of the same priority appear in the queue, they are served in the order in which they appeared. In most cases, the value of the element itself is used to determine priority. Only comparable elements are covered by the priority queue, which ensures that the elements are either ordered in ascending or descending order. The authors of Pardo and Fuente (2007) have used Fuzzy Set Theory to provide a much clear and practical definition of priority-discipline queueing models. It creates and streamlines two fuzzy queueing models with priority discipline: one consisting of non-pre-emptive priorities and the other consisting of pre-emptive priorities, which are denoted by $M I/M I/1$ and $M I/Fi/1$, respectively. The procedures to build the fuzzy queueing model and function costs too are represented by membership functions that fully retain the ambiguity of the original information when some of the model's parameters are fuzzy. The approach suggested by the authors allows for rational solutions to be found for every situation, that has various levels of probability varying from the most negative to the most positive scenario. More detail is further given to aid in the design of fuzzy priority-discipline queueing systems.

After triage, a fresh perspective to manage the arrival of patients otherwise known as the patient flow in EDs is proposed by the authors in Cildoz et al. (2019). The new accumulative priority queue with finite horizon (APQ-h) queue discipline is an extension of the accumulative priority queue (APQ) discipline that considers not only the patient's acuity level and waiting time, but also their stage of treatment. The APQ-h discipline accumulates precedence from the moment of waiting for the initial physician appointment until the waiting period meets the upper time limit established to visit the physician following the patient's arrival. The authors have implemented this queueing model in a functioning ED by developing a simulating model. The results show that APQ-h discipline outperforms the conventional queue model in terms of patient flow and their acuity level.

2.2 Hospital Beds' Allotment

Using the RSR approach and M/M/n simulation model, the authors of Yang (2011) have analyzed one hospital bed structure and used the queueing theory to create a statistical model with the right of priority service, and also used linear programming to find sickbed distribution solutions with the shortest average stay period. RSR method is one among many prevalent comprehensive assessment methods. It represents the impact and effectiveness of medical care and overall hospital quality more conclusively and precisely, with a higher sensitive degree and strong epitome capacity. The authors claim that this model has been implemented in hospitals, resulting in a significant reduction in patient waiting times, the resolution of challenging hospitalization issues, and an increase in the use of hospital beds.

This paper (Azcarate et al., 2020) includes a survey of the medical and mathematical works based on patient discharge decisions, and a proposal for an innovative simulation

system that would allow for more practical mathematical modeling of the real-world patient discharge process. When developing the model, the authors took into account three significant factors or key points: the time it takes to make a discharge decision, the number of patients who will require intense postoperative care, and the patient's health status. As a result of current patient health state, bed occupancy level, and number of patients, a simulation-based optimization strategy too is suggested as a means of discovering optimum discharge decisions. The authors conclude that by evaluating discharge policies in a more realistic setting, this new simulation paradigm allows for a more accurate analysis.

This paper (Garcia-Vicuña et al., 2020) describes the development of the first Intensive Care Unit Management Flight Simulator (ICU). It can be used to examine the decision-making of the physician in connection to the admission and discharge of patient, as well as to learn and train new physicians. The designed discrete event simulation model simulates actual ICU admission and discharge procedures, as well as patient health condition. The authors were able to show physician heterogeneity in decision-making regarding the last bed problem, and it is addressed in a wider sense: it's not only about the allocation of the last available ICU bed; it's more about how physicians make patient admission and discharge choices when the ICU fills up.

2.3 Outpatient Queuing

In (Shan et al., 2013), the authors have applied queuing theory to a hospital queuing framework to analyze the outpatient department's mechanism and create queuing models using MATLAB. The hospital's queuing system (especially in the outpatient department) is made up of numerous stations that link the head to the tail in a logical manner. In general, the following is the order in which a patient accepts medical treatment: register, assess, fee, check, and take medication. The authors have concluded that M/M/1 and M/M/c queuing models are used in the hospital queuing systems referencing a Zhuzhou hospital.

Readmissions for heart failure are a major source of cost and mortality. Post-discharge follow-up is an evidence-based technique that has been shown to minimize readmission rates, but it is inherently difficult to enforce. The authors of Mutharasan et al. (2017) have determined the cost-effectiveness of adding more clinic slots under the BPCI (Bundled Payments for Care Improvement) that would make room for more timely post-discharge visits. Using queueing theory techniques, the influence of Heart Failure Discharge Clinic Capacity on Wait Time to an Appointment was modeled. Over the course of the 100-day experiment, 566 HF discharges were made, average 5.66 discharges

per day or 39.6 discharges per week. Only 31.9% of clinic appointments are booked within seven days, leading in a projected 22.3% rehospitalization rate and a \$3187 per patient price. The findings suggest that BPCI could improve care while also lowering costs.

Increasing the number of workers could not always be the best solution to solve the issue of patient waiting time. Since an increase in the number of staff could be more expensive than the average cost of patient waiting. The arrival rate (λ) and service rate (μ) are two essential metrics of queuing theory analysis utilized in Weerakoon et al. (2019), and they were utilized to assess current mean waiting times and furthermore to forecast mean waiting times as the number of servers expanded. At the end, a new technique to reducing average outpatient department wait times was implemented (OPD). The empirical results show that the proposed system will significantly reduce the mean waiting times of the OPD system. Furthermore, this rate was obtained by theoretical research, and a virtual patient management system is recommended as a means of reducing queue wait times.

2.4 Patient Flow Management

The authors of Zhu et al. (2013) constructed a general multi-stage queuing network model along with feedback patient flow to investigate the dynamics of patient flow between stages and the performance of hospital emergency departments (ED). The queuing network is divided into several single-queuing systems, such as tandem and closed queuing. The expression of patient flow between the stages is determined in a steady state using the system's routing probability, and the dynamic interaction between nearby stages is examined. Using the parameters computed from the observations, two key performance indicators are obtained. The entire performance metrics is considered based on the model's interpretation, and a revenue model for the ED in a hospital is said to be built.

The relationship between the different parameters such as patient satisfaction, waiting time, staff satisfaction, and service time is investigated in this paper (Komashie et al., 2015). It employs a number of models to help achieve change in terms of both experiential and organizational healthcare targets. A model based on waiting times is used to predict patient satisfaction levels. Experience and efficiency in care or treatment delivery are considered as the mandatory factors to develop a healthy wholesome environment. The authors developed a model for measuring staff satisfaction with service time. The course of development of this paradigm allows for a clear link between employee happiness and patient satisfaction. By examining the relationship between staff and patient happiness, the Effective Satisfaction Level (ESL) was developed. The authors conclude that

maintain ESL will lead to a more satisfied and holistic environment.

Unreasonably long wait times have an adverse effect on human resources as well as the patient's precious time, and patient dissatisfaction. It would be ideal if the patient could receive an appropriate care plan diagnosis and know ahead of time how long he or she would be in the hospital. As a result, the authors of Savanth and Mohan Babu (2017) suggest an approach known as Patient Care Time Prediction (PTTP), which measures the waiting time of patient for each treatment. The purpose of the Hospital Queuing-Recommendation (HQR) system is to suggest an effective care plan for each patient in the queue, thereby reducing the patient's waiting time. The results show that the proposed system assists patients in completing all their treatments in a predictable time period, thus assisting hospitals in efficiently scheduling each treatment task queue and avoiding the overcrowding of patients and inefficient queues.

Patients may be in a variety of phases during treatment, such as waiting for the first appointment, doing clinical assessments, waiting for a second consultation, and so on. As a result, choosing the next patient by a physician is not easy, particularly when several quality targets must be met at the same time. In (Cildo et al., 2018), the authors have used simulation-based optimization to discover the optimum APQ regulations. The authors have put both types of regulations to the test in a real-world scenario. Implementing appropriate regulations for patient flow management is crucial for the quality of healthcare, as the simulation model's findings reveal. The APQ discipline that delivers the best quality goals is attained using an SBO approach. The ideal parameters that describe the adopted APQ policy are then established by the quality objectives and the type of patients.

2.5 External Factors

The authors have offered a method for situating vehicle base stations in this study (Amorim et al., 2019), which uses scenario-based customized optimization to handle regular traffic and demand variations generated by city dynamics. The model helps us to acknowledge how frequent changes impact the response system of an urban emergency medical service (uEMS) in a better way. The model has been evaluated in two steps. To locate vehicle base stations, the initial phase uses scenario-based optimization and survival function theory, while the other phase uses agent-based simulation to evaluate the solution's efficiency and weigh it up to average-period and non-survival prone alternatives. The proposed models were tested in a range of scenarios utilizing real-world data from Porto, which was collected over the course of a year with a total of 33,736 events between May 10th 2012 and May 9th 2013. The results have shown that

the parameters of survival functions had an effectual influence on station positioning.

The authors of Joseph (2020) have attempted to learn how to avoid some major mistakes that administrators often make when allocating resources in the ED. To address the pitfalls and the trade-offs, they have used queuing theory. They have discussed the potential reasons and various areas of implementation of the queuing theory in the healthcare field.

The authors have described a technique for tracking occupational stress in emergency department (ED) physicians in real time in this work (Cildo et al., 2020). It's part of a Decision Support System (DSS) that assigns patients to doctors based on their triage results. To acquire the consensus views of ED physicians, a unique methodology including stress factor analysis, questionnaire design, and statistical analysis of expert opinions is used to construct a work stress function. Depending on workload data from ED physicians' whiteboards, the resulting stress score can be used to assess job stress. The data obtained by the previous function is used in a real-world setting. The authors conclude that using DSS to manage ED patient flow reduces job stress and distributes it evenly throughout the full team of physicians, as well as improving other important ED performance indicators including arrival-to-provider time and compliance percentage with patient waiting time targets.

2.6 COVID-19 Pandemic and Its Impact in EDs

According to an article by WHO, dated 13 October 2020, the economic and societal impacts of the epidemic are severe. Billions of people are on the verge of falling into severe poverty, with the amount of malnourished people expected to climb to 132 million by the completion of the year considered.

This paper (Mac et al., 2021) discusses about the various models that were developed so far based on the COVID-19 pandemic. It talks about cohort and individual level models, phenomenological models, Transmission Dynamic Models, Decision-analytic models, state-transition models, and the various challenges that are faced when acquiring data for training and evaluating these models.

The minimum mean mortality rate assessed by age- and sex-specific life expectancies is required, as age and sex are key drivers of death according to Covid-19 and, unlike other priority metrics, are easily accessible to health professionals. The authors of Bonneuil (2021) state that age has been a substantial effect in ventilator allocation at this period of scarcity. Only those who require a ventilator should be considered if the goal is to increase the total number of years of life saved, which is the mean rate of mortality weighted by age- and sex-specific life expectancies but not by

co-morbidities who are in association to the population of ventilator-dependent patients. They have also calculated an optimal cut-off age based upon the patient's flow rate as well as the ventilator availability.

People on the go, as well as inhabitants of squatter settlements in cities, are among the most vulnerable communities in the Global South. During the COVID-19 response, the authors of Raju et al. (2021) have taken India into account to illustrate the problems faced by refugees and urban slum dwellers, and to make human rights-based recommendations to take action immediately for the protection of these vulnerable communities. The occurrences in India during COVID-19 demonstrate how the security of migrants and other ultra-vulnerable urban communities has teetered between two stools. The authors believe that the lockdown should have been well planned and publicized to avoid unnecessary psychological distress and stressful events.

As summary of the papers reviewed, the different types of queuing models are reviewed, and the accumulative priority queue is the most effective queue used thus far. When it comes to outpatient queuing and patient flow management, different types of models have been developed that are specific to the case or the hospital taken into account. The external factors such as the vehicles used, and the allocation of staff also contribute to the queuing in emergency departments. As the ongoing pandemic is humongous, the following idea in machine learning is proposed.

The limitations observed in the existing methods can be put down as follows. Most of the models have taken a particular Emergency department in a particular hospital as a part of their case study and for the implementation of their proposed methodology. This restricts their model to that considered use case.

3 Methodology

The input dataset is taken from GitHub and is being constantly updated till date (<https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv>). The dataset consists of 5 columns which record the date, country, confirmed cases, recovered patients, and the deaths of approximately 1 lakh records. The data analysis of the dataset represents the global spread of covid-19 on the first date that was recorded. We could see that the initial range of cases range between 0 and 500. The cases are mainly recorded in China, United States, and Thailand respectively.

Then the data goes on to show the spread globally recorded on a recent date now. It is clearly seen that the range of cases being recorded now are in the millions. This clearly indicates the spread of the pandemic in a drastic

manner. The countries which had no cases initially have recorded millions of cases over the past few months.

The regression models used in this methodology are linear regression, polynomial regression, and support vector regression. Basic analysis to show the rise of the global pandemic has been carried out below.

Figure 2 represents the rise of infection in India spanning from March 2020 to May 2021. We see a tremendous spike in the level of infection spread across the country.

Figure 3 represents the raise in the number of confirmed cases on a day-to-day basis in Italy, USA, Spain, and India.

A linear model is one in which the input variables and the single output variable are assumed to have a linear relationship. The output variable is determined from a linear combination of the input variables, in more detail. Here the output variable is the confirmed cases. The input variables are the spread of cases across the different months or the time period. The output variable is determined as a linear combination of the input variables.

Linear regression model is formulated using,

$$\text{weight} = B_0 + B_1 * \text{height}$$

where B_0 is the bias coefficient and B_1 is the coefficient for the height column.

Polynomial regression is a kind of linear regression that uses an nth degree polynomial to predict the association. It's an example of multiple linear regression in action. Polynomial Regression is sensitive to outliers, therefore the presence of one or two of them can have a negative impact on the findings.

The formula for this regression is,

$$Y = b_0 + b_1X + b_2X^2 + \dots + b_mX^m + \text{residual error}$$

where b_0 , b_1 , b_2 , and b_m are the regression coefficients.

The supervised learning algorithm, Support Vector Regression is implemented to predict discrete values. Both SVMs and Support Vector Regression are built on the same foundation. The basic point of SVR is to find the best-fitting line. In SVR, the hyperplane with the highest number of points is the best fit line.

The equation of the line in Support Vector Regression is,

$$y = wx + b$$

which is identical to Linear Regression. This straight line is also known as hyperplane in SVR.

The three regression models which are used for the prediction of confirmed cases are Linear Regression, Polynomial Regression, and Support Vector Machine regression. Out of these three predictors from the results obtained below, the polynomial regressor overfits, the linear regressor is falling apart and the SVM regressor is either low or high.

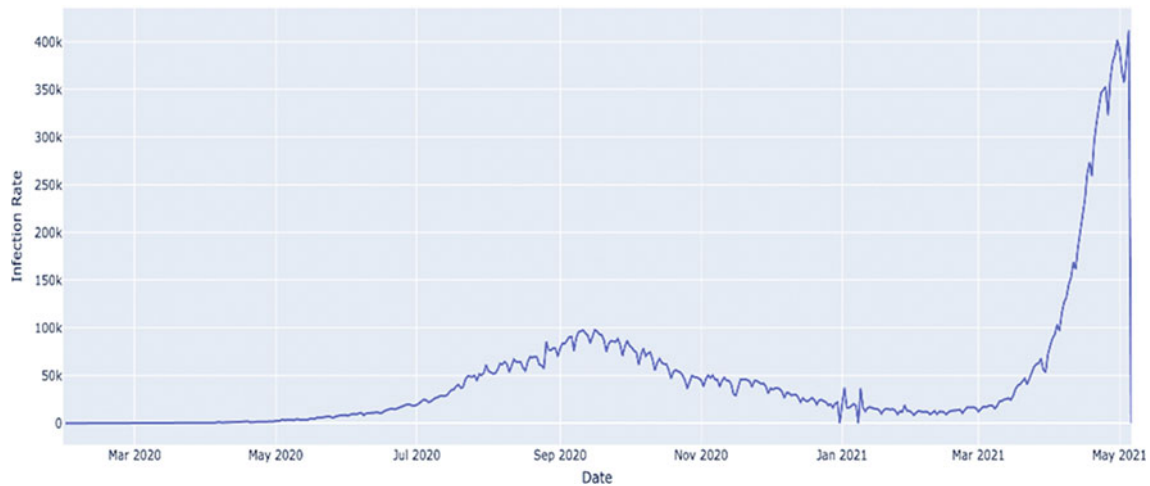
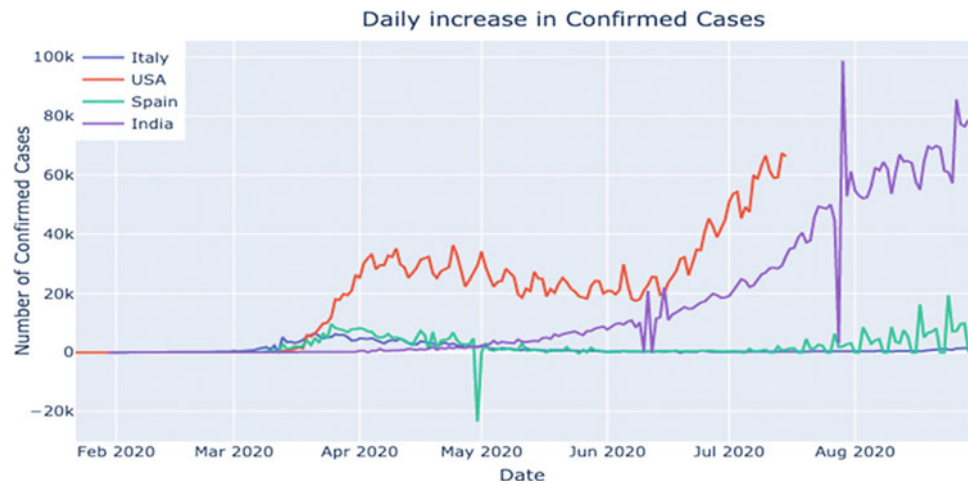


Fig. 2 Infection curve in India

Fig. 3 Daily increase in confirmed cases



4 Results and Discussion

Figure 4 represents the confirmed cases prediction using Linear Regression model. The Linear Regression Model obtained here is absolutely falling apart. As it is clearly visible that the trend of Confirmed Cases is absolutely not Linear.

Figure 5 represents the confirmed cases prediction in Polynomial Regression.

Figure 6 represents the confirmed cases prediction using Support vector machine.

The prediction obtained from the regression models ends up in one definitive conclusion. As inferred from the graphs, the linear regression model is kind of falling apart, the polynomial regression model is overfitting the results and the support vector machine regression model shoots up really high or falls flat. So, the support vector regression models fit the model better.

With the prediction of number of confirmed cases, the patient flow management in the Emergency Departments can be regulated prior to the arrival of the patients or the queue being formed. With the patients arriving in an Emergency Department, the door to doc time usually prolongs with the rate of patient flow. When these values can be predicted, the allocation of the patients becomes easier and their length of stay shortens.

5 Challenges

The first major issue or challenge to be tackled at the current moment is the inflow of patients on a daily basis and the rate of admission of people in the Intensive Care Units (ICU) or Emergency Departments (ED). With the daily cases and the mortality rates surging at an alarming rate, especially in India and other parts of the world, it is important to model a queuing system that effectively distributes patients to the treating physicians in an efficacious manner.

Fig. 4 Confirmed cases in linear regression prediction

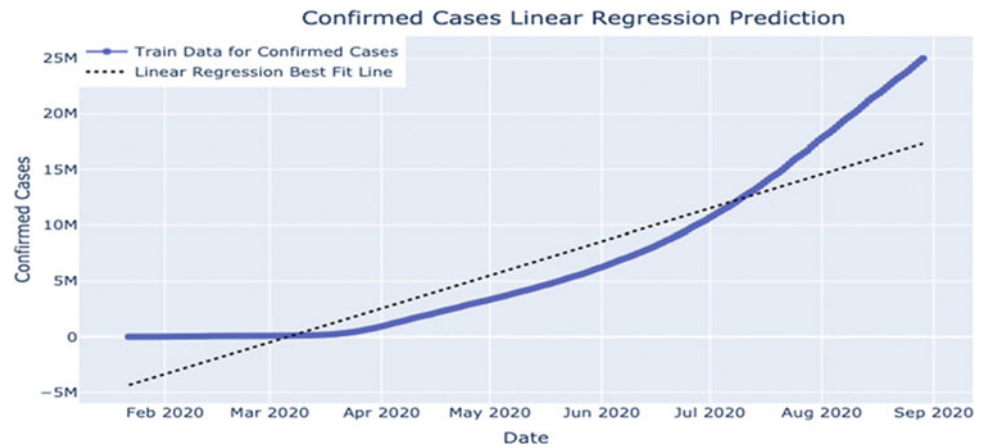


Fig. 5 Confirmed cases in polynomial regression prediction

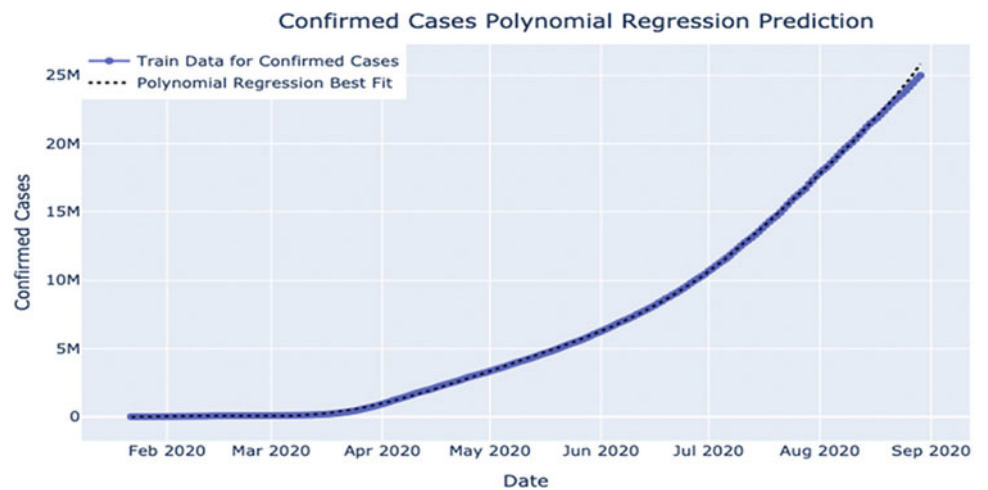
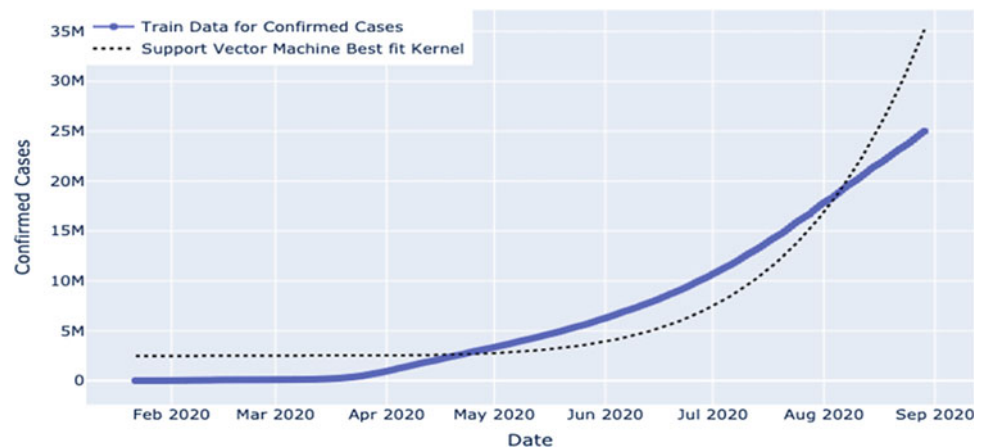


Fig. 6 Confirmed cases in SVM regression prediction



Another issue faced by the patients in India is the demand in the supply of Oxygen Cylinders to the ICU wards. The main issue is a lack of oxygenated ICU beds and an oxygen supply shortage. Many patients with low oxygen saturation have been unable to obtain either a bed or an oxygen cylinder, resulting in death. Non-COVID-19

patients, such as people with acute pulmonary disorders, who rely on a steady supply of oxygen to go about their everyday lives, have also been affected by the crisis. Many senior citizens use oxygen at home on a daily basis. The majority of them are unexpectedly finding themselves on their own.

With the number of cases of mounting up, an efficient model to tackle the oxygen supply demand would be highly appreciated. Also, the number of patients crowding in the outpatient department must be handled efficiently to avoid overcrowding there.

6 Conclusion

With a surging pandemic on the rise, researchers across the world are trying to model a system that would effectively aid physicians to treat their patients. The queuing models that are being implemented in the conventional emergency environments of today are to be mounted up to meet the need of the day. Also, only the confirmed cases are considered when it comes to analyzing the emergency departments. In the future studies, the number of patients being tested will also be taken into account. In this way, the patient flow management can be regulated further.

References

- Amorim, M., Ferreira, S., & Couto, A. (2019). How do traffic and demand daily changes define urban emergency medical service (UEMS) strategic decisions?: A robust survival model. *Journal of Transport and Health*, 12, 60–74. <https://doi.org/10.1016/j.jth.2018.12.001>
- Azcarate, C., Esparza, L., & Mallor, F. (2020). The problem of the last bed: Contextualization and a new simulation framework for analyzing physician decisions. *Omega*, 96, 102120. <https://doi.org/10.1016/j.omega.2019.102120>
- Bonneuil, N. (2021). Optimal age- and sex-based management of the queue to ventilators during the COVID-19 crisis. *Journal of Mathematical Economics*, 93, 102494. <https://doi.org/10.1016/j.jmateco.2021.102494>
- Cildoz, M., Mallor, F., & Ibarra, A. (2018). Analysing the ED patient flow management problem by using accumulating priority queues and simulation-based optimization. In *2018 Winter Simulation Conference (WSC)*. <https://doi.org/10.1109/wsc.2018.8632323>
- Cildoz, M., Ibarra, A., & Mallor, F. (2019). Accumulating priority queues versus pure priority queues for managing patients in emergency departments. *Operations Research for Health Care*, 23, 100224. <https://doi.org/10.1016/j.orhc.2019.100224>
- Cildoz, M., Ibarra, A., & Mallor, F. (2020). Coping with stress in emergency department physicians through improved patient-flow management. *Socio-Economic Planning Sciences*, 71, 100828. <https://doi.org/10.1016/j.seps.2020.100828>
- Dataset. <https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv>
- García-Vicuña, D., Esparza, L., & Mallor, F. (2020). Safely learning intensive care unit management by using a management flight simulator. *Operations Research for Health Care*, 27, 100274. <https://doi.org/10.1016/j.orhc.2020.100274>
- Joseph, J. W. (2020). Queuing theory and modeling emergency department resource utilization. *Emergency Medicine Clinics of North America*, 38, 563–572. <https://doi.org/10.1016/j.emc.2020.04.006>
- Komashie, A., Mousavi, A., Clarkson, P. J., & Young, T. (2015). An integrated model of patient and staff satisfaction using queuing theory. *IEEE Journal of Translational Engineering in Health and Medicine*, 3, 1–10. <https://doi.org/10.1109/jtehm.2015.2400436>
- Lakshmi, C., & Iyer, S. A. (2013). Application of queuing theory in health care: A literature review. *Operations Research for Health Care*, 2, 25–39. <https://doi.org/10.1016/j.orhc.2013.03.002>
- Mac, S., Mishra, S., Ximenes, R., et al. (2021). Modeling the coronavirus disease 2019 pandemic: A comprehensive guide of infectious disease and decision-analytic models. *Journal of Clinical Epidemiology*, 132, 133–141. <https://doi.org/10.1016/j.jclinepi.2020.12.002>
- Mutharasan, R. K., Kansal, P., Jackson, H. A., et al. (2017). Heart failure care transitions: Queuing theory-based cost-effectiveness analysis of outpatient clinic capacity sizing. *Journal of the American College of Cardiology*, 69, 2508. [https://doi.org/10.1016/s0735-1097\(17\)35897-7](https://doi.org/10.1016/s0735-1097(17)35897-7)
- Pardo, M. J., & de la Fuente, D. (2007). Optimizing a priority-discipline queueing model using fuzzy set theory. *Computers and Mathematics with Applications*, 54, 267–281. <https://doi.org/10.1016/j.camwa.2007.01.019>
- Raju, E., Dutta, A., & Ayeb-Karlsson, S. (2021). Covid-19 in India: Who are we leaving behind? *Progress in Disaster Science*, 10, 100163. <https://doi.org/10.1016/j.pdisas.2021.100163>
- Savanth, S.S., & Mohan Babu, K. N. R. (2017). Hospital queuing-recommendation system based on patient treatment time. In *2017 International conference on intelligent computing and control systems (ICICCS)*. <https://doi.org/10.1109/iciccs.2017.8250606>
- Shan, X., Jing, L., & Zhifeng, L., et al. (2013). The study and application of intelligent queuing in outpatient department. In *2013 third international conference on intelligent system design and engineering applications*. <https://doi.org/10.1109/isdea.2012.372>
- Weerakoon, W. M. N. B., Vasanthapriyan, S., & Ishanka, U. A. P. (2019). A queuing model for outpatient department to reduce unnecessary waiting times. In *2019 14th Conference on industrial and information systems (ICIIS)*. <https://doi.org/10.1109/iciis47346.2019.9063348>
- Wu, K., Zhu, X., Zhang, R., & Liu, S. (2019). Hospital bed planning in a single department based on Monte Carlo Simulation and queuing theory. In *2019 IEEE international conference on industrial engineering and engineering management (IEEM)*. <https://doi.org/10.1109/ieem44572.2019.8978497>
- Yang, O. (2011). The inquiry in hospital beds arrangement based on queuing theory. In *2011 6th International Conference on Computer Science and Education (ICCSE)*. <https://doi.org/10.1109/iccse.2011.6028666>
- Zhu, H., Gong, J., & Tang, J. (2013). A queuing network analysis model in emergency departments. In *2013 25th Chinese control and decision conference (CCDC)*. <https://doi.org/10.1109/ccdc.2013.6561230>

Machine Learning in Wind Energy: Generation to Supply

Bhavya Bhardwaj and M. Ganesan

Abstract

The need for clean and green power is increasing daily, to satisfy these Nuclear Plants, Solar Plants and Windfarms are on the up-rise. Windfarms become an obvious choice near coastal regions where the wind speed is adequate for the efficient and reliable working of Wind Turbines. Focusing on the needs of a smart city, it becomes necessary for the management of Windfarms to also become smart, with foreplanning, layout optimization and smart transmission. Since the output of windfarms is largely seasonal and depends on wind speed, the need to forecast wind becomes a necessity. The forecasting of wind using machine learning algorithms can help estimate future power generation. This work will compare the performance of various Machine Learning Algorithms in forecasting wind speeds and direction. Windfarms need to be optimized to offset the wake effect caused by the overlapping wake regions of the windmills, an ill-placed windmill can cause loss of efficiency and capacity. To combat the loss in efficiency and power in Windfarms, this work will present ways to optimize the performance of Windfarms using methods like Genetic Algorithm, Particle Swarm Optimization, Nelder-Mead and Random Search. The optimization performed will be subject to constraints such as Turbine type, Layout, Proximity and seasonal wind trends. The work will also present a novel approach to optimization derived from the existing work presented and will compare their performance. Genetic Algorithm and Nelder-Mead-Genetic Algorithm yield the best results, while moderately computationally expensive, they can help increase efficiency by 1–1.3%,

Nelder-Mead Particle Swarm Optimization can improve efficiency by 0.6% and Random Search by 0.36% in 50 iterations. To allow for uninterrupted power supply, Machine Learning models will be used to classify faults in High-power Transmission Lines to help predict the nature and condition of fault.

Keywords

Windfarm • Wind forecasting • Optimization • Particle swarm optimization • Nelder-Mead • Genetic algorithm • Nelder-Mead-PSO • Nelder-Mead-GA • Transmission lines

1 Introduction

As we burn through crucial natural resources to satisfy energy and electricity requirements, it becomes necessary to sustain development and optimize the use of natural resources and increase dependency on renewable sources of energy. Windfarms can help provide the solution. Table 1 shows the wind generation capacity of India (<https://www.mnre.gov.in/wind/current-status/>; Majid, 2019). The importance of windfarms will be discussed in Sect. 2. The performance of any windfarm is affected by the wind in the region. The ability to predict wind speeds and direction becomes very useful when trying to predict and accommodate future power demands. Several factors can affect the wind speed in a region, not limited to Temperature and proximity to coast. Use of Machine Learning algorithms to calculate estimated wind speeds and direction will be discussed in Sect. 3 and will help in better resource management and allocation in our smart cities. The performance of Windfarms is also affected by the wake effect, which is the change in wind speed due to the impact of wind turbines. These wake effects can reduce the energy generation capacity of the windfarm. To combat this, the following

B. Bhardwaj · M. Ganesan (✉)
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Coimbatore, India
e-mail: m_ganesan1@cb.amrita.edu

B. Bhardwaj
e-mail: bhavya1705@yahoo.com

work suggests the use of Mathematical Optimization to autogenerate the windfarm layout (Donovan, 2005; Samorani, 2013). The problem of optimization can be understood as follows. Windfarms, a collection of wind turbines are generally located in regions that have high wind speeds and ample land available. This provides the first constraint—limited land size, and the first variable wind speed. As windfarms have to run throughout the year, they are also affected by seasonal trends, high wind speeds and low wind speeds cannot be sustained; providing us the second constraint—seasonal wind speeds (Prabha et al., 2019; Raphael & Vanitha, 2018). Windfarms also cannot be placed close to each other, this provides us the third constraint—inter turbine distance, here limited to 400 m. The last and fourth constraint—the distance from boundary, here assumed to be 50 m (Donovan, 2005). The coordinates in relative Cartesian plane, which provide the input vector, that will be changed to reach an optimized solution. This work will detail the optimization of Windfarms, a source of green and clean energy to improve productivity and efficiency using different optimization techniques and discuss their performance. In subsequent sections, the need for windfarms will be discussed. Section 4 will discuss the methodology, Sect. 5 Wind Forecasting, Sect. 6 existing windfarm optimization techniques, including a novel approach using Nelder-Mead-Genetic Algorithm. Once power has been generated it needs to be transmitted and distributed. With electric grids and transmission lines becoming bigger in size, it becomes important to automate certain procedures and decrease manual effort. This work will detail the working of a diagnostic system for transmission lines which will allow to predict the condition and type of fault in the system, Sect. 7. Relying on Machine Learning for Prediction. A smart transmission system will allow better maintenance and uninterrupted power flow from the windfarms. Section 8 will present the metrics for evaluation. Section 9 will compare the results obtained from Sect. 5, 6 and 7 which will be concluded in Sect. 10.

2 Need for Windfarms

India is a developing country, with the demand for electricity and power increasing exponentially. Current needs of electricity generation are fulfilled using Thermal Power Plants fueled by Coal (53%), Lignite (1.7%), Gas (6.6%) and Diesel (0.1%), Hydroelectric Power Plants (12.2%), Nuclear Energy (1.8%), and Renewable Energy Sources include Small Hydro Project, Biomass Gasifier, Biomass Power, Urban & Industrial Waste Power, Solar and Wind Energy (24.5%) (<https://powermin.gov.in/en/content/power-sector-glance-all-india>). The suggested trends in upcoming energy sources as seen in Fig. 1 (IEA & Power, 2000–2040), show Solar and Wind

Energy to be the largest contributor to Indian Energy demands. This in line with the Paris Climate Accords that India is a signatory to. Increased dependency on Solar, Wind and Nuclear Energy will help combat Green House Emissions and help reduce pollution. Wind energy is the fastest-growing renewable energy technologies. In 2016, 16% of the electricity generated by renewable sources of energy was by wind. Wind power is cost effective, it promotes growth and jobs in manufacturing, installation, maintenance and support. It is a domestic source of energy, with reduced dependency on foreign imports. It is sustainable and falls in line with the goal of sustainable development and smart cities. One of the concerns of Wind plants has been sound and aesthetic pollution. Windfarms not being the most profitable use of land are also concerning. This can be overcome by installing offshore plants, of which India has 17.

3 Smart Cities and Wind Power

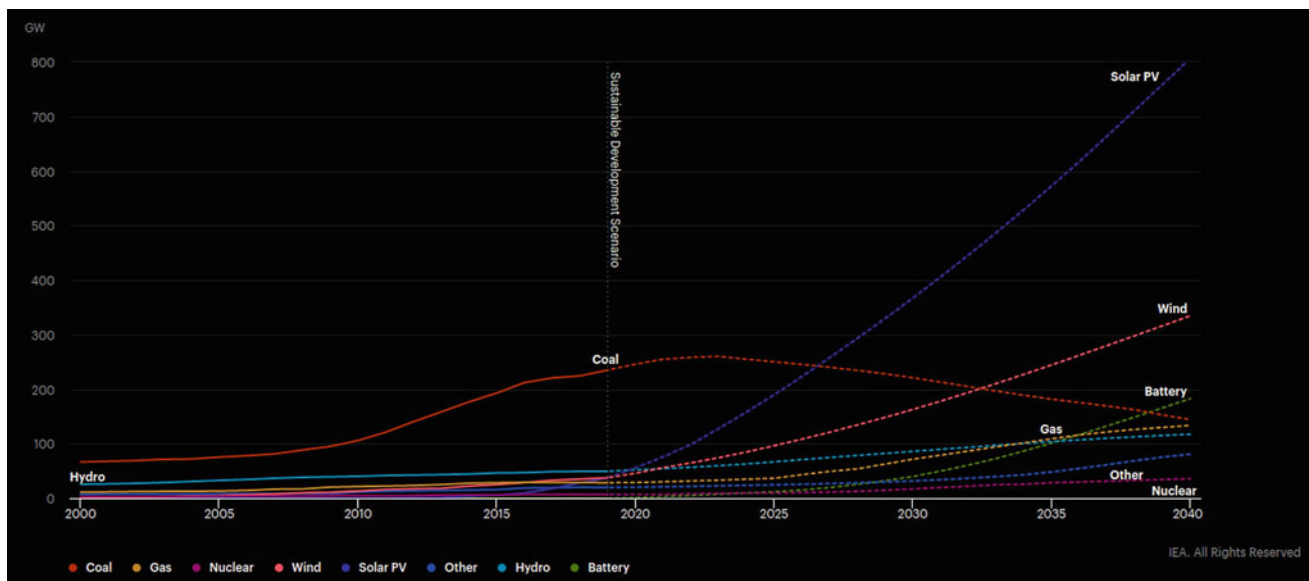
“The purpose of the Smart Cities Mission is to drive economic growth and improve the quality of life of people by enabling local area development and harnessing technology, especially technology that leads to Smart outcomes.”—Vision Smart Cities, India (<https://smartcities.gov.in/>). Coastal and riverine regions have always been the hubs of Human Civilization, they provide easy accessibility to water, arable lands, easy means of transport and trading and a clean source of energy. Keeping economic growth and quality of living in mind most smart cities focus on providing reliable power supply. This power supply can come from non-renewable sources (generally polluting) or renewable sources. Of the 100 smart cities in India, 2020 at least 15 are located near the Coast Line, making them a candidate for offshore and onshore windfarms. Windfarms are non-polluting and renewable source of energy. Reduced pollution contributes to improved quality of life. As one of the fastest growing source of energy, Wind Power is expected to be a large source of employment. With reduced cost of operations and improved productivity aided by Machine Learning and Optimization, Windfarms can contribute immensely economically. As such, the integration of Wind Power as a basic Power Generation component of a smart city is a must. This work will detail ways to improve production and supply in Wind Power Stations using Machine Learning.

4 Methodology

This work will aim to optimize the generation and supply of Wind Power using Optimization Algorithms and Machine Learning. The workflow of this work can be briefly understood from the flowchart in Fig. 2. Wind Forecasting in

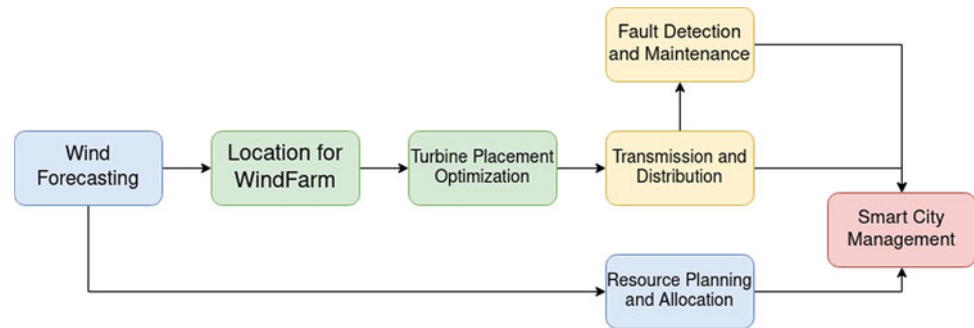
Table 1 Wind potential India

S. No	State	Wind potential (MW)
1	Gujarat	84,431.33
2	Rajasthan	18,770.49
3	Maharashtra	45,394.34
4	Tamil Nadu	33,799.65
5	Madhya Pradesh	10,483.88
6	Karnataka	55,857.36
7	Andhra Pradesh	44,228.6
8	Kerala	1699.56
9	Telangana	4244.29
10	Odisha	3093.47
11	Chhattisgarh	76.59
12	West Bengal	2.08
13	Puducherry	152.83
14	Lakshadweep	7.67
15	Goa	0.84
16	Andaman & Nicobar	8.43
	Total in MW	302,251.49
	Total in GW	302

**Fig. 1** Energy projections—India (IEA & Power, 2000–2040)

addition to existing data, is used to assess the feasibility of a location for a future windfarm. Once a windfarm is planned, its layout is designed and optimized as discussed in Sect. 5. The optimization of the windfarm layout leads to higher thorough put and improved revenue. Once energy is generated, it is transmitted through transmission lines, whose faults and damages are predicted using Machine Learning

Diagnostic Systems. The Wind forecasting data is then used to estimate future generation and compared with demand to better analyze and allocate the electrical power resources of the smart city. The outputs from the Diagnostic Device are utilized by City Management to better understand the shortcomings and advantages of the system in place, allowing for quick upgradation and repairs.

Fig. 2 Wind power smart system

5 Wind Forecasting

Several methods have been proposed for Wind Speed and direction forecasting (Prabha et al., 2019; Raphel & Vanitha, 2018). The ability to predict wind patterns is not only beneficial in windfarms but also in meteorology, aviation, agriculture. Wind forecasting plays a very important role in the establishment and operation of a windfarm. Studying the current and future trends in wind patterns allows for choosing suitable location for a windfarm. As they are heavily depended and constrained by wind speed, the ability to estimate performance is very much appreciated. Wind in a region can be affected by several different factors such as Temperature, Humidity, Wind Currents, proximity to the coast, heating of earth's surface, proximity to leeward regions, pressure. Affected by so many factors, accurate modeling of wind patterns is a difficult and cumbersome task. Here the forecasting of wind is done using existing wind parameters at varying heights, with the assumption that the forementioned factors contribute to current wind patterns, which will impact future wind patterns. The dataset taken from Kaggle (<https://www.kaggle.com/natasha23/wind-forecasting-dataset>) was of size $16,800 \times 6$ with meridional and zonal wind velocity (m/s) at 10 and 100 m height, as seen in Table 2. 15 Machine Learning Models of varying algorithms were trained and tested, the results for their performance can be seen in Table 3 and Fig. 3.

Table 2 Wind forecasting dataset

Property	Wind forecasting dataset
Rows	16,800
Columns	6
Features	Meridional wind speed (1) 10 m (2) 100 m Zonal wind speed (1) 10 m (2) 100 m
Validation split	70–30

From the above Table 3 it can be easily seen that the Boosted Tree Regression Model gives the best results. It is closely followed by Random Forest and Decision Tree. While more complex methods exist for estimating wind patterns, the scope of this work is limited to comparing the performance of some common Machine Learning Algorithms discussed above.

6 WindFarm Optimization

There exist various methods to optimize windfarms. Some commonly used methods are using Particle Swarm Optimization (PSO), Genetic Algorithm, and using evolutionary methods. Here we will discuss four methods for Windfarm optimization and compare their performance. The methods considered here will be Random Search method as proposed by Feng et al., (Feng & Shen, 2015), Genetic Algorithm proposed by Park et al., (2019) and PSO-Nelder-Mead proposed by Bhardwaj et al., (2021) and the novel approach that will be the culmination of Park et al., (2019) and (Bhardwaj et al., 2021). The windmills used here have a rotor diameter of 100 m and height of 100 m.

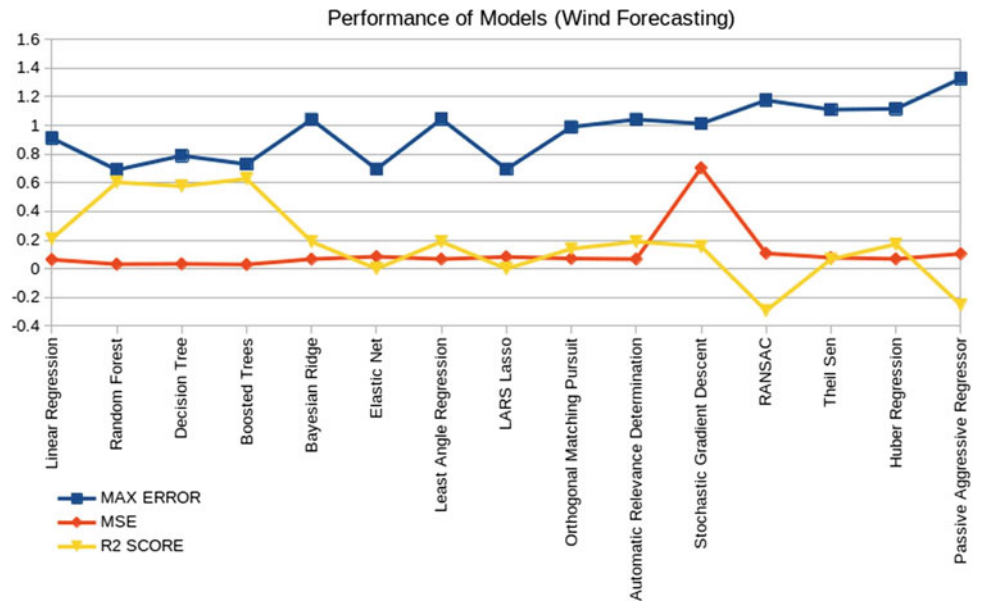
6.1 Optimization Using Random Search

The random-search method proposed by Feng & Shen (2015) uses an improved Random Search algorithm that

Table 3 Performance of models

S. No.	Model	Max. error	MSE	R2 score
1	Linear regression	0.9115	0.0645	0.2098
2	Random forest	0.6901	0.0323	0.6035
3	Decision tree	0.7891	0.0346	0.5766
4	Boosted trees	0.7302	0.0304	0.6278
5	Bayesian ridge	1.0418	0.0677	0.1894
6	Elastic net	0.6954	0.0835	0
7	Least angle regression	1.0444	0.0677	0.1894
8	LARS Lasso	0.6953	0.0835	0
9	Orthogonal matching pursuit	0.9896	0.0718	0.1397
10	Automatic relevance determination	1.0417	0.0677	0.1894
11	Stochastic gradient descent	1.0124	0.0704	0.1564
12	RANSAC	1.1756	0.108	-0.2924
13	Theil sen	1.1104	0.0779	0.0672
14	Huber regression	1.1163	0.0691	0.1717
15	Passive aggressive regressor	1.3277	0.1047	-0.2528

Fig. 3 Performance of wind forecasting models



retains the last good sample to help improve energy production. The wake effect is modeled using the Jensen Model (Shakoor et al., 2016). The constraints are boundary of the Windfarm and Minimal feasible distance. The algorithm used in the scope of this work is as proposed in Feng and Shen (2015). Unlike the other methods, here only the

position of a single turbine is changed. The AEP using this method changes from 505.4506 GWh to 507.2882 GWh.

Step-1 Select Initial Layout S_0 .

Step-2 Select any wind turbine at random and move in random direction with random step.

Step-3 If improvement in Cost function, New Layout = $S_0 + \delta(S)$, else $S_0 = S_0$.

Step-4 Check constraints, if constraints failed repeat Step-3.

Step-5 Check Layout constraints, if constraints failed repeat Step-3.

Step-6 If improvement in Cost function continues with same wind turbine, else choose another turbine.

6.2 Optimization Using Genetic Algorithm

The Genetic Algorithm proposed by Park et al., (2019) uses a Binary Genetic Algorithm to optimize the windfarm, with the Daegwallyeong windfarm, South Korea as reference. The Constraints applied are the number of turbines and the cost function is the annual energy Production (AEP). The AEP using this method changes from 505.4506 GWh to 510.6519 GWh.

Step-1 Form a Binary vector from the initial layout.

Step-2 The dimension of the vector = (Length of Windfarm/Length of Block) \times (Width of Windfarm/Width of Block).

Step-3 Total Number of Gene with value 1 = Total Number of Turbines (indicates presence of windmill).

Step-4 Pass the Initial Sample as parent to the Genetic Algorithm.

Step-5 Choose the Best Offspring, with fitness metric = AEP.

6.3 Optimization Using Nelder-Mead and PSO

This method as proposed by Bhardwaj et al., (2021) uses PSO and Nelder-Mead. The wake effect is modeled using the Jensen's model. The inputs are the boundary constraints, dimensions of windfarm, limited to rectangular shape and number of turbines. In the scope of this work, the PSO model used will be derived from Kennedy and Eberhart (1995), Shi and Eberhart (1998), Mouna et al., (2018). The AEP using this method changes from 505.4506 GWh to 508.5639 GWh.

Step-1 Create vector of size Number of Turbines \times 2, with coordinates of turbines.

Step-2 Pass as initial sample to PSO.

Step-3 Optimize till convergence.

Step-4 Fine Search and Optimization using Nelder-Mead.

Step-5 Stop at convergence.

6.4 Novel Approach: Optimization Using Nelder-Mead and Genetic Algorithm

This approach is similar to the work done in Bhardwaj et al. (2021) but uses Binary Genetic Algorithm (Park et al., 2019) instead of PSO and Nelder-Mead (Singer & Nelder, 2009) for fine search. The use of Nelder-Mead for fine search helps reduce loss in precision caused due to formation of fixed block size (that indicates position of Turbine, here $50 \times 50 m^2$) in Binary Genetic Algorithms. This technique is less computationally expensive than (Park et al., 2019) and yields data with greater precision. The AEP using this method changes from 505.4506 GWh to 512.0828 GWh.

Step-1 Form a Binary vector from the initial layout.

Step-2 The dimension of the vector = (Length of Windfarm/Length of Block) \times (Width of Windfarm/Width of Block).

Step-3 Total Number of Gene with value 1 = Total Number of Turbines (indicates presence of windmill).

Step-4 Pass the Initial Sample as parent to the Genetic Algorithm.

Step-5 Choose the Best Offspring, with fitness metric = AEP.

Step-6 Local and Fine Search using Nelder-Mead.

Step-7 Stop at convergence.

In this work, the wake effect will be modeled using Jensen Model (Shakoor et al., 2016), which is compatible with existing academic and commercial work and has been used in Works mentioned above. All optimization techniques are run for 50 iterations. The runtime and AEP is plotted in Figs. 4, 5, and 6.

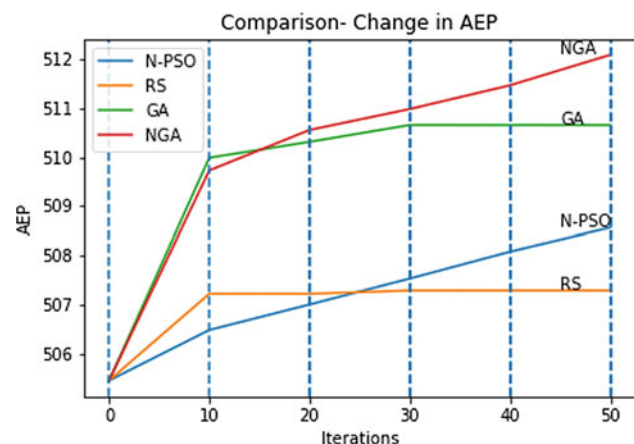


Fig. 4 AEP comparison

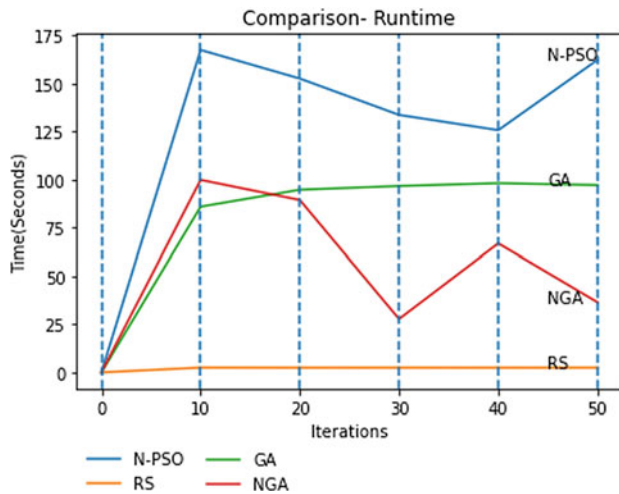


Fig. 5 Runtime comparison

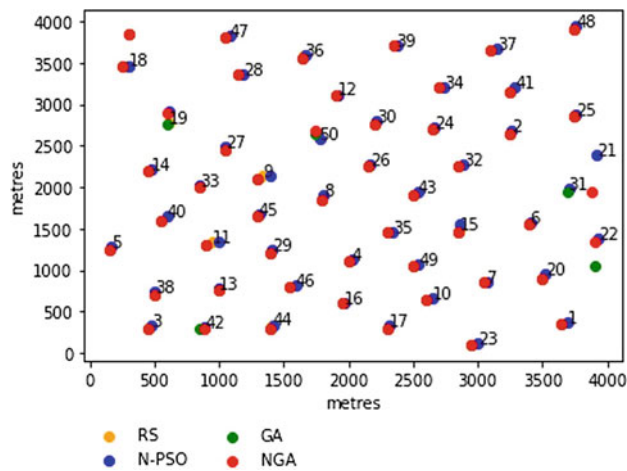


Fig. 6 Placement of wind turbines

7 Fault Diagnosis in Transmission Line

Transmission Lines are the veins of the electrical distribution system. They are responsible for bringing power from power plants to Industries, Hospitals and millions of domestic households. Any interruption in the flow of power can lead to massive losses of revenue, impact on household activities, reduced industry production. Future smart city would need to have a reliable transmission system that reduces chances of fault, while reducing dependence on manual labor. Machine Learning algorithms (Hasan et al., 2017) can quickly learn distinctive patterns in Transmission Line Telemetry to identify faults that may be caused due to asynchronization, earthing, short or breakage. Frequency analysis of Carrier Signals, can further help pinpoint the location of the fault. The dataset (kaggle) (<https://www.kaggle.com/esathyaprakash/electrical-fault-detection-and->

[classification](#); Jamil et al., 2015) used for training the ML models had a size of 7861 rows \times 10 columns, with 4 rows indicating the presence of error in Transmission line (Multiclass) A, B, C and Ground, and 6 columns showing the Current and Voltage flowing through them. A second dataset of size 12,001 rows \times 6 columns was used to train and identify whether the Transmission Line is faulty or not. The dataset can be understood from Table 4. The results of the Models trained for both cases are shown below in Tables 5 and 6 and Figs. 7 and 8.

A quick study of the results from Tables 5 and 6 and Figs. 7 and 8 show the general competency of Machine Learning Algorithms in classifying Faults in the Transmission Line. They can be used to judge the general health of the line (Table 5, Fig. 7) or predict and diagnose faults (Table 6, Fig. 8). We also observe that the performance of Boosted Trees and Naive Bayes Algorithm lead with sufficiently good results, with High Yield and Precision. Their good performance across both methods make them a good starting point for developing models that can diagnose, more complicated faults in very large systems. Here we have abstained from using Deep Learning Methods like Deep MLP and CNN- LSTM as mentioned in Fan et al., (2019).

8 Metrics

The following metrics are used for evaluating model performances.

$$Accuracy = \frac{\text{Number of Incorrect Classifications}}{\text{Total Number of samples}}$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$f1 \text{ score} = 2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$f\beta \text{ score} = \frac{(1 + \beta^2) \text{precision} \cdot \text{recall}}{\beta^2 (\text{precision} + \text{recall})}$$

$$MaxError = \max|\text{Actual Value} - \text{Predicted Value}|$$

$$MeanSquaredError = \frac{1}{n} \sum_0^n (y_i - \hat{y}_j)^2$$

$$R2 \text{ score} = 1 - \frac{\text{Sum of squares of residuals}}{\text{Total sum of squares}}$$

Table 4 Dataset transmission line

Property	Phase error dataset	Transmission line error dataset
Rows	7861	12,001
Columns	10	6
Features	Current (1) Channel A (2) Channel B (3) Channel C Voltage (1) Channel A (2) Channel B (3) Channel C	Current (1) Channel A (2) Channel B (3) Channel C Voltage (1) Channel A (2) Channel B (3) Channel C
Classes	(1) Line Grounded (2) Channel A Faulty (3) Channel B Faulty (4) Channel C Faulty	(1) Faulty (2) Not-faulty
Validation split	70–30	70–30

Table 5 Performance of models (Binary classification)

S. No.	Model	Accuracy	f1 score	f_{β} score	Precision	Recall
1	Logistic classifier	0.6957	0.5204	0.5204	1	0.3517
2	Decision tree	0.9851	0.9839	0.9839	0.9951	0.9729
3	Random forest	0.9856	0.9845	0.9845	0.9963	0.9729
4	Boosted trees	0.9867	0.9857	0.9857	0.9976	0.9741
5	Gaussian Naive Bayes	0.9789	0.9789	0.9793	0.9797	0.9790
6	Bernoulli Naive Bayes	0.5753	0.5617	0.5642	0.5710	0.5754
7	Multinomial Naive Bayes	0.6505	0.6502	0.6500	0.6500	0.6505
8	Complement Naive Bayes	0.6466	0.6465	0.6464	0.6464	0.6466
9	Boosted trees (XGBoost)	0.9938	0.9937	0.9938	0.9938	0.9938
10	Boosted trees (LightGBM)	0.9965	0.9965	0.9965	0.9965	0.9965
11	Boosted trees (CatBoost)	0.9953	0.9953	0.9953	0.9953	0.9953
12	Stochastic gradient descent	0.9892	0.9892	0.9892	0.9893	0.9892
13	Passive aggressive classifier	0.8815	0.8785	0.8895	0.9028	0.8815
14	Perceptron	0.9894	0.9894	0.9894	0.9895	0.9894

Table 6 Performance of models (Multiclass classification)

S. No.	Model	Accuracy	f1 score	f_{β} score	Precision	Recall
1	Logistic classifier	0.3449	0.1496	0.1496	0.5730	0.2114
2	Decision tree	0.8356	0.7938	0.7938	0.8072	0.7977
3	Random forest	0.8330	0.7968	0.7968	0.7978	0.7964
4	Boosted trees	0.8508	0.8182	0.8182	0.8189	0.8180
5	Gaussian Naive Bayes	1	1	1	1	1
6	Bernoulli Naive Bayes	1	1	1	1	1
7	Multinomial Naive Bayes	0.6430	0.6381	0.6560	0.6775	0.6430
8	Complement Naive Bayes	0.3450	0.2093	0.1800	0.4842	0.3450
9	Boosted trees (XGBoost)	1	1	1	1	1
10	Boosted trees (LightGBM)	1	1	1	1	1
11	Boosted trees (CatBoost)	1	1	1	1	1
12	Stochastic gradient descent	0.7157	0.6628	0.6360	0.6197	0.7157
13	Passive aggressive classifier	0.6389	0.5488	0.5121	0.4919	0.6398
14	Perceptron	0.5872	0.5540	0.5436	0.5389	0.5872

Fig. 7 Performance of models (Binary classification)

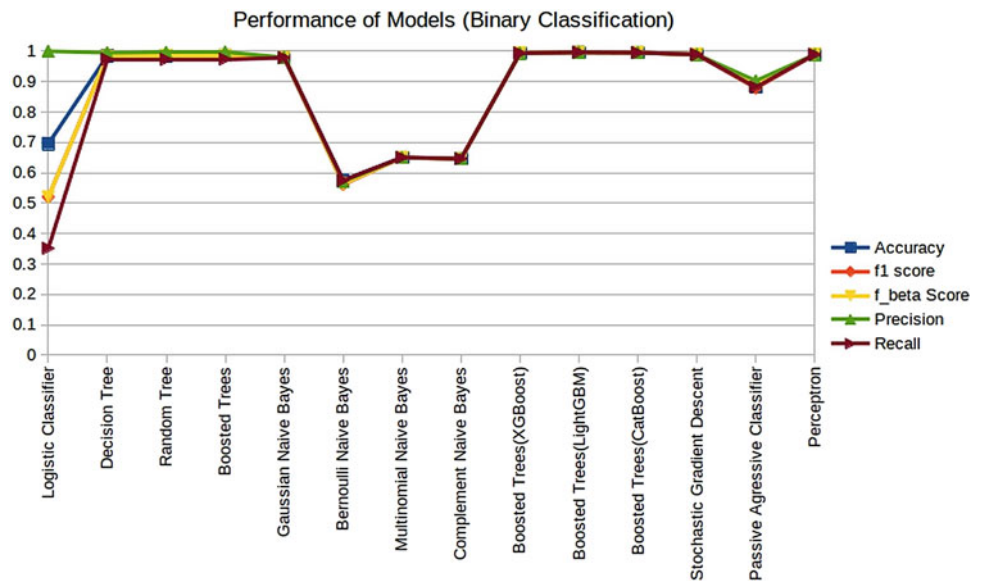
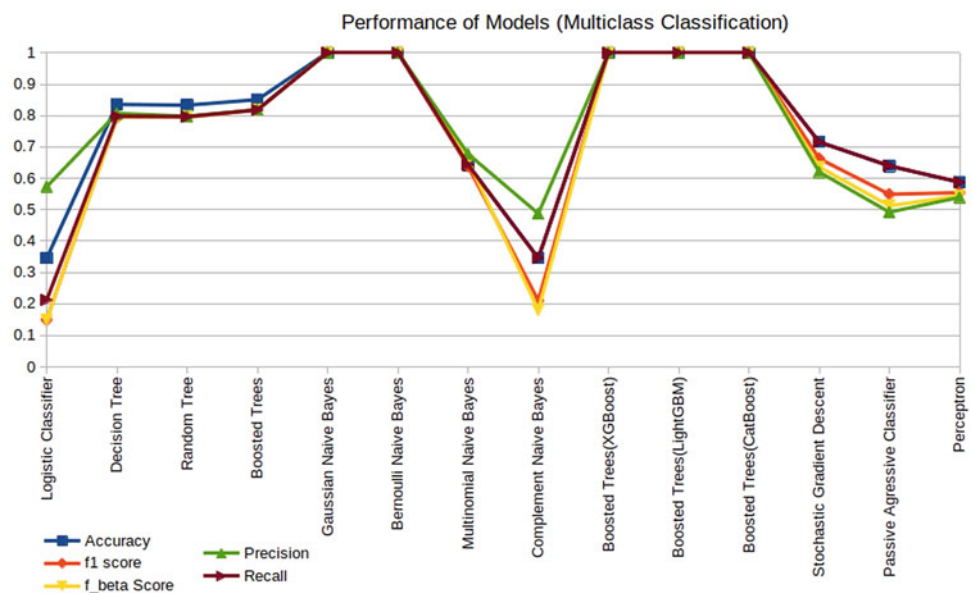


Fig. 8 Performance of models (Multiclass classification)



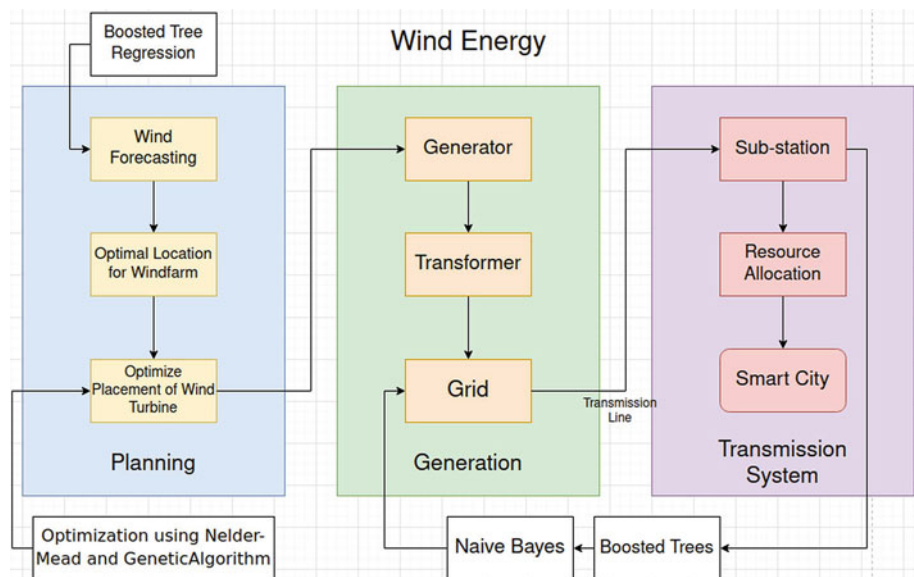
9 Results and Discussion

The wind forecasting dataset provided by Kaggle (<https://www.kaggle.com/natasha23/wind-forecasting-dataset>) was used to train and test the efficacy of 15 different machine learning algorithms, and the results were tabulated in Table 3 and Fig. 3. The best model from these 15 was found to be one made using Boosted tree method with max error of 0.7302, mean squared error of 0.0304 and R2 score of 0.6278. The tree architecture had 10 trees and maximum tree depth 6. Random Forest and Decision Tree also performed well giving similar results. The performance of Random Sample Consensus (RANSAC) model and Passive

Aggressive Regressive model was found to be the worst as indicated by their negative R2 score.

In WindFarm optimization problem, the initial dataset comprises of 50 wind turbines placed in a windfarm of dimension $4000 \times 400 \text{ m}^2$, with the minimum distance from boundary 50 m, and minimal feasible distance between windmills to be 400 m. The AEP of this dataset is 505.4506 GWh. The results of the four optimization methods are given below. Figure 6 shows the placement of windmills after optimization. The AEP obtained from Random Search is 507.2882 and the time taken for 50 iterations is 12.27 s. Of all the methods Random Search is the least computationally expensive as can be seen from Fig. 4. It also has the least slope i.e., it has the least change in AEP

Fig. 9 Wind energy: generation to supply



across iterations as seen in Fig. 5. The AEP obtained from Genetic Algorithm is 510.6519 and time taken for 50 iterations is 472.68 s. It is moderately expensive procedure, that requires conversion of Cartesian coordinates to binary chromosomes. The AEP obtained from Nelder-Mead PSO is 508.5639 and time taken for 50 iterations is 741.68 s. It has a high slope that is comparable to Nelder-Mead-Genetic Algorithm as seen in Fig. 4. The AEP obtained from Nelder-Mead-Genetic Algorithm is 512.0828 and time taken for 50 iterations is 320.47 s. Of all optimization techniques it yields the maximum AEP, as seen in Fig. 4, and is ranked 3rd in total time taken as in Fig. 5, making Nelder-Mead-Genetic Algorithm the preferred choice for optimization. The Genetic Algorithm had a crossover rate of 0.9 and mutation rate of 0.00016.

The performance of Classification Algorithms was considerably good at identifying and classifying faults in Transmission lines as is seen from Tables 5 and 6. Models based on Boosted Trees and its derivatives, Gaussian Naive Bayes, Bernoulli Naive Bayes were found to be extremely good at classification, scoring almost 100% in Accuracy, Precision and Recall. The boosted trees algorithm has 10 trees and maximum tree depth of 6. Figure 9 explains the Wind Energy Planning, Generation and Transmission System and the Machine Learning Models used for them.

10 Conclusion

The above work details the use of Machine Learning to develop a smart Wind Power Generation System to power smart cities. It explains the need for Wind forecasting and its uses to improve resource allocation. Several different

algorithms are compared and their metrics are shown for a uniform dataset. Boosted Tree classifier gives the best results with the lowest MSE. The work explains the optimization of windfarms using four different methods and compares their performance in terms of AEP and Time. From Figs. 4 and 5, it can be seen that the results obtained from Genetic Algorithm and Nelder-Mead-Genetic Algorithm are comparable in early iterations, with the growth in Nelder-Mead-Genetic Algorithm surpassing Genetic Algorithm at > 30 iterations. Random Search Algorithm yields the minimum change in AEP, but also consumes the least amount of time, approximately 40 times less. Nelder-Mead PSO, yields intermediate results, but is computationally very expensive. The change in AEP across iterations is similar in Nelder-Mead-Genetic Algorithm and Nelder-Mead PSO. A comparative study of Machine Learning Models in diagnosing and classifying faults in Transmission Lines is done in Sect. 6. The results obtained from Naive Bayes Algorithm and Boosted Trees show promise. Naive Bayes can be used for Multiclass classification whereas Boosted trees yield good results in binary classification. The results show the overall excellent performance of simple Machine Learning Algorithms in solving the problem.

References

- Bhardwaj, B., Jaiharie, J., Dadhich, R. S., Ahmed, S. I., & Ganesan, M. (2021). Windfarm optimization using Nelder-Mead and particle Swarm optimization. In *2021 7th international conference on electrical energy systems (ICEES)* (pp. 524–529). IEEE.
- Donovan, S. (2005). Wind farm optimization. In *Proceedings of the 40th annual ORSNZ conference* (pp. 196–205).
- Fan, R., Yin, T., Huang, R., Jianming Lian, and Shaobu Wang. (2019). Transmission line fault location using deep learning techniques. In *2019 North American power symposium (NAPS)* (pp. 1–5). IEEE.

- Feng, J., & Shen, W. Z. (2015). Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, 78, 182–192.
- Hasan, A. N., Eboule, P. S. P., & Twala, B. (2017). The use of machine learning techniques to classify power transmission line fault types and locations. In *2017 international conference on optimization of electrical and electronic equipment (OPTIM) and 2017 intl aegean conference on electrical machines and power electronics (ACEMP)* (pp. 221–226). IEEE.
- <https://powermin.gov.in/en/content/power-sector-glance-all-india>
- <https://smartcities.gov.in/>
- <https://www.datarevenue.com/en-blog/machine-learning-for-energy-transmission>
- <https://www.kaggle.com/esathyaprakash/electrical-fault-detection-and-classification>
- <https://www.kaggle.com/natasha23/wind-forecasting-dataset>
- <https://www.mapsofindia.com/government-of-india/smart-citiesproject.html>
- <https://www.mnre.gov.in/wind/current-status/>
- IEA, Power capacity in India by source in the Sustainable Development Scenario. (2000–2040). IEA, Paris. <https://www.iea.org/data-and-statistics/charts/power-capacity-in-india-by-source-in-the-sustainable-development-scenario-2000-2040>
- Jamil, M., Sharma, S. K., & Singh, R. (2015). Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus*, 4(1), 1–13.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (vol. 4, pp. 1942–1948). IEEE.
- Majid, M. A. (2019). Wind energy programme in India: Emerging energy alternatives for sustainable growth. *Energy and Environment*, 30(7).
- Miranda, L. J. V. (2020). Py Swarms documentation.
- Mouna, H., Mukhil Azhagan, M. S., Radhika, M. N., Mekaladevi, V., & Nir mala Devi, M. (2018). Velocity restriction-based improvised particle swarm optimization algorithm. In *Progress in advanced computing and intelligent engineering* (pp. 351–360). Springer.
- Park, J. W., An, B. S., Lee, Y. S., Jung, H., & Lee, I. (2019). Wind farm layout optimization using genetic algorithm and its application to Daeg-wallyeong wind farm. *JMST Advances*, 1(4), 249–257.
- Prabha, P. P., Vanitha, V., & Resmi, R. (2019). Wind speed forecasting using long short term memory networks. In *2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICT)* (vol. 1, pp. 1310–1314). IEEE.
- Raphel, D., & Vanitha, V. (2018). Seasonal wind speed forecasting using ANFIS. In *IEEE biennial international conference on power and energy systems: Towards sustainable energy (PESTSE)*.
- Samorani, M. (2013). The wind farm layout optimization problem. In *Handbook of wind power systems* (pp. 21–38). Springer.
- Shakoor, R., Hassan, M. Y., Raheem, A., & Wu, Y.-K. (2016). Wake effect modeling: A review of wind farm layout optimization using Jensen s model. *Renewable and Sustainable Energy Reviews*, 58, 1048–1059.
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)* (pp. 69–73). IEEE.
- Singer, S., & Nelder, J. (2009). Nelder-mead algorithm. *Scholarpedia*, 4(7), 2928.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., & Burovski, E. et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272.

Multi-class Segmentation of Trash in Coastal Areas Using Encoder-Decoder Architecture

S. Surya Prakash, V. Vengadesh, M. Vignesh,
and Satheesh Kumar Gopal

Abstract

Trash accumulation in beaches affects the ecosystem of coastal lines. Different types of trash can affect beaches in a variety of ways and manual identification of this trash might become laborious. So, it becomes important to devise a method to facilitate the localization of this trash without human intervention. In this paper, we propose a deep learning-based solution for multi-class segmentation of trash objects using Unmanned Aerial Vehicles. But the problem with the aggregated orthoimages is the low foreground-to-background ratio which tends to render very high false positives (FP) during classification. To counter this, we propose a random data generation method to generate synthetic data over real backgrounds. The best performing model among the chosen candidate architectures (U-Net and SegNet) on both the real and synthetic datasets are evaluated and results are compared using various segmentation metrics. Later the segmentation masks are transposed onto a map to facilitate localization.

Keywords

Semantic segmentation • Synthetic data • Trash detection • U-Net • SegNet • ResNet-50

S. Surya Prakash · V. Vengadesh · S. K. Gopal
Department of Mechanical Engineering, SSN College
of Engineering, Chennai, 603110, India
e-mail: suryaprakash040900@gmail.com
surya18173@mech.ssn.edu.in

V. Vengadesh
e-mail: vengateshvengadesh@gmail.com
vengadesh18180@mech.ssn.edu.in

S. K. Gopal
e-mail: satheeshkumarg@ssn.edu.in

M. Vignesh (✉)
Department of Electronics and Communication Engineering,
SSN College of Engineering, Chennai, 603110, India
e-mail: vignesh.desmond@gmail.com; vignesh18181@ece.ssn.edu.in

1 Introduction

Water bodies and human settlements go hand in hand as history points out. Beaches recently have been the only source of inspiration left out as most of the inland water bodies are drained or wiped out with the trash. This is specifically true for the urban citizens than for the others. Government efforts, NGOs, and volunteers have always directed their efforts to keep the beaches clean regularly and at various instances of time. Armed with the technologies of the yesteryears or sometimes not with appropriate technologies, the trash takes over the beaches in no time. An upgradation of technology is the need of the hour. Robotics, AI, IoT, and automation are the interdisciplinary research areas where the answers could be sought. Some problems are unique by their geographical location and beach debris is one such problem. Universally applied solutions may not work here as it is. One needs to create, validate and demonstrate an appropriate technology for the society to put it to use effectively.

Several land-based autonomous beach cleaning (Ichimura & Shin-ichi, 2016; Wattanasophon & Ouitrakul, 2014) and indoor cleaning robots (Pinheiro et al., 2015) are designed which employ varied terrain navigation and applications involving path planning. Also, we have robots that use the techniques of Machine and Deep Learning (Thiagarajan et al., 2019) to localize the beach debris and also engulf them thereby tidying the area. But one of the shortcomings of going with land-based localization and cleaning would be the time and inability of these multi-terrain vehicles to reach certain secluded spots (over the rocks on beaches and sometimes excavated spots), that is, the land-based rover systems might fail to analyze altitudes and depths which might lead to the missing or incorrect classification of beach debris. In these scenarios, aerial vehicles can come to aid. Many studies and research have been carried out over the years when people attempted to monitor the amount of coastal litter (Kako et al., 2018; Nakashima, 2011; Park &

Kang, 2005) and carried out the work of debris detection using the processes of image thresholding (Bao & Sha, 2018; Jang et al., 2011) and deep learning (Bak et al., 2019).

Object detection using Machine Vision has seen humongous growth over the past few years. Several object detection techniques have engaged the interests of many research scientists who constantly work towards the growth of Artificial Intelligence (AI) in the field of Autonomous Vehicles, Medical Image Analysis, etc. Two broad divisions of Object detection namely Bounding Box detection and Semantic Segmentation are continuously being researched and improved. Semantic Segmentation focuses on the aspect of pixel-level prediction of an image which facilitates the class-wise clustering of pixels (Fig. 1).

Later, the predicted image and the ground truth image will be compared using the metric called Intersection over Union (IoU) which would give an insight into the goodness of the fit. The objective of this research can be decomposed into two major parts,

- Development of a deep learning model which will enable the semantic segmentation of coastal debris.
- Georeferencing and overlay of segmentation masks over a satellite map.

The two segmentation architectures used in this research namely U-Net (Ronneberger et al., 2015) and Seg-Net (Badrinarayanan et al., 2017) will be discussed in the later sections.

2 Proposed Methodology

The methodology proposed for multi-class segmentation and localization of beach debris is given in Fig. 2.

Fig. 1 Sample image (*left*) and segmentation label (*right*) (<https://www.tugraz.at/index.php?id=22387>)



3 Data

3.1 Study Area

The area of study for this research is chosen as Ennore Beach, Tamil Nadu, in Southern India. This study area was chosen because of the presence of enormous amounts of debris. This location has the coordinates of latitude and longitude as $13^{\circ}12' 23.4864''$ N and $80^{\circ}19' 38.0100''$ E respectively.

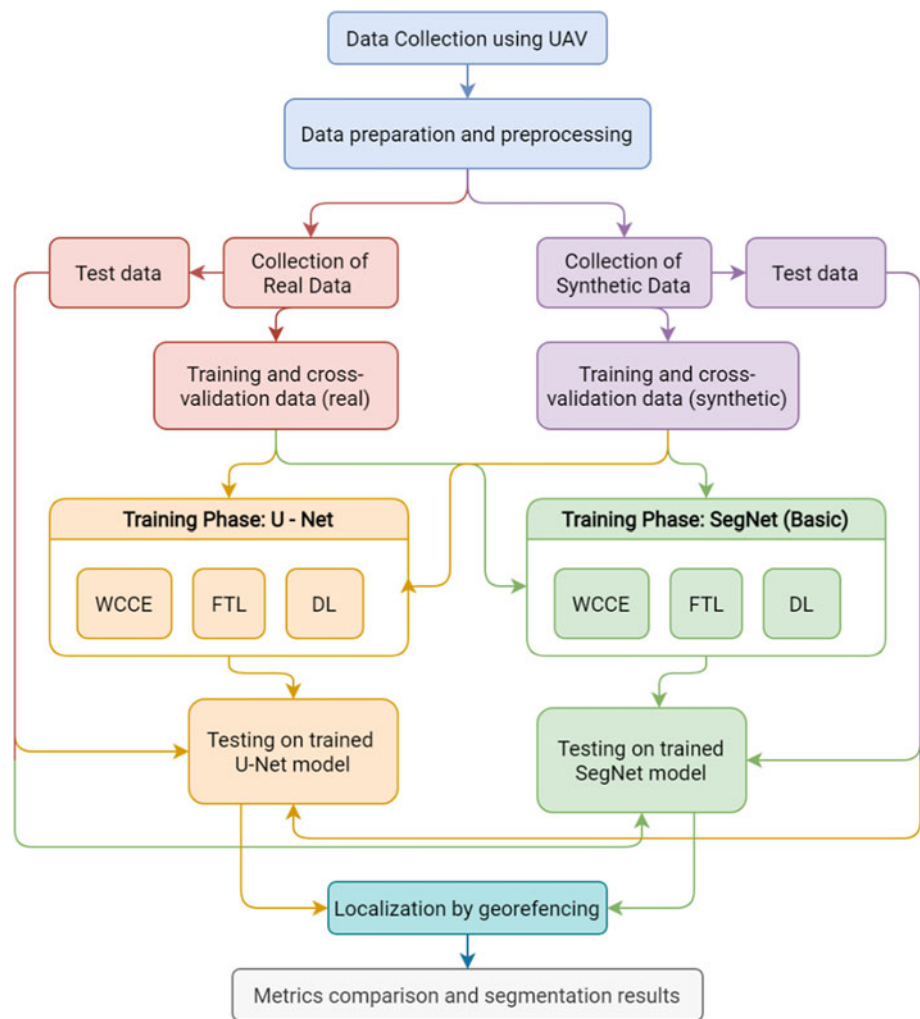
3.2 UAV Details

The UAV used for this task is DJI Tello with a front-facing camera of a resolution of 1280×720 p at a Field of View (FOV) of 82.6° . The drone has four propeller blades with a maximum flight height of 15 m. The drone has a 5 MP camera sensor. Aerial videos are taken only during daylight at 30 frames per second at a height of 5 m from which around 600 frames were extracted.

3.3 Real Data

Real data (<https://github.com/SrihariM1809/beach-garbage-dataset>) is the collation of images with little or no modifications to the pixel-level information. From the recorded video, a total of 387 frames are validated and picked. Some images containing empty backgrounds are also picked and kept aside for synthetic data generation. These images were manually annotated based on the below-listed classes. The aggregated data are thoroughly analyzed and a total of 7 different classes were picked based on the frequency of occurrence which is listed below,

Fig. 2 Flowchart of the proposed methodology



- Beach (Background)
- Foam
- Wood and Plants
- Clothes
- Plastic and Polymers
- Other objects
- Other backgrounds.

3.4 Synthetic Data

One of the important concerns which arise during the aggregation of real data is the dominance of beach (background) pixels over the class (foreground) pixels. This inherently posits an imbalance of collected data which could greatly affect the training. Various oversampling techniques prove to be effective in this case (Chawla et al., 2002; Ho & Samuel, 2020)

¹ The distribution of classes is given in numbers and is not to be confused with pixels.

which could evade this problem. Oversampling techniques like the weighted class approach are also employed widely. Looking closely at the dataset, we can find that the physical and geometrical nature of the classes is highly inconsistent and this may lead to subpar segmentations. To solve this, we propose the generation of synthetic data.

Generating synthetic data includes major modifications to the nature of distribution, frequency of debris, and characteristics of the debris classes. In this process, we attempt to manually place trash objects over the empty beach backgrounds which were collected earlier to counter the low foreground-to-background pixel ratio. For the construction of this dataset, we use the extracted trash object images from the TrashNet (<https://github.com/garythung/trashnet>) dataset (Figs. 3 and 4).

TrashNet consists of around 2500 images of trash spanning over 6 different classes that are captured over varied lighting conditions and constant background. For extraction, this dataset is again manually annotated and three classes of debris are chosen which are listed below,

Fig. 3 Sample image from real data (left) and its segmentation label (right)



- Glass
- Metal
- Plastic.

These extracted images are then placed over the empty beach backgrounds which were collected during the real data preparation process using two different routines which are discussed in the following sections (Fig. 5).

3.5 Randomly Generated Data (RGD)

This object placement mechanism involves the random placement of trash objects with no constraints except for the out-of-bounds constraint. The extracted images from the TrashNet dataset are randomly placed over the empty beach background images and a total of 620 images are generated using this process. The segmentation masks are also

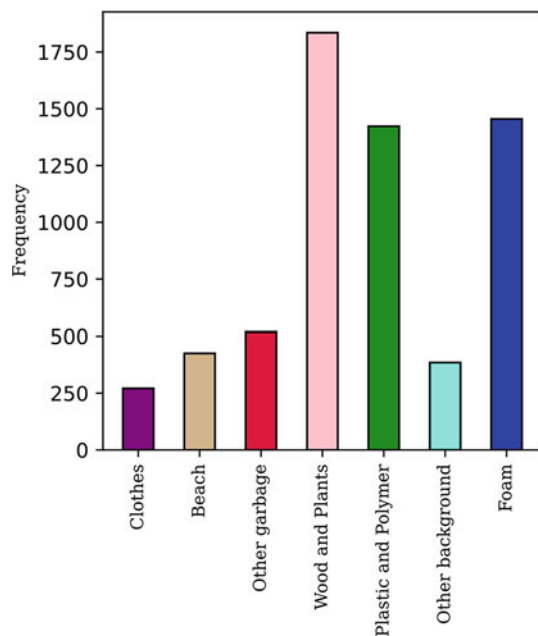


Fig. 4 Distribution of classes over the entire real dataset¹

simultaneously generated during this process. One of the disadvantages of this mechanism is that it does not take into consideration the nature of the background. Hence, any objects which are placed over backgrounds other than beach backgrounds are selectively discarded (Figs. 6 and 7).

Each generated image has a total of 5 different classes which are mentioned below,

- Beach
- Other Background
- Glass
- Metal
- Plastic.

4 Data Preprocessing

All the image data were resized down to 512×512 and then fed into the model for training. No other image augmentation or image manipulation techniques were employed during this phase. For the baseline FCN-32 ResNet test, 320×320 images were used.

5 Deep Learning and Segmentation

Deep learning is one of the major subfields of machine learning, and neural networks (NN) are the current standard. The physics of neural networks mimic the Central Nervous System of the human body. Like the neurons, which transmit electrochemical stimuli through their dendritic fractals to the other neurons for perception, neural networks try to develop a comprehension of the data that is being fed. Deep Neural Network (DNN) architectures have three unique structures namely the input layer, the hidden layers, and the output layer. And each layer possesses several neurons or nodes which perform a specific task upon activation. Each layer also has non zero randomized weight vector which gives importance to specific data. The performance of a DNN is governed by various parameters like the number of hidden

Fig. 5 Sample images from the TrashNet dataset along with the corresponding segmentation masks



Fig. 6 Sample image from RGD (left) and its segmentation label (right)

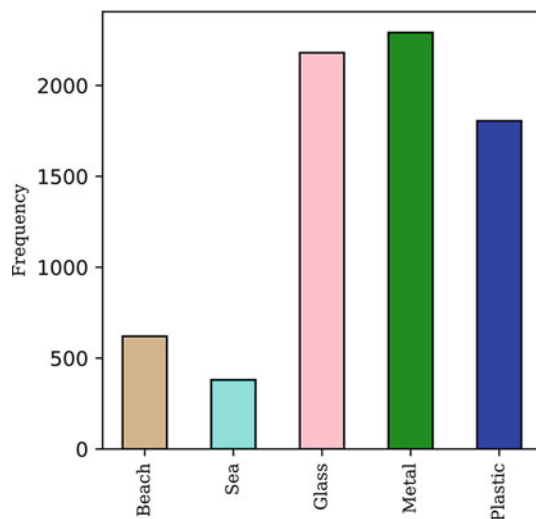


Fig. 7 Distribution of classes for RGD

layers, nodes in each layer, and the activation function that is used to trigger a particular neuron.

The DNN works in two phases namely forward propagation and backward propagation. The forward propagation or pass is the phase where multidimensional data is fed into the input layers and it propagates through the hidden layers and finally reaches the output layer to carry out suitable tasks (regression or classification). During this phase, the neurons in each layer learn about the data that is being fed and tries to reflect upon the same when tested. The second phase is the backpropagation phase where the weights corresponding to each hidden layer are updated to minimize the total loss.

These phases repeat for a fixed number of iterations until a minimal loss value is obtained.

Convolutional Neural Network (CNN or ConvNet) is a type of DNN used mainly for visual data. It consists of convolutional layers which perform convolution with filters or kernels to generate feature maps. The repeated systematic application of such kernels results in the enhancement of interesting features present in the image, which can then be used for classification, detection, segmentation, and so on. In CNN, the filter values act as the weights and are updated every epoch in the training phase, similar to DNN. There are many architectures proposed over the years, most successful being AlexNet, LeNet, VGG, Inception, and ResNet. However, many new architectures are proposed since then and each has its purpose and advantages. An important application of CNN is semantic segmentation, where each pixel of the image is classified, thus partitioning the image into distinct regions representing their respective classes. There are many traditional methods for semantic segmentation, making use of spatial and spectral features of the image. However, CNN's rich understanding of scene context and spatial invariance led to its dominance in the field of segmentation.

6 Encoder-Decoder Architectures

Encoder-Decoder Architectures are widely used in many areas of deep learning for mapping or associating inputs to certain outputs, namely in sequence transduction and inverse

problems such as super-resolution and image denoising. The encoder is the part of the network that receives the input and outputs a feature map or tensor. These feature maps retain the essential information from the input in the context of a given problem. The decoder is the part of the network that upscales the feature map to give the desired output.

In the context of semantic segmentation, encoder-decoder networks are supervised as the images are compared with corresponding segmentation masks. The weights are updated by using a loss function, of which there are many. For this research, we decided to explore 3 loss functions and study their results.

Usually, the encoder and the decoder are based on a pre-existing CNN architecture, known as a backbone network. VGG, ResNet, Inception are some of the widely used backbone networks. For this study, we will use ResNet-50 as our backbone network, discussed in the following section.

6.1 ResNet—50 (Backbone)

ResNet-50 (He et al., 2016) is a smaller version of the baseline ResNet-152, containing 48 convolutional layers. ResNet is a short name for Residual Network, as it introduced the concept of Residual Learning (Fig. 8).

ResNet is a powerful architecture due to the presence of residual skip connections (see Fig. 9). In CNNs with lots of layers (known as deep CNNs), the back-propagated values required to change the weights, which arises from the loss function calculated at the end of the network, becomes very small at the initial layers. This is known in the literature as the vanishing gradient problem. Another problem is that adding layers increases the parameter space and such makes the model find appropriate weights much harder. This is

known as the degradation problem. In ResNet, the skip connections let the layers fit a residual mapping $g(x)$ and the nonlinear layers fit another mapping $f(x) := g(x) - x$ thus making the original mapping $g(x) := f(x) + x$. This addition of identity mapping solves the vanishing gradient problem.

In all variants of ResNet, there is a Max Pooling layer present after Conv1 (first convolutional layer). Max Pooling is a down-sampling strategy used in CNNs to reduce parameter space and induce robustness to variations. It is also used to achieve translational invariance to small shifts in the input image. ResNet uses 3×3 Max Pooling with stride 2, which means the maximum value out of a 3×3 window is chosen and the window slides across horizontally and vertically across the image, 2 pixels at a time. For a $7 \times 7 \times 64$ input, after max pooling, we get a $3 \times 3 \times 64$ tensor as the output which has far fewer parameters, as shown in Fig. 10.

Similarly, in all variants of ResNet, there is a Global Average Pooling (GAP) layer before the dense fully connected layers. However, while using ResNet as a backbone, the GAP layer is dropped.

The backbone network is initialized with weights pre-trained on the ImageNet dataset, a dataset containing over 14 million images consisting of everyday objects divided into more than 1000 classes. The rationale behind training the model on top of weights from a pre-trained model instead of training the model from scratch is as follows (He et al., 2019). Pretrained models like ImageNet are much better at detecting features in images such as faces and edges compared to models trained with randomly initialized weights. A model trained on a large enough dataset like the ImageNet serves as a generic model for any visual classification. The pre-trained weights also provide some regularization effect. It also saves a lot of computations and training

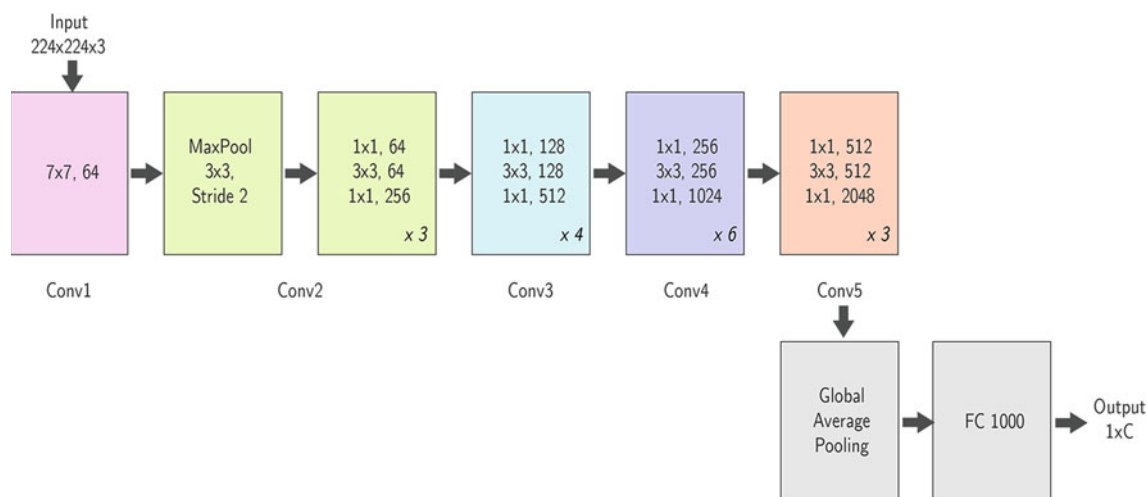


Fig. 8 Simplified representation of the ResNet—50 architecture. For a backbone network, only the Conv(1–5) blocks are used (colored) and other layers are dropped (grey)

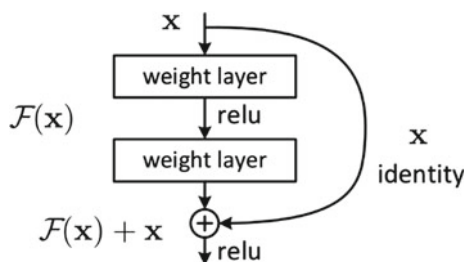


Fig. 9 Representation of residual skip connection

1	2	1	3	4	1	1	4	4	5
1	2	4	1	2	2	1	6	7	5
3	3	1	4	1	5	1	5	7	3
1	5	6	1	4	4	1			
1	4	1	7	1	2	3			
5	2	2	1	1	1	2			
1	2	2	1	1	3	3			

Fig. 10 Visualization of 3×3 max pooling with stride = 2 for a 7×7 image

time compared to training the model from scratch, as these networks have millions of parameters. For this study, no input layers are frozen and all layers are set as trainable layers.

6.2 U-Net

U-Net is an encoder-decoder architecture originally proposed for biomedical image segmentation but later found use in many other fields due to its highly accurate predictions with very little training data. In U-Net, the feature maps obtained at the layers of the encoder are used to reconstruct the required data when upsampling at the decoder (Fig. 11).

The U-Net architecture is shaped like a U, justifying its name. The original U-Net architecture has blocks of repeated 3×3 convolutions followed by a ReLU (Rectified Linear Unit) activation function and a 2×2 Max Pooling layer with stride 2. The reverse happens at the decoder, where 2×2 up-convolution takes place that reduces the number of channels by 2. Additionally, the output at each level of the encoder block is concatenated with the decoder block. Note

that the output image is not the same dimensions as the input image.

For our research, we use ResNet-50 as our backbone and so modifications must be made. In this modified U-Net, the input image tile is of size 512×512 and the output of each ResNet block is concatenated with the ResNet block in the decoder where successive convolutions and up-convolutions are made.

6.3 SegNet

SegNet is designed to be an efficient encoder-decoder architecture for semantic segmentation. It was primarily designed for road scene segmentation, but like U-Net, found its way into other fields. Similar to U-Net, SegNet has a backbone encoder network. The original authors used VGG-16 as their backbone (Fig. 12).

In SegNet, each convolution is followed by a ReLU activation function and a 2×2 Max Pooling layer with stride 2, similar to U-Net. Generally, successive max-pooling results in loss of spatial resolution in the feature maps, and thus boundary accuracy is reduced. This can be mitigated if the information is conserved during subsampling. In U-Net, the entire feature maps are concatenated before subsampling. However, this is not memory efficient. Instead, only the max-pooling indices can be sent to the decoder before subsampling, where upsampling would now take place with obtained indices.

For this research, we will be using ResNet—50 as the backbone for SegNet which allows for a more standard way to compare with U-Net. Since ResNet—50 has no max-pooling layers, we make use of SegNet-Basic, a variant of SegNet which has the same encoder network but has a decoder network akin to FCN. SegNet has considerably fewer parameters than U-Net as feature maps are not concatenated in the decoder, and thus require lesser training data than U-Net. The original SegNet had an input image tile size of 360×480 , but we will use a tile size of 512×512 .

7 Loss Function

To train the encoder-decoder network, we require loss functions that can mathematically represent the delta between the predicted segmentation and the ground truth mask. There are many widely used loss functions for semantic segmentation (Jadon, 2020). For this research, we'll evaluate the performance and metrics of the networks when trained using the following loss functions.

Fig. 11 U-Net architecture

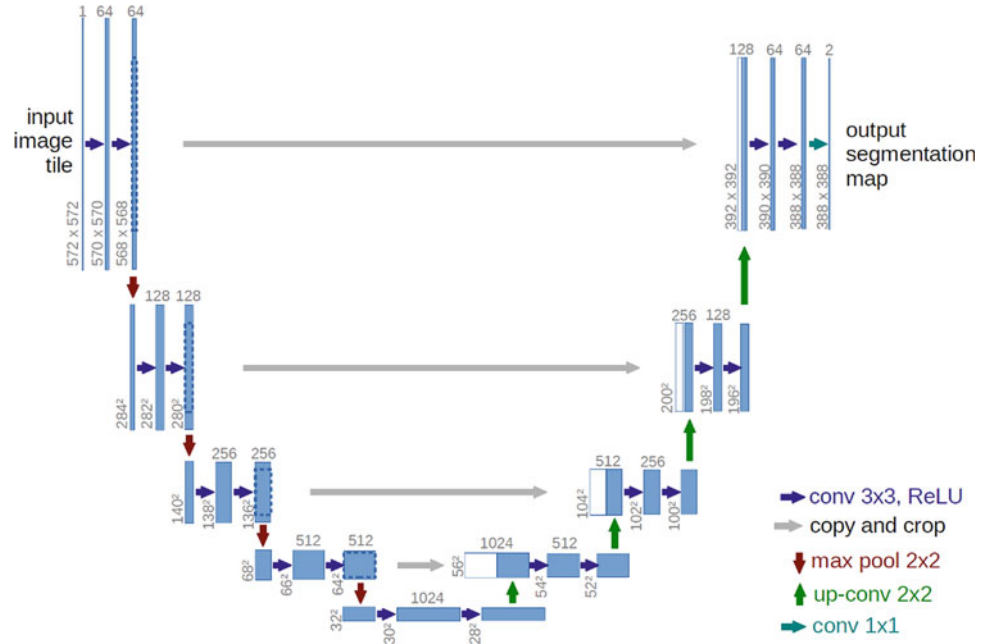
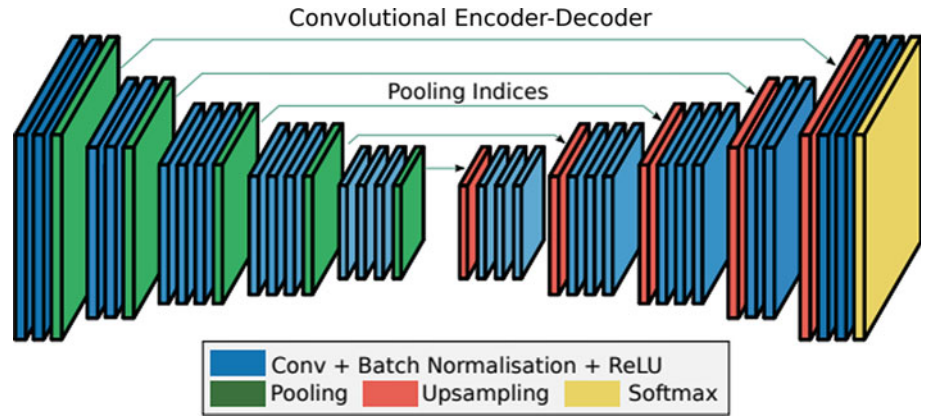


Fig. 12 SegNet architecture



7.1 Weighted Categorical Cross-Entropy

Weighted Categorical Cross-Entropy Loss (WCCE) (Pihur et al., 2007) is a variant of the more general Categorical Cross-Entropy Loss (CCE). CCE, also known as SoftMax Loss. CCE is defined as follows:

$$L_{CCE}(y, \hat{y}) = - \sum_{i=0}^C y_{o,i} \log \hat{y}_{o,i} \quad (1)$$

where C is the number of categories and o is the current observation (i.e., pixel). Note that the actual loss is the average of the L_{CCE} value for all pixels. In the case of WCCE, each category is assigned a weight ω_i , which can be calculated naively from the distribution of pixels per each category in the entire dataset, and can be modified based on trial and error.

$$L_{WCCE}(y, \hat{y}) = - \sum_{i=0}^C \omega_i y_{o,i} \log \hat{y}_{o,i} \quad (2)$$

Usually, the weights are assigned in such a way that the model is forced to learn harder examples and not get biased towards background classes which are usually a lot easier to learn.

7.2 Dice Loss

Dice loss (Sudre et al., 2017) is based on the well-known Dice similarity index (also known as Dice coefficient). Dice coefficient is used in various computer vision models to calculate the similarity between predicted and ground truth labels. The dice coefficient is given as follows:

$$D(y, \hat{y}) = \frac{2y\hat{y}+1}{y+\hat{y}+1} \quad (3)$$

where y is the ground truth and \hat{y} is the prediction. Here, 1 is added to both numerator and denominator so that the coefficient is not undefined when $y = \hat{y} = 0$. The dice loss is simply the complement of dice coefficient to the maximum value 1 and is defined as:

$$L_{\text{DICE}}(y, \hat{y}) = 1 - D(y, \hat{y}) \quad (4)$$

Similar to CCE and WCCE, the actual loss is the average of the L_{DICE} value for all pixels. Note that the dice loss is binary and thus for multi-class classification, the loss can be calculated per class and summed up.

7.3 Focal Tversky Loss

Focal Tversky Loss (Abraham et al., 2019) is a combination of Focal Loss (Lin et al., 2020) and Tversky index (Salehi et al., 2017). Focal loss is a special loss function designed to tackle the class imbalance problem, where easy negative classes like the background class are present in abundance. It introduces 2 parameters α and γ to better control the training of the model. α refers to class weights that are assigned to each class and γ is a focusing parameter that smoothly down-weights the easier examples and focuses the model on the harder ones.

$$L_F(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (5)$$

Here p_t is the ground truth probability and $(1 - p_t)^\gamma$ is known as the modulating factor, which plays a main role in countering class imbalance. Note that this loss is defined only for binary classification. Tversky Index is a generalization of the Dice coefficient seen earlier. It adds weight to the false positives and false negatives using a parameter β .

$$TI(y, \hat{y}) = \frac{y\hat{y}}{y\hat{y} + \beta(1-y)\hat{y} + (1-\beta)y(1-\hat{y})} \quad (6)$$

If $\beta < 0.5$, then the index decreases more with false negatives, thereby countering class imbalance. If $\beta = 0.5$, then the index reduces to the familiar Dice coefficient. Focal Tversky loss is the attempt to combine the advantages of both Focal loss and the Tversky Index to make the model focus on harder-to-learn examples with very small segmentation masks w.r.t background class.

$$L_{FT}(y, \hat{y}) = \sum_{i=0}^C (1 - TI(y, \hat{y})_i)^\gamma \quad (7)$$

8 Evaluation Metrics

To evaluate the performance of any model, metrics are necessary. In the context of segmentation, the metric should represent the pixel-wise accuracy and should be maximum

when the predicted and ground truth labels overlap. Two metrics are widely used when evaluating segmentation:

- IoU (Intersection over Union), also known as Jaccard Index
- Dice Coefficient, also known as Sørensen–Dice coefficient.

8.1 IoU

Intersection over Union, or simply IoU, is a well-known and widely used metric to evaluate the overlap between 2 objects. Assuming P and Q are two segmentation masks we need to compare, the IoU is calculated as follows:

$$\text{IoU}(P, Q) = \frac{\|P \cap Q\|}{\|P \cup Q\|} \quad (8)$$

Here, \cap and \cup refer to the intersection and union operations and $\|x\|$ refers to the norm of x . In segmentation, the norm simply refers to the area covered in pixels. IoU is a very simple and intuitive yet accurate metric to evaluate segmentation. The IoU metric can also be rewritten to show its relation to true and false detections.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (9)$$

Here:

- TP = True Positive
- FP = False Positive
- FN = False Negative.

In multi-class segmentation, the IoU is calculated for each class (i.e., class-wise IoU) and averaged over all classes. This is known as mean IoU or simply mIoU and is the base metric for the evaluation of many segmentation models.

8.2 Dice Coefficient

We have already seen the definition and use of the Dice coefficient in dice loss (Sect. 7.2). This section mainly aims to add to the previous section by comparing with IoU. Given masks P and Q , the dice coefficient is calculated as follows:

$$\text{Dice}(P, Q) = \frac{2\|P \cap Q\|}{\|P\| + \|Q\|} \quad (10)$$

The dice coefficient has an interesting property. In the literature of machine learning, precision and recall are two important metrics in classification. Precision and Recall are defined as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

If we take the harmonic mean of precision and recall, we obtain the dice coefficient as follows.

$$\text{Dice} = \left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}} \right)^{-1} = \left(\frac{TP+FP}{TP} + \frac{TP+FN}{TP} \right)^{-1} \quad (13)$$

$$\text{Dice} = \frac{2TP}{2TP+FP+FN} \quad (14)$$

Similar to IoU, the dice coefficient is calculated per class and then averaged over all classes, known as mean Dice. Both IoU and Dice coefficient is bounded in the range [0, 1], 0 when the masks have no overlap and 1 when the masks have perfect overlap. IoU tends to penalize trivial mistakes more than Dice. It can be shown that

$$\frac{\text{Dice}}{2} \leq \text{IOU} \leq \text{Dice} \quad (15)$$

From this, we can see that IoU is always lower than the Dice coefficient. So, it can be seen that IoU is an indicator for worst-case performance, and the Dice coefficient is an indicator for average performance.

9 Training

Training of the model is done in three phases. During the first phase of training the proposed network architectures are trained over the real data using the three mentioned loss functions concurrently and the results are logged. Similarly, in the next phase, the networks are trained over the synthetic data and final results are collected. The accumulated results are then compared using the discussed metrics and the predicted segmentations are visualized.

The training is carried out on the Kaggle platform which offers a weekly GPU quota. The GPU used is Nvidia Tesla P100 packed with a GPU memory of 16 GB @ 1.32 GHz. The available physical memory is around 12 GB with a disk space of 5 GB.

TensorFlow and Keras framework was chosen to carry out the training. The code used for the experiment is a modified version of the image-segmentation-keras library (<https://www.tugraz.at/index.php?id=22387>). Throughout the entire training process, the setting of model hyperparameters was undisturbed to achieve consistency with the results. So, Adam is chosen as the optimization routine. Learning parameters like the learning rate is fixed at 10^{-3} , and the momentum parameters are fixed at $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the value of $\epsilon = 10^{-7}$. A batch size of 16 is fixed and the number of

steps per epoch is also fixed at 32. No other callbacks like Early Stopping and Reduce Learning Rate on Plateau are employed. For WCCE, the class weights for real data are given in Table 1 and the class weights for synthetic data are given in Table 2. For Focal Tversky Loss, $\beta = 0.3$ and $\gamma = 1$ are fixed throughout the training. Results like train and validation accuracy, loss, and mIoU are obtained at the end of every epoch. A constant train-validation-test data ratio of 0.7:0.25:0.05 is maintained throughout the entire training i.e., 270, 96, 21 images from the real dataset, and 434, 108, 78 images from the synthetic dataset are used for training, validation, and testing respectively. Each training step was carried out for a total of 50 epochs.

10 Results

To visualize the rate of convergence and its nature for any model, it is common practice to plot the metrics against the epoch for both training and cross-validation datasets. Figures 13, 14, 15 and 16 show the metrics (Accuracy and mIoU) versus epoch for two models: U-Net and SegNet, each trained with three losses: Weighted Categorical Cross-Entropy (WCCE), Dice Loss (DL), and Focal Tversky Loss (FTL).

From the plots, it can be inferred that WCCE loss struggles to learn the hard examples and the mIoU plateaus in all cases. For all the loss functions tested, training accuracy increased within a small number of epochs and was consistently above 90%. However, the training mIoU for the loss functions shows a steady linear increase. Due to the inherent imbalance of pixels per class, easier examples contribute to most of the accuracy (calculated pixel-wise) whereas mIoU provides more insight into how the model learns less frequent classes with low pixel footprint. FTL's training mIoU is better than other losses in the real dataset, demonstrating the novel loss's use of modulating factor gamma pushing the model to learn harder examples. The distinction between FTL and DL is much less in the synthetic dataset, where the dataset is slightly more balanced. The validation graphs reinforce the points made above. Occasional dips in the validation graphs can be simply attributed to the randomness of the train-validation split. Both U-Net and SegNet have similar learning trends and only differ in the metric values.

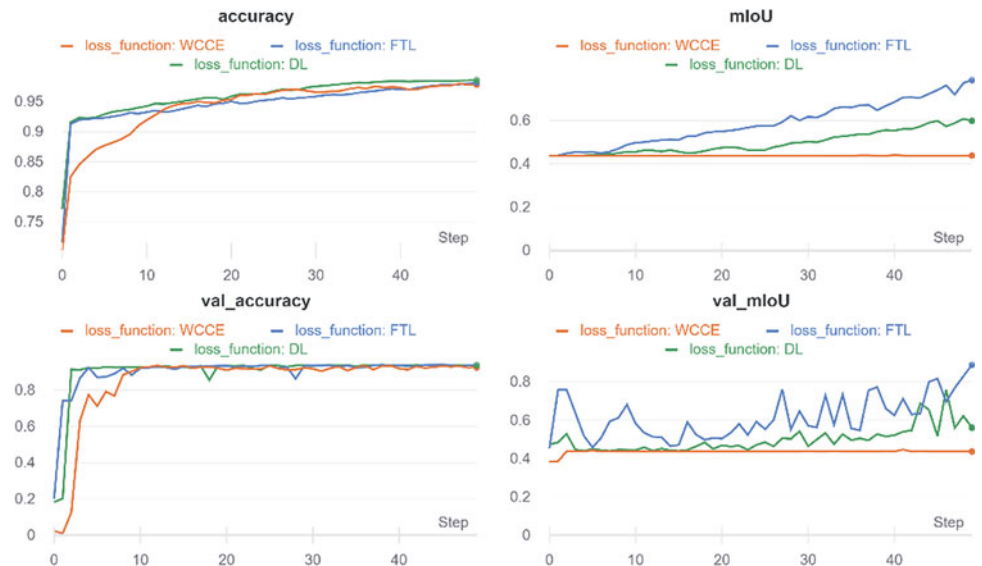
Comparing the tables for real and synthetic data (Tables 3 and 4), we can interpret that the mIoU values are significantly higher for synthetic than real and this can be attributed to two facts. One is the prevalence of class imbalance in real data than in synthetic data. And the other reason is due to the presence of a smaller number of classes in synthetic data than in real data, inherently making the segmentation task easier.

Table 1 Class weights for real data used for WCCE

Class	Class weight
Beach	0.005
Other background	0.005
Wood	0.18
Plastic and polymer	0.18
Foam	0.18
Clothes	0.18
Other	0.17

Table 2 Class weights for synthetic data used for WCCE

Class	Class weight
Beach	0.005
Other background	0.005
Glass	0.326
Metal	0.330
Plastic	0.334

Fig. 13 Training and validation graph for U-Net (Real)

11 Localization

After we obtain the model segmentation for a given input image, localization can be done by doing an affine transformation of the segmentation from the local frame of reference of the UAV to a global frame of reference. This is known in the literature as georeferencing. To do so, we would require information about the camera specifications, UAV's flying height, and the angle of inclination if any. These values can be used to obtain the corner coordinates of an image.

² Base map: "13°13' 13.8648" N, 80°19' 35.4426" E. Ennore Region, Tamil Nadu, India" Google Maps, Google.

In general, it is a difficult task to accurately localize multiple images due to the difficulty involved in obtaining the corner coordinates due to the complexity of transformation, first obtaining the image's central coordinates and then the corner coordinates given the parameters mentioned above. A simpler alternative would be to mosaic all images taken from the drone, and then localize the single large image. Figure 17 shows the image mosaic of segmentation outputs of 24 images taken from the UAV at the study area, overlaid on top of satellite map taken from Google Maps. These georeferenced segmentations can then be used for beach litter monitoring and cleaning purposes.

Fig. 14 Training and validation graph for SegNet (Real)



Fig. 15 Training and validation graph for U-Net (Synthetic)



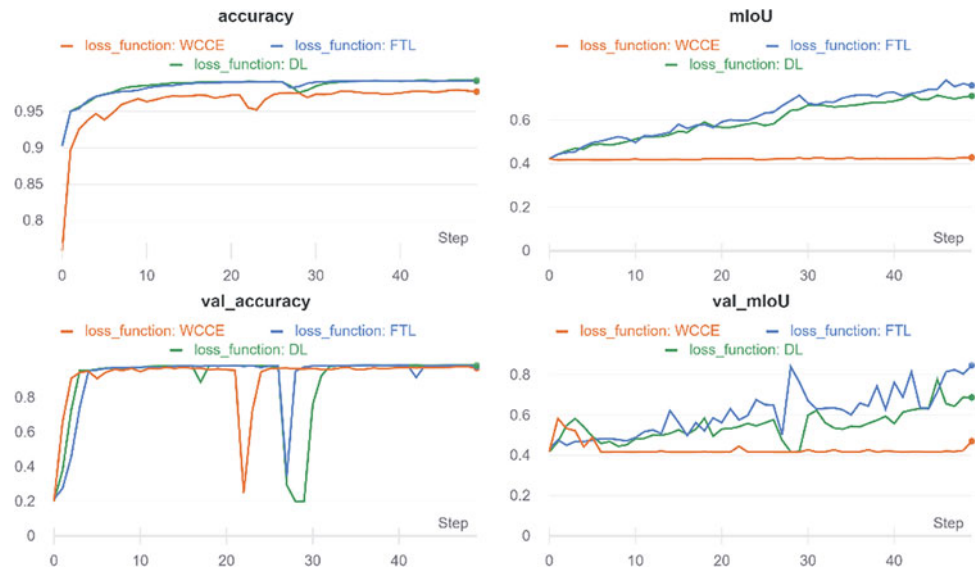
12 Discussion

Inferences from both graphs and prediction metrics give us some insight into the performance of chosen architectures and loss functions, along with the type of data used.

12.1 Architecture

Looking at the results from a prediction on test data, U-Net has managed to outperform SegNet in almost all cases. This can be asserted since the decoder levels of the proposed SegNet architecture consisted of upsampling layers rather than reusing the corresponding Max Pooling indices from

the encoder. This version of SegNet is also termed SegNet-basic as seen before. Both models' performance is considerably better on synthetic data than real data. For comparative analysis, a Fully Connected Network (FCN-32) with ResNet-50 backend (Shelhamer et al., 2017) was chosen as a baseline metric. FCN is one of the first encoder-decoder architectures with wide usage in multiple fields of segmentation. However, it suffers from a myriad of faults due to a lot of parameters in the network making it harder to learn over small datasets. It also requires a lot of memory to train, whereas our candidate networks do not suffer from out-of-memory issues. For our comparative analysis, we used FTL as the loss function for training FCN-32 and other hyperparameters the same as before. The results obtained from the baseline show much poorer results

Fig. 16 Training and validation graph for SegNet (Synthetic)**Table 3** Evaluation metrics for real dataset

Model	mIoU	mDice	Avg	Class wise IoU					Class wise dice				
				W	P	F	C	O	W	P	F	C	O
U-Net (WCCE)	0.413	0.522	0.923	0.442	0.295	0.208	0.261	0.297	0.613	0.455	0.344	0.414	0.458
SegNet (WCCE)	0.407	0.514	0.919	0.399	0.290	0.174	0.335	0.248	0.570	0.450	0.297	0.503	0.398
U-Net (DL)	0.427	0.536	0.883	0.431	0.322	0.171	0.319	0.339	0.603	0.487	0.292	0.484	0.506
SegNet (DL)	0.407	0.515	0.879	0.368	0.231	0.288	0.321	0.222	0.537	0.375	0.448	0.487	0.364
U-Net (FTL)	0.412	0.516	0.881	0.429	0.236	0.155	0.348	0.290	0.601	0.382	0.269	0.517	0.450
SegNet (FTL)	0.409	0.515	0.880	0.372	0.234	0.125	0.340	0.295	0.577	0.380	0.245	0.511	0.459

Table 4 Evaluation metrics for synthetic dataset

Model	mIoU	mDice	Avg	Class wise IoU			Class wise dice		
				Glass	Metal	Plastic	Glass	Metal	Plastic
U-Net (WCCE)	0.679	0.744	0.983	0.688	0.711	0.740	0.815	0.831	0.851
SegNet (WCCE)	0.629	0.710	0.972	0.596	0.615	0.684	0.746	0.762	0.812
U-Net (DL)	0.689	0.750	0.988	0.705	0.715	0.735	0.827	0.834	0.847
SegNet (DL)	0.660	0.730	0.986	0.630	0.664	0.693	0.773	0.798	0.819
U-Net (FTL)	0.692	0.751	0.989	0.703	0.722	0.744	0.826	0.838	0.853
SegNet (FTL)	0.598	0.681	0.980	0.580	0.510	0.528	0.734	0.675	0.691

compared to SegNet and U-Net, with some unbalanced classes getting very low class-wise IoU and Dice scores. In terms of boundary pixel prediction, both SegNet and U-Net provide much better results than the baseline FCN, which

usually outputs the beach and background pixels. U-Net is recommended in most cases for accurately segmenting beach debris, with SegNet performing slightly poorer in some cases but much better than FCN-32.



Fig. 17 Overlay of model segmentation over the survey area²

12.2 Loss Functions

As for the loss functions, it can be said that Dice loss performed better in most cases for both models. This shows the effectiveness of Dice loss in multiclass segmentation and its compatibility with multiple segmentation models. Focal Tversky loss has provided great results in both real and synthetic data, especially with U-Net, however, mIoU and mDice values of FTL are quite low in real data. This may be attributed to the fact that real data is highly imbalanced. Further experimentation on FTL parameters is also needed. As for WCCE, the mIoU and mDice values are consistently lesser than the other two losses. This might be attributed to the fact that the weights of WCCE are extremely hard to tune due to the nature of the dataset, and thus proves little leeway for the model to converge. WCCE loss should be generally avoided for complex datasets as it struggles to adapt to the uneven class and pixel distribution due to the hardcoded class weights, however, it can be used for datasets with even class distribution, which could be the case in sparse marine debris segmentation. DL is recommended for cases with debris segmentation with multiple classes and hard-to-segment data with a very low pixel to background ratio. FTL performs marginally better than DL but its focusing index can help tune the learning process to target difficult examples and makes it a great segmentation loss function for datasets with high-class imbalance, which is the case in our custom dataset.

12.3 Segmentation Results

Figure 18 shows the predicted segmentation for given test input images along with the ground truth labels. The predicted segmentations reflect the evaluation metrics, U-Net has better segmentation labels than SegNet in most cases and segmentations obtained from models trained with FTL resemble the closest to ground truth masks with smooth boundary segmentation and DL also providing satisfactory predictions.

13 Summary

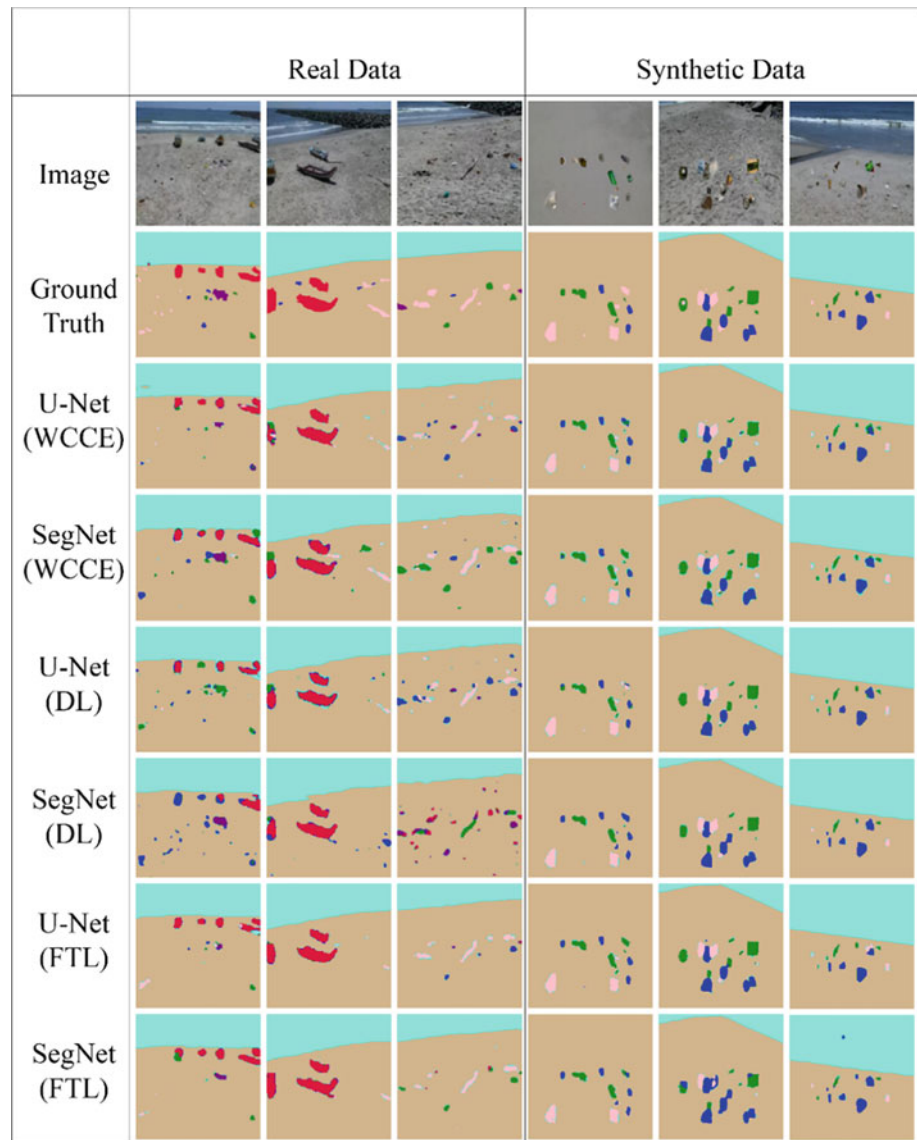
The following general points can be derived from the above comparisons:

- U-Net performs better than SegNet-basic in most cases, and both networks perform considerably better than the baseline FCN-32 model.
- Models trained with FTL and DL provide great results with DL having higher mIoU and mDice in most cases and FTL providing better segmentation for beach debris with very few false positives. Models trained with WCCE fail to adapt to imbalanced datasets.
- Synthetic data generation makes model learning easier by increasing overall debris data which reduces pixel to background ratio and combating class imbalance in datasets.

14 Conclusion

In this work, multiclass segmentation of coastal debris has been presented. Since the acquired data has a class imbalance, we propose a random data generation method to improve model accuracy. The graphs and plots obtained narrate the performance of chosen models over the gathered data. The evaluation metrics obtained for the models trained under various loss functions are then compared and discussed. Segmentation predictions are also presented which dictates the better performance of the models on synthetic data over real data. These findings can be used in the precise localization of beach debris which will prove very useful in beach litter monitoring and cleaning. Further works on this

Fig. 18 Prediction results for real and synthetic data along with original images and ground truth labels



topic may include evaluation of new and upcoming models for semantic segmentation along with newly proposed loss functions, along with better synthetic data generation methods. Finally, the segmentation mask is also localized and overlaid onto a map which facilitates detection and monitoring and can also locate debris even at inaccessible sites.

References

- Abraham, N., & Khan, N. M. (2019). A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)* (pp. 683–687).
- Aerial Semantic Segmentation Drone Dataset, <https://www.tugraz.at/index.php?id=22387>
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481–2495.
- Bak, S. H., Hwang, D. H., Kim, H. M., & Yoon, H. J. (2019). Detection and monitoring of beach litter using uav image and deep neural network. In *ISPRS—International archives of the photogrammetry, remote sensing and spatial information sciences, XLII-3/W8*, 55–58.
- Bao, Z., Sha, J., Li, X., Hanchiso, T., & Shifaw, E. (2018). Monitoring of beach litter by automatic interpretation of unmanned aerial vehicle images using the segmentation threshold method. *Marine Pollution Bulletin*, 137, 388–398.
- Beach Garbage Dataset: GitHub repository, <https://github.com/SrihariM1809/beach-garbage-dataset>
- Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002) SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research (JAIR)*, 16, 321–357.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).

- He, K., Girshick, R., & Dollár, P. (2019). Rethinking ImageNet pre-training. In *IEEE/CVF international conference on computer vision (ICCV)* (pp. 4917–4926).
- Ho, Y., & Wookey, S. (2020). The Real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8, 4806–4813.
- Ichimura, T., & Nakajima, S. v Development of an autonomous beach cleaning robot “Hirottaro”. In *IEEE international conference on mechatronics and automation* (pp. 868–872).
- Image Segmentation Keras: GitHub repository: <https://github.com/divamgupta/image-segmentation-keras>
- Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)* (pp. 1–7).
- Jang, S. W., Lee, S. K., Oh, S. Y., Kim, D. H., & Yoon, H. J. (2011). The application of unmanned aerial photography for effective monitoring of marine debris. *Journal of the Korean Society of Marine Environment Safety*, 17(4), 307–314.
- Kako, S. I., Isobe, A., Kataoka, T., & Yufu, K. (2018). Sequential webcam monitoring and modeling of marine debris abundance. *Marine Pollution Bulletin*, 132, 33–43.
- Lin, T., Goyal, P., Girshick, R., He, K., & Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 318–327.
- Nakashima, E. (2011). Using aerial photography and in situ measurements to estimate the quantity of macro-litter on beaches. *Marine Pollution Bulletin*, 62(4), 762–769.
- Park, S. S., & Kang, H. Y. (2005). The quantity and characteristics of marine debris collected from the coastline in Jeonnam. *Journal of Korea Society of Waste Management*, 22(2), 203–212.
- Pihur, V., Datta, S., & Datta, S. (2007). Weighted rank aggregation of cluster validation measures, a monte carlo cross-entropy approach. *Bioinformatics*, 23(13), 1607–1615.
- Pinheiro, P., Cardozo, E., Wainer, J., Rohmer, E. (2015). Cleaning task planning for an autonomous robot in indoor places with multiples rooms. *International Journal of Machine Learning and Computing*, 5(2), 86–90
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *LNCS*, 9351, 234–241.
- Salehi, S. S. M., Erdogmus, D., & Gholipour, A. (2017). Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging* (Vol. 10541, pp. 379–387). Springer.
- Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 640–651.
- Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In Cardoso, M. et al. (Eds.), *Deep learning in medical image analysis and multimodal learning for clinical decision support. DLMIA 2017, ML-CDS 2017. Lecture Notes in Computer Science* (Vol. 10553, pp. 240–248). Springer, Cham.
- Thiagarajan, S., & Satheesh, K. G. (2019). Machine learning for beach litter detection. Advances in Intelligent Systems and Computing In M. Tanveer & R. Pachori (Eds.), *Machine intelligence and signal analysis* (Vol. 748, pp. 259–266). Springer.
- TrashNet: Github repository—<https://github.com/garythung/trashnet>
- Wattanasophon, S., & Ouitrakul, S. (2014) Garbage collection robot on the beach using wireless communications. In *3rd international conference on informatics, environment, energy and applications* (Vol. 66, pp. 92–96).



Learning Analytics for Smart Classroom System in a University Campus

Tasneem Hossenally, U. Kawsar Subratty,
and Soulakshmee D. Nagowah

Abstract

Learning Analytics (LA) allows a student to see his progress and allows an instructor to see comparative progress of the different learners involved in the class. LA is thus being incorporated in different educational settings such as in smart classrooms where students' behaviour and performance are observed and analyzed. This chapter presents LA in a smart classroom using predictive models to assess formative assessment, attendance, and behavioural performance of students. The smart classroom is modelled by an automated attendance using RFID tags and an activity detection system using camera and sound sensor to record attendance and detect different behaviours of students. Moreover, an attendance model was implemented to analyze the students' attendance record and their behavioural performance from the activity detection system using a predictive model. In addition, an automated quiz system was implemented in a web application to assess students along with its model to predict their formative assessment performance. To perform prediction, the three models have been trained using different machine learning algorithms where the most accurate models are deployed in the web application. The attendance model was trained using five different multiclass classification algorithms namely, decision tree (100%), logistic regression (78.7%), Naïve Bayes (83.2%), KNN (98.4%) and Adaboost M1 (73.5%). Similarly, the formative performance model was trained using decision tree (99.1%), logistic

regression (80.8%), Naïve Bayes (92.5%), KNN (94.6%), and Adaboost M1 (87.1%). The behaviour model is evaluated using four multiclass machine learning algorithms namely, decision tree (100%), logistic regression (91%), Naïve Bayes (95%), and KNN (100%). Since both decision tree and KNN have the same accuracy, fivefold cross-validation technique is used to differentiate their accuracy. Moreover, the chosen prediction models are evaluated by comparing the accuracy along with the f1-score, precision, and recall. The smart classroom system consists of additional functionalities like addition or removal of a quiz, ranking of students, quiz history, and graphical display of performance for both students and lecturers to see.

Keywords

Internet-of-Things • Smart classroom • Sensors • Learning analytics • Machine learning

1 Introduction

Technological innovations have turned the world into a global city. Without a dispute, in this new era advanced technologies have influenced every industry and the system of education is no exception. The introduction of better educational experiences, career growth for students and educators, and the establishment of a quality education infrastructure are all ways to enhance educational standards. Smart classrooms have innovative integrated study materials and technologies such as smart boards, laptops, tablets, and audio devices to assist in the goal of creating a powerful learning atmosphere (Zhao, 2006). The key goal of adopting smart classrooms is to substitute obsolete learning curriculum with cutting-edge technologies. The aim of smart classroom is to make learning and classroom management simpler. Internet of Things (IoT) has supported in the

T. Hossenally · U. K. Subratty · S. D. Nagowah (✉)
Software and Information Systems Department, Faculty of
Information, Communication and Digital Technologies,
University of Mauritius, Reduit, Mauritius
e-mail: s.gurbhurrun@uom.ac.mu

T. Hossenally
e-mail: tasneem.hossenally@umail.uom.ac.mu

U. K. Subratty
e-mail: umme.subratty@umail.uom.ac.mu

modernization of classrooms and has liberated conventional classrooms from their limitations (Nagowah et al., 2019). IoT links any object through a network in which devices transfer data remotely. Similarly, in a modern classroom, the lighting, the door, and even the board can be interconnected to a network, enabling for smart practice. IoT has introduced a whole new definition to education, elevating teaching, and learning practices to a whole new degree.

LA is the way of evaluating and comprehending an environment in order to bring enhancement to it. Likewise, LA in a smart classroom is the analysis of individual students' behaviour, whereby academic achievement can be controlled (Nagowah et al., 2019). LA in smart classrooms strives to individualize teachings depending on every student's ability (Aguilar et al., 2018; Nagowah et al., 2019). In this way, students' learning strategies can be reinforced and improved. Therefore, to improve students' performance, it is of utmost importance that the system predicts the performance of students (Aguilar et al., 2018). LA has the potential to predict and boost student performance especially because it facilitates faculty, universities, and learners to make data-driven choices about student performance and engagement (Olmos & Corrin, 2012; Smith et al., 2012). LA therefore provides further 'personalized learning', which allows learners to have even more authentic learning experiences, among several other advantages (Greller & Drachslrer, 2012). Analytics can be used to enhance learning, teaching, and academic performance. For instance, the University of Alabama used first-year student data files to establish a retaining framework focusing on various metrics such as English course grade and total hours obtained. LA is one of the strategies for not only reporting student achievement but also for offering support that facilitates the forms of process improvement that accrediting bodies pursue (Dietz-Uhler & Hurn, 2013). The rest of the chapter is structured as follows: Sect. 2 presents related work. Section 3 describes the proposed system. The use of learning analytics in the smart classroom system is illustrated in Sect. 4. Performance evaluation and model selection of different machine learning models are illustrated in Sect. 5. Implementation details of the system prototype and results are presented in Sect. 6. The proposed system is evaluated in Sect. 7. Finally, Sect. 8 presents the conclusion and future works.

2 Related Work

There are some smart classroom systems that have applied LA. One example is a Smart Classroom-Social and behavioural analysis intelligence framework that has been implemented (Temkar et al., 2016). The prime objective of

this implementation is to oversee students' levels of satisfaction throughout a lecture by identifying jiggling and noises as they are the main actions of showing lack of interest (Gandhi, 2017). To evaluate the different behavioural patterns during a lecture, various parameters such as sound frequency, motion level, and noise must be identified, evaluated, and controlled. Sensors including such PIR sensors, motion sensors, microphones, cameras, and sound sensors are used to capture those metrics. Further to that, a smart band is used to track student attendance based on pulse rate (Temkar et al., 2016).

Another system in which LA has been used, is a smart classroom that detects and regulates the degree of interest in a lecture, enabling the lecturer to notice and regulate the level of involvement in a lecture (Gligoric et al., 2015). Initially, lecturers' teaching methods were assessed through surveys collected during a lecture, and although these approaches supported lecturers in adjusting their teaching strategies, they did not facilitate the application of improvements during the lecture. As a result, to implement the smart classroom, selection, derivation, and understanding of the parameters are enabled in a more convenient approach. The main parameters identified are the movements and the interaction between the teacher and the students during a lecture. To capture those parameters, different sensors like camera, a 3-axis accelerometer, and broadband NT2A multipattern capacitor microphone are used (Gligoric et al., 2015).

A conventional classroom can be turned into a smart classroom by using technology such as artificial intelligence and cloud models to tailor the learning experience to different students. LA tasks are also accountable for data analysis in a smart classroom (Larsson et al., 2014). In another system, LA tasks are introduced in a smart classroom using a combination of cloud and multiagent samples. Besides that, machine learning techniques and mathematical approaches are used when they determine the details of how a student learning process works. The main purpose is to assess how students learn in order to improve teaching and learning experiences (Aguilar et al., 2018).

A successful development of predictive analytics has been achieved where the system consists of performance prediction, skill estimation, and behaviour detection. For the performance prediction, score of each student needs to be calculated (Kellen et al., 2013; Sin & Muthu, 2015). Based on this score, the involvement and interaction of students can be identified. Therefore, a good score implied the student has a good performance whereas a low score indicates poor performance. Consequently, students with poor performance need more attention. Skill estimation is attained by the interaction of students with the system. Behaviour detection recognizes the facial expressions of students, which are used

to predict how satisfied, contented or unhappy the students are with the teaching methods (Grafsgaard et al., 2014; Sin & Muthu, 2015).

3 Proposed System

This section describes the proposed system, which includes a smart attendance system, activity detection system, sleep detection of students and performance assessment system in the classroom as shown in Fig. 1.

Smart Attendance. The attendance system is implemented using a Radio Frequency Identification (RFID) scanner that records attendance of each student. Students scan their cards at the RFID reader located at the entrance to the classroom, and the time the card is scanned, are logged.

The smart attendance system computes presence, absence, and lateness for each semester and stores the information in a database.

Activity Detection/Drowsiness and Yawn Detection. Student activities are detected by live streaming via a web cam, where activities such as standing, sitting, leaning, and laying down are recognized. To detect these activities, a pretrained Faster R-CNN model has been used. By detecting yawn and drowsiness of students in class, the learning states can be determined and used by lecturers to improve lectures (Wang et al., 2019). Students' drowsiness and yawns are thus identified using a library called Dlib, which comprises of a predictor file, and upon detection, an alert is displayed on the screen. Moreover, a KY-037 sound sensor has been used to collect sound data from the classroom. All the collected data from attendance, activity, drowsiness, and yawn

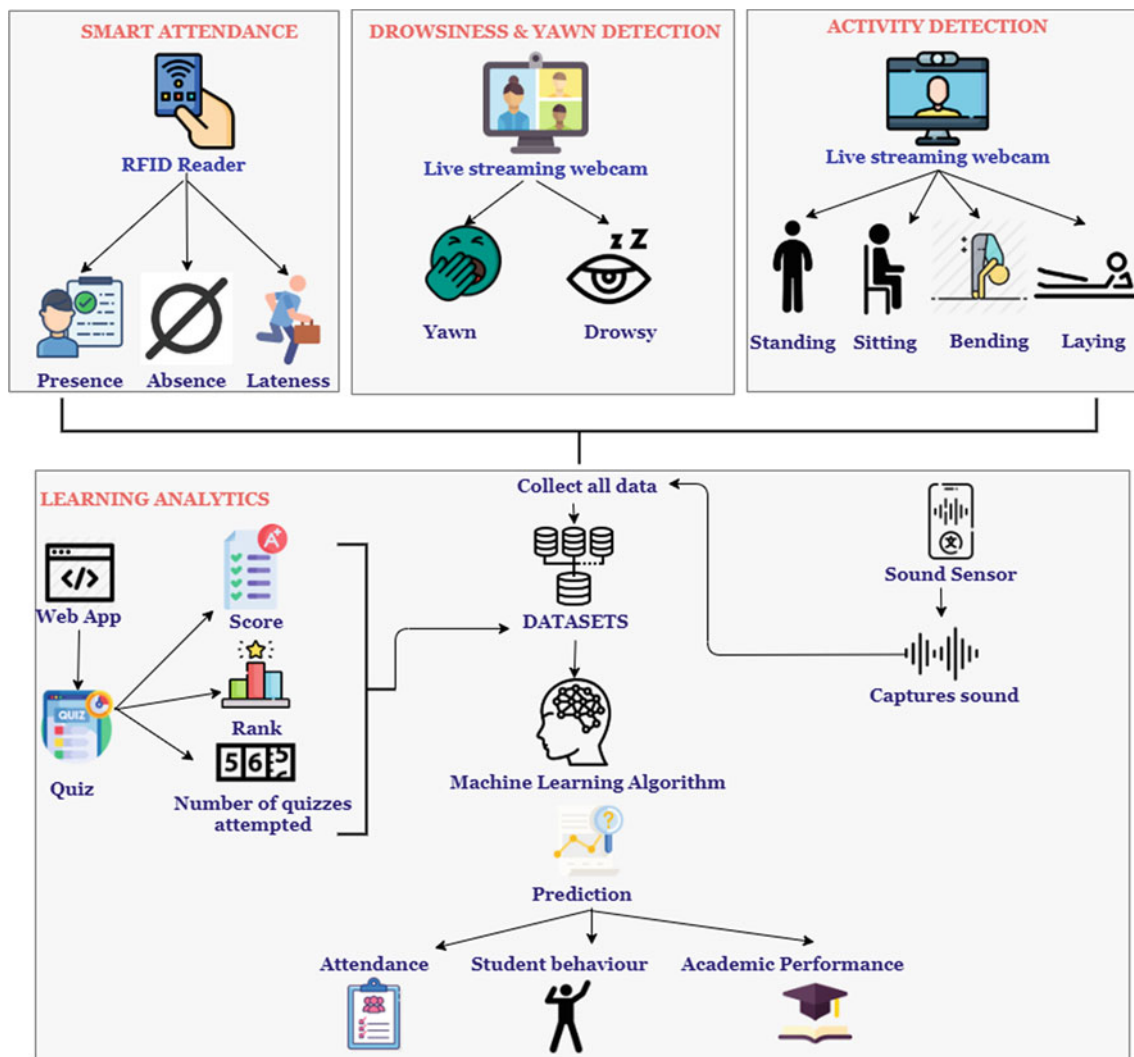


Fig. 1 Architecture diagram of proposed system

detection are used as datasets to simulate attendance and student behaviour in the classroom using various algorithms.

Performance Assessment. The system additionally makes use of formative assessment, which is a type of assessment that provides guidance to instructors about the level of understanding and progress of students (Nagowah & Nagowah, 2009). Formative assessment normally consists of quizzes or Multiple-Choice Questions (MCQs) carried out on a regular basis (Nagowah & Nagowah, 2009; Nagowah et al., 2010). The system allows students to take interactive quiz questions using a web application, and formative performance can be predicted based on overall score, rank, and number of quizzes taken. Different machine learning algorithms are used to train the quiz model and the one with the highest accuracy is deployed. The quiz system includes an administration part where lecturers are able to set questions and view students' scores and prediction graphs of attendance, formative assessment, and behaviour analysis. With these graphs, lecturers can analyze students well and might improve their way of teaching, therefore applying learning analytics to the education sector.

For predicting attendance of students, the attributes absence, presence, and lateness are used. These details are collected by the RFID system and are stored in a database. A graph of prediction against time is plotted. Additionally, another predictive model is trained and deployed to analyze the different behaviours of students in a class during a lecture. The data gathered from the activity detection system, that is, the sound level detected, whether a student is sleeping, standing, sitting, or laying down as well as the yawning and drowsiness factor, are used to predict the different ways that a student can behave in class. This prediction is displayed graphically to help ease of analysis. All predictive models are implemented using different machine learning algorithms and the one having the highest accuracy is deployed to get the most accurate prediction about a student. To sum up, models are trained to perform prediction on attendance, students' behaviour, and formative performance and through this prediction, improvements can be introduced in classrooms thus rising educational standards.

4 Learning Analytics in Smart Classroom

LA allows a student to see his progress and allows an instructor to see comparative progress of the different learners involved in the class. In doing so, the quality of teaching and learning might be improved which will accomplish the purpose of learning analytics (Siemens & Long, 2011). LA is thus being incorporated in different educational settings such as in smart classroom where students' behaviour and performance are observed and

analyzed. As mentioned in the Sect. 3, three machine learning models are implemented to predict the different aspects of students in a smart classroom using data collected through sensors and a web application. Different machine learning techniques are used to build the models and analyzed to choose the most accurate ones to be deployed.

To perform a better prediction, students' id number along with their card number for the RFID card are stored. The parameters that are being considered for the three models are namely presence, absence, lateness, number of drowsiness and yawn in a lecture, sound level, score of quiz, rank and number of quizzes attempted. Prediction of attendance is performed using presence, absence, and lateness. As mentioned in Sect. 3, these data have been captured by the RFID reader and Arduino. Similarly, prediction of student behaviour in classroom is accomplished using the number of yawn and drowsiness and the sound level. Number of yawn and drowsiness are captured through a camera placed in the classroom while sound level is recorded using the sound sensor and Arduino. Ultimately, prediction for formative performance is achieved using the parameters namely total score, rank, and number of quizzes attempted. These data have been obtained from an automated quiz system implemented in a web application. The prediction for formative performance model and attendance are classified as four classes, which are, excellent, good, bad, and very bad. For the behaviour model, the classification prediction classes are bored, neutral, and concentrated. The parameters collected from the sensors and web application are then processed in Jupyter Notebook of python to provide a prediction by finding patterns in the unprocessed data. After the training of the model, the accuracy are analyzed and compared to choose the best model for prediction. When the model is selected, it is saved and deployed in the web application via a python API code. The API then takes data from the database and finds the prediction. The prediction is then returned to the database where the web application accesses those data to plot a graph.

4.1 Application of ML Algorithms for Prediction

Classification algorithms are used for the different prediction as the expected output of the model should be a label, that is, for both formative performance and attendance models, the prediction is classified as excellent, good, bad, or very bad, which represent the performance of each student. For the behaviour model, the prediction is classified as concentrated, neutral, or bored, which represent the overall behaviour of students in a smart classroom. Different multi-class classification algorithms are used to train those models namely Logistic Regression, K-Nearest Neighbours (KNN), Decision Tree, Adaboost M1 and Naive Bayes. Multinomial

logistic regression is a simple continuation of binary logistic regression that includes more than two categories of the dependent or output variable. The logistic function (sigmoid function) is an S-shaped curve for data discrimination across multiple classes and this algorithm is used as it can easily cater for multinomial classes instead of binary classes. It is also easy to implement and is fast at classifying unseen data (Starkweather & Moske, 2011). KNN algorithm assumes that related things exist in closeness, that is, “*similar things are near each other*” (Brownlee, 2016). This algorithm assigns the value of K to number of neighbours chosen, then calculate the distance between the current and query data. When a prediction is required, the k-most similar records to a new record from the training dataset are then located. From these neighbours, a summarized prediction is made and KNN is much faster than other algorithms as it does not need to be trained to learn the patterns but stores the training dataset and learns from it when making real time predictions, thus causing the algorithm to have an ability to add new data continuously without affecting the accuracy (Brownlee, 2016). Decision Tree algorithm uses a graphical representation given a certain condition to get all possible solutions to a problem. It uses attribute selection measure to find the best attribute in the dataset. It therefore breaks down the root node into subtree that contains possible values for the best attributes and eventually creates the decision tree node, which consists of the best attribute. The reason for choosing this algorithm is the effectiveness of the model that can predict data with minimum pre-processing time (Brownlee, 2014). Adaboost M1 algorithm which stands for Adaptive Boosting is an ensemble technique to boost the prediction efficiency of a classifier. The weight is updated at each iteration according to the performance, with higher weights to incorrectly classified higher instances until new data is created and specified condition is met. It also provides low generalization error, that is, a type of error when measuring the accuracy of an algorithm by analyzing if it can predict output values for undetected data (Naseer et al., 2020). Naïve Bayes algorithm considers each of the features contributed independently to the probability that the input is the class variable. The algorithm calculates two probabilities that is, the class probabilities: the probabilities of having each class in training dataset and the second is the conditional probabilities: the probabilities of each input feature giving a specific class value. Naïve bayes is chosen as it does not require much training data and predicts well with real time data (Mukherjee & Sharma, 2012).

4.2 Techniques Used to Train the Model

No matter what dataset is used, it needs to be split into training and testing data for training any machine learning

model. Neither of them should be too high to prevent more dependency on the amount of data used nor too small, otherwise no split will give satisfactory variance (Nguyen et al., 2021). For both formative performance model and attendance model, the dataset is split into 80% training set and 20% testing set, which is considered the ideal way to split the dataset using the Scikit-learn’s `train_test_split` module in python.

Moreover, in case two algorithms give the same accuracy for behaviour model, another technique called the K-fold cross validation is used to differentiate the accuracies. This technique is basically used to evaluate a machine learning model. A single variable called k is present in this technique which “*refers to the number of groups that a given data sample is to be split into*” as said by Jason Brownlee (Brownlee, 2016). Moreover, this technique prevents biased and optimistic evaluation of models compared to other methods (Brownlee, 2019). It does not depend on how the dataset is split, that is, the 80:20 ratio split (Nguyen et al., 2021).

5 Performance Evaluation and Model Selection

In this section, different models are compared and evaluated to choose the best ones for better prediction and accurate value to be deployed in the web application. The accuracy is discussed along with the precision, recall, and f1-score of the different algorithms used.

Precision is the measure of correctly predicted positive class from the total positive class predicted (Hossin & Sulaiman, 2015). It is calculated by the following formula,

$$\text{Precision} = \text{True positive} / (\text{True Positive} + \text{False Positive}) \quad (1)$$

Recall is the actual number of positive class predicted (Hossin & Sulaiman, 2015) as per the following formula,

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{True Negative}) \quad (2)$$

F1-Score represents the harmonic mean between precision and recall (Hossin & Sulaiman, 2015). A model is considered more accurate when the F1-score value is nearly equal to 1 as computed by the following formula,

$$\text{F1-Score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (3)$$

Moreover, different confusion matrix graphs for each algorithm are plotted to compare the predicted output along with the actual one.

The Formative Performance Model. The formative performance model is used to predict the formative performance of each student by taking quiz. More than 800 data records are used as training data to train this model using the algorithms namely Decision Tree, KNN, Naïve Bayes, Adaboost and Logistic Regression and about 224 data records for testing data, excluding the records obtained from the web application. Table 1 shows the different metrics used for analyzing the different algorithms used for formative performance model. Figure 2 shows the graphs of predicted output against actual output. The number 0 represents the ‘*excellent*’ class, 1 represents the ‘*good*’ class, the number 2 represents the ‘*bad*’ class, and lastly 3 represents the ‘*very bad*’ class for the formative performance model. In Fig. 2, for the decision tree algorithm, 82 test data for the class ‘*very bad*’ is correctly classified, 59 test data is correctly classified for the class ‘*bad*’, for the class ‘*good*’, 56 test data are predicted correctly, for the class ‘*excellent*’, 25 test data are correctly classified and lastly in the overall predicted outputs, only 2 test data are wrongly predicted. So, according to the Table 1 and Fig. 2, the algorithm which best fits the model is the Decision Tree algorithm as all the analysis parameters are the highest and the predicted output against the actual one is nearly the same.

The Attendance Model. The attendance model is used to predict the attendance performance of a student during a semester by using the records from the RFID. About 2000 data records are used as training data to train this model using the algorithms namely Decision Tree, KNN, Naïve Bayes and Logistic Regression and 768 data records for testing data, excluding the records obtained from the RFID attendance. Table 2 shows the different metrics used for analyzing the different algorithms used for attendance model. Figure 3 shows the graphs of predicted output against actual output. The number 0 represents the ‘*excellent*’ class, 1 represents the ‘*good*’ class, the number 2 represents the ‘*bad*’ class, and lastly 3 represents the ‘*very bad*’ class for the formative performance model. In Fig. 3, for the decision tree algorithm, 532 test data for the class ‘*very bad*’ is correctly classified, 182 test data is correctly classified for the class ‘*bad*’, for the class ‘*good*’, 54 test data are predicted correctly, for the class ‘*excellent*’, 3 test data are correctly classified and lastly in the overall predicted

outputs, only 3 test data are wrongly predicted. Therefore based on findings from Table 2 and Fig. 3, the algorithm which best fits the model is the Decision Tree algorithm as all the analysis parameters are the highest and the predicted output against the actual one is nearly the same.

The Behaviour Model. The behaviour model is used to predict the different behaviours of students during a lecture using the data obtained from the camera used in activity detection. More than 800 data records are used as training data to train this model using the algorithms namely Decision Tree, KNN, Naïve Bayes, and Logistic Regression and about 150 data records for testing data, excluding the records obtained from the activity detection system. Table 3 shows the different metrics used for analyzing the different algorithms used for behaviour model. Figure 4 illustrates the graphs of predicted output against actual output. The number 0 represents the ‘*bored*’ class, 1 represents the ‘*neutral*’ class, and lastly the number 2 represents the ‘*concentrated*’ class. Since the accuracy and other parameters are the same for both Decision Tree and KNN algorithms, as shown in Table 3 and Fig. 5, 5-fold cross validation (CV) technique is used and a mean score for testing data and training data are calculated and evaluated as shown in Table 4. Figure 5 illustrates the graphs of predicted output against actual output for KNN with CV and Decision Tree with CV. In Fig. 5, for the decision tree algorithm which uses the fivefold CV technique, 112 test data for the class ‘*concentrated*’ is correctly classified, 34 test data is correctly classified for the class ‘*neutral*’, for the class ‘*bored*’, 6 test data are predicted correctly and lastly in the overall predicted outputs, only 1 test data is wrongly predicted. Additionally, in Fig. 5, for the KNN algorithm which uses the fivefold CV technique, 112 test data for the class ‘*concentrated*’ is correctly classified, 28 test data is correctly classified for the class ‘*neutral*’, for the class ‘*bored*’, 2 test data are predicted correctly and lastly in the overall predicted outputs, 8 test data are wrongly predicted. So, according to the fivefold CV used, the algorithm which best fits the model is the Decision tree algorithm as all the metrics analyzed are the highest, the predicted output against the actual one is nearly the same and the wrongly predicted test data are less compared to the KNN algorithm.

Table 1 Metrics for formative performance model

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	92	91	92	92
KNN	95	95	95	95
Adaboost	87	81	87	83
Logistic	81	80	80	80
Decision tree	99	99	99	99

Fig. 2 Predicted output versus actual values for formative performance model

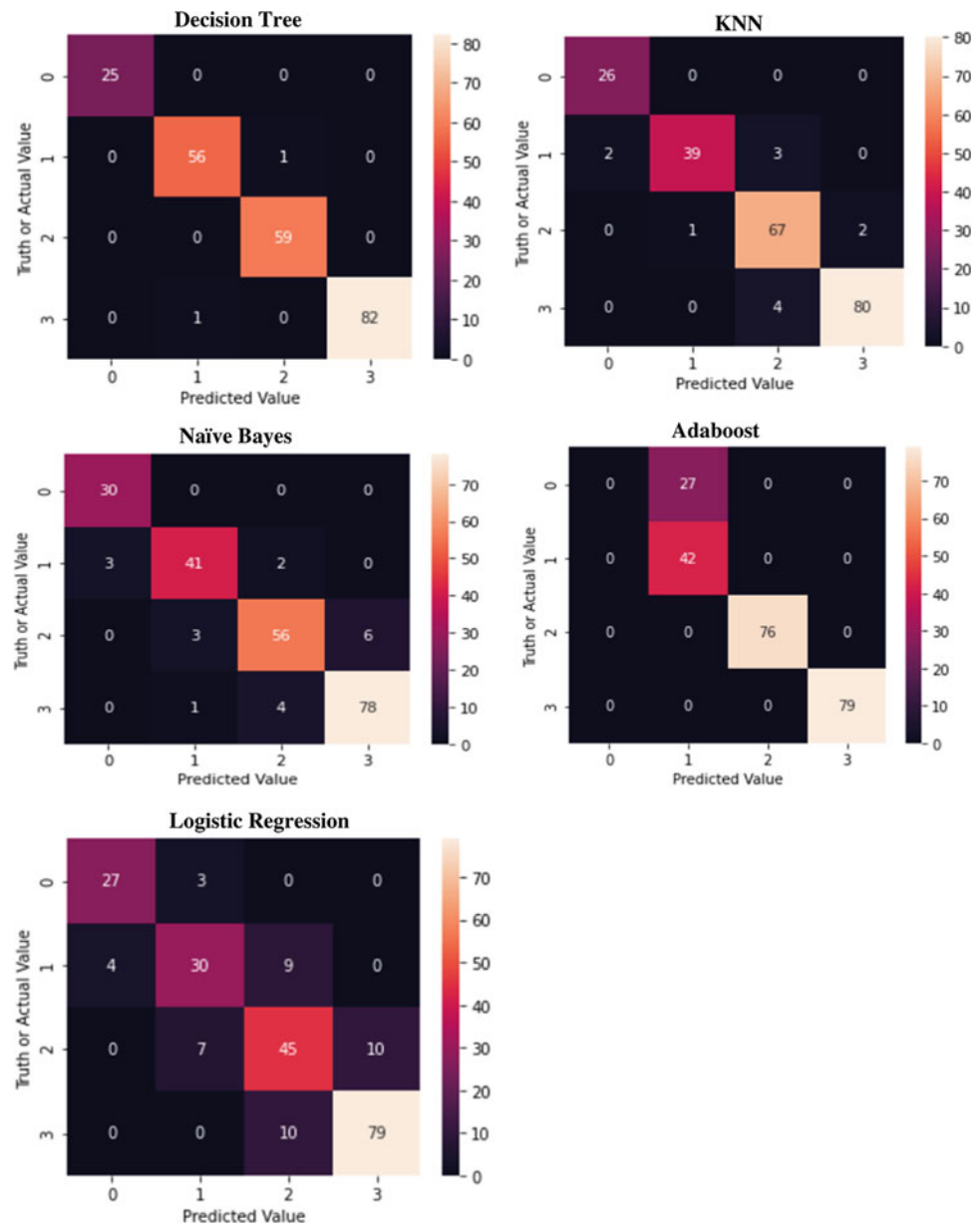


Table 2 Metrics for attendance model

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	85	85	85	83
KNN	98	98	98	97
Logistic	81	80	81	80
Decision tree	99	100	100	100

6 System Prototype

In this section, the different tools used for setting up the system prototype are described.

6.1 Hardware and Software Used

The hardware used for the implementation of the web application and detecting yawn and drowsiness of students is

Fig. 3 Predicted output versus actual values for attendance model

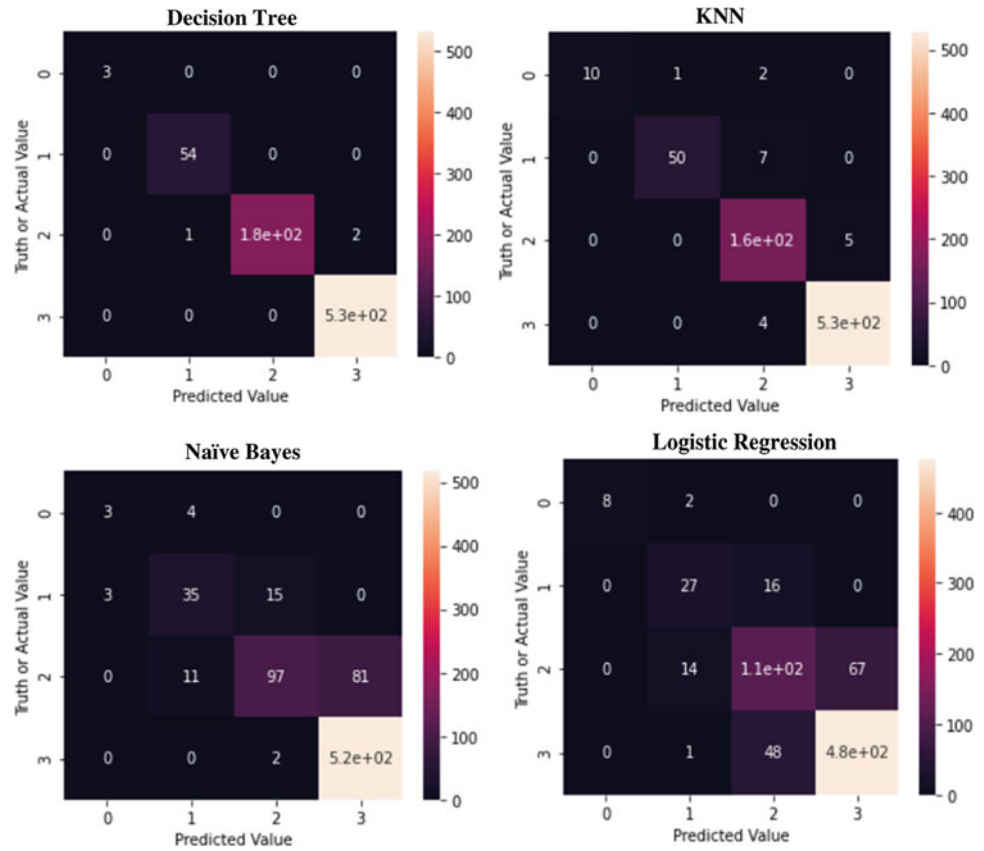


Table 3 Metrics for behaviour model

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	95	95	95	95
KNN	100	100	100	100
Logistic	91	90	91	90
Decision tree	100	100	100	100

a personal computer, where the webcam was used as camera. The details of the computer are shown in the Table 5.

Table 6 shows the IoT devices used in the implementation of the attendance and activity detection system.

Table 7 shows the software that were used for the implementation and where they were used.

6.2 Implementation of Different Modules

This section describes the different modules implemented.

The RFID Attendance Module. The RFID attendance module is implemented using RFID readers and RFID cards, which make it easy for students to scan their cards before entering the class. The attendance details in terms of the presence, absence, and lateness of the student are recorded in MySQL database using C# codes in Arduino IDE. Rules are set for the cards and python codes are used to record the

mentioned parameters in the database. If the student is not authorized to a class, a red LED will light up with a delay of 10 s and a buzzer alarm will be switched on and a message will be displayed on the screen. Otherwise, a green LED will be light up with 10 s delay and a message stating the student’s information will be displayed. The data records in the database are used for the prediction of the attendance performance of the students. Figure 6 shows the setup of the IoT based smart attendance system. The smart attendance system has the aim of taking attendance and recording time of entry along with calculating total number of absences in a semester and the amount of lateness. The smart attendance system has been implemented using an Arduino Uno, a RFID reader RC522, a Servo, a buzzer and a breadboard.

The Yawn and Drowsiness Module. This module is implemented to detect the number of students yawning and feeling drowsy in a class. With the webcam, Dlib and predictor libraries in python are used for this purpose. An Eye

Fig. 4 Predicted output versus actual values for behaviour model

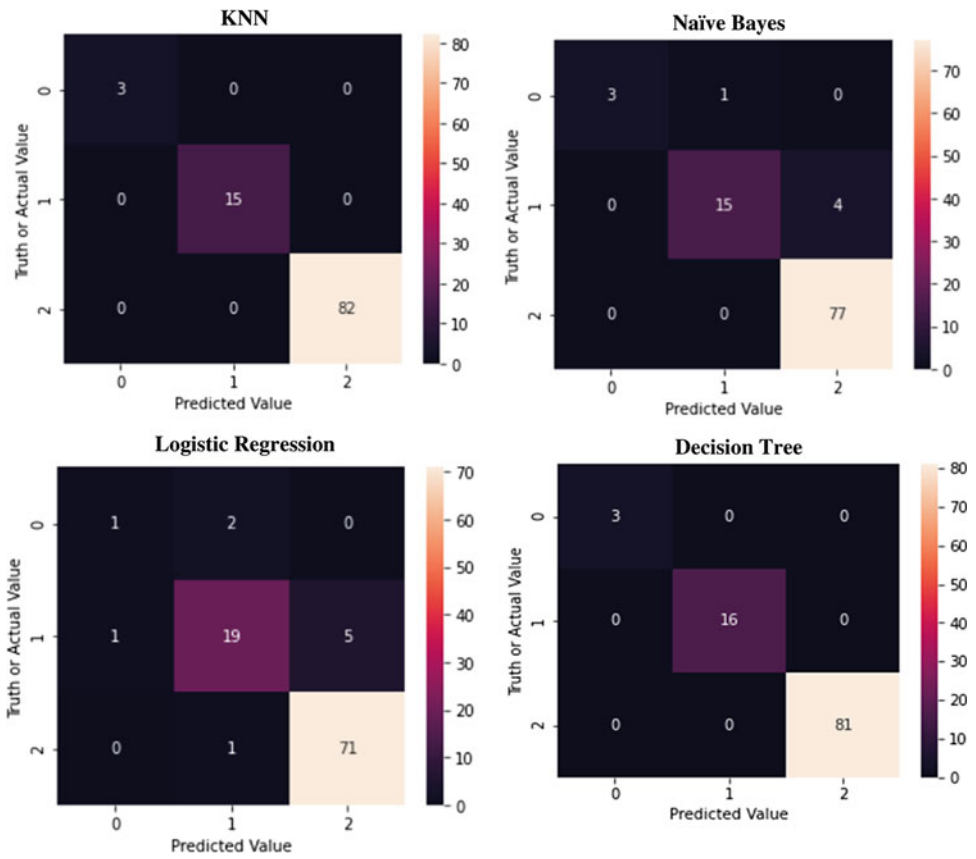


Fig. 5 Predicted output versus actual values for behaviour model using 5-Fold CV

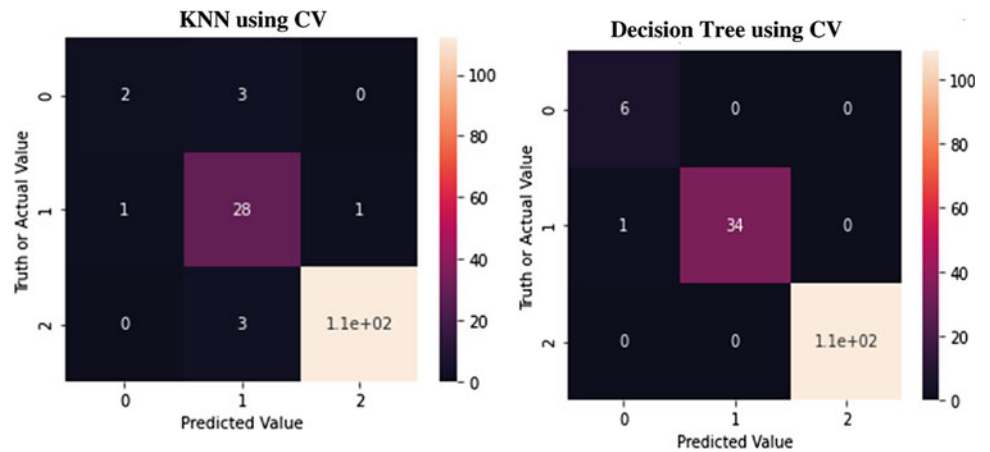


Table 4 Metrics for behaviour model using 5-Fold CV

Algorithms	Mean accuracy for training data (%)	Mean accuracy for testing data (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	94.2	94.7	95	95	94
Decision tree	95.3	99.3	99	99	99

Aspect Ratio (EAR) value is set, which determines the distance between the eyelids of a person. Thus, the threshold value for EAR is set to 0.3. If the value recorded is below the

threshold, the student is detected to be drowsy as the distance between his eyelids is smaller than the threshold. Otherwise, the student is considered not to be drowsy.

Table 5 Personal computer specification

Specification	Personal computer
Processor	Intel® Core™ i7-6500U CPU @ 2.50 GHz 2.60 GHz
RAM	16.0 GB
Hard disk	908 GB
System type	64-bit operating system

Table 6 IoT devices

IoT device	Description
RFID reader and RFID cards	A RFID reader and RFID cards will be used because it is efficient and practical for students. It is affordable and easily available on the market. Moreover, RFID requires less human effort and is not time-consuming
Arduino Uno	Arduino Uno is open source in both hardware and software. It is cheap and is suitable for beginners. Furthermore, it is cross-platform and numerous sketches are already available
LEDs	LEDs are cheap and easily available. It is also very easy to programme its function
Buzzer	Buzzer is easily available and not costly. It can be programmed using a simple function
Sound sensor	Sound sensor is easily available and can collect real-time data. It captures intensity of a noise effortlessly. Additionally, it is a low power device and does not consume much energy

Table 7 Software specification

Software name	Description	Dependencies
Komodo edit	It is a software used for web-based development using many different programming languages	Used for coding the web application in PHP, JavaScript, HTML, and CSS
Anaconda navigator	Anaconda navigator is a graphical user interface (GUI) that enables the start of conda packages and environments	N/A
Spyder	The scientific python development environment (Spyder) is an IDE included with Anaconda which enables users to do analyzing, interactive testing, fixing, and self-analysis of features	Used for the API code, the RFID attendance and the activity detection system using Dlib and predictor libraries and a pretrained model
Xampp	It is a PHP development environment	For hosting the web application on a personal computer and using MySQL database embedded for activity detection and RFID attendance
Jupyter notebook	It is an application that allows you to create and edit documents that display the input and output of a Python or R language script	Used for the training of machine learning models using several algorithms
Arduino IDE	Arduino IDE uses C# programming language. It is used for Arduino projects and has inbuilt libraries and sketches	Used for RFID attendance and sound detection in activity detection system

Moreover, the yawn threshold is set to be 20, which is the distance between the upper lips and bottom lips of a person. If the value detected is above the threshold value, the person is said to be yawning. Otherwise, the person is detected not be yawning. After detecting this behaviour, the data are recorded in MySQL database to be used later for predicting

the behaviour of student in class. Figure 7 shows the drowsiness and yawn alert upon detection of drowsiness and yawn in classroom. The value for EAR is 0.13 meaning that the EAR is below the threshold, which is 0.3, thus displaying a drowsiness alert on the screen. The value for yawn is 34.83 and it exceeds the threshold which is 20. Therefore, it

Fig. 6 Sensors used for RFID attendance

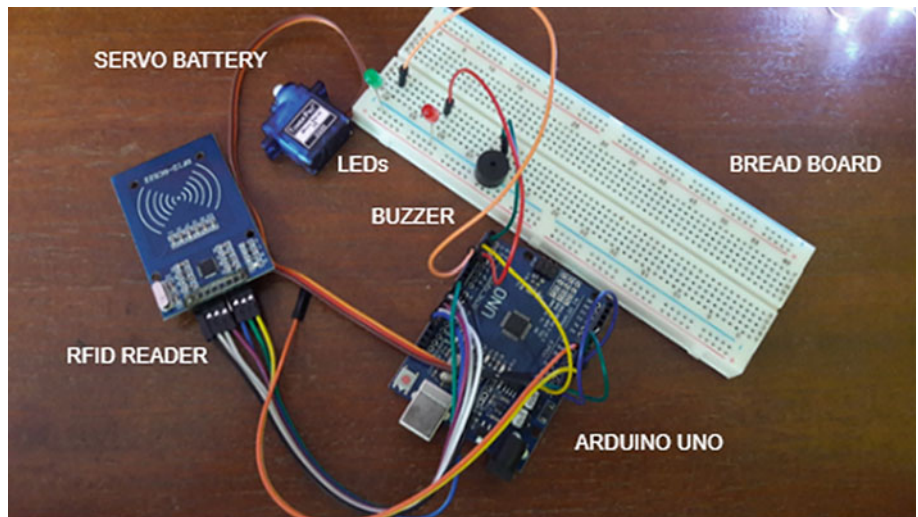


Fig. 7 Drowsiness and yawn alert



Fig. 8 Activity detection in classroom



considers that the individual is yawning and hence displays a yawn alert on the screen.

The Activity Detection Module. In this module, the activity of students like standing, sitting, sleeping, and bending are detected. Using a pretrained model in python, these activities can be detected. After detecting those motion, a detection box is displayed on the screen showing the specific activity of the student along with its accuracy. By implementing this, lecturers can analyze the activities of the students. Figure 8 shows the different activities captured by the camera namely standing, sitting, bending, and sleeping.

The Sound Detection Module. The sound detection module uses a sound sensor to record the sound in a class and switch on a red LED upon detection of a loud noise and display a peak on the serial plotter graph in Arduino IDE. After this detection, a record is stored in MySQL database as true or false. True for the detection of the sound and false for not detecting any loud noise. This record is then added with yawn and drowsiness records to better analyze the behaviour of students with the behaviour model. Figure 9 shows the setup for sound detection system. The sound sensor KY-037 captures the noise and then displays it in a graph on the serial plotter of Arduino IDE. Through this, we can detect the sound level in the classroom. The detection of sound level has been implemented using an Arduino Uno, a KY-037 sound sensor and a breadboard.

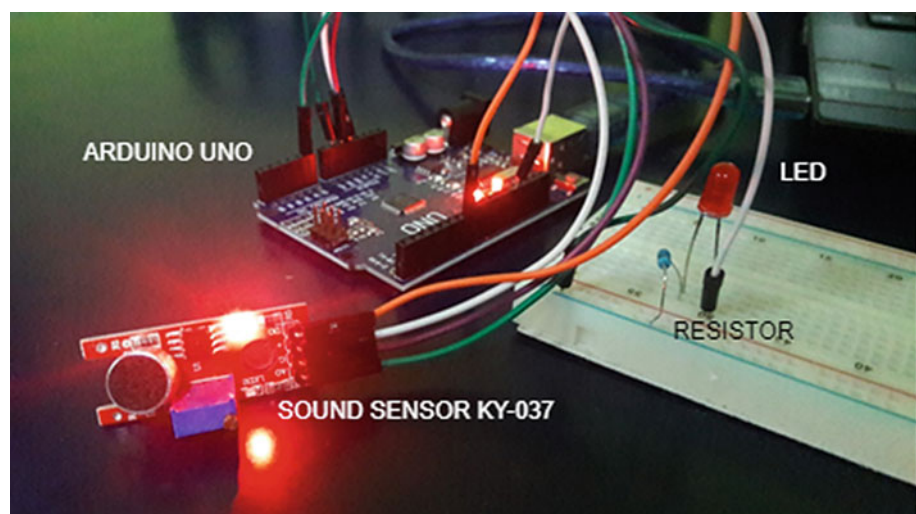
The Formative Assessment Module. This module implements an automated quiz system as formative assessment for the students in a web application. This web application allows the students to take a maximum of 5 quizzes which are set by lecturers themselves in the application. After taking each quiz, the student's scores are recorded. The maximum score of the student is 100 and the minimum score is 0. The ranking depends on the total

number of students registered in the web application. Using these parameters, the formative performance can be predicted by the machine learning model.

Deployment of Machine Learning Models. Once the data are collected from the RFID attendance system, the activity detection, yawn, drowsiness, and sound system, and the automated quiz in the web application, they are recorded in MySQL database which is present in Xampp. Therefore, a PHP code is written to transfer the data collected from the database to the model for prediction. Arriving in the python code, the parameters are decoded and used as testing data for the prediction of the models. The prediction value is encoded in JSON file and sent back to the PHP code. The CURL function is used for sending and receiving data from python to PHP and vice versa. The prediction values are then decoded and stored in the database. Moreover, the chart.js script from Google charts is called to draw the graphs using the prediction value recorded before.

Figures 10 and 11 show the graphs from the web application that predict student's attendance and formative performance in the proposed system. The x-axis of the graph represents the date and time when the student's formative and attendance performance were predicted and the y-axis represents the performance level that is, the value 1 represents the 'excellent' class, the value 2 represents the 'good' class, the value 3 represents the 'bad' class and the value 4 represents the 'very bad' class. As both formative model and attendance model prediction are classified as the same classes, the blue line graph represents the formative performance, and the red line graph represents the attendance performance of the student. Moreover, the user can select a particular point on the graph to show the specific time and predicted performance of that point, as shown in Figs. 10 and 11. Figure 10 shows the formative performance predicted as 2, which is classified as good performance and

Fig. 9 Sensors used for sound detection



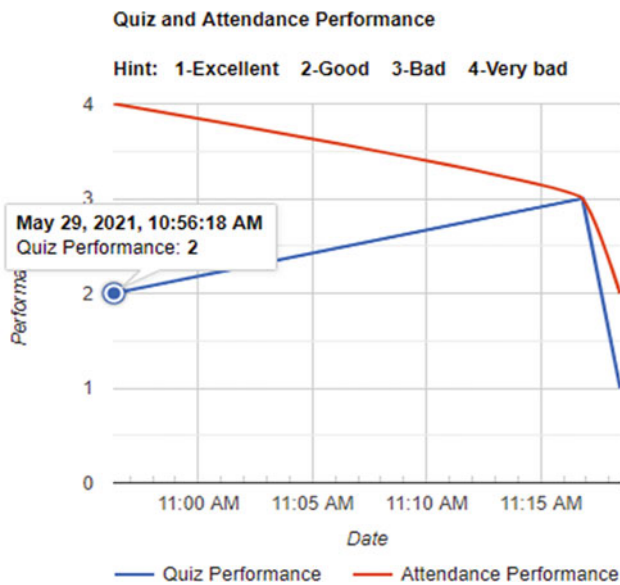


Fig. 10 Formative performance graph

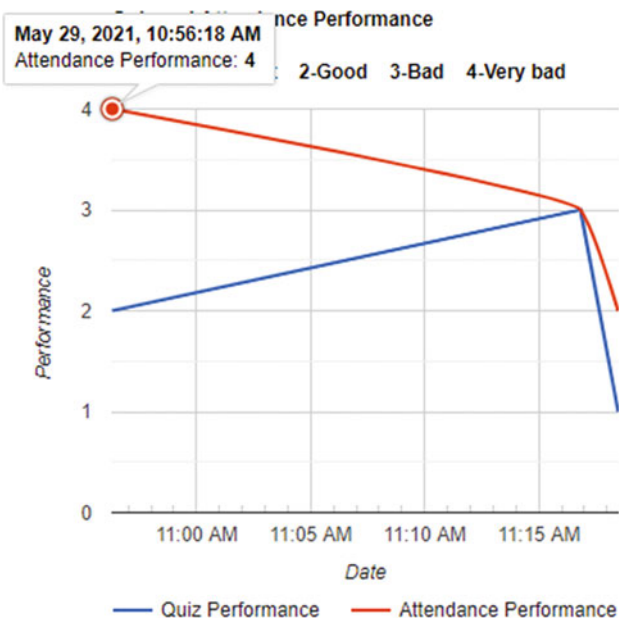


Fig. 11 Attendance performance graph

Fig. 11 shows the attendance performance as 4, which is classified as very bad performance.

Figure 12 represents the overall behaviour of the class. The x-axis of the graph represents the date and time when the students' behaviour performance is predicted and the y-axis represents the performance level that is, the value 1 represents the 'concentrated' class, the value 2 represents the 'neutral' class and the value 3 represents the 'bored' class. In addition, the instructor can select a particular point on the graph, where the behaviour predicted for the students will be

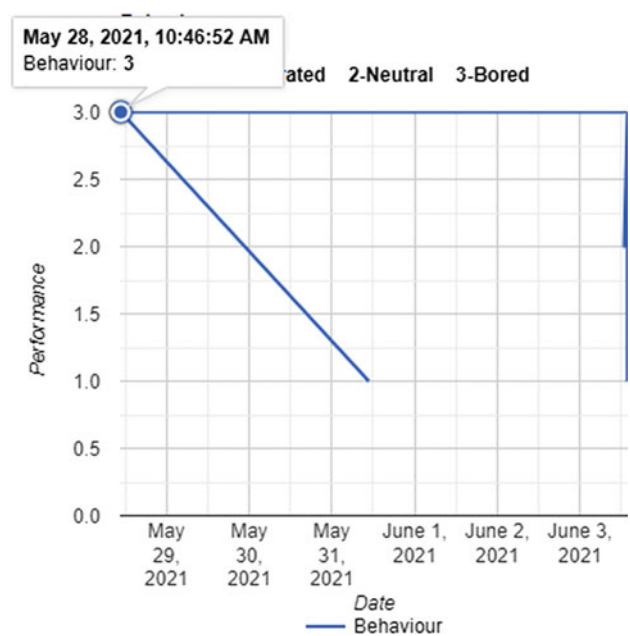


Fig. 12 Behaviour performance graph

displayed as well as the date and time. As shown in Fig. 12, the behaviour predicted is 3, which represents a bored behaviour of students.

7 System Evaluation

In Table 8, the comparison between the existing systems and the proposed system is made. It is observed that three existing systems (Aguilar et al., 2018; Gligoric et al., 2015; Temkar et al., 2016) in Sect. 2 do not use quizzes as formative assessment like the proposed system to better apply LA, while (Sin & Muthu, 2015) uses games as formative assessment. All of systems (Aguilar et al., 2018; Gligoric et al., 2015; Sin & Muthu, 2015; Temkar et al., 2016) concentrate on detecting the level of interest of students, which is the basic goal of LA. As we can see in Table 8, Gligoric et al. (2015) and Temkar et al. (2016) use camera to detect motion only whereas Aguilar et al., (2018) and Sin & Muthu, (2015) use camera to detect faces for either facial expression or attendance. The proposed system in this chapter combines both aspects, that is, uses camera to detect both facial expression and motion of students, thus providing better understanding of the students' behaviour, which can therefore help the instructors to improve their way of teaching or provide help. Moreover, Gligoric et al. (2015), Temkar et al. (2016) and the proposed system use the microphone to detect sound level of students as voice channel relates mutually with human behaviour expressions (Pentland, 2004; Vinciarelli et al., 2009). Temkar et al. (2016) uses biometric attendance,

Table 8 Comparison between the existing systems and the proposed one

Feature	Temkar et al., (2016)	Gligoric et al., (2015)	Aguilar et al., (2018)	Sin and Muthu, (2015)	Proposed system
Detect level of interest using learning analytics	✓	✓	✓	✓	✓
Learning analytics face recognition or facial expression	×	×	✓	✓	✓
Motion detection using:	✓	✓	×	×	✓
(a) Camera					
(b) 3-axis accelerometers	×	✓	×	×	×
(c) PIR sensor	✓	×	×	×	×
Sound recording using:	✓	✓	×	×	✓
(a) Microphone					
(b) Sound sensor	✓	×	×	×	×
Skill estimation by interaction with system through games	×	×	×	✓	×
Knowledge feedback loop (Learning Analytics)	×	×	✓	×	×
Biometric attendance (using pulse rate)	✓	×	×	×	×
Performance prediction using score	×	×	×	✓	✓
Formative performance using automated quiz	×	×	×	×	✓
Model prediction to apply LA	×	×	✓	✓	✓
RFID attendance	×	×	×	×	✓

that is, detecting pulse rate of students using pulse rate sensor to take attendance. However, with this concept, as students are under observance, they may not behave naturally as they are conscious, so this is not considered as the best solution (Temkar et al., 2016). So as a result, an RFID attendance is instead implemented in the proposed system. Furthermore, Gligoric et al. (2015) used the 3-axis accelerometer to analyze communications between lecturers and students which can be inaccurate as level of interest is normally not detected by communication only. Lastly, the proposed system, Aguilar et al. (2018) and Sin & Muthu (2015), use the concept of model prediction for the application of LA. By providing prediction of students' behaviour, LA can be applied to improve learning experience of students and can provide help to instructors.

8 Conclusion and Future Works

This chapter presents an IoT-based smart classroom system. The smart classroom consists of an automated attendance system, which has been setup using IoT devices namely, Arduino Uno and RFID reader. The attendance system

captures presence, calculate absence and lateness of students. Using data obtained from the RFID reader, a prediction is made on attendance of students. Furthermore, a camera is setup in the classroom which detects if students are sleepy during lectures by identifying drowsiness and yawns. A sound sensor has been used to capture sound level in the classroom. Based on the data obtained from the camera and sound sensor, behaviour of students is predicted. With these predictions, learning experience and level of interest can be improved among students during a lecture. The prediction models are trained using different algorithms to be able to make comparison for choosing the best one to obtain better and accurate results. Moreover, by helping lecturers to identify the performance and behaviour of students, the value of teaching is enhanced as time is being saved, thus satisfying the purpose of LA. Additionally, students can assess themselves by viewing their performance through graphs, which might be a help for future improvement. As future works, an accelerometer can be considered to detect if a student is raising hand. The movements of the body can be used as an additional parameter to improve the quality of analysis of behaviour. Moreover, a summative assessment system can be implemented where open-ended questions

may be used instead of multiple-choice questions. Analyzing the answers might be another way of applying LA.

References

- Aguilar, J., Sánchez, M., Cordero, J., Valdiviezo-Díaz, P., Barba-Guamán, L., & Chamba-Eras, L. (2018). Learning analytics tasks as services in smart classrooms. *Universal Access in the Information Society*, 17(4), 693–709.
- Brownlee, J. (2014). Feature selection in python with scikit-learn. *Machine Learning Mastery*, 14.
- Brownlee, J. (2016). Master machine learning algorithms: Discover how they work and implement them from scratch. *Machine Learning Mastery*.
- Brownlee, J. (2019). A gentle introduction to k-fold cross-validation. *Machine Learning Mastery*.
- Dietz-Uhler, B., & Hurn, J. E. (2013). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning*, 12(1), 17–26.
- Gandhi, S. L. (2017). Smart education service model based on IOT technology. In *International interdisciplinary conference on science technology engineering management pharmacy and humanities*.
- Glorigic, N., Uzelac, A., Krco, S., Kovacevic, I., & Nikodijevic, A. (2015). Smart classroom system for detecting level of interest a lecture creates in a classroom. *Journal of Ambient Intelligence and Smart Environments*, 7(2), 271–284.
- Grafsgaard, J., Wiggins, J., Boyer, K. E., Wiebe, E., & Lester, J. (2014). Predicting learning and affect from multimodal data streams in task-oriented tutorial dialogue. In *Proceedings of the 7th international conference on educational data mining*.
- Greller, W., & Drachslrer, H. (2012). Translating learning into numbers: A generic framework for learning analytics. *Educational Technology and Society*, 15(3), 42–57.
- Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process*, 5(2), 1.
- Kellen, V., Recktenwald, A., & Burr, S. (2013). Applying big data in higher education: A case study. *Cutter Consortium*, 13(8), 1–39.
- Larusson, J. A., & White, B. (Eds.) (2014). *Learning analytics: From research to practice* (Vol. 13). Springer.
- Mukherjee, S., & Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technology*, 4, 119–128.
- Nagowah, S. D., & Nagowah, L. (2009). Assessment strategies to enhance students' success. In *Proceedings IASK international conference, teaching and learning* (pp. 382–389).
- Nagowah, S. D., Meghoo, L., & Gaonjur, K. (July 2010). A mobile framework for formative assessment for first year students. In *2010 3rd International conference on computer science and information technology* (Vol. 4, pp. 492–499).
- Nagowah, S. D., Ben Sta, H., & Gobin-Rahimbux, B. A. (2019). An ontology for an IoT-enabled smart classroom in a university campus. In *2019 international conference on computational intelligence and knowledge economy (ICCIKE)* (pp. 626–631).
- Naseer, M., Zhang, W., & Zhu, W. (2020). Prediction of coding intricacy in a software engineering team through machine learning to ensure cooperative learning and sustainable education. *Sustainability*, 12(21), 8986.
- Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakash, I., & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*.
- Olmos, M., & Corrin, L. (2012). Learning analytics: A case study of the process of design of visualizations. *Journal of Asynchronous Learning Networks*, 16(3), 39–49.
- Pentland, A. (2004). Social dynamics: Signals and behavior. In *International conference on developmental learning* (Vol. 5).
- Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *Educause Review*, 46(5), 31–40.
- Sin, K., & Muthu, L. (2015). Application of big data in education data mining and learning analytics-A literature review. *ICTACT Journal on Soft Computing*, 5(4).
- Smith, V. C., Lange, A., & Huston, D. R. (2012). Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses. *Journal of Asynchronous Learning Networks*, 16(3), 51–61.
- Starkweather, J., & Moske, A. K. (2011). Multinomial logistic regression.
- Temkar, R., Gupte, M., & Kalgaonkar, S. (2016). Internet of things for smart classrooms. *International Research Journal of Engineering and Technology*.
- Vinciarelli, A., Pantic, M., & Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12), 1743–1759.
- Wang, Z., Jiang, F., & Shen, R. (2019). An effective yawn behavior detection method in classroom. In *International conference on neural information processing* (pp. 430–441). Springer Cham.
- Zhao, J. (2006). Research university faculty perceptions of smart classroom technologies (pp. 1–105). University of Arkansas



Predictive Analytics for Smart Health Monitoring System in a University Campus

Zakia N. S. H. Mohung, B. Unayza Boodoo,
and Soulakshmee D. Nagowah

Abstract

The Internet-of-Things (IoT) is modifying the infrastructure of technologies through interactions among various modules and components. It has enabled the setting up of complex systems such as smart homes, smart traffic control systems and smart environments. After COVID-19 pandemic, it is becoming more and more difficult to maintain a healthy and secure environment on university grounds. This chapter presents an IoT-based smart health system implemented on a university campus. The smart health system allows people on campus to closely keep track of their health status. A web application has been developed to provide real-time information of their vitals through medical sensors connected to a microcontroller (Arduino) for data acquisition. For disease prediction, a disease prediction module uses the sensor data and a health form to predict three main diseases: cold flu, hypertension and diabetes. To perform prediction, three models namely the cold flu model, hypertension model and diabetes model have been trained on different machine learning algorithms where the most accurate models are deployed in the web application. The cold flu model is evaluated using five different non-linear classification algorithms namely, decision tree (99%), random forest (99.5%), naïve bayes (94.9%), K-Nearest-Neighbour (89.7%) and SVM (55.3%) while hypertension model having a linear distribution is evaluated using three linear classification algorithms namely, logistic regression (86.0%), linear SVM (99.3%) and stochastic gradient descent (49.6%). Besides, the diabetes model is evaluated using logistic regression (88.7%), linear SVM (93.3%), decision tree (98.0%) and KNN (93.3%). The user is

alerted of his diagnosis by email. Moreover, the IoT-based smart health system consists of features such as online booking of appointments, health history and a medication section. Proper treatment can therefore be administered based on the users' health details, diagnosis and medication, if any.

Keywords

Internet-of-Things • Smart health • Sensors • Disease prediction • Machine learning

1 Introduction

The Internet-of-Things (IoT) relates to the practice where devices, applications and sensors are interconnected via a network (Banka et al., 2018). This connectivity allows data to be fetched and manipulated to obtain a desired output. Through IoT-based systems, several life-threatening diseases such as cardiovascular disease (CVD) or diabetes could be monitored. There are situations where doctors could not be reached or alerted on time during emergency cases and they fail to deliver necessary health assistance. Introducing IoT in the world of healthcare helps in keeping the patients under observation remotely through computing platforms and thus, nowadays IoT is being heavily implemented in the intensive care unit. The physiological parameters such as: blood pressure, oxygen saturation and body temperature of the patient are constantly monitored and compared with standard threshold values to identify the positive or negative presence of anomalies indicating a possibility of disease detected (Siam et al., 2019). Tracking sensor data keeps medical practitioners updated about patients' health conditions and assists them in real-time decision-making so that suitable actions can be taken and medical help can be sent on time (Bhatia & Sood, 2017). Incorporating IoT in health assistance has helped to achieve constant healthcare delivery

Z. N. S. H. Mohung · B. U. Boodoo · S. D. Nagowah (✉)
Software and Information Systems Department, Faculty of
Information, Communication and Digital Technologies,
University of Mauritius, Reduit, Mauritius
e-mail: s.ghurbhurrin@uom.ac.mu

anytime and anywhere and hence, efficiently diminishing preventable death rate (Senthamarasi et al., 2018).

Several smart health systems make use of machine learning algorithms and cloud-based databases. Such a measure is viewed as a significant asset in the decision-making process as it has the ability to manage large volume of data from various datasets. Cloud computing has the ability to handle busy and heavy processing of machine learning in a meaningful way. The standard analysis modelling includes neural network model, classification model and clustering methods with effective algorithms such as Fuzzy algorithm, Naïve Bayes algorithm and Random Forest algorithm among others (Kumar et al., 2018). Data can be populated from multiple sources with specific data types such as image, text and categorical data through the IoT devices. These data are then forwarded to a secured cloud repository and this information can be accessed via a web application by respective users logging in the system with their valid credentials. Several health systems using the classification approach, pursue the learning phase by mapping the data into two classes such as ‘Normal’ and ‘Disease Affected’ and identify the severity if an anomaly is positively detected (Kumar et al., 2018). CVD tops the chart of cause of million death worldwide every year (Jan et al., 2018). Predictive classifier models were developed to tackle these health issues where artificial intelligence, machine learning and data mining (DM) were used for medical diagnosis. DM is the extraction of meaningful data from large amount of data. To enhance the performance, the system first filters the collected data to eliminate noise and redundant data (Al-Makhadmeh & Tolba, 2019). Furthermore, these data are classified by labels, that is, categorical classes (for example, presence or absence). The predictive models are then trained using the training set (following the Pareto Principle where the dataset is split into training set (80%) and test set (20%)). Once the training data are analysed by the model, the classifier is used for classifying forthcoming data. Several researchers have adopted different algorithms and techniques such as classification, regression, Artificial Neural Network (ANN), Decision Tree (DT) or K-Nearest-Neighbour (KNN) to support doctor’s decisions by improving their accuracy of diagnosis.

This chapter demonstrates how a full-fledge smart health system on a university campus was implemented to monitor the student’s health parameters by capturing data through medical sensors, for instance: Infra-Red (IR) temperature sensor, pulse rate sensor, blood pressure (BP) sensor and a glucose sensor using a microcontroller (Arduino). Members of the university campus, that is, lecturers, students, administrative or other staff can register and have access to their respective data ensuring that privacy is maintained. The goal of this system is to collect both sensor data and health details from a health form in order to predict if the person is

healthy or if an anomaly is detected. Three models are trained using different machine learning algorithms namely DT, Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), KNN and Regression and the model with highest accuracy was implemented on the system for the prediction of three main diseases: cold flu, hypertension and diabetes. Upon abnormality detection, the system sends an alert and notifies the user of his current health condition. Based on each health checkup, a track record of the health vitals is created. For further assistance, the user may book an appointment with a registered doctor in the system by selecting an appropriate time slot. The rest of the chapter is structured as follows: Sect. 2 presents related work. Section 3 describes the proposed system. The use of predictive analytics in the healthcare system is illustrated in Sect. 4. Performance evaluation and model selection of different machine learning models are illustrated in Sect. 5. The implementation details of the system prototype are provided in Sect. 6. The proposed system is evaluated in Sect. 7. Finally, Sect. 8 presents the conclusion and future works.

2 Related Work

There exist several works that have been implemented by many researchers in the past by using various techniques for monitoring and disease prediction. Different systems have been analysed and reviewed in this section.

Kumar et al. (2018) proposed a three-phased cloud-based IoT system where in the first phase, medical data were acquired through: (1) wearable IoT devices attached to the patient where data were captured at regular intervals and compared with the normal values, sending an alert message to physician, (2) UCI repository dataset where values were mapped with actual IoT data and (3) medical records, where patients data for diabetes disease were obtained from multiple hospitals. Variety in data helps for severity analysis. In the second phase, the medical data were then stored in a secured cloud database providing adequate space accommodating large volume of data. Secure storage and retrieval algorithms (AES and DES algorithms) were used for securing the medical data when manipulated. The third phase has the responsibility in predicting the disease with its severity level and diagnose it. A Fuzzy rule-based classifier was proposed for effective decision-making by using fuzzy rules. The accuracy of the system was determined by calculating accuracy, precision and recall (sensitivity) function.

Ali et al. (2020) suggested a framework consisting of different layers for heart disease prediction using a deep-learning model and an ontology approach for data presentation. Physiological data such as electrocardiogram (ECG), an electroencephalogram (EEG), an electromyogram

(EMG), the heart rate, blood pressure (BP), position, activities, respiration rate, blood sugar, oxygen saturation and cholesterol levels of the patient were captured through sensors and sent via Bluetooth and WIFI devices. Observation reports containing medical, smoke and diabetes history were also analysed (structured/unstructured data). Both structured and unstructured data from sensors were combined in a comma-separated value (CSV) file. Based on the compiled CSV file, the following steps were carried out: 1. Data fusion, 2. Pre-processing, 3. Deep-learning heart disease prediction, 4. Ontology-based recommendation. The pre-processed data were passed to the deep-learning classifier and the model was trained using a heart disease dataset. Mathematical functions (probability) converted the heart disease features into values and calculated the weight for each value used in the deep-learning model. The ontology then displays a plan for dietary intake or activities as per the patient's current condition.

Pandey and Prabha (2020) came up with an IoT monitoring and disease prediction framework where pulse rate and temperature sensors data were acquired through a microcontroller (Arduino) and different machine learning algorithms were implemented for heart disease prediction. First, the sensors were attached to the finger of the patient in order to retrieve a pulse reading. After information pre-processing, that is, converting the raw data and making it more suitable for the machine learning model, the data values were passed into a model, which was trained using a dataset. 40 samples of data were collected and different machine learning classifiers such as SVM, NB, RF, DT and KNN were applied to the model, leading to an accuracy of 86%, 83%, 83%, 74% and 75%, respectively. The SVM has returned a better accuracy for the heart disease prediction.

Bhatia and Sood (2017) proposed an IoT-assisted smart workout with a healthcare prediction using ANN. The system is divided in different layers, each having a specific task. Firstly, numerous IoT devices such as smart wearable sensors, actuators and bio-sensors, RFID are used to gather real-time health-oriented (physiological) and non-health-oriented (environmental and behavioural) variables. Data about dietary intake are obtained by RFID tags. These data are then forwarded to a cloud storage for classification and feature extraction. Since collected data are classified as sensitive, SSL (Socket Shell) protocol and authentication was used for data security. The data are then classified into two classes (Vulnerable Health-oriented Dataset Class and Non-Vulnerable Health-oriented Dataset Class) based on probabilistic health-influencing parameter known as Degree of Vulnerability (DoV). To determine the DoV values, Bayesian Belief Network (BBN) model is used. The abstraction layer carries out temporal mining (extract data values and patterns from database) of real-time data to

identify vulnerabilities in the form of a probabilistic value called Probabilistic State Vulnerabilities (PSoV). Furthermore, the PSoV values are utilised to train the ANN model through back propagation ANN to predict health services during exercise sessions.

3 Proposed System

This section presents a full-fledge smart health system on a university campus that was implemented to monitor staff and student's health parameters. Due to the COVID-19 pandemic, this prototype was developed to constantly monitor the health status of each individual on campus, which result in efficient temperature screening and ensuring a secure environment on university grounds. Figure 1 illustrates the smart health system. The vitals' data are captured through the following medical sensors as shown in Fig. 2: Infra-Red (IR) temperature sensor, pulse rate sensor, blood pressure (BP) sensor and a glucose sensor using a microcontroller (Arduino board) and are stored in a database. Owing to the unavailability of BP sensor and a non-invasive glucose sensor due to COVID-19 circumstances and also absence of these two sensors components in Arduino simulators such as Tinkercad, an electronic blood pressure monitor and a normal invasive glucose monitor were used as alternatives and the data were entered manually. The values are then displayed through a web application. Most sensors made use of IR and are highly sensitive to motion. A switch was thus included in the circuit where each user needs to switch on/off when required after vitals' parameters are successfully captured. Members of the university campus can register and have access to their respective data and other services, ensuring that privacy is maintained through strong authentication. The goal of this system is to use both collected sensors' numerical data and the user's health details regarding health symptoms gathered from a health questionnaire for predicting whether a person is healthy or not. If an anomaly is detected, a diagnosis is then rendered. In addition, these recorded parameters are being utilised for the prediction of diabetes, cold flu and hypertension, which are common ailments among young adults. Furthermore, for the sensor monitoring section, the software tools used to acquire the sensors were Arduino IDE, used to write and upload commands for the sensors and Spyder python version 3.8 from Anaconda, to forward the data to database. For the development and setting up of virtual local server for the web application, Komodo IDE and XAMPP were used, respectively. Moreover, other features in particular: online booking of appointments, health history and a medication section were implemented. To ensure accurate treatment, user's health details, diagnosis and medication, if any, were used.

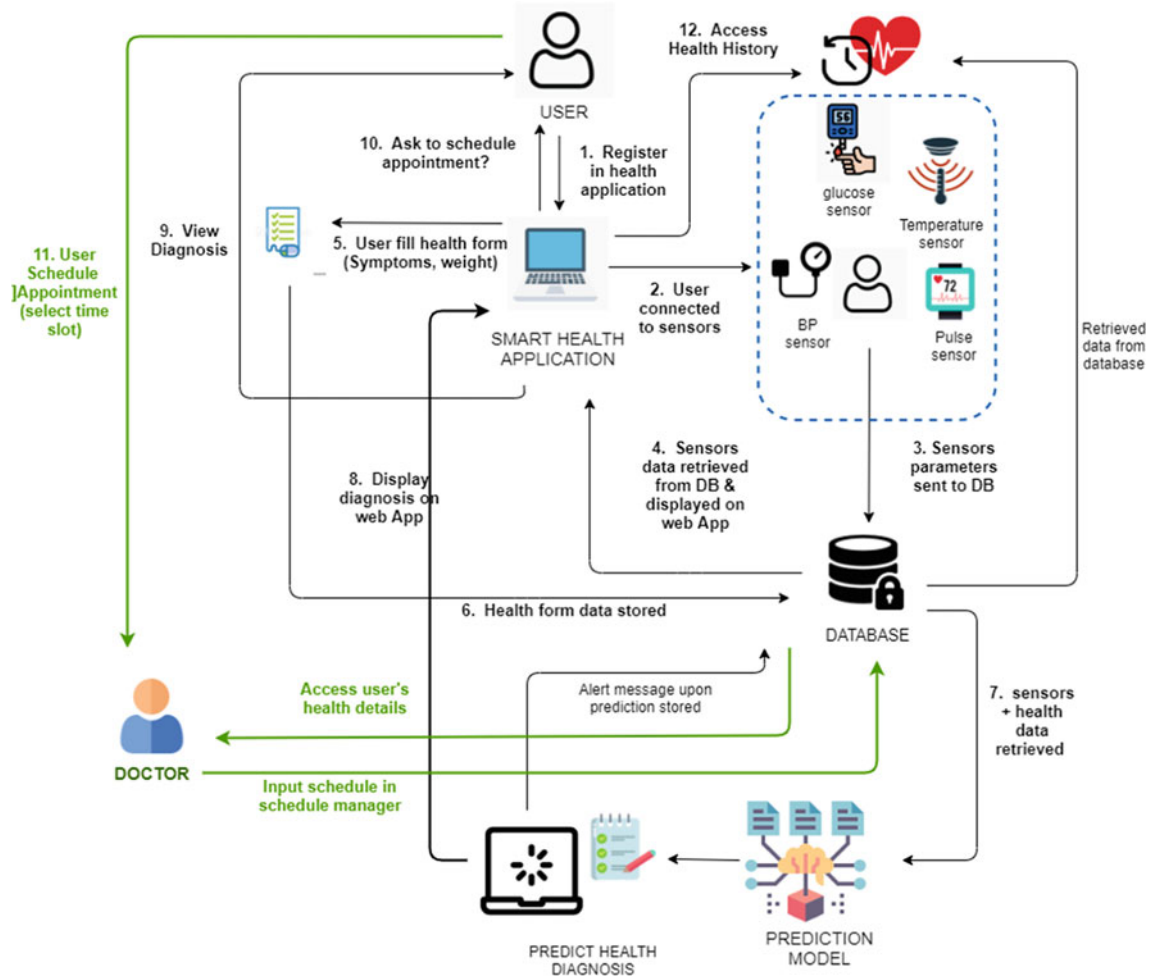
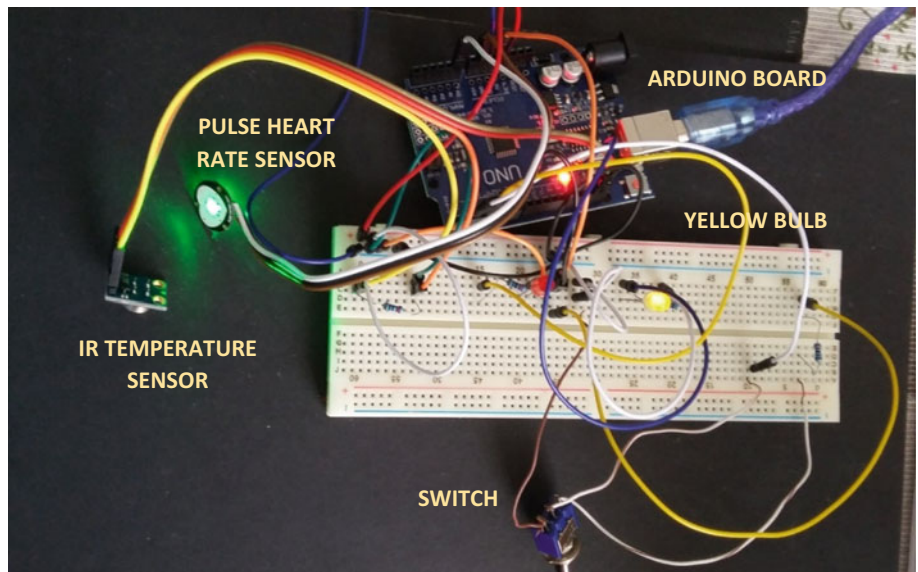


Fig. 1 Diagram of proposed system

Fig. 2 Arduino set up with sensors



In view of disease prediction, the system constitutes three models namely the cold flu model, the hypertension model and the diabetes model. The cold flu model, having a lot of overlapping, is evaluated using non-linear algorithms namely, DT, RF, NB, KNN and SVM while hypertension model, having a linear distribution is evaluated using Logistic Regression (LR), linear SVM and Stochastic Gradient Descent. However, the diabetes model is evaluated using both linear and non-linear classification algorithms namely: LR, linear SVM, Decision Tree and KNN due to some overlapping of data. The dataset for each model was manually populated. When applying the 70/30 rule, there will be sufficient data for the training set (70%) to train the model and 30% as a test set which determines the accuracy of the model. Upon diagnosis obtained, if a disease is detected, the system sends an alert message via email notifying the user of his current health condition. For further assistance and clarification, the user may schedule an online appointment with a registered doctor in the system by selecting an appropriate time slot. For users already under medication, a medication section is available prompting the users to enter their daily dosage if any, so that proper and accurate healthcare can be administered. Medical practitioners wishing to extend their help can register in the smart health system, where a schedule is provided. Based on each health checkup, a track record of the health status is generated in the health history.

4 Predictive Analytics in Smart Health

This section provides detailed explanation on how the predictive models, namely: cold flu model, hypertension model and diabetes model have been implemented. In addition, a brief description of machine learning algorithms considered are provided.

4.1 Applying Machine Learning Algorithms for Disease Prediction

Machine Learning learns from the surrounding environment and existing data to imitate human intelligence by using computational algorithms (Naqa & Murphy, 2015). Supervised machine learning can be categorised as classification and regression. In classification, the predicted outcome should be of discrete values such as ‘cold flu’ or ‘no cold flu’, ‘hypertension’ or ‘no hypertension’ and ‘diabetes’ or ‘no diabetes’ while regression uses continuous values to be predicted. Classification can further be split into linear and non-linear classification. Algorithms considered for linear classification are LR, Stochastic Gradient Descent

(SGD) and Linear SVM while the considered algorithms for non-linear classification are DT, RF, KNN, NB and Non-Linear SVM. LR is an algorithm where the output has a binomial distribution (0 and 1) (Rymarczyk et al., 2019). The cost function, which is the Sigmoid Function, is used to draw the curve, which separates the two classes. LR calculates a probability to determine to which category it belongs to (Rymarczyk et al., 2019). SGD is an iterative algorithm that aims at optimising loss functions (Elshoush & Dinar, 2019; Nawaz et al., 2021). For each iteration, a sample of data is selected randomly instead of the whole dataset to calculate the gradient (Nawaz et al., 2021). SVM can be both linear and non-linear. In Linear SVM, the data is divided linearly by a hyperplane (Suthaharan, 2016) having a maximum marginal distance between the support vectors of both categories (Pisner & Schnyer, 2020). In non-linear SVM, the data is separated by having each axis in the scatter plots defined by a feature space using the kernel functions, which will provide the separation of data (Suthaharan, 2016). DT is a tree structure, which contains a root node, decision nodes, leaf nodes and branches. At each decision node, the branches will represent a result in binary output (Yes or No). The tree is split using variables such as Gini Index, entropy, gain ratio, classification error and information gain (Song & Ying, 2015). RF is an ensemble classifier, which uses a set of Decision Trees for prediction (Belgiu & Drăguț, 2016; Petkovic et al., 2018). DT is trained by using sample data from the datasets (Belgiu & Drăguț, 2016). New data passes through all the DT and the outcome is based on votes (Petkovic et al., 2018). KNN will have a set of data (training data) classified. When the model gets an unknown data, the model determines the neighbours by calculating the distances and classifies the data by the majority class of the neighbours (Zhang et al., 2017). NB assumes all the parameters as independent variables and calculates the conditional probability of each class given each feature (Jiang et al., 2010). The conditional probability for both classes is then compared for prediction outcomes.

4.2 Dataset Construction

Datasets had to be constructed for the three models as some of the available datasets contained data that could not be measured. Besides, as our system focuses on a campus environment, the factors that will determine the disease, particularly among students and staff are identified to create the dataset. The three models are bi-level classification models thereby, the dataset is constructed having only two possible outcomes, which are, 0 and 1. Due to Covid-19 pandemic, real data could not be obtained from hospitals or clinics. The data in the dataset have been manually inserted

and then verified by a medical practitioner to ensure correctness of data.

A. Cold Flu Dataset

According to Eccles (2005, 2009), the common symptoms of cold flu are: sore throat, runny nose, headache, cough, muscle pain and fever. In addition, Long (2020) stated that these respiratory diseases along with other illnesses can cause your heart rate to increase slightly compared to your normal heart rate. Considering those articles, the cold flu dataset is built with the following parameters: body temperature, heart rate, respiratory rate, sore throat, body pain, runny nose, headache and cough. The cold flu dataset consists of 648 rows having 324 rows representing 'cold flu' and the other 324 rows representing 'no cold flu'.

B. Hypertension Dataset

Based on Ali et al. (2018); AlKaabi et al., (2020); Sooklall et al., (2018); Tadesse & Alemu, (2014); Tegnah et al., (2019), it is noted that the common risk factors of having hypertension among adult students are BMI, Systolic Blood Pressure, Diastolic Blood Pressure, Smoke, Physical Activity Level and Heredity. The dataset for this model is thus constructed using these attributes. The hypertension dataset consists of 496 rows having 248 rows representing hypertension and the other 248 rows representing no hypertension.

C. Diabetes Dataset

Considering the studies conducted by Al Amiri et al. (2015); Antwi et al., (2020); Khlaifat et al., (2020); Nagowah & Joaheer, (2018); Moonian et al., (2020), the factors that can determine the possibility for young adults to have diabetes are: Glucose Level, Systolic Blood Pressure, Diastolic Blood Pressure, BMI, Physical Activity Level and Heredity. The dataset for the diabetes model is thus built based on these factors. The diabetes dataset consists of 512 rows having 256 rows representing 'diabetes' and the other 256 rows representing 'no diabetes'.

5 Performance Evaluation of Each Model and Selection

In this section, the three models (Cold Flu, Hypertension and Diabetes) are evaluated using different algorithms based on the classification metrics namely: accuracy score, Precision, Recall and F1-score.

Precision is the measure of correctly predicted positive class from the total positive class predicted (Hossin & Sulaiman, 2015). It is given by the formula,

$$\text{Precision} = \text{True positive}/(\text{True Positive} + \text{False Positive}) \quad (1)$$

Recall is the actual number of positive classes predicted (Hossin & Sulaiman, 2015). It is given by the formula,

$$\text{Recall} = \text{True Positive}/(\text{True Positive} + \text{True Negative}) \quad (2)$$

F1-Score is the harmonic mean between precision and recall (Hossin & Sulaiman, 2015). A model is considered more accurate when the F1-score value is nearly equal to 1. It is given by the formula,

$$\text{F1-Score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \quad (3)$$

True Positive can be defined as the case where the model has correctly predicted a positive class while *True Negative* is the case where the model has correctly predicted a negative class. *False Negative* is the outcome where the model has incorrectly predicted a negative class and *False Positive* is the outcome where the model has incorrectly predicted a positive class.

In addition, to have a visual representation of how the different models are working, an ROC (Receiver Operating Characteristic) curve is plotted. AUC (Area under the curve) can be used to compare the performance of different classifiers (Hossin & Sulaiman, 2015; Liu et al., 2014). ROC is a 2-dimensional graph where true positive rate is plotted against false positive rate (Liu et al., 2014), which shows the overall performance of the models (Hossin & Sulaiman, 2015).

5.1 Cold Flu Model

To determine which algorithms to apply, the dataset is visualised as shown in Fig. 3 which demonstrates how each parameter relates to each other based on the outcome (cold flu or not cold flu). It is noted that there is a lot of overlapping, especially between pulse rate and other parameters such as temperature, respiratory rate, running nose, headache and cough. Due to a lot of overlapping, non-linear algorithms are used to train the model. All the features (temperature, pulse rate, respiratory rate, sore throat, body ache, running nose, headache and cough) from the dataset are extracted to train the model as they all contribute in the possibility of having cold flu. The dataset is divided into training and testing data in a ratio of 7:3. 453 rows are used for training the model and the remaining 195 rows are used to test the model. The model is trained by importing each algorithm from the 'sci-kit learn' library where the train set is fitted and the score for each model is calculated using the test set. For the KNN algorithm, the number of k is

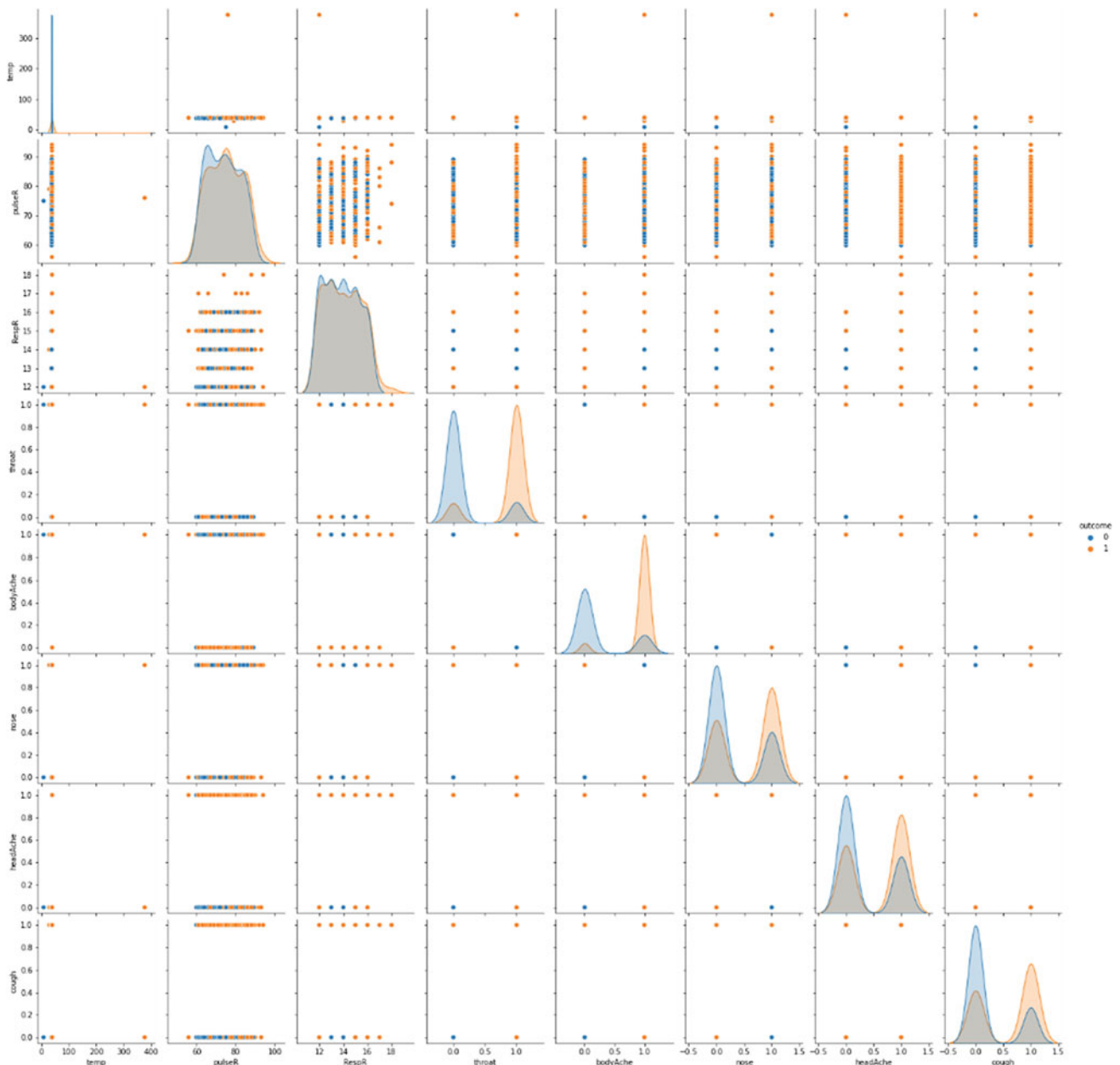


Fig. 3 Cold flu data visualisation

calculated by applying square root to the total number of training data. The testing data is used to evaluate the different algorithms as shown in Table 1.

From Table 1, it is noted that RF has the highest accuracy. A cross-validation, with five folds, is applied on all the algorithms to ensure that the model is not overfitting. In cross-validation, the model does not depend on how the dataset is split (Bengio & Grandvaet, 2004). In this technique, the dataset is separated into k number of blocks of the same size, in this case, the dataset is separated into five blocks. The training data set will consist of four blocks while the testing data set will consist of one block of data. The

models will then be trained and tested iteratively with different training and testing data set. The accuracy score of the five folds is compared in Table 2.

From Table 2, it is clear that the model is not overfitting as all the unknown test data have got similar accuracy. The possible reason for having 100% accuracy score is because of the way the algorithms work. They make use of a tree structure that makes decision at each node and each representing a rule. As RF is performing better than DT with different set of test data, RF is chosen to be deployed.

Figure 4 shows the ROC graph, which compares the different models based on the area under the curve. The

Table 1 Evaluation metrics for cold flu model

Algorithms	Accuracy score (%)	Precision	Recall	F1-score
DT	99.0	0.98	1.00	0.99
RF	99.5	0.99	1.00	0.99
KNN	89.7	0.92	0.86	0.89
NB	94.9	0.99	0.90	0.94
SVM	55.3	0.51	0.86	0.64

Table 2 Accuracy score with different test data

Number of folds	DT (%)	RF (%)	NB (%)	KNN (%)	SVM (%)
1st fold	100	100	97.6	96.9	60.2
2nd fold	98.1	100	96.1	91.5	64.4
3rd fold	99.0	99.5	98.9	93.1	56.9
4th fold	96.0	97.8	96.1	94.6	57.9
5th fold	95.3	98.9	91.5	89.8	54.7

Fig. 4 ROC curve of cold flu algorithms

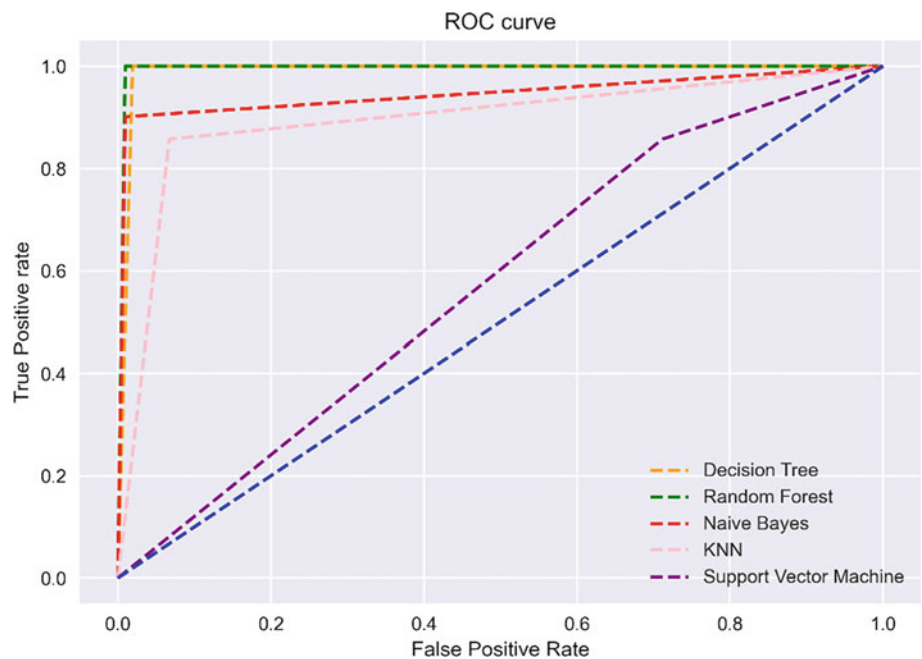


figure aims at demonstrating how well certain algorithms are working being above the straight diagonal line and having a greater area compared to other algorithms. Figure 4 shows that RF is performing slightly better than DT by having the perfect curve where it has the highest area under curve compared to DT, NB, KNN and SVM.

5.2 Hypertension Model

The hypertension dataset is visualised using the ‘seaborn’ library to identify the algorithms to be used as shown in Fig. 5. The plotted diagram shows how each parameter

relates to each other based on the output (hypertension or not hypertension). Since the data has a linear distribution, linear algorithms are considered. The three linear algorithms: LR, SGD and SVM, are used to train the model. The dataset is loaded using the ‘pandas’ library where all the features (systolic, diastolic, BMI, smoking, physical activity level, heredity) from the dataset are used to train the model. The dataset is separated into training and testing datasets in a ratio of 7:3 by using the ‘train_test_split’ function from the ‘sci-kit learn’ library. 347 rows are used as training set and 149 are used as testing set. The hypertension model is trained with each algorithm imported from the ‘sci-kit learn’ library, by fitting the training data set to them. The models

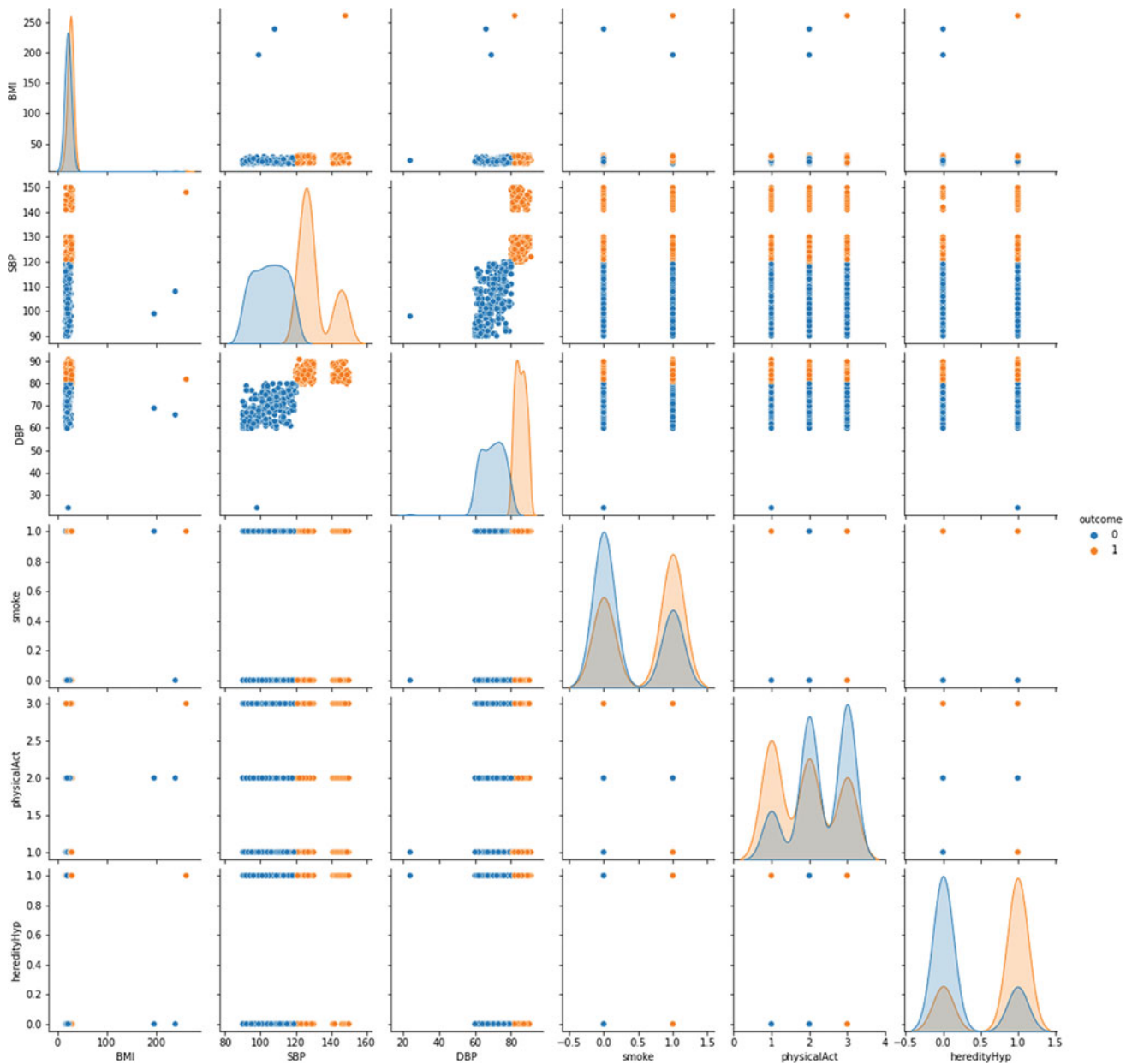


Fig. 5 Hypertension data visualisation

are then evaluated based on the classification metrics mentioned above in Sect. 5.0 using the test data. Table 3 shows the different metrics for the hypertension model.

From Table 3, it is evident that the best model is SVM as it is 99.3% accurate and has a F1-score of 0.99. There is a clear separation between the two outcomes and each feature

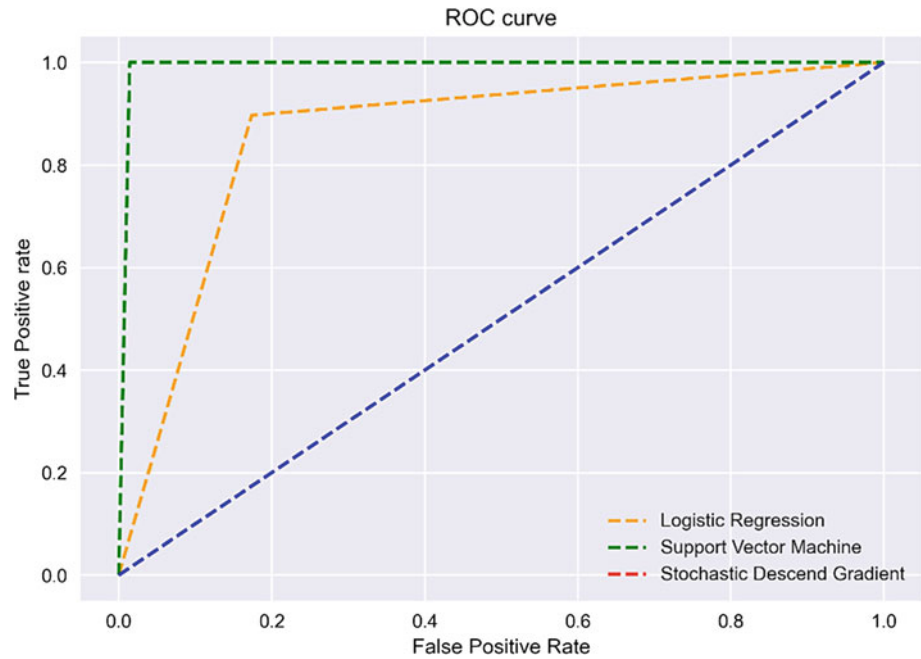
contributes equally for the prediction of the outcome, based on the ‘feature importance’ function, thus resulting in high accuracy score and F1-score value.

The ROC curve is shown in Fig. 6 to demonstrate the performance of the LR, SGD and SVM algorithms. It is proven that the performance of SGD is low as its line lies in

Table 3 Evaluation metrics for hypertension model

Algorithms	Accuracy score (%)	Precision	Recall	F1-score
LR	86	0.84	0.90	0.87
SGD	49.6	0.50	1.00	0.66
SVM	99.3	0.99	1.00	0.99

Fig. 6 ROC curve of hypertension algorithms



the same blue diagonal line, which is used as a reference. SGD overlaps with the blue diagonal line, resulting in the red line (SGD) not visible. It is also proven how well is SVM performing by having nearly the perfect line with the highest area under curve compared to the other two algorithms. It is noted that the LR algorithm has its curve drawn below the SVM curve, which eventually results in having its area smaller than the SVM algorithm. Likewise, for SGD, its curve is even lower than SVM and LR, which will result in having an even smaller area.

5.3 Diabetes Model

The same concept is applied to the diabetes model where the dataset is visualised as shown in Fig. 7 using the pair-plot diagram from the 'seaborn' library to determine the algorithms that will be used. The diagram describes the relationship between each parameter based on the outcome (diabetes or no diabetes). It is noted that most data are well classified as diabetes and no diabetes. However, there are some data overlapping with BMI against physical activity level and heredity. As these little overlapping should not be neglected, a combination of both linear and non-linear classification algorithms is used.

The training dataset is fitted in the following algorithms: LR, SVM, DT and KNN. All features (glucose level, systolic, diastolic, BMI, physical activity level and heredity) are extracted to train the model as all the features are important factors that affect the model. The dataset is then loaded in the notebook using the 'pandas' library. Those algorithms are

called from the 'sklearn' library where the 70% of the training set data (358 rows) are applied to each algorithm and the remaining 30% data (154 rows) are used to evaluate the performance of the different algorithms using the classification metrics as shown in Table 4.

From Table 4, it is noted that DT has the highest accuracy. The model is evaluated with different test data using cross-validation technique to ensure that the model is not overfitting as shown in Table 5.

From Table 5, it is noted that DT is performing better with the different test data and has the highest accuracy with the test data in Table 4. Therefore, the DT model is saved and deployed on the web application.

In addition, an ROC curve is plotted to compare the performance of the different algorithms as shown in Fig. 8. Since the accuracy of SVM and KNN is the same (based on the test data in Table 4), the two algorithms' lines overlap with each other. Hence, only the KNN line is visible in the graph where SVM also lies. DT has the best accuracy value as demonstrated in Fig. 8. The DT line is closely drawn to the ideal curve which should be a right-angled line. SVM and KNN having similar accuracy values using test data from Table 4, results in their lines overlapping with each other. Ultimately, DT has the highest area under the curve compared to LR, SVM and KNN.

6 System Prototype

This section presents the implementation details for the system prototype.

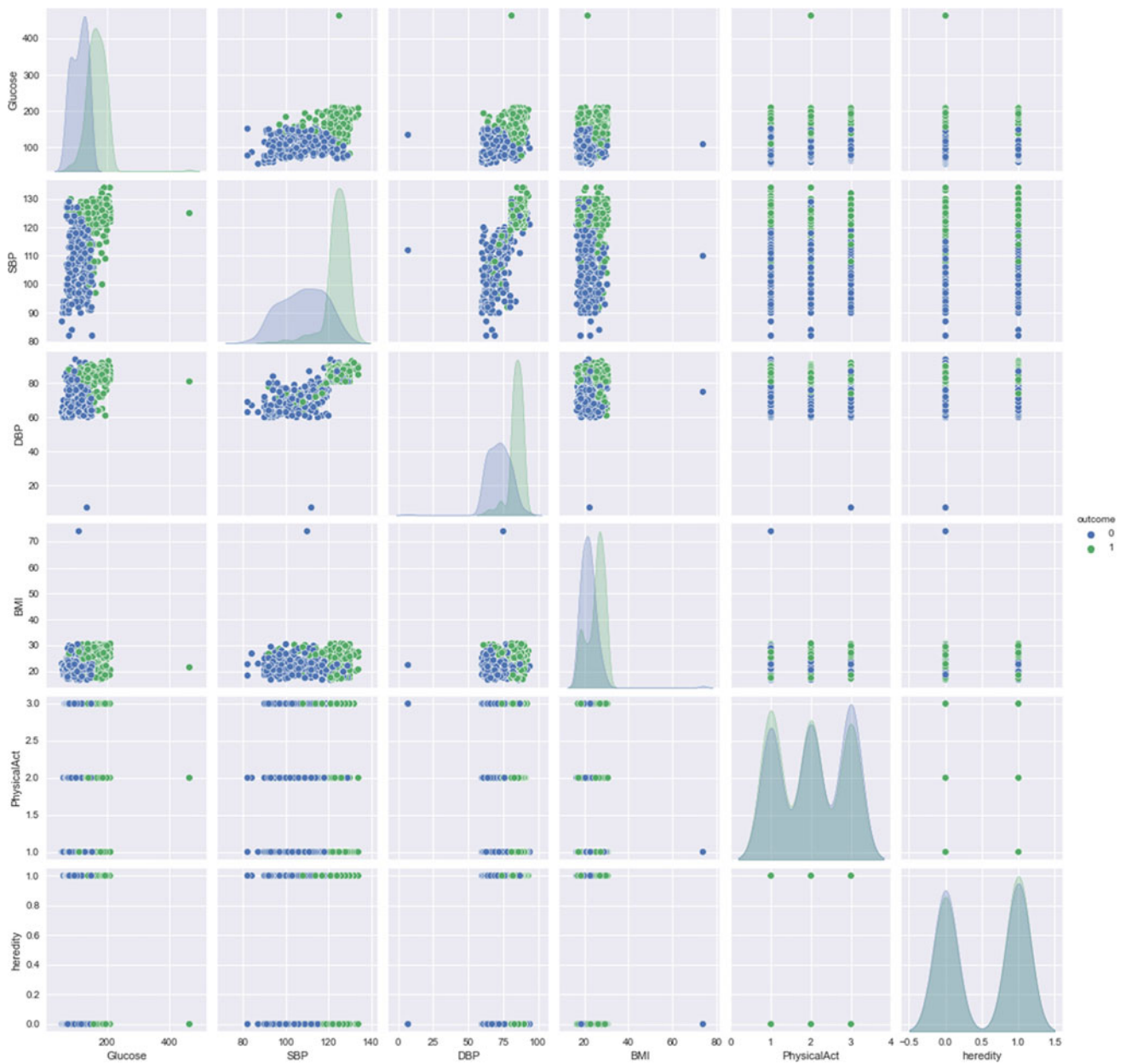


Fig. 7 Diabetes data visualisation

Table 4 Evaluation metrics for diabetes model

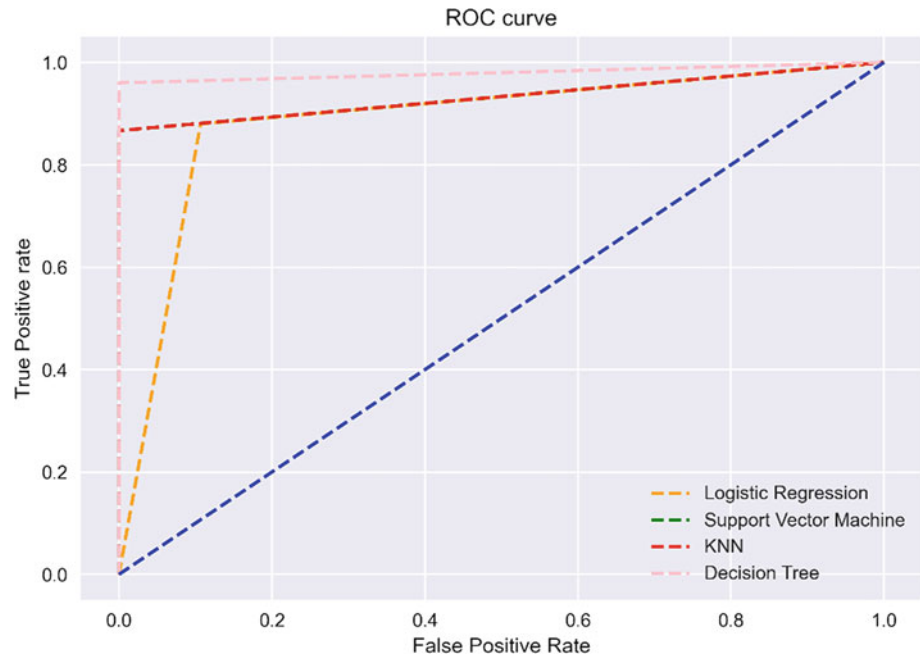
Algorithms	Accuracy score (%)	Precision	Recall	F1-score
LR	88.7	0.89	0.88	0.89
SVM	93.3	1.00	0.87	0.93
DT	98.0	1.00	0.96	0.98
KNN	93.3	1.00	0.87	0.93

On the sensor monitoring side, to capture the parameters to be used to identify cold flu, hypertension and diabetes, a temperature sensor, pulse rate sensor, glucose monitor and a BP monitor have been used to predict a diagnosis. The data

collected via sensors were forwarded to the database by a python script using Spyder version 3.8 by Anaconda Navigator. The libraries to be imported are serial, time and MySQLdB. Serial library has been used to get data from the

Table 5 Accuracy score with different test data for diabetes

Number of folds	LR (%)	SVM (%)	DT (%)	KNN (%)
1st fold	97.0	99.0	99.0	98.0
2nd fold	94.0	96.0	98.0	97.0
3rd fold	82.0	87.0	94.0	84.9
4th fold	82.7	89.9	92.0	88.9
5th fold	82.9	97.0	91.0	97.0

Fig. 8 ROC curve for diabetes algorithms

Arduino serial monitor, time library has been used to obtain the serial monitor data at intervals and mySQLdb library has been used to send Arduino values to phpMyAdmin database. Arduino IDE has been utilised to write command scripts and upload on the sensor to perform operations. A web application has been developed and hosted using Komodo IDE and XAMPP server, respectively. Moreover, the collected data from sensors are displayed on the application. Other parameters such as glucose level, systolic and diastolic for BP, weight and height for BMI and respiration rate were manually inserted. A health form has been designed to record additional health symptoms, which are signs that cannot be measured, for instance, headache and both sensors and form data have been used for disease detection.

The three models, cold flu model, hypertension model and diabetes model, have been trained and evaluated as explained in Sects. 5.1, 5.2 and 5.3. These models are implemented using *Jupyter Notebook* from *Anaconda* (<https://docs.anaconda.com/anaconda/user-guide/faq/>). The best models, RF for cold flu, SVM for hypertension and DT for diabetes are then deployed on a python file by using the ‘pickle’ library. This file will serve as a web API by using

the ‘flask’ framework. The web application allows the user to collect real-time sensor data such as temperature, pulse rate, blood pressure and glucose level and to fill the health questionnaire where the user chooses which symptoms of cold flu he is having, his physical activity level, whether he smokes or not and lastly, whether his close relatives have hypertension and diabetes. In addition, the user is prompted to enter his respiratory rate, height and weight. These data are then sent to the URL where the python file is running using the CURL function in PHP. The file gets the request and the arguments consisting of the sensor data and health form data. These data are then partitioned and sent to their respective models, that is, temperature, pulse rate, respiratory rate, sore throat, body ache, running nose, headache and cough are passed to the cold flu model, BMI, systolic blood pressure, diastolic blood pressure, smoke, physical activity level and hypertension heredity are sent to the hypertension model and at last, glucose level, systolic blood pressure, diastolic blood pressure, BMI, physical activity level and diabetes heredity are passed to the diabetes model. The three models will then predict the possibility of having each disease, respectively. The API will then return the result in json

Table 6 Hardware specifications

Features/Devices	Laptop 1	Laptop 2
Processor	Intel® Core™ i5-7200U @2.50 GHz 2.71 GHz	Intel® Core™ i3-6006U CPU @ 2.00 GHz 2.00 GHz
RAM	8.00 GB	16.0 GB
Hard disk	916 GB	916 GB
Operating system	Windows 10	Windows 10
System type	64-bit operating system	64-bit operating system

Table 7 Software specifications

Component	Specification
OS	Windows 10
Development platform	– Jupyter notebook from Anaconda – Spyder from Anaconda – Komodo edit
Web server	XAMPP
Database	PhpMyAdmin
Sensor platform	Arduino IDE

format to the web application where the data are decoded and displayed. The user may send the result to his email where each prediction is retrieved from database and sent using the *PHPMailer* library. The web application is tested using two laptops whose specifications are given in Table 6. The software that is used to implement the web application and to develop the models are provided in Table 7.

accurate predictions. Most systems have considered only one algorithm to implement the prediction model, which has the ability to predict only one disease. It is noticed that (Bhatia & Sood, 2017) has not predicted any disease but rather it predicts the health status vulnerability. In addition, the common metric used to evaluate the performance of the model is accuracy. Kumar et al. (2018) have used two additional metrics, that is, precision and recall which focus only on the true positive prediction.

7 System Evaluation

Table 8 shows the comparison of the different features of the existing systems in Sect. 2 against the proposed system.

It is noted that only (Kumar et al., 2018) has considered a health form along with the sensor data to make more

This chapter has proposed a model, which has the possibility of predicting more than one disease namely cold flu, diabetes and hypertension. Our proposed model made use of a health form filled by the user, which is used along with the sensor data to make more efficient and accurate predictions. The three predictive models proposed are all compared with

Table 8 Feature comparison with existing systems

Features	Kumar et al., (2018)	Ali et al., (2020)	Pandey and Prabha, (2020)	Bhatia and Sood, (2017)	Proposed model
Sensor data collection	✓	✓	✓	✓	✓
Health form	✗	✓	✗	✗	✓
Alert system	✗	✗	✗	✗	✓
ML techniques	Fuzzy rule-based classifier	Deep learning	KNN, SVM, NB, DT, RF	BBN	LR, SGD, SVM, DT, RF, KNN, NB
Performance metric used	Accuracy, precision, recall	Unknown	Accuracy	unknown	Accuracy, precision, recall and F1-score
Disease predicted	Diabetes	Heart disease	Heart disease	PSoV	Cold flu, diabetes, hypertension

Your sensor's data		Symptoms
Temperature:	37.1	Throat Pain Headache Smoke: No Physical: Active Relative Hypertension: No Relative Diabetes: Yes
Pulse Rate:	75	
Respiratory Rate:	14	
Systolic BP:	123	
Diastolic BP:	83	
Glucose:	135	
BMI:	24.1	

Fig. 9 Student health parameters data

Results:		
You are fine! No flu detected.	There is a high risk of having Hypertension!	There is a low risk of having Diabetes.

Fig. 10 Prediction result of the student

different algorithms based on their data distribution. The most effective model is determined by calculating their accuracy, precision, recall and F1-score, which is then deployed on the web application. However, the datasets are constructed by random values whose outcome can be 0 or 1. The ideal way to construct the dataset would have been to get real data from the hospital. Figure 9 illustrates a table showing the real-time sensor values on the left-hand side and symptoms recorded from a health questionnaire on the right-hand side, for a particular student. These displayed values are then used for the diagnosis prediction. Figure 10 shows how the prediction output of the three predictive models: cold flu, hypertension and diabetes are displayed in terms of its intensity that is, high or low risk, based on acquired sensors and form data from Fig. 9, of that student. As there is a large audience in a campus, communicable diseases can easily be spread. Therefore, a monthly graph is plotted as shown in Fig. 11, which displays the total number of people having cold flu in the current month, on campus. The cold flu rate is determined by the registered number of cold flu cases detected in that month.

8 Conclusion and Future Works

This chapter explains how a smart health system has been implemented in a university campus. The proposed system collects data from sensors along with health form data to make prediction on mainly three diseases: cold flu being the most common communicable disease in a campus along with hypertension and diabetes, which are increasingly affecting youngsters. A variety of algorithms are selected to train the three predictive models. The most accurate being RF for cold flu, SVM for hypertension and DT for diabetes are deployed. These models predict the diseases with high accuracy score. The proposed system additionally allows users of the campus to monitor their health status without the intervention of a doctor. The system also helps to maintain a healthy environment in the university campus as each individual health can be monitored. Besides, a healthy environment will eventually ensure better performance of students. In the future, the system can incorporate more sensors and more diseases to predict. Moreover, the dataset

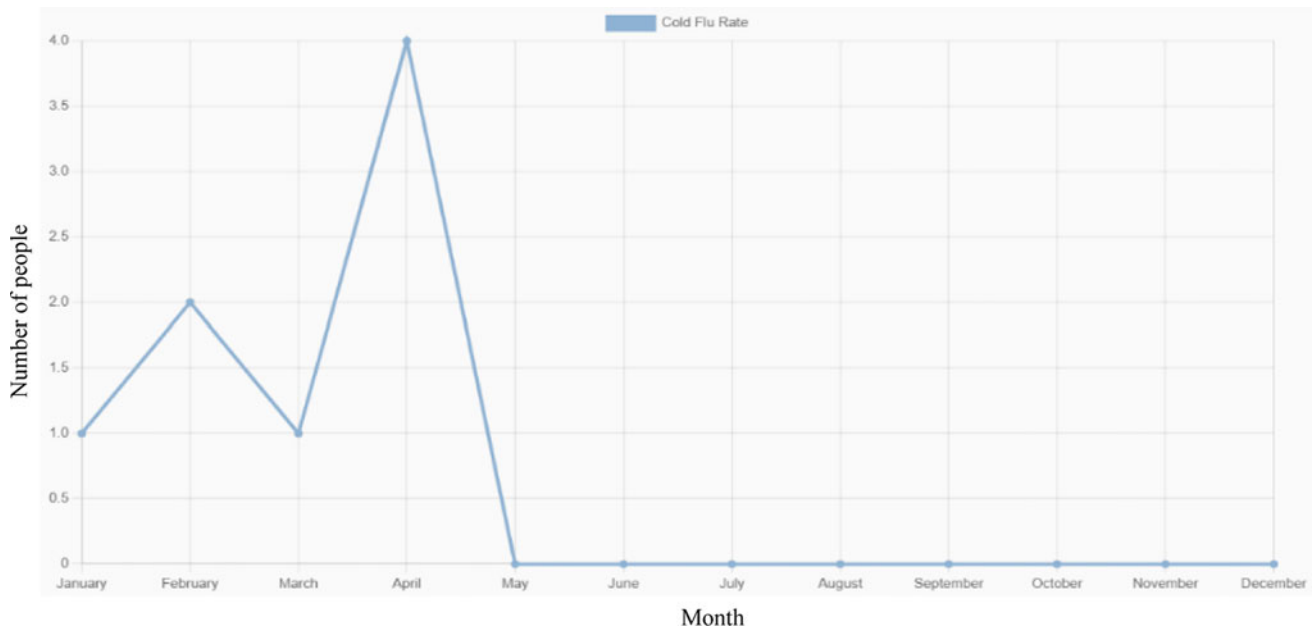


Fig. 11 Cold flu rate graph

can be improved by using real patients' data to get more accurate results. The system implemented is flexible and embedded with general features and therefore, can be further extended at any workplace and not only on a campus.

References

- Al Amiri, E., Abdullatif, M., Abdulle, A., Al Bitar, N., Afandi, E. Z., Parish, M., & Darwiche, G. (2015). The prevalence, risk factors, and screening measure for prediabetes and diabetes among Emirati overweight/obese children and adolescents. *BMC public Health*, *15* (1), 1–9.
- Ali, F., El-Sappagh, S., Islam, S. R., Kwak, D., Ali, A., Imran, M., & Kwak, K. S. (2020). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, *63*, 208–222.
- Ali, N., Mahmood, S., Manirujjaman, M., Perveen, R., Al Nahid, A., Ahmed, S., Khanum, F. A., & Rahman, M. (2018). Hypertension prevalence and influence of basal metabolic rate on blood pressure among adult students in Bangladesh. *BMC Public Health*, *18*(1), 1–9.
- AlKaabi, L. A., Ahmed, L. S., Al Attiyah, M. F., & Abdel-Rahman, M. E. (2020). Predicting hypertension using machine learning: Findings from Qatar biobank study. *PLoS ONE*, *15*(10), e0240370.
- Al-Makhadmeh, Z., & Tolba, A. (2019). Utilizing IoT wearable medical device for heart disease prediction using higher order Boltzmann model: A classification approach. *Measurement*, *147*, 106815.
- Antwi, J., Lavin, R., Sullivan, S., & Bellavia, M. (2020). Perception of and risk factors for type 2 diabetes among students attending an upstate New York college: A pilot study. *Diabetology and Metabolic Syndrome*, *12*, 1–8.
- Banka, S., Madan, I., & Saranya, S. S. (2018). Smart healthcare monitoring using IoT. *International Journal of Applied Engineering Research*, *13*(15), 11984–11989.
- Belgiu, M., & Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31.
- Bengio, Y., & Grandvaet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, *5*, 1089–1105.
- Bhatia, M., & Sood, S. K. (2017). A comprehensive health assessment framework to facilitate IoT-assisted smart workouts. A predictive healthcare perspective. *Computers in Industry*, *92*, 50–66.
- docs.anaconda.com. (n.d.). Frequently asked questions—Anaconda documentation. [online] Available at: <https://docs.anaconda.com/anaconda/user-guide/faq/>. [Accessed 30 July 2021].
- Eccles, R. (2005). Understanding the symptoms of the common cold and influenza. *The Lancet Infectious Diseases*, *5*(11), 718–725.
- Eccles, R. (2009). Mechanisms of symptoms of common cold and flu. In *Common cold* (pp. 23–45). Birkhäuser Basel.
- Elshoush, H. T., & Dinar, E. A. (2019). Using adaboost and stochastic gradient descent (sgd) algorithms with R and orange software for filtering e-mail spam. In *2019 11th computer science and electronic engineering (CEECE)* (pp. 41–46). IEEE.
- El Naqa, I., & Murphy, M. J. (2015). What is machine learning?. In *Machine learning in radiation oncology* (pp. 3–11). Springer, Cham.
- Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process*, *5*(2), 1.
- Jan, M., Awan, A. A., Khalid, M. S., & Nisar, S. (2018). Ensemble approach for developing a smart heart disease prediction system using classification algorithms. *Research Reports in Clinical Cardiology*, *9*, 33–45.
- Jiang, L., Cai, Z., & Wang, D. (2010). Improving naive Bayes for classification. *International Journal of Computers and Applications*, *32*(3), 328–332.
- Khlaifat, A. M., Al-Hadid, L. A., Dabbour, R. S., & Shoqirat, N. (2020). Cross-sectional survey on the diabetes knowledge, risk perceptions and practices among university students in South Jordan. *Journal of Diabetes and Metabolic Disorders*, 1–10.

- Kumar, P. M., Lokesh, S., Varatharajan, R., Babu, G. C., & Parthasarathy, P. (2018). Cloud and IoT based disease prediction and diagnosis system for healthcare using Fuzzy neural classifier. *Future Generation Computer Systems*, 86, 527–534.
- Liu, Y., Zhou, Y., Wen, S., & Tang, C. (2014). A strategy on selecting performance metrics for classifier evaluation. *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, 6(4), 20–35.
- Long, S. (2020). How to tell if you have a common cold or a full-blown case of the flu—SheKnows. [online] Sheknows.com. Available at: <https://www.sheknows.com/health-and-wellness/articles/842117/the-common-cold-versus-the-seasonal-flu-whats-the-difference/>. [Accessed 25 April 2021].
- Moonian, O., Jodheea-Jutton, A., Khedo, K. K., Baichoo, S., Nagowah, S. D., Nagowah, L., Mungloo-Dilmohamud, Z., & Cheerkoot-Jalim, S. (2020). Recent advances in computational tools and resources for the self-management of type 2 diabetes. *Informatics for Health and Social Care*, 45(1), 77–95.
- Nagowah, S. D., & Joaheer, R. (2018). A model for classifying people at risk of diabetes mellitus using social media analytics. In *International conference on emerging trends in electrical, electronic and communications engineering* (pp. 195–204). Springer, Cham.
- Nawaz, M. S., Shoaib, B., & Ashraf, M. A. (2021). Intelligent cardiovascular disease prediction empowered with gradient descent optimization. *Heliyon*, 7(5), e06948.
- Pandey, H., & Prabha, S. (2020). Smart health monitoring system using IOT and machine learning techniques. In *2020 sixth international conference on bio signals, images, and instrumentation (ICBSII)* (pp. 1–4). IEEE.
- Petkovic, D., Altman, R. B., Wong, M., & Vigil, A. (2018). Improving the explainability of Random forest classifier-user centered approach. In *PSB* (pp. 204–215).
- Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine learning* (pp. 101–121). Academic Press.
- Rymarczyk, T., Kozłowski, E., Kłosowski, G., & Niderla, K. (2019). Logistic regression for machine learning in process tomography. *Sensors*, 19(15), 3400.
- Senthamilarasi, C., Rani, J. J., Vidhya, B., & Aritha, H. (2018). A smart patient health monitoring system using IoT. *International Journal of Pure and Applied Mathematics*, 119(16), 59–70.
- Siam, A. I., Abou Elazm, A., El-Bahnasawy, N. A., El Banby, G., Abd El-Samie, F. E., & Abd El-Samie, F. E. (2019). Smart health monitoring system based on IoT and cloud computing. *Menoufia Journal of Electronics Engineering Research*, 28, 37–42.
- Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2), 130.
- Sooklall, R., Tengnah, M. A. J., & Nagowah, S. D. (2018). A proposed framework for hypertension in Mauritius. *Journal of Health Informatics in Africa*, 5(1), 16–27.
- Suthaharan, S. (2016). Support vector machine. In *Machine learning models and algorithms for big data classification* (pp. 207–235). Springer.
- Tadesse, T., & Alemu, H. (2014). Hypertension and associated factors among university students in Gondar, Ethiopia: A cross-sectional study. *BMC Public Health*, 14(1), 1–5.
- Tengnah, M. A. J., Sooklall, R., & Nagowah, S. D. (2019). A predictive model for hypertension diagnosis using machine learning techniques. In *Telemedicine technologies* (pp. 139–152). Academic Press.
- Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1774–1785.



SysML-Based Design of Autonomous Multi-robot Cyber-Physical System Using Smart IoT Modules: A Case Study

Qasem Abu Al-Haija

Abstract

Generally, cyber-physical systems (CPS) are composed of a group of computing devices that communicate with one another and interact with the physical world via sensors and actuators in a feedback loop. Indeed, CPS systems became available everywhere, such as the team of autonomous mobile robots with prescribed tasks and missions. This chapter proposes a detailed architectural model-based design for a cyber-physical system of autonomous multi-differential drive robots (DDRs) to visualize the map in their path through a forest environment with vertical cliffs on each side in a regular time-based manner. The robots shall travel toward the east with a maximum velocity of 15 miles per hour to end the trip at a pre-specified point located at 15 miles-linear distance from the starting point. Several system specifications, constraints, and requirements were considered in developing this CPS system. Finally, SysML modeling is designed for the proposed system and reported in this report, including requirement, activity, state machine, block definition, internal block definition, and parametric diagrams. Hence, the proposed architecture design provides a generic model-based system engineering (MBSE) for robotic designers to develop customized autonomous DDRs in a collaborative network with arbitrary hardware components and software configurations.

Keywords

Smart city applications • Cyber physical system • Differential robot • SysML modeling • MagicDraw • Visual paradigm • Microcontroller • Ultrasonic sensors • Motors • GSM module • H-Bridge controller • GPS module

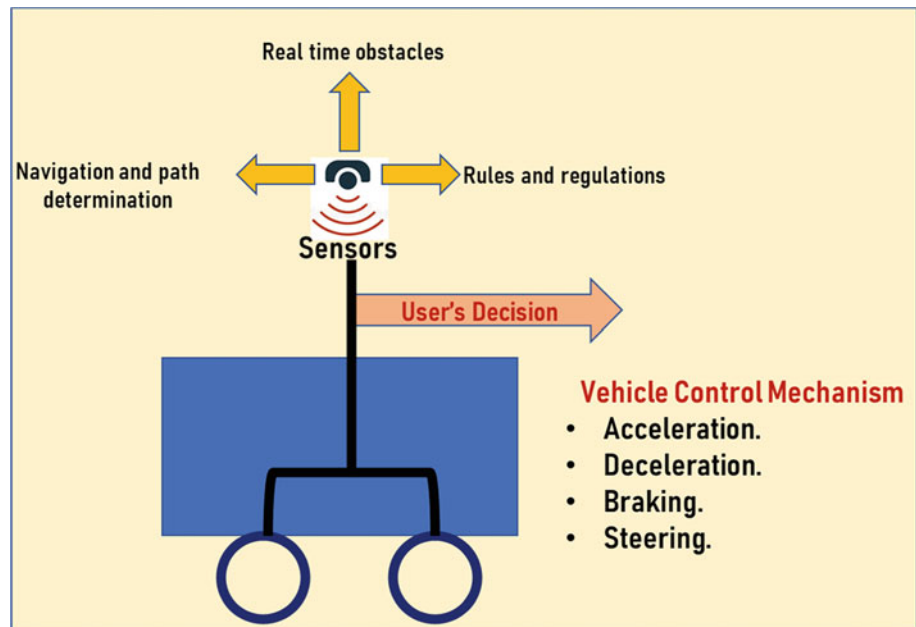
1 Introduction

Recent advances in machine learning, control, and sensing technology offered an essential infrastructure to develop enhanced autonomous vehicles and robotics (AVR) for different real-life applications (Hruska, 2018). AVR functionality relies heavily on utilizing low-cost sensing, powerful embedded system hardware, and communication networks to develop an extensive range of different applications (Bekey, 2005; Interim Report & on 21st Century Cyber-Physical Systems Education, Division on Engineering & Physical Sciences, 2015). Nowadays, many industries and governments are employing these types of autonomous robots (Alur, 2015) in real-time monitoring and data collection applications such as environmental monitoring, automation, security, and surveillance.

Unlike ordinary vehicles, autonomous vehicles such as the differential drive robot (DDR) system integrate a processor unit with several sensors, motors, and some other arbitrarily chosen elements along with vehicle control mechanism as well as a navigation strategy to function in a specific environment within prescribed operational requirements and constraints (RobotShop Community Blogs, <https://www.robotshop.com/>). However, autonomous vehicles or mobile robots (Al-Haija et al., 2020) have many challenges related to navigation control and decision-making strategy. Figure 1 depicts the functionality of the ordinary vehicle under human control and the autonomous vehicle under the control sensors and processor.

Q. Abu Al-Haija (✉)
Department of Computer Science/Cybersecurity, King Hussein
School of Computing Sciences, Princess Sumaya University for
Technology (PSUT), Amman, Jordan
e-mail: q.abualhaija@psut.edu.jo

Fig. 1 Human-based vehicle versus autonomous vehicle



According to Fig. 1, two types of vehicles are illustrated, namely ordinary and autonomous vehicles. Both vehicles consider three parameters to be configured at the input side of the controller and four outcomes/decisions to be generated accordingly at the output side of the vehicle's controller. The input parameters include navigation and path determination, real-time obstacles, and rules and regulations. The output control decisions are related to vehicle acceleration, deceleration, braking, and steering. The significant difference between ordinary and autonomous vehicles is that the control is wholly established through human intervention in the typical vehicle. In contrast, the control is utterly established in the autonomous vehicle through a collection of sensors and embedded processor/microcontroller intervention.

In this chapter, we consider a cyber-physical system (Alur, 2015) that comprises a team of autonomous mobile robots (differential drive robots) tasked with identifying and retrieving a target inside a forest with an unknown path plan. Indeed, we propose a comprehend SysML model-based system for this cyber-physical system (CPS) based on the proposed top view diagram given in Fig. 2 (along with the DDR system description provided in the next section). Such a model is built as a general model that can be customized with any elements and subsystems.

Smart cities depend on information technology, the Internet of things, and artificial intelligence to improve the quality of life at all levels, such as government service, transportation systems, medical systems, indoor and outdoor citizen welfare, and others. For example, autonomous vehicles have a significant impact on improving the life inside cities by self-operating and accomplishing indispensable tasks deprived of human involvement. Autonomous vehicles can help reduce the number of crashes on the roads of intelligent cities. It has

been identified that autonomous vehicles have reduced the number of hits by 94% due to driver behavior/error. Indeed, autonomous vehicles have many applications that make our lives and cities more intelligent. The revolution of autonomous vehicles is illustrated in Table 1.

This chapter is organized as follows: Section 2 reviews the recent related research work correlating machine learning with cyber-physical systems. Section 3, system description and requirements, describes system features, specifications, and requirements. Section 4, system modeling using SysML, provides all the graphical modeling environments to visualize the multi-robot system's structure, behavior, requirements, and parametric. Finally, Section 5, conclusions and remarks, concludes the work in this chapter.

2 Machine Learning-CPS-Related Research

At present, machine learning techniques have emerged and engaged with several aspects of real-life applications such as information technology applications, medical healthcare applications, pharmaceutical applications, industrial and automation applications, robotics applications, and many others. Primarily, it has been coupled with several cyber-physical systems (CPSs) to provide autonomous and intelligent services for several CPSs and IoT systems such as instruction detection (Al-Haija et al., 2020), driver assistance systems (Kurihata et al., 2005), cybersecurity (Al-Haija, 2021), weather detection for autonomous vehicles (Al-Haija et al., 2020), transport control systems (Tatarinov & Kir-sanov, 2019), and others. For instance, the authors in Al-Haija et al. (2020) suggested a high-performance structure for weather identification in the autonomous vehicles'

Fig. 2 Top view diagram for DDR system entities

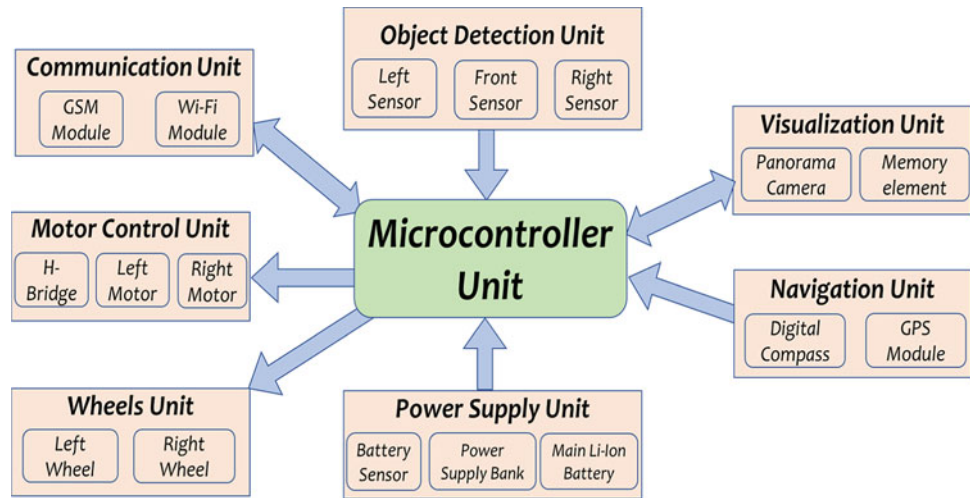


Table 1 Revolution of autonomous vehicles (self-driving): Three eras growth

Era 1	Era 2	Era 3
2015–2020	2020–2030	2030–2050
Fully autonomous vehicles (AVs) being developed for consumers	Consumers begin to adopt autonomous vehicles (AVs)	Autonomous vehicles (AVs) become the primary means of transport system
<ul style="list-style-type: none"> Autonomous vehicles (AVs) are already a reality in industrial fleets Car OEMs (original equipment manufacturer) begin to assess strategic impact New mobility models begin to emerge 	<ul style="list-style-type: none"> The after-sales service landscape is reshaped Insurers shift from covering individuals to covering technical failures Supply chain and logistics are redefined 	<ul style="list-style-type: none"> AVs free up to 50-min a day for drivers Parking space is reduced by billions of square meters Vehicles crash fall by 90% saving billions of dollars AV technology accelerates development of robots for consumer use

cyber-physical system by employing deep learning techniques such as ResNet-18 convolutional neural network. Their developed scheme uses transfer learning approaches of the deep ResNet-18 CNN pre-trained on ImageNet to classify the weather for an autonomous vehicle into four labels, namely: sunrise, shine, rain, and cloudy. As a result, their model scored a remarkable classification accuracy of 98.22% that outperformed several other states of art schemes.

Also, Pereira and Thomas (2020) have utilized machine learning (ML) techniques to automate the control of safety-critical cyber-physical systems (CPS) for critical application areas. They presented an inclusive and operational investigation for the safety hazards which might appear during the ML life cycle and might compromise the innocent process of ML-based CPS. To better illustrate their advanced solution, they have demonstrated the recognized hazards using a real-world CPS system, namely the autonomous shop-floor vehicle. Eventually, the authors claimed that their work forms the basis for many future holistic safety approaches for ML-based cyber-physical systems, especially in critical CPS applications. In a counter study, authors of Dreossi et al. (2019) presented an investigation study of

compositional falsification of cyber-physical systems with machine learning components. According to their research, since CPSs have started to rely heavily on employing machine learning (ML) mechanisms at a different level of development. Therefore, the precision and accuracy of CPSs are governed by the characteristics of the internal machine learning components. Thus, the authors have addressed the compositional falsification of correlating CPS with ML by proposing a compositional falsification framework where a temporal logic falsifier and a machine learning analyzer cooperate to find falsifying executions of the considered model. To prove the effectiveness of their proposed model, they have implemented with automatic emergency braking cyber-physical system utilizing the deep neural networks.

Moreover, McKee et al. (2018) introduced a comprehensive study building intelligent autonomous control cyber-physical systems by adopting machine learning paradigms in automating, augmenting, and integrating systems across the domains of smart cities, autonomous vehicles, energy efficiency, smart manufacturing in Industry 4.0, and health care. Besides, they have provided a review for several ML-CPS challenges such as safety and security, big data, a

model of reality, augmentation of systems, and computation. Another intelligent IoT system for real-world traffic light systems has been developed (Al-Haija & Jebri, 2020). The author proposed a bright ARM Cortex M4-based traffic light system to ensure safe traffic flow and minimize accidents for the ground vehicles. Their implementation results exhibited their intelligent system's effectiveness in insuring and organizing road traffic and evading car accidents.

Furthermore, Boursinos and Koutsoukos (2001) developed a machine learning-based conformal prediction for assurance monitoring of cyber-physical systems. To validate their predictive model, they employed the German Traffic Sign Recognition Benchmark and a robot navigation dataset that exhibits small error rates and alarms. Thus, the approach is encouraging for assurance monitoring of CPS. In addition, Tran et al. (2020) proposed a neural network verification (NNV) tool for deep learning-enabled cyber-physical systems (CPS) using feed-forward neural networks algorithms. They assessed the NNV scheme using several practical techniques, such as the safety verification of a deep learning-based adaptive cruise control system.

Finally, the literature is full of cyber-physical systems employing several machine learning techniques at different development layers of the CPSs (Castaño et al., 2017; Dartmann et al., 2019; Ruchkin et al., 2016; Sokolov et al., 2020; Wickramasinghe et al., 2018). In our work of this chapter, we propose a comprehensive SysML-based architectural model for designing a cyber-physical system of an autonomous multi-differential drive robot that can visualize the map in their path within a forest environment with vertical cliffs on each side in a regular time-based manner. The proposed system is intelligent, and machine learning and fuzzy logic methods (Wang et al., 2018) can be adopted at several modules of the CPS, such as the autonomous motion can be developed using fuzzy-based motion model, the autonomous obstacle avoidance can be designed using supervised machine learning techniques such as convolutional neural network (CNN) (Fei-Fei, 2019), K-nearest

neighbors (KNN) (Meneses et al., 2019), decision tree method (DTM) (Song & Lu, 2015), majority voting method (MVM) (Tama & Rhee, 2017), support vector machine (SVM) (Ghose, 2017), and the autonomous braking system can be developed using reinforcement learning techniques in which the DDR can learn from the environment.

3 System Description and Requirements

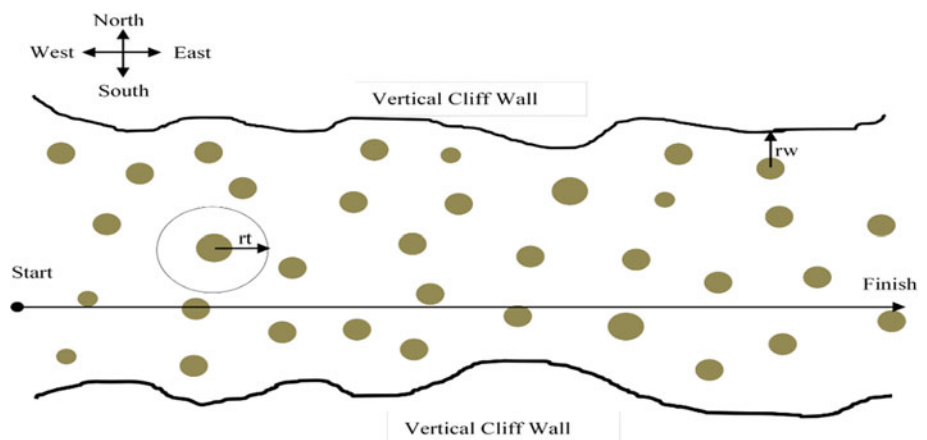
Indeed, the construction of any cyber-physical system requires designing strategies for control, computing, and communication collaboratively. The proposed multi-robot system to be developed to meet the objectives mentioned above can be described in terms of the forest environment, operational specifications, and differential robot specifications. Figure 3 illustrates the forest operating environment for DDR, composed of two-sized trees and surrounded by two cliff-walls on both sides (North and South).

The general characteristics for the forest environment are summarized as follows:

- The circumference base of the biggest tree is 3 feet, and the smallest is 1 foot.
- Each tree will have a minimum radius r_t of 15 feet to the closest trees.
- Each tree will have a minimum radius r_w of 10 feet to the cliff wall.
- The start to finish linear distance of the environment is 15 miles.
- All trees have no branches lower than 20 feet above the ground.

Besides, the operational specifications for system are summarized as follows: The system shall have a maximum velocity of 15 miles per hour, visual maps of the area 360° around the system shall be taken every one minute, while the visual mapping is occurring, the location of the system must

Fig. 3 System environment diagram



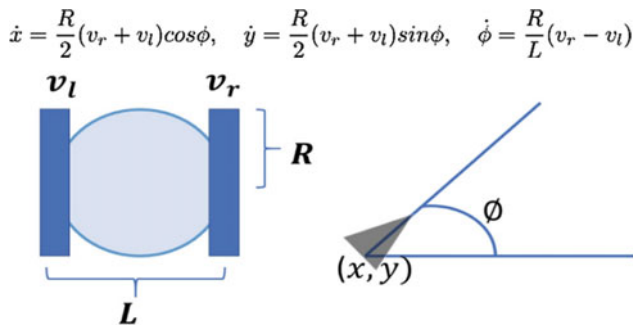


Fig. 4 DDR velocity and heading diagram

not change, the navigation of the environment shall be less than 2 h, and the proximity sensor shall have a linear range not less than 5 feet. Moreover, Fig. 4 shows the velocity and heading diagram for the differential drive motor.

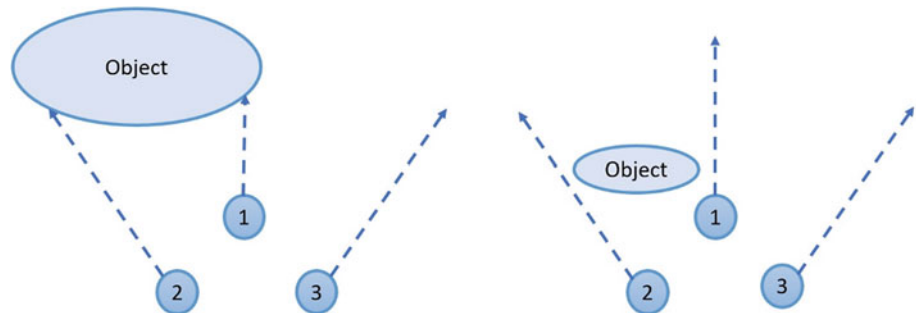
Furthermore, the design specifications and configurations for the differential drive robot are listed as follows:

- The navigation control scheme of proximity sensors is presented in Fig. 5 and used only for obstacle avoidance and control strategy to maintain a heading of due east. Sensor configuration must account for operational specifications and the environment characteristics.
- Must include a visual system and must account for operational specifications and the environment characteristics.
- Robot configuration such as shape, size, weight, and components must be derived in the SysML.
- The voltage of motor will have a range from 0 to 12 V.
- Modeling of DDR motors can be kept at a minimum such as the first-order system where:

$$\omega(t) = \frac{K_m}{\tau \cdot s + 1} \text{ and } \omega(t) = K_{om}V(t) \quad (1)$$

where: $\omega(t)$ is the angular velocity (rad/sec) of the motor, K_m is the gain constant of the motor, τ is the motor time constant, and k_{om} is the conversion constant for the motor voltage to the angular velocity.

Fig. 5 DDR example: three sonar sensors diagram



4 SysML Modeling of Autonomous Mobile Robots System

System Modeling Language (SysML) is a model-based system engineering (MBSE) tool (Friedenthal et al., 2009; Delligatti, 2014) that provides a graphical modeling environment to visualize the structure, behavior, requirements, and parametric of a system and communicate that information to others. SysML defines nine kinds of diagrams to express various aspects of system design information as illustrated in SysML hierarchy diagram shown in Fig. 6.

Indeed, several visual modeling tools are existing nowadays to provide rich graphical architectural models for systems engineering at the system development phase prior to the physical implementation of system elements and entities. One of the leading and powerful system modeling tools is the MagicDraw software platform (No Magic Inc., 2015). MagicDraw 19.0 provides a SysML plugin that includes SysML profile, template, and diagrams with specific SysML menus, tool bars, diagrams, specification dialogs, and user interface. In this work, we have utilized the capabilities and features of MagicDraw 19.0 to build up the SysML modeling for the prescribed autonomous differential drive robot using seven diagrams: package diagram, block definition diagram, internal block diagram, activity diagram, state machine diagram, parametric diagram, and requirement diagram.

4.1 System Package Diagram

The package diagram (*pkg*) is used to display the overall organization of the system model by grouping the model elements into a name space (Friedenthal et al., 2009). The system model can be organized in multiple ways including: by system hierarchy (e.g., enterprise, system, and component), by diagram kind (e.g., requirements, use cases, and behavior), and by viewpoints to augment model organization (by integrated product teams (IPT)). The package diagram for the proposed DDR system is shown in Fig. 7 in which the system model elements are organized by diagram kind.

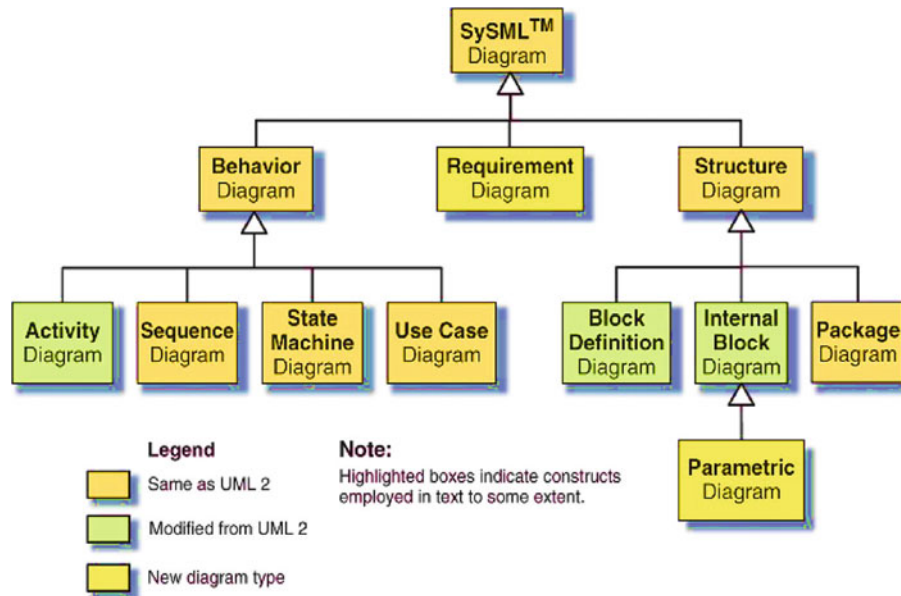


Fig. 6 SysML taxonomy of diagram types (Batarseh & McGinnis, 2012)

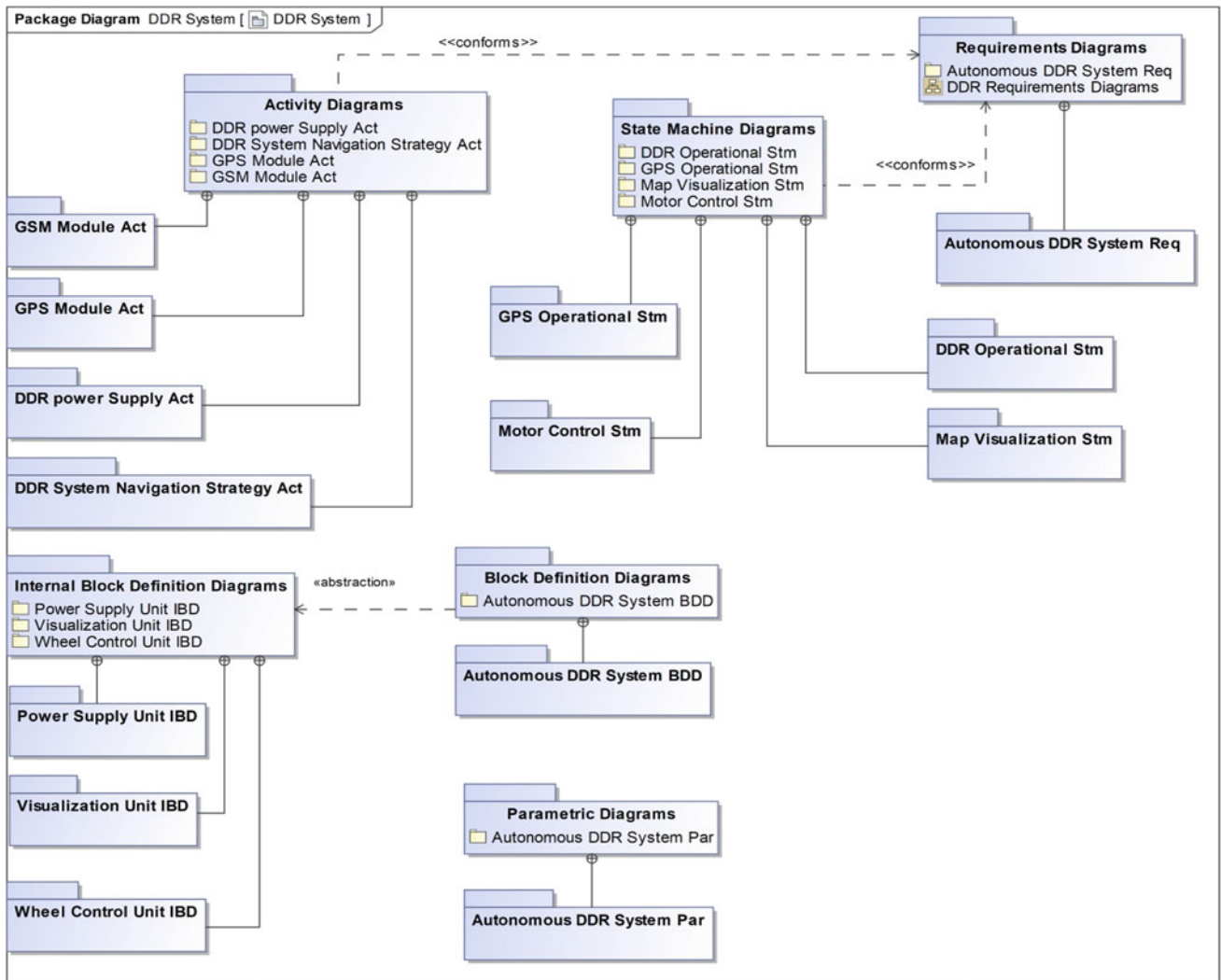


Fig. 7 Autonomous DDR system package diagram

4.2 System Requirements Diagram

The requirement diagram (**req**) is used to display the text-based requirements of the system being modeled and their relationships to other model elements (Friedenthal et al., 2009). Several dependency relationship types can be specified by the requirement diagram such as satisfy, trace, verify, refine, and derive. The requirements diagram for the proposed DDR system is shown in Fig. 8 which clearly specifies the system requirements (both provided and derived) and their possible relationships such as tractability between maximum velocity requirement and the navigation duration, or the requirements for power supply which should satisfy the navigation duration which may last for two operational hours. Another example is that the 360° map visualization should be verified to be taken with fixed system location.

such as hardware, software, and actors (persons) (Friedenthal et al., 2009). Each block in the structure can be characterized by multiple standard compartments such as values, operations, constraints, and references. The block definition diagram for the proposed DDR system is shown in Fig. 9. It decomposes the DDR system into eight subsystems (units) such as communication unit, microcontroller, and object detection unit. For each unit, the bdd shows the associated elements and other characteristics. For example, the object detection unit is a realization of its three sonar sensors since all of them inherit its values such as the linear detection range. Another example, the power supply subsystem has a possible constraint to be considered such as the battery life, while the navigation unit can be specified by its operations such as *get direction information ()* and *get location information ()*.

4.3 System Block Definition Diagram

The block definition diagram (bdd) provides a unifying concept to describe the structure of a system components

4.4 System Internal Block Diagrams

The internal block diagram (*ibd*) is used to specify the internal structure of a single block to be emphasized using a

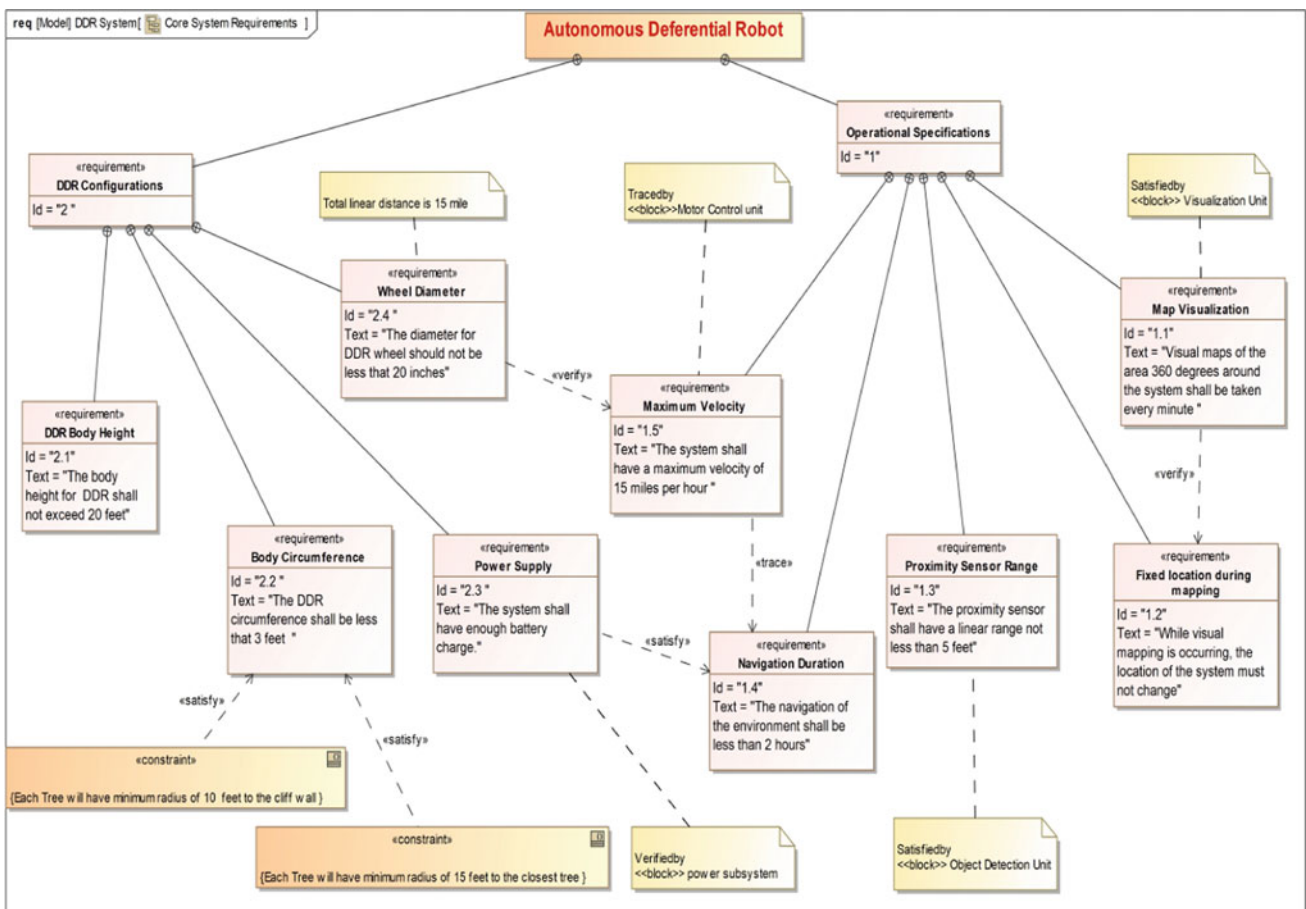


Fig. 8 Autonomous DDR system requirements diagram

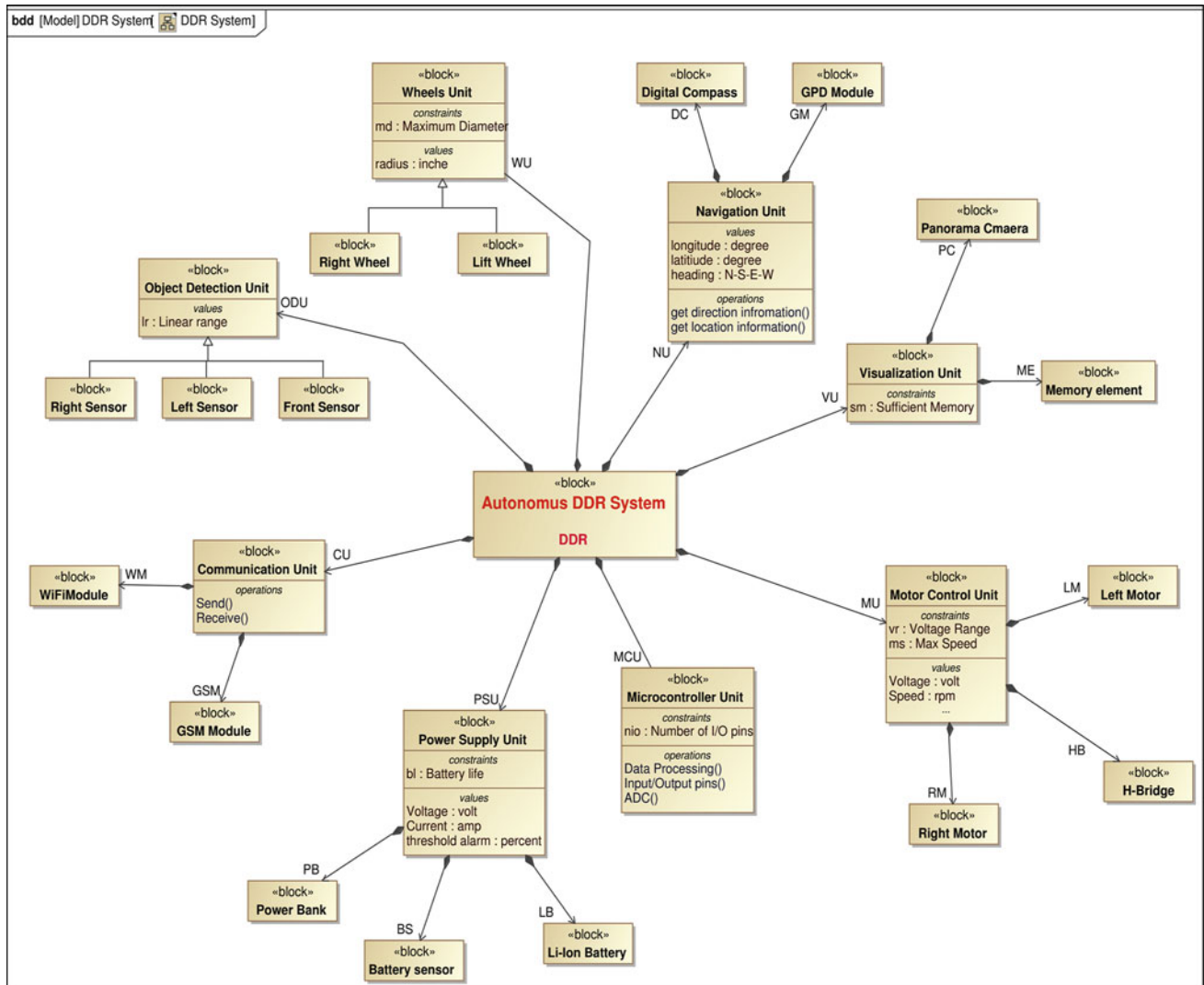


Fig. 9 DDR system block definition diagram

specific set of connections among block’s properties. In this subsection, we provide three *ibds* to comprehend the internal configuration of three major subsystems of DDR system. For instance, Fig. 10 shows the internal block diagram for power supply subsystem strategy: Starting from the battery sensor (BS) as a reference part, it generates a trigger (p1) through the microcontroller unit to activate (p2) the power bank to replace the main battery in case at 20% threshold.

Also, Fig. 11 illustrates the ibd diagram for the structural block configuration of visualization subsystem. The process connects the system operator (administrator) to request a time-stamped stored panorama image (360° map) by issuing a send () command through the GSM module (received by Rx pin) which communicate the command to the

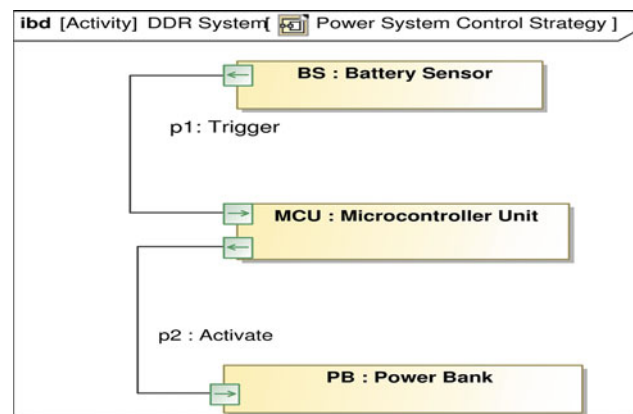


Fig. 10 Internal block diagram for power supply subsystem

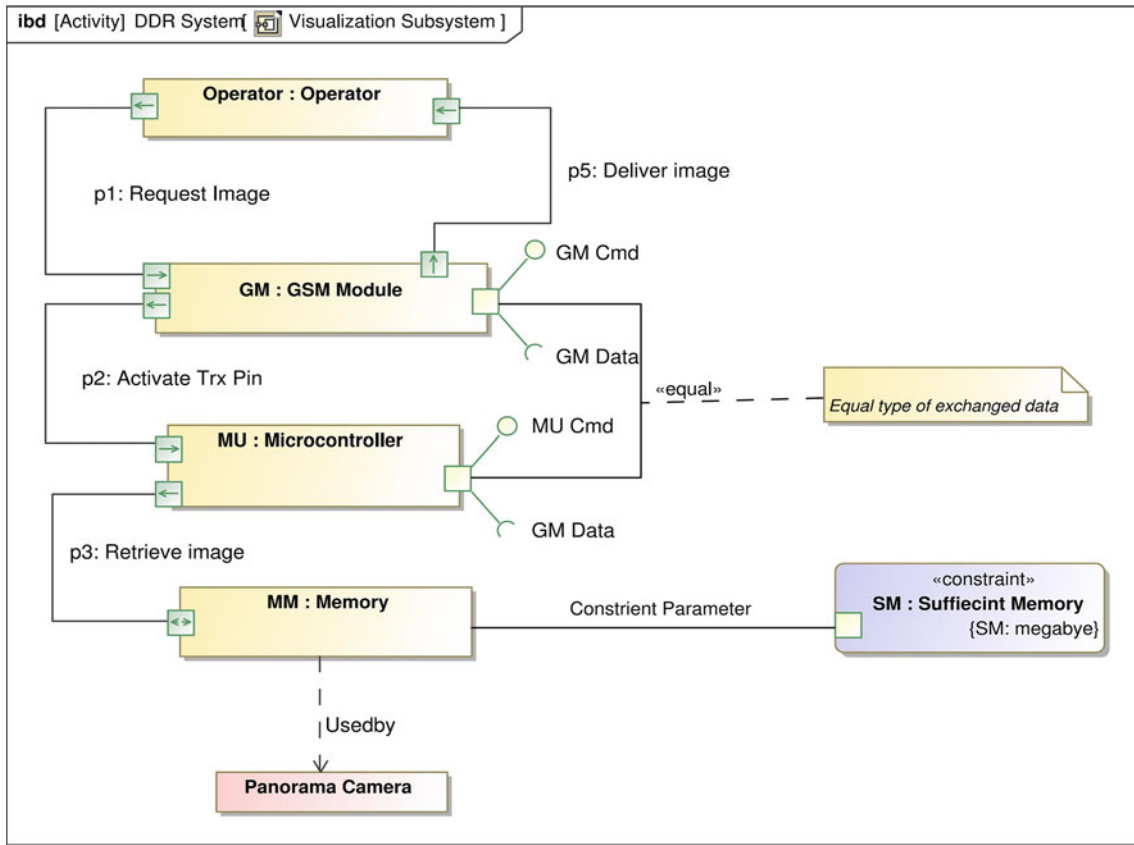


Fig. 11 Internal block diagram for map visualization subsystem

microcontroller unit via Tx pin to retrieve the requested image form the memory element. Also, it is noted that both microcontroller and GSM module require to use two interfaces to call the operations 10 and to provide the operation.

Moreover, it is assumed here that the mapping activity (explained in the activity diagrams section) is using the memory element to store the 360° panorama image maps. Moreover, Fig. 12 describes the *ibd* diagram of wheel control subsystem. The system is composed from two wheels coupled with two motors (left and right) interfaced with the microcontroller unit via a relay (H-Bridge) unit. The rely component provides separate rotation information for each motor according to the parameters given in the parametric diagram. Accordingly, each motor receives its data through data interface and thus provides the commands through CMD interface to the wheels such as the rotation speed and heading.

4.5 System Activity Diagrams

The activity diagram (act) is a dynamic view of the system that expresses sequences of behaviors and event occurrences over time (Delligatti, 2014). This is accomplished

by specifying the transformation of inputs to outputs through a controlled sequence of actions. Here, we provide the act diagrams of three system dynamics processes. Figure 13 elaborates the complete navigation strategy of DDR system. In this act diagram, we have used a conditional loop node that is constrained on incremental timer to emphasize that the complete navigation process should finish within a maximum of two hours or less ($t_i = 120$ min). Then, the activity starts by recording the initial location and direction information followed by a horizontal fork to parallelize the processes of both visualization and motion control units.

For motion control (navigation) process (on the right of fork), the DDR start moving toward east (i.e., forward) with continuous update on GPS/compass information, it checks if there is any signal received on the HIGH pin of the front sensor (i.e., within 5 feet), then an interrupt action is called (stop wheels) to check if there exists any incoming signal on the HIGH pin of the left sensor, then if yes, this means that the obstacle is observed by front and left sensors, thus, the DDR should move the left wheel to get away from obstacle (i.e., move toward right), and then, it goes back to move forward again with updated GPS location and compass heading; otherwise, the DDR will

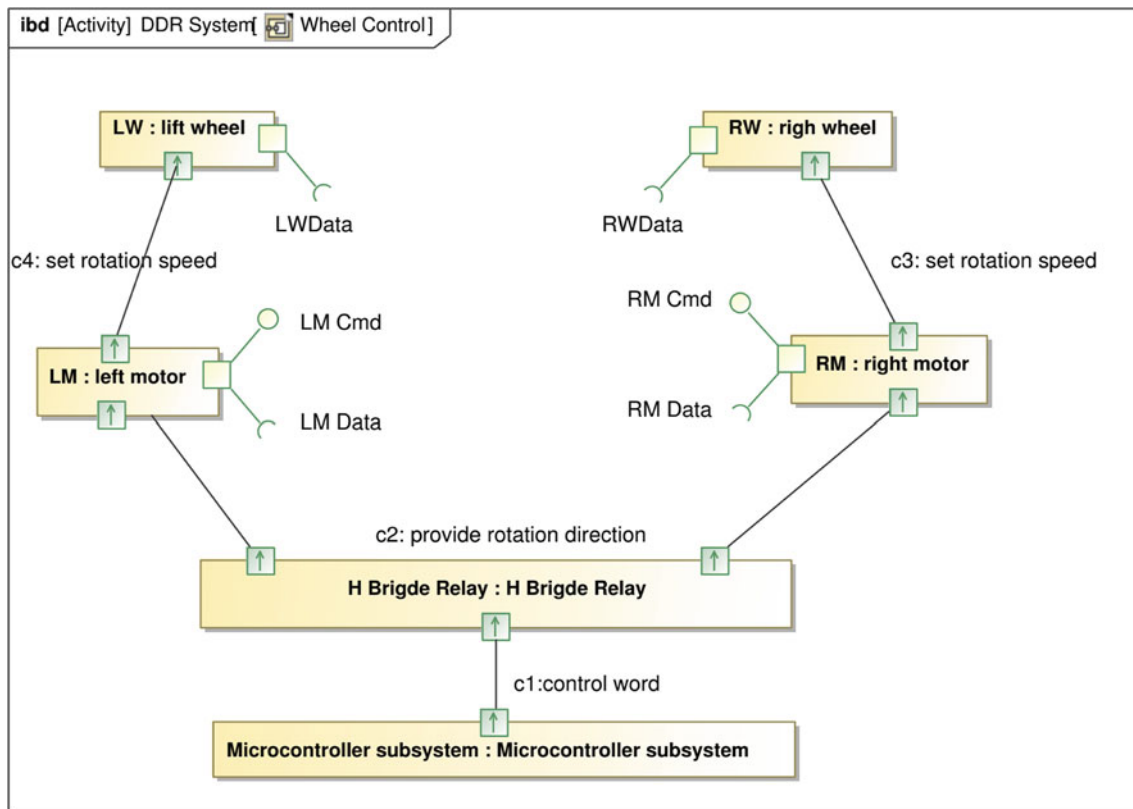


Fig. 12 Internal block diagram for wheel control subsystem

have to move the right wheel and repeat with the rest of process. The motion parameters obtained from the DDR velocity and heading parametric diagram. Regarding the visualization activity (left of fork), the panorama camera calls the stop wheel actions in order to capture and store a new 360° map image with time stamped on each image (captured every minute), and if there is a signal received on the receive pin, it responds to that by sending the requested image.

Also, Fig. 14 presents the activity diagram for power supply subsystem. At the beginning, the system starts operating with the main battery with battery sensor installed to keep tracking of the battery level. Therefore, if the battery level reached to 20%, the system will activate the backup power bank and generate an alert to the operator as two parallel actions expressed by the horizontal fork. After then, a horizontal joint is used to clear the battery sensor readings and start over.

Moreover, Fig. 15 shows the activity diagram for the GSM module. Once the GSM module is turn on, it enables the full power mode in order establish a connection to the network. Once connected, if data received then send the message and enter low power mode until woken up for data to be sent.

4.6 System State Machine Diagrams

The state machine diagram (STM) is a behavior diagram used to represent discrete to the life cycle behavior of a block or system element. It supports event-based behavior such as change event, time event, and signal event. In this subsection, here we provide the STM diagrams of DDR system in addition to three STM diagrams for specific system blocks. The STM diagram for DDR system is presented in Fig. 16, and once operated on, the DDR switches from inactive state to the composite state called "Active". During the active state of DDR, it fluctuates between the following states:

- Wander state: It is the navigation state of DDR.
- Avoid state: It means the obstacle avoidance state, activated once an object gets sensed by the sensors.
- Localize state: When the DDR starts to get the updated GPS location and compass direction.
- Visualize state: This state is the map visualization process, periodically executed.

Also, the map visualization is emphasized more in Fig. 17. In addition to the idle state, the visualization process

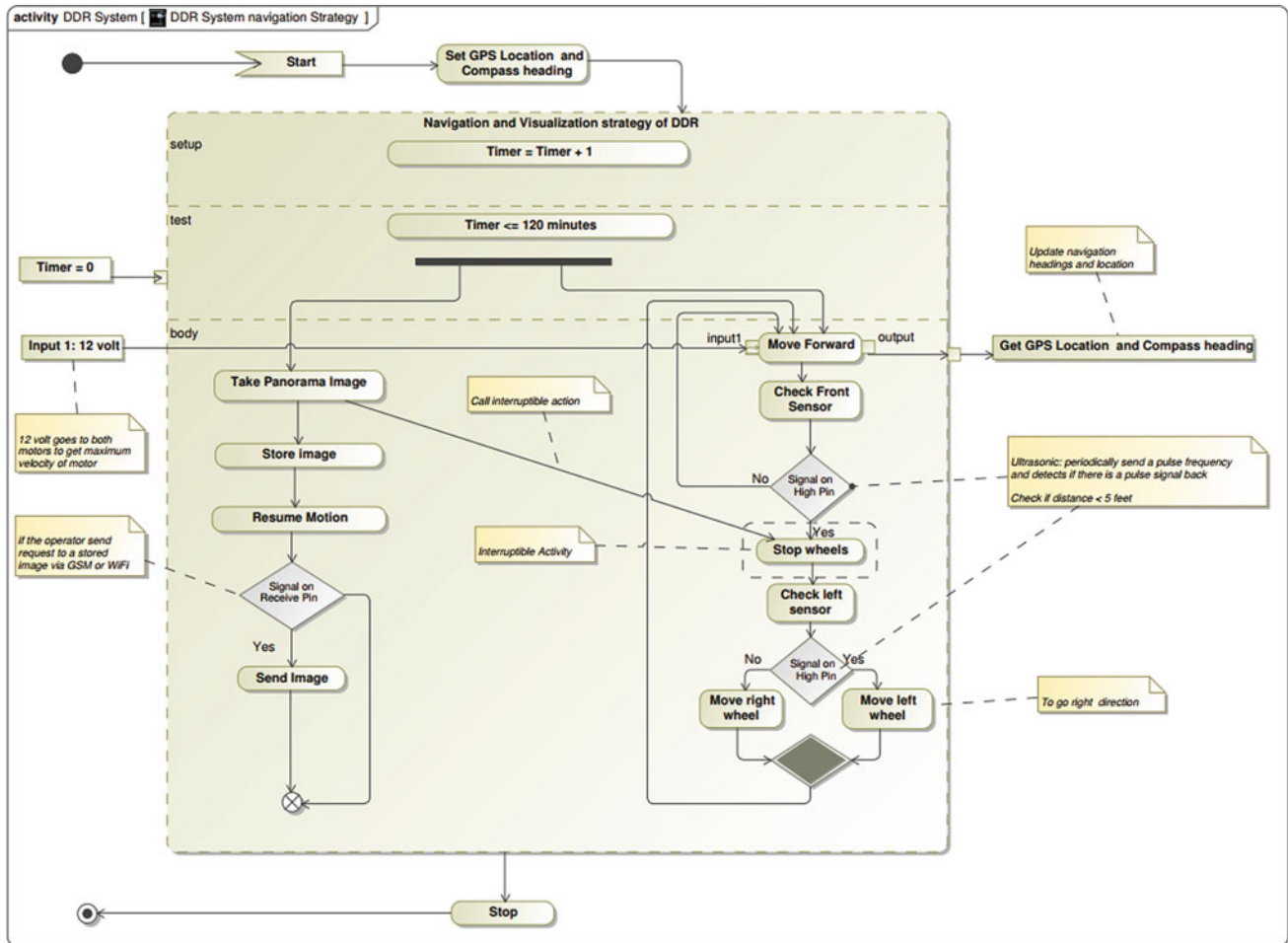


Fig. 13 DDR navigation system activity diagram

has two major states executed in order before end of mission state, which are the map imaging using panorama camera followed by save (store) the map on the memory card.

In addition, Fig. 18 shows the main states of the DC motor control block. Once operated, it shifts from being in the idle state to rotate state with a certain speed (i.e., RPM) set by the DDR system, and when it is called to halt (due to obstacle detection or map visualization), it switches to stop state, and so on.

Furthermore, Fig. 19 shows the main states of the GPS module unit. Beside the idle state, the GPS module can be in one of the following states:

- Initialize state: This state is to connect with satellite system once it gets a fix.
- Read state: this state starts after the connection with sat is established, then GPS asks to get the location information, i.e., longitude and latitude.
- Calculate state: This state calculates the location using the provided longitude and latitude.

Besides, Fig. 20 displays the main states of the power supply module unit. In addition to main battery, the system employs a supplementary sensor to measure the battery level for the system supply. Based on the sensor reading, the sensor will keep reading and calibrating by either keeping the system operating on the main battery if the battery level is more than 20%; otherwise, the sensor will generate a couple of warning messages, and thus, the power supply bank will be activated to support the overall system power.

4.7 System Parametric Diagrams

The parametric diagram (par) is generally used to express information about system constraints in the form of mathematical models that determine the set of valid values within a running system (Delligatti, 2014). However, the parameters regarding the DDR system such as DDR velocity and heading as well as dimensions are clearly stated in the

Fig. 14 DDR power supply subsystem activity diagram

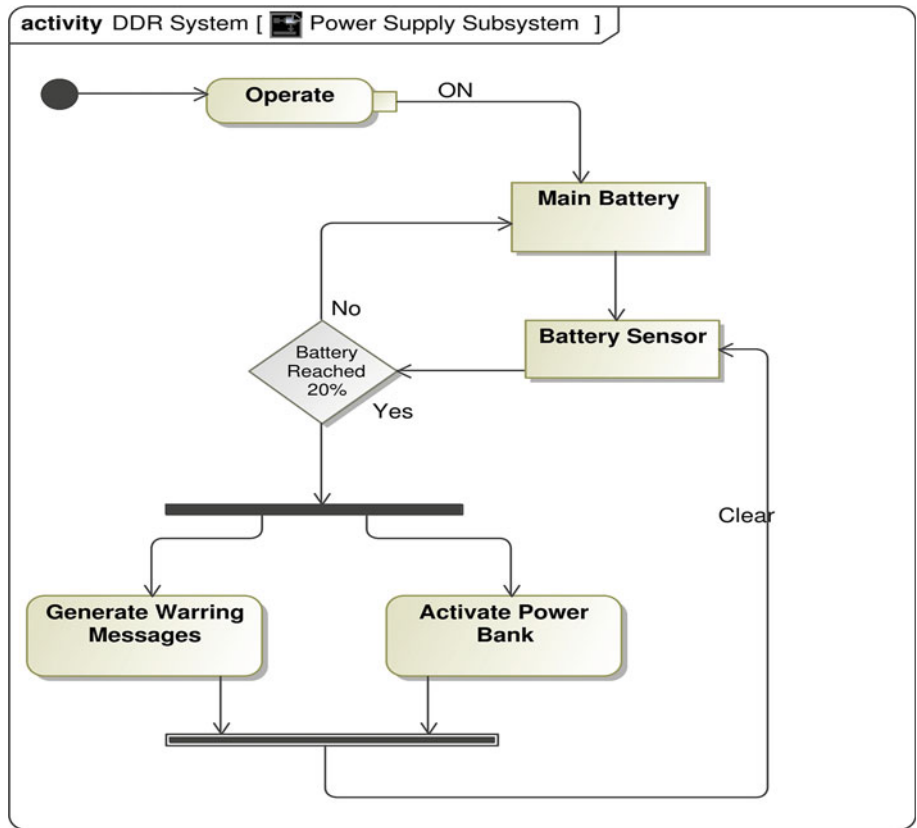
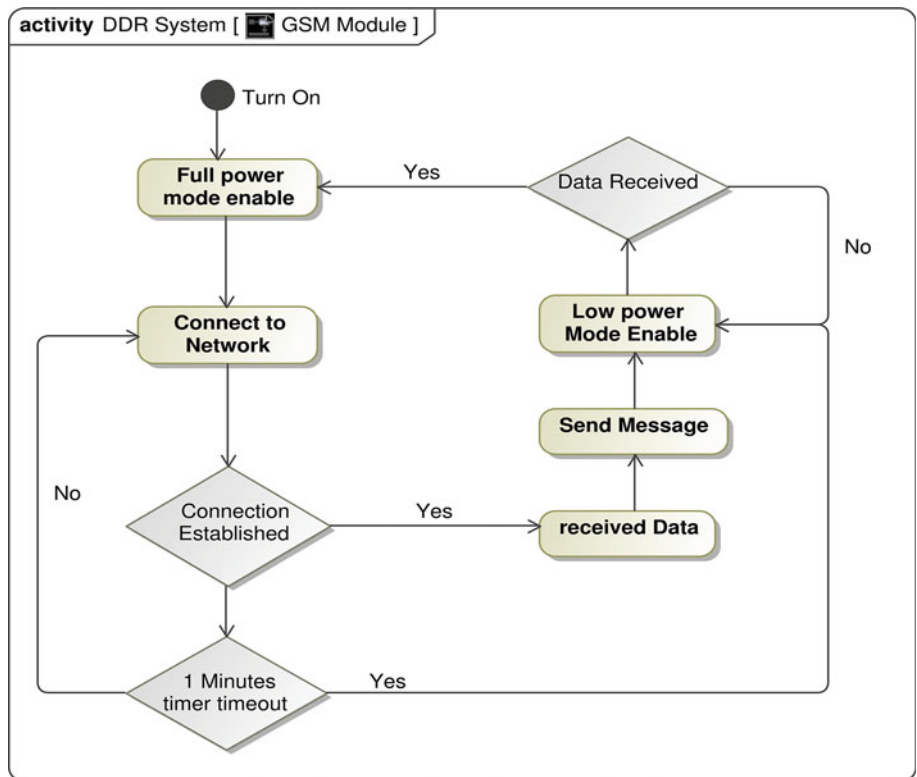


Fig. 15 GSM module activity diagram



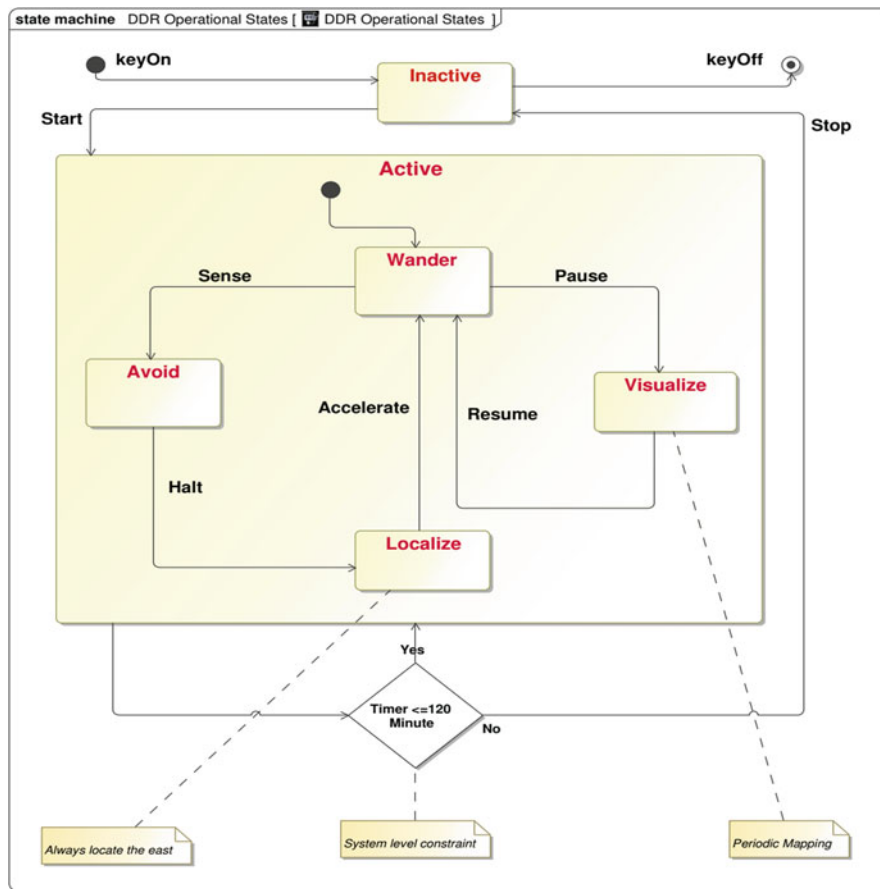


Fig. 16 DDR system operational states

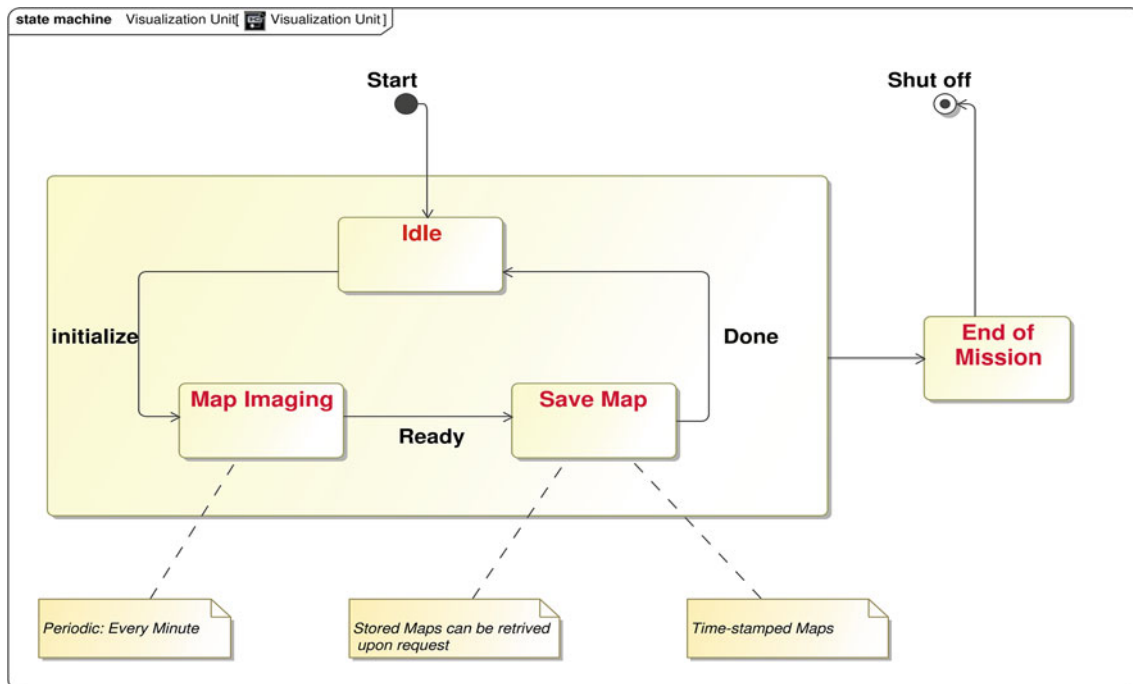
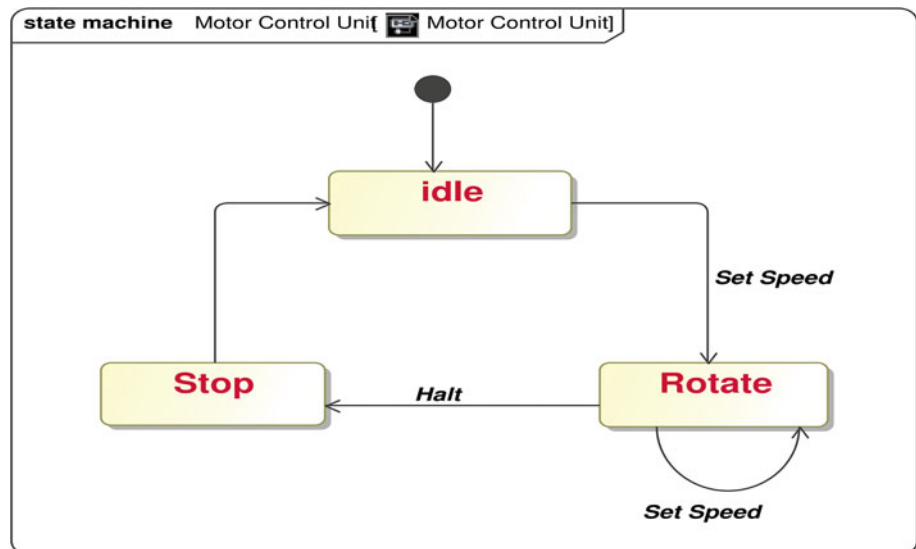
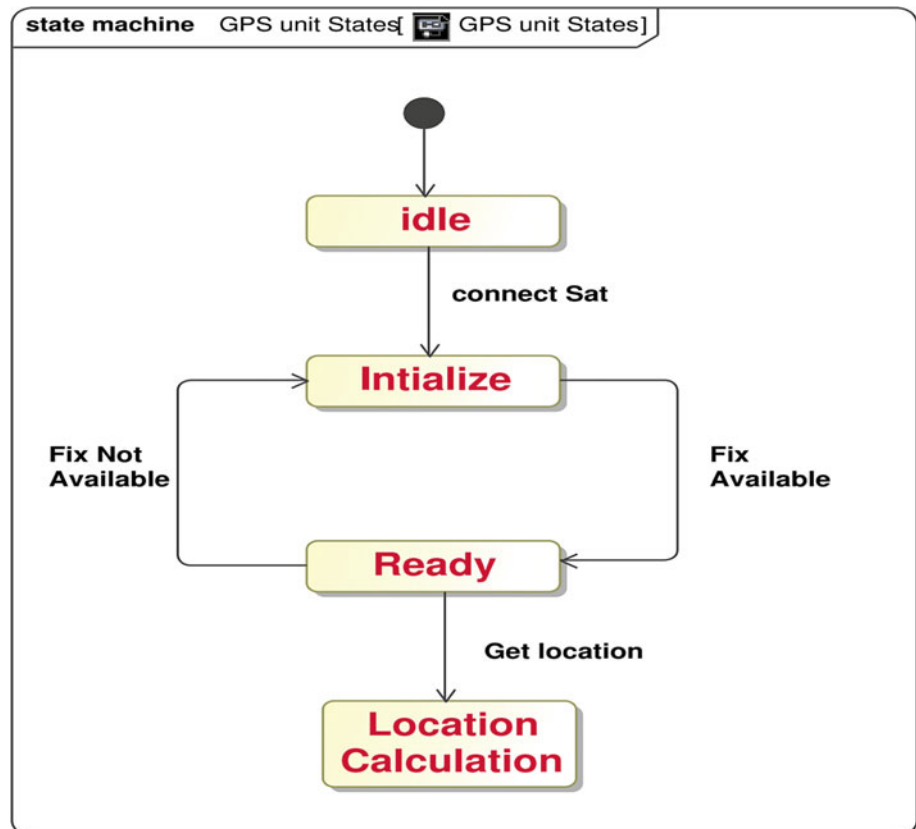


Fig. 17 Map visualization states

Fig. 18 DC motor control states**Fig. 19** GPS unit states

parametric diagram of Fig. 21. While Fig. 22 represents the parametric diagram for the sonar/proximity (i.e., ultrasonic in this work) sensor parameters. Both diagrams formulate the mathematical model equations, constraints, variables, and constants that need to be followed and satisfied by the proposed DDR system.

For parametric diagrams, we have used another SysML development tool called visual paradigm 15.1 (Visual Paradigm International Inc., 2018) since the MagicDraw does not offer tools for parametric diagram design. However, MagicDraw is more powerful and well-known tool for systems modeling visualization.

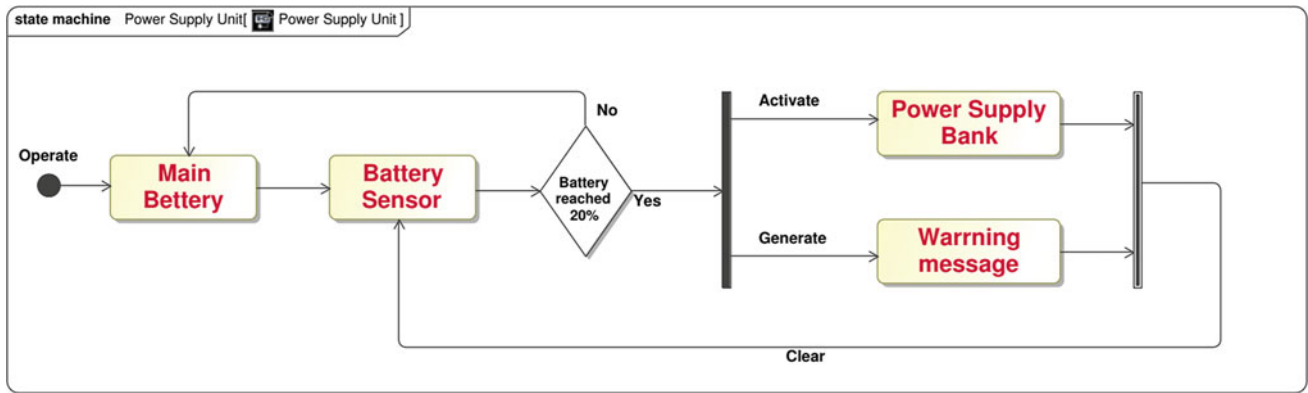


Fig. 20 Power supply unit states

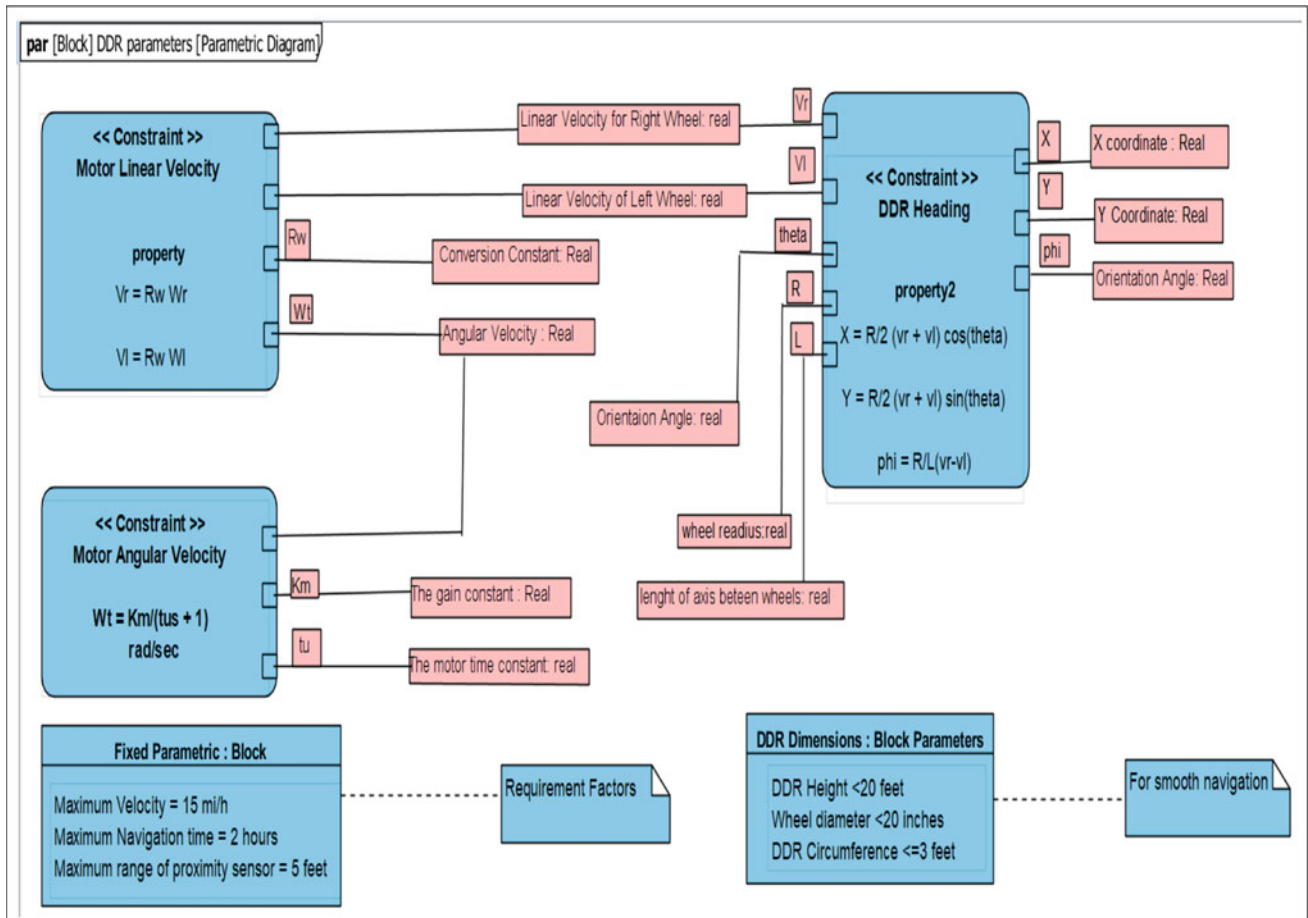


Fig. 21 Parametric diagram of DDR system parameters

5 Conclusions and Remarks

In this chapter, a SysML model-driven engineering-based autonomous differential drive robot (DDR) has been developed using MagicDraw 19.0 software package (MBSE tool).

The proposed DDR is capable of autonomously navigate through a forest environment with vertical cliffs on each side with a continuous operation of two hours' duration before recharging the battery with effective obstacle detection and avoidance mechanisms. Besides, the DDR can periodically visualize and store a 360° panorama images for the

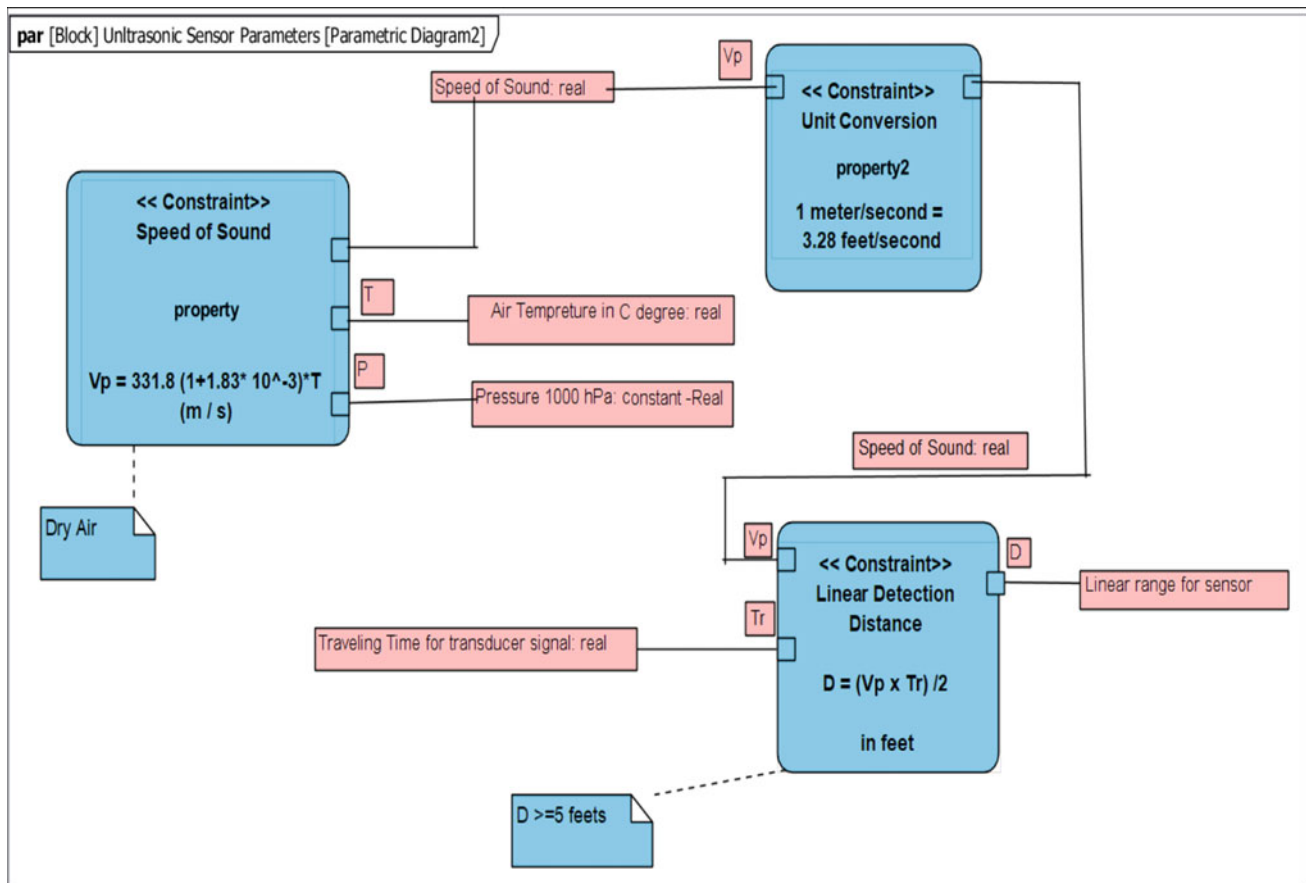


Fig. 22 Parametric diagram of ultrasonic sensor

surrounding environment and accordingly send them via GSM or Wi-Fi modules to the system operator upon receiving the commands. Eventually, the application of SysML in this chapter demonstrated that MBSE techniques can provide information models to capture system aspects and facilitate the functional design of industrial automation and systems.

References

- Al-Haija, Q. A. (2021). *On the security of cyber-physical systems against stochastic cyber-attacks models*. 2021 IEEE International IOT Electronics and Mechatronics Conference (IEMTRONICS), pp. 1–6. <https://doi.org/10.1109/IEMTRONICS52119.2021.9422623>
- Al-Haija, Q. A., & Jebri, N. A. (2020). *ARM cortex based modelling and implementation of a self-controlled traffic light system*. 3rd Smart Cities Symposium (SCS 2020), IET Digital Library, pp. 607–612.
- Al-Haija, Q. A., & Zein-Sabatto, S. (2020). An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks. *Electronics*, 9(12), 2152. <https://doi.org/10.3390/electronics9122152>
- Al-Haija, Q. A., Smadi, M., & Zein-Sabatto, S. (2020). *Multi-class weather classification using ResNet-18 CNN for autonomous IoT and CPS applications*. IEEE 7th Annual Conference on Computational Science & Computational Intelligence (CSCI20), Las Vegas, USA.
- Alur, R. (2015). *Principles of cyber-physical systems*. The MIT Press. 978-0-262-02911-7.
- Batarseh, O., & McGinnis, L. F. (2012). System modeling in sysml and system analysis in arena. In *Proceedings of the 2012 Winter Simulation Conference (WSC)* (pp. 1–12). IEEE.
- Bekey, G. A. (2005). *Autonomous robots: From biological inspiration to implementation and control*, by Intelligent Robotics and autonomous agents series. MIT Press. ISBN: 9780262025782.
- Boursinos, D., & Koutsoukos, X. (2020). *Assurance monitoring of cyber-physical systems with machine learning components*. arXiv preprint [arXiv:2001.05014](https://arxiv.org/abs/2001.05014)
- Castaño, F., Beruvides, G., Haber, R. E., & Artuñedo, A. (2017). Obstacle recognition based on machine learning for on-chip LiDAR sensors in a cyber-physical system. *Sensors*, 17(9), 2109.
- Dartmann, G., Song, H., Schmeink, A. (Eds.). (2019). *Big data analytics for cyber-physical systems: machine learning for the internet of things*. Elsevier.
- Delligatti, L. (2014). *SysML distilled: A brief guide to the systems modeling language*. Addison-Wesley Professional.
- Dreossi, T., Donzé, A., & Seshia, S. A. (2019). Compositional falsification of cyber-physical systems with machine learning components. *Journal of Automated Reasoning*, 63(4), 1031–1053.
- Fei-Fei. (2019). *CS231n: Convolutional neural networks for visual recognition*. Computer Science, Stanford University. Available online: <http://cs231n.stanford.edu>

- Friedenthal, A. M., & Steiner, R. (2009). *OMG systems modeling language-OMG SysML tutorial*, by Object Management Group (OMG). published and used by INCOSE.
- Ghose, A. (2017). *Support vector machine (SVM) tutorial: Learning SVMs from examples*. Medium: Towards Data Science
- Hruska, J. (2018, October). *Self-driving cars still can't handle snow, rain, or heavy weather*. ExtremeTech, By Ziff Davis, LLC.
- Interim Report on 21st Century Cyber-Physical Systems Education, Division on Engineering and Physical Sciences, Computer Science and Telecommunications Board. (2015). National Academies Press.
- Kurihata, H., Takahashi, T., Ide, I. et al., (2005, June). *Rainy weather recognition from in-vehicle camera images for driver assistance*. Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 205–210
- McKee, D. W., Clement, S. J., Almutairi, J., & Jie, Xu. (2018). Survey of advances and challenges in intelligent autonomy for distributed cyber-physical systems. *CAAI Transactions on Intelligence Technology*, 3(2), 75–82.
- Meneses, J. S., Chavez, Z. R., & Rodriguez, J. G. (2019). Compressed kNN: K-nearest neighbors with data compression. *Entropy*, 21(3), 234. <https://doi.org/10.3390/e21030234>
- No Magic Inc. (2015). *MagicDraw, architecture made simple: user manual 18.1*. No Magic Inc. Retrieved on-line from: <http://www.nomagic.com/files/manuals/>
- Pereira, A., & Thomas, C. (2020). Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction*, 2(4), 579–602. <https://doi.org/10.3390/make2040031>
- RobotShop Community Blogs, <https://www.robotshop.com/>
- Ruchkin, V., Fulin, V., Kostrov, B., Taganov, A., & Kolesenkov. A. (2016). *Forest fire monitoring by means of cyber-physical system*. 2016 5th Mediterranean Conference on Embedded Computing (MECO), pp. 30–34. IEEE.
- Sokolov, S., Zhilenkov, A., Chernyi, S., Nyrkov, A., & Glebov, N. (2020). Hybrid neural networks in cyber physical system interface control systems. *Bulletin of Electrical Engineering and Informatics*, 9(3), 1268–1275.
- Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives Psychiatry*, 27(2), 130–135. PMID: 26120265; PMCID: PMC4466856.
- Tama, B. A., & Rhee, K. H. (2017). An extensive empirical evaluation of classifier ensembles for intrusion detection task. *International Journal Computer Systems Science and Engineering*, 32(2), 149–158.
- Tatarinov, V., & Kirsanov, A. (2019). Enhancement of monitoring systems for the transport of dangerous goods by road. *IOP Conference Series: Material Science Engineering*, 492, 012017. <https://doi.org/10.1088/1757-899X/492/1/012017>
- Tran, H. D. et al. (2020). NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In Lahiri, S., & Wang, C. (Eds.), *Computer aided verification. CAV 2020. Lecture Notes in Computer Science* (Vol. 12224). Springer. https://doi.org/10.1007/978-3-030-53288-8_1
- Visual Paradigm International Inc. (2018). *Visual paradigm, design and management tool for business IT development: tutorials*. Retrieved on-line from: <https://www.visual-paradigm.com/tutorials/>
- Wang, W., Maio, F., & Zio, E. (2018). Hybrid fuzzy-PID control of a nuclear cyber-physical system working under varying environmental conditions. *Nuclear Engineering and Design*, 331, 54–67.
- Wickramasinghe, C. S., Daniel, L. M., Amarasinghe, K., & Manic, M. (2018). *Generalization of deep learning for cyber-physical system security: A survey*. IECON 2018–44th Annual Conference of the IEEE Industrial Electronics Society, pp. 745–751. IEEE.



Vulnerabilities and Ethical Issues in Machine Learning for Smart City Applications

K. Martin Sagayam[✉], Roopa Jeyasingh, J. Jenkin Winston[✉],
and Tony Jose[✉]

Abstract

In modern era, technology is transforming throughout the world fast-paced with numerous technological advancements and developments. Any use of technology, however, brings with it new concerns and obstacles. Many aspects of smart cities rely on data transfer, storage and learning technology and cyber-security issues have an impact on how they operate. First part of this chapter is focused on different aspects of vulnerabilities of machine learning in privacy, data security, privacy and public administration, scarcity of skilled professionals, huge capital, unemployment, economic disparity and artificial stupidity. Secondly, different ethical issues of ML in societal, implementation and data collection levels are presented. The use cases of smart city applications are presented to emphasize the vulnerabilities and ethical issues and to address and regulate such use of information.

Keywords

Vulnerabilities • Ethical issues • Machine learning • Deep learning • Smart city • Cyber security

K. M. Sagayam (✉) · R. Jeyasingh · J. J. Winston · T. Jose
Karunya Institute of Technology and Sciences, Coimbatore,
Tamil Nadu 641114, India
e-mail: martinsagayam@karunya.edu

R. Jeyasingh
e-mail: roopa@karunya.edu

J. J. Winston
e-mail: jenkinwinston@karunya.edu

T. Jose
e-mail: tonyjose@karunya.edu

1 Introduction

Smart cities are intended to provide a better standard of living for all its residents. The community should be able to reap the fruits of the immense shift towards the smart implementations. The tremendous modifications and installations that are required to convert a conventional urban area to a smart city, let alone build a new one come at exorbitant costs. The wealth that is spent on such advanced projects comes from people of all layers in the society. The richest and the poorest have their shares in the development and because of the same reason, it should be accessible for everyone at fair prices. There are a lot of ethical concerns in the implementation of such projects. Some of them are concerns about the privacy of people, fairness in usage, optimum usage of funds, etc. Other than the ethical issues, the so-called most advanced installations are still prone to a lot of vulnerabilities. There is no point in implementing something at such high costs finally to see it fail miserably in terms of security and safe operation. Imagine a situation where the system going rogue, or it gets captured by the wrong entities. Something that is built for the wellness of the people will become the reason for the destruction of a society itself.

Hence a lot of considerations must be taken in this, especially in the design phase itself. The system vulnerabilities may come in unforeseeable ways as the system itself is new to us. Whatever designs are developed, whatever technologies are proposed, they must be tested rigorously in all possible to dimensions to minimize such vulnerabilities. However, there are always possibilities of intrusion, no matter how safe the system may seem to be. So, it is always advisable to have mechanisms to minimize the damage too. The vulnerabilities may come as individual attacks on certain nodes of service, targeted attacks against certain individuals or institutions, or it can be even a system-wide attack which could take the system down. The problem with such kind of attacks on such big missions like smart cities is that a

lot of automated mechanisms are involved, and once they fall into the wrong hands, it can be used to hurt the citizens, a lot of them, at once, before anyone could find a solution to stop the attack.

Considering the plans to implement the smart cities on a global scale, it is high time we discussed about the vulnerabilities and ethical issues associated. Once the design phase is over, and the systems are up and running, there is nothing much we can do to correct the system, none at affordable costs. This chapter intends to provide an insight to some of the major vulnerabilities and ethical issues of machine learning (ML) for smart city applications. Section 2 discusses the major problems with having vulnerabilities in the system. The major points of vulnerability are discussed. Section 3 tries to draw the attention towards the major ethical issues related to the implementation of smart cities. There are, in fact, a variety of perspectives in which the system implementation must be viewed on. It cannot be overlooked as a simple demonstration of advancement of technologies or as an attractive stepping stone to a modern world. There are multiple hidden layers of ethical issues which may be evident only after the implementation. So, it is always better to inspect all possibilities before the implementation than to rush into the implementation of something which does not have absolute structure or organization yet.

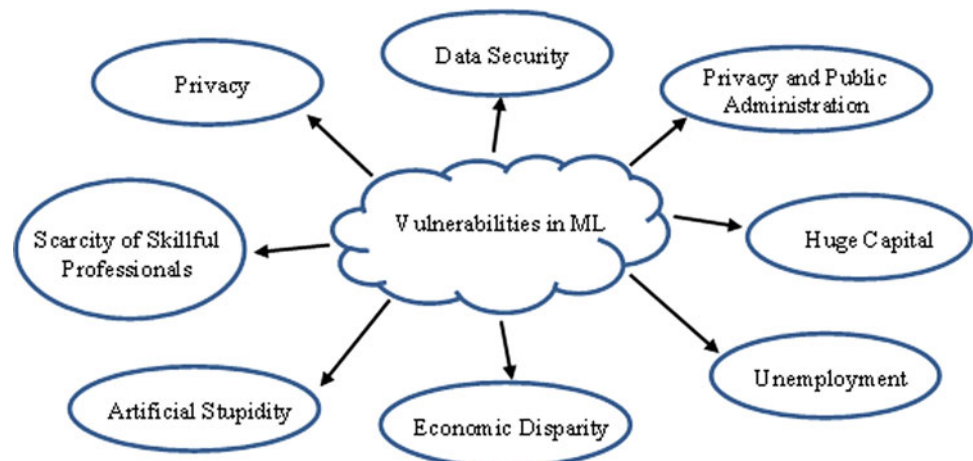
2 Vulnerabilities of ML in Smart City

Different services, technologies and protocols are used by Internet of Things (IoT) devices. As a result, maintaining future IoT infrastructure without exposing the system to undesired vulnerabilities will become extremely difficult. IoT networks and applications are vulnerable to assaults for a variety of reasons. To begin with, most IoT devices have limited resources (e.g. computing power and memory) and hence have limited processing capability. Second, IoT

devices are connected via various protocols, and the growing number of IoT devices increases cloud centre latency. Third, IoT devices are sometimes left unattended, allowing an intruder to physically get access to them. Fourth, the majority of data communication is wireless, making it vulnerable to eavesdropping. As a result, traditional Intrusion Detection System (IDS) frequently fail to accurately detect IoT attacks. Thus, an attacker can use susceptible IoT devices to connect to smart city routers and devices in a variety of locations, including homes, retail malls, restaurants, hotels and airports. An attacker who compromises these IoT devices may get sensitive data such as credit card details, video streams and other personal information.

Application layer interacts directly with the user and is easy to access; the application layer of smart cities is the most vulnerable. Phishing viruses, worms, Trojans, spywares, Denial of Service (DoS) attacks, software vulnerabilities and malicious scripts are examples of application-layer attacks. Phishing is a type of data theft that takes advantage of a user's gullibility. A link similar to a user service is made and delivered to the user to fill out, incentivizing the user to divulge sensitive personal data. Attackers can easily find software and online application vulnerabilities since they can test the systems. In this sense, Structured Query Language (SQL) injection-based cyber-attacks are also common. Many smart cities depend on antiquated security software and encryption approaches that are far too vulnerable to attack. Even more secure methods, such as blockchain, have been targeted by hackers. Many bitcoin-based services have experienced the same vulnerability and millions of dollars were stolen. Furthermore, Software Defined Networking (SDN) vulnerabilities allow attackers to mask their identity in the system and watch system behaviour through the backdoor of the SDN. Figure 1. shows the vulnerabilities of machine learning (ML) in a smart city application. A detailed discussion on these vulnerabilities is presented below.

Fig. 1 Vulnerabilities in ML for smart city applications



2.1 Privacy

The risks of machine learning in smart cities include the erosion of privacy through mass surveillance at a granular level, notification and permission are primarily a haphazard activities. The absence of acquisition systems, system brightness and automation may create blurring, reduce the monitoring, which will produce unexpected duplication. Other issues include new broad forms of governance; predicting a profile to treat people differently; celebratory practices; data biases; lack of transparency, control and accountability in artificial intelligence (AI) systems, etc. Issues may also arise in redesign, digital segmentation and separate access to digitally connected services; ground-breaking decisions and consultation with limited citizens; marketing of private and public infrastructure and services; data theft, etc. (Habibzadeh et al., 2019).

2.2 Data Security

A major issue that may arise with the implementation of smart cities is the amount of sensitive data to be collected and handled. The safe storage and use of such private data is of utmost importance. Since the decisions to be made through most of the AI applications depend on availability of varieties of such information, the collection of sensitive data cannot be avoided altogether as well. The data could be simple raw CCTV footage to sensitive personal images of individuals which are captured knowingly and unknowingly. Data security systems with latest technologies that can withstand online attacks must be developed and used for the safeguard of citizens' data. The government and the custodians of the data are indebted to make sure that the data is intact. If there is a breach of security and leakage of such data, citizens will lose their trust in the system and slowly the model would fail (Sengan et al., 2020, 2021).

2.3 Privacy and Public Administration

Please Another problem is in collecting and using private data in the name of public administration and public security. The governments and the ruling bodies can bring in legislation that would entitle them to collect various data of private nature under the local and national security banners. As we have seen in many incidents in the history, the governments and ruling bodies may not be working in the sole interest of the society and the data that is collected can be misused widely (Lebrument et al., 2021). This may range from targeting political opponents to personal vendettas.

2.4 Scarcity of Skillful Professionals

To deploy and successfully run any smart city project, skilled workers are most required. As the technology for running the smart cities are fairly new and still developing, there are not much experienced professionals in the field. This will put a limitation on the implementation, development and smooth running of the smart cities. Effective training programmes must be conducted regularly for the workers, to equip them with the knowledge of the state-of-the-art technologies (Alghamdie, 2021). Trying to run the systems without adequate knowledge will not only cause poor efficiency and damage to the system but also is a waste of invaluable resources.

2.5 Huge Capital

The conversion of an existing urban area to a smart one will require a lot of effort and modifications in its infrastructure, all of which come at very high expenses. The structures need to be rebuilt with a lot of sensors incorporated, running lots of data lines, attaching very large numbers of wired and wireless transceivers, etc. All these devices are state-of-the-art and are costly. High-speed internet facilities must be available in every nook and corner to support all the data transfer required. Even the conventional systems like water supply and public transport have to be rebuilt from scratch to incorporate such changes. All of this will require huge capital which has to be raised from the society itself (Chen et al., 2021).

2.6 Unemployment

The implementation of smart cities will bring its inherent advantages of full automation of almost all systems. This means that a lot of workers will lose their manual jobs and be replaced by automation systems. This in turn will affect the labourers and their families. So, unemployment will be an adverse effect of smart cities. However, efforts can be made to train the workers in the new arenas related to technology and to relocate them to better jobs (Huang & Marefati, 2020).

2.7 Economic Disparity

The societal economic system runs on the basis of compensation for the work done. To be specific, hourly wages are provided for workers for their manual labour. Among the labourers, the hourly pay is again layered according to their skill sets. During the advancement of technology, like the implementation of smart cities, those who have proficiency

in technology and related works will be paid better while the manual labourers will be paid least or removed from their jobs. This will bring in economic disparities in society. Ones who are well equipped with skills related to technology will survive while the others will struggle. Also, there will be accumulation of wealth in the hands of people who run the services and who own the AI-related firms (Alanne, 2021).

2.8 Artificial Stupidity

Intelligence has to be imparted by the process of teaching and learning. This is applicable for humans as well as machines. In the case of machines and automated systems, they are trained to do certain operations. Machine learning is one of the popular training mechanisms through which the system is trained to do intended operations with minimum human intervention (Liu & Zhang, 2021; Liu et al., 2020). However, there are certain vulnerabilities that may incur during the training phase. Some of them are listed below.

2.8.1 Racist Robots

Robots can operate at accelerated pace and have the power to process data that is far beyond human potential. However, it cannot be trusted to operate unbiased and impartial consistently. These intelligent systems are developed by humans, who could be discriminatory and deprecative. If only such intelligent systems are handled by people with the motive to advance the society, AI can become an agent in making a positive change in the society.

2.8.2 Lack of Security

The advancement of technology makes it more powerful. With more power, more responsibility comes to protect it from vindictive biosphere. It can used either for wicked reasons or rational motives. These robots like autonomous weapons and soldiers which can cause damage if operated

maliciously can be threat to the mankind. The cyber-security task force will have to ensure that these intelligent systems are protected to be deployed in battlefields. Otherwise, it may lead to adverse after effects.

2.8.3 Robot Rights

When deploying robots that handle real-world situations, like facing an accident, the robots may have to take decisions on risking the negative outcomes to mitigate the misery. The advancement in technology has to support everyone. This has to be in mind while developing such robots. Mass production of robots with artificial intelligence are possible with sophisticated manufacturing units. In densely populated areas these robots with smart technologies can ease the daily routines of human life. But proper care and planning has to be taken in implementing this technology in highly safe and secure manner. Apart from approaching towards providing better solutions, the developer should also have the notion how it can harm the people who come in contact with it. To ensure quality of life for everyone, the technology, governance and people must be in harmony. This can really make a city smart.

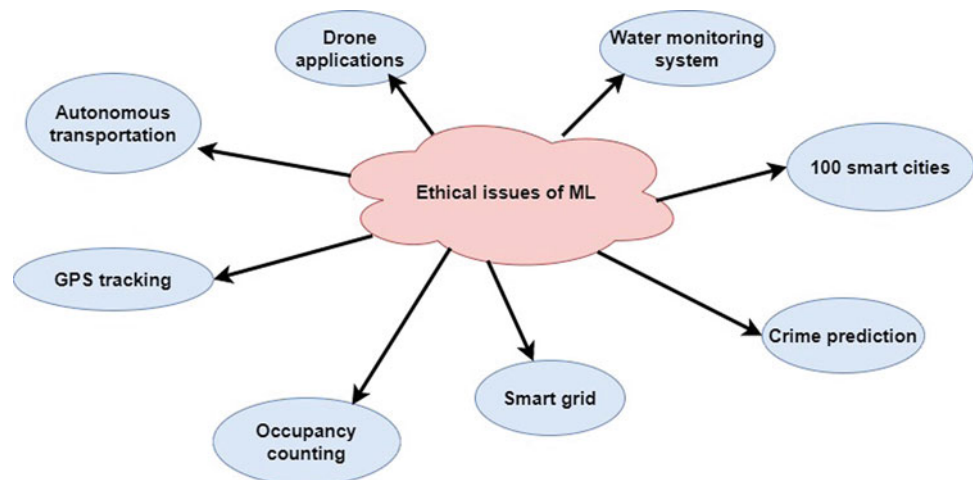
3 Ethical Issues

In the smart city context, many applications have emerged with ethical implications associated with them as shown in Fig. 2. The following are some of the use cases of smart cities towards the contribution, design and research phase in which how ethical issues may arise.

3.1 Water Monitoring System

A water monitoring system was implemented in a Korean city to automate the water supply system. The system

Fig. 2 Ethical issues towards the smart city applications



was built to fully automate the distribution of water to the whole city, and it was also intended to monitor the quality of water. It was implemented using sensors and software systems from the corporate companies for evaluation and monitoring. When a sensor fails, it may lead to shut down of the system or it may have to interact with other systems that are maintained by other vendors. Thus, there will be time delay in the entire process, during which the service is not available for the public. This may have happened due to the improper operation and maintenance of the system. This compromised the credibility of the system. E.g., South Korean (U-City) (Kim, 2015).

3.2 100 Smart Cities

In 2014, initiatives have started to implement 100 smart cities pan India. This was to improve the quality of human life, infrastructure, mass transit, free pollution, efficient energy and governance. These implementations were targeted to nine cities with four million of population. There was an imbalance in distribution of the facilities between the lower, middle and upper classes of the people in the society. It affected different strata of society as only some of them were benefitted (Souvanic, 2016).

3.3 Crime Prediction

Crime prediction is geared to identify potential risks in certain locations, people groups and communities towards unlawful activities. The software is used to find the anonymizing mobile data within the demographic and geographic location. This may be used to predict the criminal activity and law enforcement towards the crime activities. However, the ethical issues may increase when data accessing and sharing information intensifies (Bogomolov et al., 2014).

3.4 Smart Grid

Smart grid is used to employ the different levels of energy distribution for an individual electrical device. This system is integrated with Internet of Things (IoT) to manage the usage and billing. But in certain situations, there could be scarcity of power to an individual system and it may have disastrous effects. Thus, it has a problem of peak usage time and peak usage throttling due to large usage (Kostyk & Herkert, 2012).

3.5 Occupancy Counting

Occupancy counting is used in managing people, especially in huge buildings. It can also be used for room allocations and dynamic distribution of resources. It could also be helpful in crucial occasions like events of fire, security and privacy breaches, etc. The ethical issue that may arise in this is that at times if a wrong data is produced, that is, if the number of people counted is different, it may even cause irreversible problems including loss of lives (Akkaya et al., 2015).

3.6 GPS Tracking

Geographical Positioning Systems (GPS) is much popular in all cities to identify location and to monitor of outdoor activities. It can effectively find the accurate location in each area. The legally collected data is stored in secure servers for further notifications. This location data may be tracked by wrong institutions for targeted attacks or even for personalized advertisements all of which are unethical (Elmaghraby & Losavio, 2014).

3.7 Autonomous Transportation

Autonomous transportation is a complementary task in smart city projects. It is supposed to provide a cheap and safe transport with intelligence. This project is carried out to avoid the human errors and slackness during the driving. Self-driving vehicles are introduced to make the travel safe and fuel-efficient. However, if a system fails in the project, human lives are at risk (Newcomer, 2022).

3.8 Drone Applications

Drones hold a large variety of potential applications in a smart city project. They can be used for multiple applications ranging from security surveillance to environmental observation. They are also used in applications like crowd surveillance, infrastructure monitoring, emergency services and even courier deliveries. However, they can also be used to spy on citizens, families or institutions which is an unethical use (Mohamed et al., 2020).

4 Discussions and Conclusion

The various aspects of smart city implementations are briefly discussed. Some of the vulnerabilities that may arise associated with the deployment of smart cities are presented below:

- Privacy
- Data Security
- Privacy and Public Administration
- Scarcity of Skillful Professionals
- Huge Capital
- Unemployment
- Economic Disparity
- Artificial Stupidity.

Some use cases of the ethical issues towards the smart city implementation is presented below:

- Water monitoring system
- 100 smart cities
- Crime prediction
- Smart grid
- Occupancy counting
- GPS tracking
- Autonomous transportation
- Drone applications.

The above-mentioned perspectives can be considered during the development of technologies for smart city implementations as the implementations are ultimately aimed to aid the society for a better living. Deployment projects without the consideration of such factors may in turn produce adverse effects than the expected positive outcomes.

References

- Akkaya, K., Guvenc, I., Aygun, R., Pala, N., & Kadri, A. (2015). *IoT-based occupancy monitoring techniques for energy-efficient smart buildings*. IEEE Wireless Communications and Networking Conference Workshops (WCNCW), New Orleans, LA, USA, pp. 58–63.
- Alanne, K. (2021). A novel performance indicator for the assessment of the learning ability of smart buildings. *Sustainable Cities Society*, 72, 103054.
- Alghamdie, M. L. (2021). A novel study of preventing the cyber security threats. *Materials Today: Proceedings*
- Bogomolov, A., Lepri, B., Staiano, J., Oliver, N., Pianesi, F., & Pentland, A. (2014). *Once upon a crime: Towards crime prediction from demographics and mobile data*. 16th International Conference on Multimodal Interaction, Turkey, 427–434.
- Chen, D., Wawrzynski, P., & Lv, Z. (2021). Cyber security in smart cities: A review of deep learning based applications and case studies. *Sustainable Cities Society*, 66, 102655.
- Clever, S., Crago, T., Polka, A., Al-Jaroodi, J., & Mohamed, N. (2018). Ethical analyses of smart city applications. *Urban Science*, 2, 96.
- Elmaghraby, A. S., & Losavio, M. M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy. *Journal of Advanced Research*, 5, 491–497.
- Habibzadeh, H., et al. (2019). A survey on cyber security, data privacy, and policy issues in cyber physical system deployments in smart cities. *Sustainable Cities and Society*, 50, 101660.
- Huang, W., & Marefati, M. (2020). Energy, exergy, environmental and economic comparison of various solar thermal systems using water and Therminol base fluids, and CuO and Al₂O₃ nanofluids. *Energy Reports*, 6, 2919–2947.
- Kim, J. S. (2015). Making smart cities work in the face of conflicts: Lessons from practitioners of South Korea's U-City projects. *The Town Planning Review*, 86, 561–585.
- Kostyk, T., & Herkert, J. (2012). Computing Ethics: Societal Implications of the Emerging Smart Grid. *Communications of the ACM*, 55, 34–36.
- Lebrument, N., et al. (2021). Triggering participation in smart cities: Political efficacy, public administration satisfaction and sense of belonging as drivers of citizens' intention. *Technological Forecasting and Social Change*, 171, 120938
- Liu, Y., et al. (2020). Analyzing the robotic behavior in a smart city with deep reinforcement and imitation learning using IoRT. *Computer Communications*, 150, 346–356.
- Liu, L., & Zhang, Y. (2021). Smart environment design planning for smart city based on deep learning. *Sustainable Energy Technologies and Assessments*, 47, 101425.
- Mohamed, N., & Al-Jaroodi, J., Jawhar, I., Idries, A., & Mohammed, F. (2020). Unmanned aerial vehicles applications in future smart cities. *Technological Forecasting and Social Change*, 153, 119293.
- Newcomer, E. (2022). *Uber CEO's commitment to self-driving cars tested by fatality*. <https://www.livemint.com/Companies/ikUlecmn1LQHPUIEWxy7DJ/Uber-CEOs-commitment-to-self-driving-cars-is-tested-by-fata.html>. Last accessed February 2, 2022.
- Sengan, S., et al. (2020). Enhancing cyber physical systems with hybrid smart city cyber security architecture for secure public data smart network. *Future Generation Computer System*, 112, 724–737.
- Sengan, S., et al. (2021). Detection of false data cyber attacks for the assessments of security in smart grid using deep learning. *Computer & Electrical Engineering*, 93, 107211.
- Souvanic, R. (2016). The smart city paradigm in India: Issues and challenges of sustainability and inclusiveness. *Social Scientist*, 5, 29–48.