

Studies in Computational Intelligence 870

Jyotsna Kumar Mandal
Somnath Mukhopadhyay
Paramartha Dutta
Kousik Dasgupta *Editors*

Algorithms in Machine Learning Paradigms

 Springer

Studies in Computational Intelligence

Volume 870

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

The books of this series are submitted to indexing to Web of Science, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink.

More information about this series at <http://www.springer.com/series/7092>

Jyotsna Kumar Mandal ·
Somnath Mukhopadhyay ·
Paramartha Dutta · Kousik Dasgupta
Editors

Algorithms in Machine Learning Paradigms

 Springer

Editors

Jyotsna Kumar Mandal
Kalyani University
Kalyani, India

Paramartha Dutta
Department of Computer
and System Sciences
Visva-Bharati University
Bolpur, West Bengal, India

Somnath Mukhopadhyay
Department of Computer Science
and Engineering, Triguna Sen
School of Technology
Assam University
Silchar, Assam, India

Kousik Dasgupta
Department of Computer Science
and Engineering
Kalyani Government Engineering College
Kalyani, West Bengal, India

ISSN 1860-949X

ISSN 1860-9503 (electronic)

Studies in Computational Intelligence

ISBN 978-981-15-1040-3

ISBN 978-981-15-1041-0 (eBook)

<https://doi.org/10.1007/978-981-15-1041-0>

© Springer Nature Singapore Pte Ltd. 2020

This work is subject to copyright. All rights are reserved 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.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Editorial Preface

With the advancement of human civilization there are various phases of transformation—Scientific, Technological, and others. Some of the phases, in general and some of the inventions/discoveries in specific have left mark in the history of mankind. Inventions of fire, electricity, wheel, are some such examples. Because of dynamism inherent in the progress of civilization newer and newer findings come up as a result of intense initiative of research community all and otherwise passive computing system—popularly known as machine intelligence. It is needless to mention that machine intelligence is the conclusive finding of the computing process that requires providing adequate training of the computing system under consideration—the process being popularly referred to machine learning. To be precise, machine learning the process framework to achieve the task of machine intelligence. Applications of machine learning encompass optimizations, health-care, agriculture, remote sensing, financial services, IoT and Analytics, Virtual Personal Assistants for example Alexa, Marketing and Sales, Recommendation systems, Social Media Services, and what not. To be precise, in the present scope of these treatise, editors find pleasure in accommodating such applications of immense significance. They are considered as follows:

In chapter one, authors have reported a priority based Hesitant-Intuitionistic Trapezoidal Fuzzy Einstein Operators and their use in multi-criteria group decision making. In the next chapter authors have proposed a model for unsupervised feature selection based on graph-oriented information-theoretic approach. In chapter three, a fact-based Expert System being proposed for supplier selection in ERP environment. In next chapter authors have modelled a fuzzy time series for handling seasonal pattern and prediction. In chapter five, authors have proposed a texture-based model for automatic classification of fruits and vegetables. In the next chapter a deep learning based early sign detection model has been proposed for proliferative diabetic retinopathy in neovascularization at the disc. In the next chapter authors have proposed a predictionpolicy for live migration in cloud computing. In the next two chapters authors have proposed a tracking and recognizing systems of changing human emotions from facial image sequence. In chapter ten, authors have devised a model to predict stable neighbor nodesin mobile ad hoc

network. In the final chapter, authors have proposed anew model for optimizing initial parameters of Lorenz attractor and its application in PRNG.

The editors desire to avail this opportunity to express their heartfelt thanks and sincere gratitude to the contributing authors for their rich technical findings reflected in their articles. But for their contributions, this initiative could not have become a reality. The task of finalizing the articles of high technical merit could not have been achieved without the active cooperation and support from the learned reviewers, thankless albeit. Last but not the least, the editors feel obliged to Springer Nature as publication partner for taking all the pain to provide the editors the opportunity of making the present endeavor in the form of an edited volume a success.

Happy reading!!!

Kalyani, India
Silchar, India
Bolpur, India
Kalyani, India

Jyotsna Kumar Mandal
Somnath Mukhopadhyay
Paramartha Dutta
Kousik Dasgupta

Contents

Hesitant-Intuitionistic Trapezoidal Fuzzy Prioritized Operators Based on Einstein Operations with Their Application to Multi-criteria Group Decision-Making	1
Arun Sarkar and Animesh Biswas	
Unsupervised Feature Selection Using Information-Theoretic Graph-Based Approach	25
Sagarika Saroj Kundu, Abhirup Das and Amit Kumar Das	
Fact-Based Expert System for Supplier Selection with ERP Data	43
Kartick Chandra Mondal, Biswadeep Deb Nandy and Arunima Baidya	
Handling Seasonal Pattern and Prediction Using Fuzzy Time Series Model	57
Mahua Bose and Kalyani Mali	
Automatic Classification of Fruits and Vegetables: A Texture-Based Approach	71
Susovan Jana, Ranjan Parekh and Bijan Sarkar	
Deep Learning-Based Early Sign Detection Model for Proliferative Diabetic Retinopathy in Neovascularization at the Disc	91
Nilanjana Dutta Roy and Arindam Biswas	
A Linear Regression-Based Resource Utilization Prediction Policy for Live Migration in Cloud Computing	109
Gopa Mandal, Santanu Dam, Kousik Dasgupta and Paramartha Dutta	
Tracking Changing Human Emotions from Facial Image Sequence by Landmark Triangulation: An Incircle-Circumcircle Duo Approach	129
Md Nasir, Paramartha Dutta and Avishek Nandi	

Recognizing Human Emotions from Facial Images by Landmark Triangulation: A Combined Circumcenter-Incenter-Centroid Trio Feature-Based Method	147
Avishek Nandi, Paramartha Dutta and Md Nasir	
Stable Neighbor-Node Prediction with Multivariate Analysis in Mobile Ad Hoc Network Using RNN Model	165
Arindrajit Pal, Paramartha Dutta, Amlan Chakrabarti and Jyoti Prakash Singh	
A New Approach for Optimizing Initial Parameters of Lorenz Attractor and Its Application in PRNG	181
Ramen Pal and Somnath Mukhopadhyay	
Author Index	195

About the Editors

Dr. Jyotsna Kumar Mandal is a Professor of Computer Science & Engineering, and former Dean of FETM, Kalyani University, India. He holds an M.Sc. in Physics from Jadavpur University, M.Tech. in Computer Science from the University of Calcutta, and was awarded a Ph.D. in Computer Science & Engineering by Jadavpur University. He has 32 years of teaching and research experience in various fields of computer science and allied areas, and has published 170 articles in journals, more than 300 articles at conferences, and edited 31 volumes and seven books. He is a Fellow of IETE, life member of CRSI and CSI, and senior member of IEEE.

Dr. Somnath Mukhopadhyay is an Assistant Professor at the Department of Computer Science and Engineering, Assam University, Silchar, India. He completed his M.Tech. and Ph.D. degrees in Computer Science and Engineering at the University of Kalyani, India. He has co-authored one book and edited six books and published over 25 papers in various international journals and conference proceedings, including three chapters in edited volumes. His research interests include remote sensing and computational intelligence. He is a life member of the Computer Society of India and currently the Regional Student Coordinator of Region II, Computer Society of India.

Prof. (Dr.) Paramartha Dutta, FIE (India), FIETE, FOSI, SMIEEEE, SMACM, SMCSI, completed his bachelor's and master's in Statistics and Master of Technology in Computer Science at the Indian Statistical Institute, and his Ph.D. in Engineering at the Bengal Engineering and Science University, Shibpur. He has co-authored eight books and twelve edited books, and published 250 papers in various peer-reviewed journals and conference proceedings as well as several book chapters. He holds six international and eleven national patents.

Dr. Dutta received (i) IRDP Lifetime Achievement Award 2018, (ii) Distinguished Scientist Award in Computer Science and Engineering 2018 conferred by the Venus International Research Foundation, (iii) Excellence in Science and Technology Award 2018–2019 conferred by the Indian Science Congress Association, Government of

India, and (iv) INSA Teacher Award 2019 conferred by the Indian National Science Academy, Government of India.

Dr. Kousik Dasgupta is an Assistant Professor of Computer Science & Engineering, Kalyani Government Engineering College, West Bengal. He completed his B.Tech. in Electronics and Power Engineering at Nagpur University in 1993, M.Tech. in Computer Science at West Bengal University of Technology in 2007, and his Ph.D. at the Department of Computer Science and Engineering, University of Kalyani, in 2017. Dr. Dasgupta has presented numerous papers at various conferences and co-authored several books. He is a member of five scientific and professional societies.

Hesitant-Intuitionistic Trapezoidal Fuzzy Prioritized Operators Based on Einstein Operations with Their Application to Multi-criteria Group Decision-Making



Arun Sarkar and Animesh Biswas

Abstract In this article, a ranking method for hesitant-intuitionistic trapezoidal fuzzy (H-ITF) numbers (H-ITFNs) is proposed. After introducing H-ITFN, the concept of score function and accuracy function of H-ITFN are defined and H-ITF prioritized weighted averaging and geometric operators based on Einstein operations are developed. Some desirable properties of the proposed operators are investigated in detail. A method for ordering the alternatives in multi-criteria group decision-making problems with H-ITF information based on different priority levels of decision-makers and criteria is presented. An illustrative example concerning academic resource deployment in educational institutions studied previously, is considered and solved. The comparison of the results with earlier methods reflects superiority of the proposed methodology.

Keywords Multi-criteria group decision-making · Trapezoidal intuitionistic fuzzy number · Hesitant-intuitionistic fuzzy number · Einstein operations · Prioritized aggregation operators

1 Introduction

Zadeh [1] introduced the concept of fuzzy sets. Afterward, the idea of intuitionistic fuzzy sets (IFSs) was presented by Atanassov [2–4]. It is now well known that the elements of IFS are characterized by a membership degree and a non-membership degree through which it represents its respective belongingness and non-belongingness to a set. Thus, IFS is more effective to deal with uncertainty and vagueness in many real-life applications. Zhang [5] introduced a family of intuitionistic fuzzy Einstein hybrid weighted averaging and geometric operators, the quasi-intuitionistic fuzzy Einstein

A. Sarkar
Department of Mathematics, Heramba Chandra College, Kolkata 700029, India
e-mail: asarkarmth@gmail.com

A. Biswas (✉)
Department of Mathematics, University of Kalyani, Kalyani 741235, India
e-mail: abiswaskln@rediffmail.com

hybrid weighted averaging and geometric operators. Sedef and Balaman [6] extended ELimination Et Choice Translating REality I (ELECTRE I) with VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) to provide complete ranking under intuitionistic fuzzy environment. Garg and Arora [7] introduced intuitionistic fuzzy soft power averaging (IFSPA) and geometric (IFSPG) operators, weighted IFSPA (WIFSPA) operator, ordered WIFSPA operator, and corresponding geometric aggregation operators. Liu and Tang [8] proposed the intuitionistic fuzzy prioritized interactive Einstein Choquet operator, which could consider the prioritization and interaction among the criteria.

Like fuzzy numbers [9], the intuitionistic fuzzy numbers (IFNs) are also special type of IFS defined on the real number set, seems to suitably describe an unknown quantity [10]. Shu et al. [11] defined the concept of a triangular IFN (TIFN) and developed an algorithm for intuitionistic fuzzy fault tree analysis. Li [12] introduced the concept of the TIFN and ranking method on the basis of the concept of a ratio of the value index to the ambiguity index as well as applications to multiple attributes decision-making (MADM) problems in depth. Wan et al. [13] proposed triangular intuitionistic fuzzy weighted average (TIF-WA) operator in which the weights of the attributes and decision-makers (DMs) are completely unknown. Then, Wang [14] defined trapezoidal intuitionistic fuzzy (TriIF) numbers (TriIFNs), which is the extension of the TIFN. Wang and Zhang [15] presented TriIF weighted averaging and geometric operators, and employed to intuitionistic fuzzy multi-criteria decision-making. Wei [16] proposed TriIF ordered weighted averaging and hybrid aggregation operators. Das and Guha [17] introduced a new ranking method of TriIFN by utilizing the concept of a newly defined centroid point. All the methods described above did not consider hesitancy associated with the IFNs.

However, in many real-life multi-criteria group decision-making (MCGDM) problems DMs are unable to determine the exact membership degree of an element to a set due to some sort of doubts among a few different possible values. To deal with such cases, Torra and Narukawa [18] and Torra [19] introduced another variant of fuzzy sets, viz., hesitant fuzzy set (HFS), allowing the membership degree having a set of possible values. Xia and Xu [20] proposed a series of operators under various situations and discussed the relationships among them. HFSs are applied to various fields of decision-making [21–28]. Yu [29] developed some operations based on Einstein operations for HFS and introduced a class of aggregation operators, viz., hesitant fuzzy Einstein weighted averaging, hesitant fuzzy Einstein ordered weighted averaging and hesitant fuzzy Einstein hybrid averaging operators and corresponding to its geometric operators to aggregate HFEs. Inspired by the concept of IFS and HFS, Zhou et al. [30] recently proposed a new IFN known as hesitant-intuitionistic fuzzy (H-IF) numbers (H-IFNs) and delivered an idea of H-IF preference relation (H-IFPR). Operational laws and comparative methods of H-IFNs are also described. To aggregate fuzzy numbers, different types of aggregation operators are available in the literature. Yager [31] first introduced the prioritized average (PA) operator, which has characterized by the ability to deal with the decision-making problems where the criteria differ in priority level. Motivated by the idea of prioritized aggregation operators, Wei [32] developed some prioritized aggregation operators for aggregating

hesitant fuzzy information, and some models for hesitant fuzzy MADM problems are also presented in which the attributes are in different priority levels. Apart from the algebraic operations, viz., algebraic sum and algebraic product, Einstein operations play an important role for aggregation. Yu et al. [33] proposed hesitant fuzzy Einstein prioritized weighted and power weighted average as well as geometric operators.

In this article, the concept of hesitant-intuitionistic trapezoidal fuzzy (H-ITF) numbers (H-ITFNs) is defined. Some operational laws of H-ITFN based on Einstein operations have been introduced. To find the ordering of H-ITFNs a score function is also defined. The proposed score definition can remove the drawback arise due to the definition of score function given by Jianqiang and Zhong [15] in the context of TrIFNs. Based on Einstein operations, H-ITF prioritized Einstein weighted averaging (H-ITFPEWA) operator and H-ITF prioritized Einstein weighted geometric (H-ITFPEWG) operator have been developed to aggregating the H-ITFNs. A method for finding the rank of alternatives in a MCGDM with H-ITF information has been proposed. Finally, a numerical example has been provided to establish the efficiency of the proposed models and achieved solutions are compared with other existing methods.

2 Preliminaries

In this section, some basic concepts, which are essential to develop the proposed methodologies, are described.

Definition 1 ([1]) Let X be a non-empty set. A fuzzy set F in X is characterized by its membership function $\mu_F : X \rightarrow [0, 1]$ such that for each $x \in X$, $\mu_F(x)$ is interpreted as the degree of membership of the element x in fuzzy set F . Therefore, it is clear that F is completely determined by the set of tuples

$$F = \{(x, \mu_F(x)) | x \in X\} \tag{1}$$

Definition 2 ([2–4]) Let $X = \{x_1, x_2, \dots, x_n\}$ be fixed. An IFS β on X can be represented as:

$$\beta = \{(x_i, \mu_\beta(x_i), \nu_\beta(x_i)) | x_i \in X\} \tag{2}$$

where the functions $\mu_\beta(x_i)$ and $\nu_\beta(x_i)$ denote, respectively, the membership and non-membership degree of x_i to X in β with the condition that $0 \leq \mu_\beta(x_i) \leq 1, 0 \leq \nu_\beta(x_i) \leq 1$ and $0 \leq \mu_\beta(x_i) + \nu_\beta(x_i) \leq 1$. For computational convenience, Xu and Yager [34] used the notation (μ_β, ν_β) to represent an IFN β .

Definition 3 ([14, 17]) An IFN β , defined on the set of real numbers is said to be an TrIFN, denoted by $\mathcal{T} = ([a, b, c, d]; u), ([a', b, c, d']; v)$ if the membership function of it is defined by

$$\mu_{\mathcal{T}}(x) = \begin{cases} f_{\mathcal{T}}^L(x) = \frac{(x-a)u}{(b-a)}, & \text{for } a \leq x \leq b, \\ u, & \text{for } b \leq x \leq c, \\ f_{\mathcal{T}}^R(x) = \frac{(d-x)u}{(d-c)}, & \text{for } c \leq x \leq d, \\ 0, & \text{for } x \leq a, x \geq d \end{cases} \quad (3)$$

and the non-membership function is defined as

$$\nu_{\mathcal{T}}(x) = \begin{cases} g_{\mathcal{T}}^L(x) = \frac{(b-x)+(x-a')v}{(b-a)}, & \text{for } a' \leq x \leq b, \\ v, & \text{for } b \leq x \leq c, \\ g_{\mathcal{T}}^R(x) = \frac{(x-c)+(d'-x)v}{(d'-c)}, & \text{for } c \leq x \leq d', \\ 1, & \text{for } x \leq a', x \geq d' \end{cases} \quad (4)$$

where u and v represent the highest membership degree and lowest non-membership degree of an element $x \in \mathcal{T}$, respectively, such that u and v satisfies the conditions:

$0 \leq u, v \leq 1$ and $0 \leq u + v \leq 1$. The function $\pi_{\mathcal{T}}(x) = 1 - u_{\mathcal{T}}(x) - \nu_{\mathcal{T}}(x)$ is called the hesitancy or indeterminacy of an element x belonging to \mathcal{T} . For computational convenience, it has been taken $a = a'$ and $d = d'$. In such a case, a TrIFN \mathcal{T} can be simply denoted by $\mathcal{T} = ([a, b, c, d]; u, v)$.

The trust degree [15] of TrIFN \mathcal{T} in between $[u, 1 - v]$ is presented as:

$$I_{\theta}(\mathcal{T}) = \frac{1}{2} \times \left(\int_0^u \{ (1 - \theta)g_{\mathcal{T}}^L(x) + \theta g_{\mathcal{T}}^R(x) \} dy + \int_0^{1-v} \{ (1 - \theta)g_{\mathcal{T}}^L(x) + \theta g_{\mathcal{T}}^R(x) \} dy \right)$$

where $\theta \in [0, 1]$ expresses the decision-makers risk preference. $I_{\theta}(\mathcal{T})$ is known as the expected value of TrIFN \mathcal{T} . If $\theta > 0.5$, the decision-makers love risk, and if $\theta < 0.5$, the decision-makers hate risk, and if $\theta = 0.5$, decision-makers have indifferent risk preferences.

From the above expression, the expected value of the TrIFN $\mathcal{T} = ([a, b, c, d]; u, v)$, is calculated [15] as $I(\mathcal{T}) = \frac{1}{8} \times [(a + b + c + d) \times (1 + u - v)]$.

For ordering the TrIFNs, Jianqiang and Zhong [15] introduced the score and accuracy function based on the expected value as follows:

Definition 4 ([15]) Let $S(\mathcal{T})$ and $A(\mathcal{T})$ denote, respectively, score function and accuracy function of a TrIFN $\mathcal{T} = [a, b, c, d]; w, u$. Then using the expected value those are defined as

$$S(\mathcal{T}) = I(\mathcal{T})(u - v) \quad (5)$$

and

$$A(\mathcal{T}) = I(\mathcal{T})(u + v) \quad (6)$$

For comparing TrIFNs, the following method is applied.

Let \mathcal{T}_1 and \mathcal{T}_2 be any two TrIFNs

- (i) if $S(\mathcal{T}_1) < S(\mathcal{T}_2)$, then $\mathcal{T}_1 < \mathcal{T}_2$;
- (ii) if $S(\mathcal{T}_1) = S(\mathcal{T}_2)$, then
 - if $A(\mathcal{T}_1) < A(\mathcal{T}_2)$, then $\mathcal{T}_1 < \mathcal{T}_2$;
 - if $A(\mathcal{T}_1) = A(\mathcal{T}_2)$, then $\mathcal{T}_1 = \mathcal{T}_2$

As an extension of fuzzy sets, Torra and Narukawa [18] and Torra [19] introduced the concept of hesitant fuzzy sets (HFSs).

Definition 5 ([18, 19]) Let X be a fixed set, an HFS defined on X can be represented in the form of a function that maps each element of X with a subset consisting of a finite number of elements of $[0, 1]$. Symbolically, it is denoted as

$$E = \{ \langle x, h_E(x) \rangle | x \in X \} \tag{7}$$

where $h_E(x)$ is a collection of some finite values in $[0, 1]$, representing the possible membership degrees of the element $x \in X$ to the set E . For simplicity, Xia and Xu [20] denoted $h = h_E(x)$ as hesitant fuzzy elements (HFE).

Considering both the concepts of IFS and HFS, Zhou et al. [30] introduced the concept of Hesitant-intuitionistic fuzzy set (H-IFS) in the recent past.

Definition 6 ([30]) If X is a fixed set, then a H-IFS K on X is defined as follows:

$$K = \{ \langle x, h_K(x), v_K(x) \rangle | x \in X \} \tag{8}$$

where $h_K(x) = \left\{ \bigcup_{\gamma \in h(x)} \gamma \right\}$ represent the membership degree of element x in K , and membership degree $h_K(x)$ is an HFE with the condition that all possible finite values of $h_K(x)$ belong to $[0, 1]$. and $v_K(x)$ define the non-membership degree.

Zhou et al. [30] called $\left(\left\{ \bigcup_{\gamma \in h(x)} \gamma \right\}, v(x) \right)$ as an H-IFN, to simplify a H-IFN Zhou et al. [30] further provided $v(x) = 1 - \max \left\{ \bigcup_{\gamma \in h(x)} \gamma \right\}$ and for convenience denoted as $k = (h, 1 - h^+)$ where $h = \left\{ \bigcup_{\gamma \in h} \gamma \right\}$ and $h^+ = \max \left\{ \bigcup_{\gamma \in h} \gamma \right\}$.

Definition 7 Zhou et al. [30] introduced the score function of a H-IFN $k = (h, 1 - h^+)$ as

$$S(k) = (1/l_h) \sum_{\gamma \in h} \gamma - 1 + h^+ \text{ and the accuracy function of } k \text{ by } A(k) = 1 + (1/l_h) \sum_{\gamma \in h} \gamma - h^+$$

Let $k_1 = (h_1, 1 - h_1^+)$ and $k_2 = (h_2, 1 - h_2^+)$ be two H-IFNs, then

- (i) $k_1 < k_2$ whenever $S(k_1) < S(k_2)$;
- (ii) if $S(k_1) = S(k_2)$, then

- $k_1 < k_2$ whenever $A(k_1) < A(k_2)$;
- $k_1 = k_2$ whenever $A(k_1) = A(k_2)$.

2.1 The Prioritized Average Operators

The prioritized average operator originally introduced by Yager [31], which has defined as follows:

Definition 8 ([31]) Let $C = \{C_1, C_2, \dots, C_n\}$ be a collection of criteria and a set of alternatives, $X = \{x_1, x_2, \dots, x_m\}$. There is a prioritization between the criteria expressed by the linear ordering $C_1 > C_2 > C_3 > \dots > C_n$, indicates criteria C_j has a higher priority than C_k if $j < k$. The value $C_j(x)$ is the performance of any alternative x under criteria C_j , and satisfies $C_j(x) \in [0, 1]$. One commonly used approach is to calculate for each alternative x , a score $C(x)$ as an aggregation of the $C_j(x)$. If

$$PA(C_j(x)) = C(x) = \sum_{j=1}^n w_j C_j(x) \quad (9)$$

Where $w_j = \frac{T_j}{\sum_{j=1}^n T_j}$, $T_j = \prod_{k=1}^{j-1} C_k(x)$, ($j = 2, \dots, n$), $T_1 = 1$. Then PA is called the prioritized average operator.

2.2 Einstein Operations

The set theoretical operators play an important role in combining fuzzy sets. All types of aggregation operators convey the general concepts of the t-norms and t-conorms, which satisfy the requirements of the conjunction and disjunction operators, respectively. There are various types of t-norms and t-conorms in the literature. Einstein operations include the Einstein sum and Einstein product, which acts as t-conorms and t-norms, respectively.

The operations are defined as follows [35].

For all $(a, b) \in [0, 1]^2$, the Einstein sum \oplus_ε and Einstein product \otimes_ε is defined respectively as:

$$a \oplus_\varepsilon b = \frac{a + b}{1 + a \cdot b} \text{ and } a \otimes_\varepsilon b = \frac{a \cdot b}{1 + (1 - a)(1 - b)} \quad (10)$$

Based on the above preliminary concepts, H-ITFN is developed in the following section.

3 Hesitant-Intuitionistic Trapezoidal Fuzzy Number

Inspired by the concept of TrIFN and H-IFN, an extended H-IFN, viz., the H-ITFNs are introduced and its Einstein-based operation is also discussed in the following section.

Definition 9 An H-IFN k , defined on the set of the real line \mathbb{R} , is said to be a H-ITFN, denoted by $\alpha = \langle ([a, b, c, d]; h), ([a', b, c, d']; 1 - h^+) \rangle$ in which the membership function $\phi_\alpha(x)$ and non-membership function $\psi_\alpha(x)$ are defined as follows:

$$\phi_\alpha(x) = \begin{cases} \frac{(x-a)h}{(b-a)}, & \text{for } a \leq x \leq b, \\ h, & \text{for } b \leq x \leq c, \\ \frac{(d-x)h}{(d-c)}, & \text{for } c \leq x \leq d, \\ 0, & \text{for } x \leq a, x \geq d \end{cases} \quad (11)$$

and the non-membership function is defined as

$$\psi_\alpha(x) = \begin{cases} \frac{(b-x)+(x-a')(1-h^+)}{(b-a)}, & \text{for } a' \leq x \leq b, \\ 1 - h^+, & \text{for } b \leq x \leq c, \\ \frac{(x-c)+(d'-x)(1-h^+)}{(d'-c)}, & \text{for } c \leq x \leq d', \\ 1, & \text{for } x \leq a', x \geq d' \end{cases} \quad (12)$$

Where $h = \left\{ \bigcup_{\gamma \in h} \gamma \right\}$ defined as a HFE, possible membership degrees, and $(1 - h^+)$ is a non-membership degree in which h^+ represents the maximum membership degree in $\left\{ \bigcup_{\gamma \in h} \gamma \right\}$.

For the sake of simplicity in computation and without any loss of generality, we considered $a = a'$ and $d = d'$. A H-ITFN α can be symbolically defined as by $\alpha = ([a, b, c, d]; h, 1 - h^+)$. If $b = c$, H-ITFN reduces to a triangular hesitant-intuitionistic fuzzy number (TH-IFN), i.e., TH-IFN is a special case of H-ITFN.

Diagrammatically, an H-ITFN is represented by the following Fig. 1.

The expectation degree of an H-ITFN α in between $\left[\frac{1}{l_h} \sum_{\gamma \in h} \gamma, h^+ \right]$ is defined as follows:

$$I_\theta(\alpha) = \frac{1}{2} \left(\int_0^{\frac{\sum_{\gamma \in h} \gamma}{l_h}} \left\{ (1 - \theta) \frac{(x-a) \sum_{\gamma \in h} \gamma}{l_h(b-a)} + \theta \frac{(d-x) \sum_{\gamma \in h} \gamma}{l_h(d-c)} \right\} dy \right. \\ \left. + \int_0^{h^+} \left\{ (1 - \theta) \frac{(x-a) \sum_{\gamma \in h} \gamma}{l_h(b-a)} + \theta \frac{(d-x) \sum_{\gamma \in h} \gamma}{l_h(d-c)} \right\} dy \right)$$

where l_h is the number of values in h and $\theta \in [0, 1]$ expresses the decision-makers risk preference.

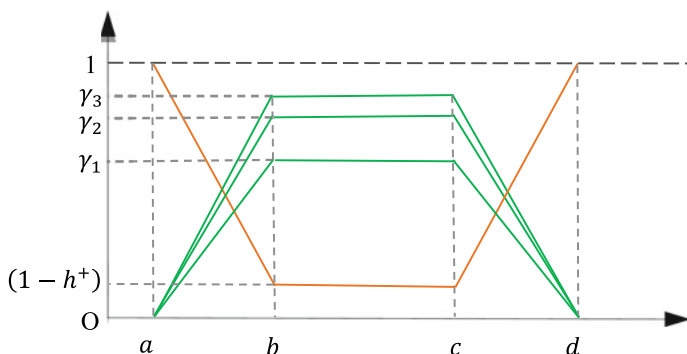


Fig. 1 Representation of H-ITFN $\alpha = \langle [a, b, c, d]; h, (1 - h^+) \rangle$

Now considering $\theta = 0.5$ in the above expression, the expected value of the H-ITFN $\alpha = \langle [a, b, c, d]; h, (1 - h^+) \rangle$, is calculated as

$I(\alpha) = \frac{1}{8} \left[(a + b + c + d) \times \left(\frac{1}{l_h} \sum_{\gamma \in h} \gamma + h^+ \right) \right]$. Considering this $I(\alpha)$ score function and accuracy function of H-ITFNs are defined as follows:

Definition 10 Let $\alpha = \langle [a, b, c, d]; h, (1 - h^+) \rangle$ be a H-ITFN. The score function $S(\alpha)$ of α is defined as

$$S(\alpha) = \frac{1}{2} \left(1 + I(\alpha) \times \left(\frac{1}{l_h} \sum_{\gamma \in h} \gamma - 1 + h^+ \right) \right) \quad (13)$$

and the accuracy function $A(\tilde{\alpha})$ of $\tilde{\alpha}$ is given by

$$A(\alpha) = \frac{1}{2} \left(1 + I(\alpha) \times \left(\frac{1}{l_h} \sum_{\gamma \in h} \gamma + 1 - h^+ \right) \right) \quad (14)$$

It is worthy to mention here that for different values of θ various scores and accuracy functions can be defined. Further, the above definition of score function includes the score function defined by Jianqiang and Zhong [15] in the context of IFN; Zhou et al. [30] in the context of H-IFNs and the score function defined by Liu [12] in the context of TrIFNs. Further, the above definition removes the drawback of the score function defined by Jianqiang and Zhong [15] as described earlier. Another advantage of the proposed score function is that the value of $S(\alpha)$ would not be negative in any circumstances.

Using the score function as defined above, the comparative laws of two H-ITFNs are described as follows:

Let α_1 and α_2 be two H-ITFNs.

If $S(\alpha_1) < S(\alpha_2)$, then $\alpha_1 < \alpha_2$;

If $S(\alpha_1) = S(\alpha_2)$, then

- (i) $\alpha_1 < \alpha_2$ whenever $A(\alpha_1) < A(\alpha_2)$
- (ii) $\alpha_1 = \alpha_2$ whenever $A(\alpha_1) = A(\alpha_2)$.

The Einstein operations on H-ITFNs are described as follows.

Definition 11 Let $\alpha_j = \left([a_j, b_j, c_j, d_j]; h_j, (1 - h_j^+) \right)$ ($j = 1, 2$) and $\alpha = \left([a, b, c, d]; h, (1 - h^+) \right)$ be any three H-ITFNs and $\lambda > 0$. Based on Einstein operations, the following operations on H-ITFNs are introduced.

- $\alpha_1 \oplus_\varepsilon \alpha_2 = ([a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2];$
- $\bigcup_{\gamma_1 \in h_1, \gamma_2 \in h_2} \left(\frac{\gamma_1 + \gamma_2}{1 + \gamma_1 \gamma_2}, \frac{(1 - h_1^+)(1 - h_2^+)}{1 + h_1^+ h_2^+} \right)$
- $\alpha_1 \otimes_\varepsilon \alpha_2 = ([a_1 a_2, b_1 b_2, c_1 c_2, d_1 d_2];$
- $\bigcup_{\gamma_1 \in h_1, \gamma_2 \in h_2} \left(\frac{\gamma_1 \gamma_2}{1 + (1 - \gamma_1)(1 - \gamma_2)}, \frac{2 - h_1^+ - h_2^+}{1 + (1 - h_1^+)(1 - h_2^+)} \right)$
- $\lambda \alpha = \left([\lambda a, \lambda b, \lambda c, \lambda d]; \bigcup_{\gamma \in h} \left(\frac{(1 + \gamma)^\lambda - (1 - \gamma)^\lambda}{(1 + \gamma)^\lambda + (1 - \gamma)^\lambda}, \frac{2(1 - h^+)^\lambda}{(1 + h^+)^\lambda + (1 - h^+)^\lambda} \right) \right)$
- $\alpha^\lambda = \left([a^\lambda, b^\lambda, c^\lambda, d^\lambda]; \bigcup_{\gamma \in h} \left(\frac{2\gamma^\lambda}{(2 - \gamma)^\lambda + \gamma^\lambda}, \frac{(2 - h^+)^\lambda - (h^+)^\lambda}{(2 - h^+)^\lambda + (h^+)^\lambda} \right) \right)$

Using the concept of the newly introduced H-ITFN and Einstein operations, the H-ITF prioritized Einstein weighted averaging (H-ITFPEWA) and the H-ITF prioritized Einstein weighted geometric (H-ITFPEWG) operators are developed in the subsequent section.

4 Hesitant-Intuitionistic Trapezoidal Fuzzy Prioritized Einstein-Based Aggregation Operators

In this section, the prioritized averaging and geometric operators based on Einstein operations under H-ITF environment are investigated. The definition of the H-ITF prioritized Einstein weighted averaging (H-ITFPEWA) operator and the H-ITF prioritized Einstein weighted geometric (H-ITFPEWG) operators are defined and some of their properties are described in details.

Definition 12 Let $\alpha_j = \left([a_j, b_j, c_j, d_j]; h_j, (1 - h_j^+) \right)$, $j = 1, 2, \dots, n$ be a collection of H-ITFNs, and also let $H-ITFPEWA : V^n \rightarrow V$, be a function such

that

$$H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) = \oplus_{\varepsilon_j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} \alpha_j \right) \tag{15}$$

Then $H-ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n)$ is called an $H-ITFPEWA$ operator, where

$$T_j = \prod_{k=1}^{j-1} S(\alpha_k) \text{ for } j = 1, 2, \dots, n; \text{ and } T_1 = 1 \tag{16}$$

and $S(\alpha_k)$ denotes the score value of the $H-ITFN \alpha_k$.

From the definition of score function, it is clear that the value of T_j would never be negative unlike the situation may arise in Jianqiang and Zhong [15].

Several properties of $H-ITFNs$ using $H-ITFPEWA$ and $H-ITFPEWG$ operators are described subsequently.

Theorem 1 *Let $\alpha_j (j = 1, 2, \dots, n)$ be a collection of $H-ITFNs$, then the aggregated value using $H-ITFPEWA$ operator represents also $H-ITFN$ and is given by*

$$\begin{aligned}
 H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) = & \left(\left[\sum_{j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} a_j \right), \right. \right. \\
 & \left. \sum_{j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} b_j \right), \sum_{j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} c_j \right), \sum_{j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} d_j \right) \right]; \\
 & \bigcup_{\gamma_i \in h_i, i=1,2,\dots,n} \frac{\prod_{j=1}^n (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^n (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}, \\
 & \left. \frac{2 \prod_{j=1}^n (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (1 + h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}} \right) \tag{17}
 \end{aligned}$$

Proof The proof of the theorem is executed using the mathematical induction method.

The theorem obviously holds for $n = 1$.

It is assumed that the theorem is true for $n = k$. Then it has to be proved that it is true also for $n = k + 1$.

Now, for $n = k$,

$$\begin{aligned}
 H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_k) = & \oplus_{\varepsilon_j=1}^k \left(\frac{T_j}{\sum_{j=1}^k T_j} \alpha_j \right) \\
 & \left(\sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^k T_j} a_j \right), \sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^k T_j} b_j \right), \right.
 \end{aligned}$$

$$\sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} c_j \right), \sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} d_j \right)$$

$$\bigcup_{\gamma_i \in h_i, i=1,2,\dots,k} \frac{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}},$$

$$\frac{2 \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}}{\prod_{j=1}^k (1 + h_j^+) \frac{T_j}{\sum_{j=1}^n T_j} + \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}}$$

So, for $n = k + 1$,

$$H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_k, \alpha_{k+1}) =$$

$$H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_k, \alpha_{k+1}) \oplus_\varepsilon \left(\frac{T_{k+1}}{\sum_{j=1}^n T_j} \alpha_{k+1} \right)$$

$$= \left(\left[\sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} a_j \right), \sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} b_j \right), \sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} c_j \right), \sum_{j=1}^k \left(\frac{T_j}{\sum_{j=1}^n T_j} d_j \right) \right]; \right.$$

$$\bigcup_{\gamma_i \in h_i, i=1,2,\dots,k} \frac{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}},$$

$$\left. \frac{2 \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}}{\prod_{j=1}^k (1 + h_j^+) \frac{T_j}{\sum_{j=1}^n T_j} + \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}} \right) \oplus_\varepsilon$$

$$\left(\left[\frac{T_{k+1}}{\sum_{j=1}^n T_j} a_{k+1}, \frac{T_{k+1}}{\sum_{j=1}^n T_j} b_{k+1}, \frac{T_{k+1}}{\sum_{j=1}^n T_j} c_{k+1}, \frac{T_{k+1}}{\sum_{j=1}^n T_j} d_{k+1} \right]; \right.$$

$$\bigcup_{\gamma_{k+1} \in h_{k+1}} \frac{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} - (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}}{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} + (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}}, \frac{2(1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}}{(1 + h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j} + (1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}} \left. \right)$$

$$= \left(\sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} a_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} b_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} c_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} d_j \right); \right.$$

$$\bigcup_{\gamma_i \in h_i, i=1,2,\dots,k,k+1} \frac{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}} + \frac{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} - (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}}{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} + (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}},$$

$$1 + \frac{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^k (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^k (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}, \frac{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} - (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}}{(1 + \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}} + (1 - \gamma_{k+1})^{\frac{T_{k+1}}{\sum_{j=1}^n T_j}}},$$

$$\frac{2 \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}}{\prod_{j=1}^k (1 + h_j^+) \frac{T_j}{\sum_{j=1}^n T_j} + \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}} \cdot \frac{2(1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}}{(1 + h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j} + (1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}} \left. \right)$$

$$1 + \left(1 - \frac{2 \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}}{\prod_{j=1}^k (1 + h_j^+) \frac{T_j}{\sum_{j=1}^n T_j} + \prod_{j=1}^k (1 - h_j^+) \frac{T_j}{\sum_{j=1}^n T_j}} \right) \left(1 - \frac{2(1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}}{(1 + h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j} + (1 - h_{k+1}^+) \frac{T_{k+1}}{\sum_{j=1}^n T_j}} \right)$$

$$= \left(\left[\sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} a_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} b_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} c_j \right), \sum_{j=1}^{k+1} \left(\frac{T_j}{\sum_{j=1}^n T_j} d_j \right) \right]; \right. \\ \left. \bigcup_{i=1,2,\dots,k,k+1} \frac{\prod_{j=1}^{k+1} (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^{k+1} (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^{k+1} (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^{k+1} (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}, \right. \\ \left. \frac{2 \prod_{j=1}^{k+1} (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^{k+1} (1 + h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^{k+1} (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} \right)$$

Hence, the above is true for $n = k + 1$ also. Thus, the theorem is true for all integers.

This completes the proof of the theorem.

Theorem 2 (Idempotency) Let $\alpha_j = ([a_j, b_j, c_j, d_j]; h_j, (1 - h_j^+))$ be a collection of H -ITFNs. If all α_j are equal, i.e., $\alpha_j = \alpha$ for all j , then

$$H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha \quad (18)$$

If put $\alpha_j = \alpha$ for all j in Theorem 1, then Theorem 2 can be easily find.

Theorem 3 (Boundary) Let α_j ($j = 1, 2, \dots, n$) be a collection of H -ITFNs. Also, let $h^{\min} = \min(h_j^-)$, where $h_j^- = \min(h_j)$; and $h^{\max} = \max(h_j^+)$ where $h_j^+ = \max(h_j)$ for $j = 1, 2, \dots, n$. Again, let $\alpha_{\min} = ([a_{\min}, b_{\min}, c_{\min}, d_{\min}]; h^{\min}, (1 - h^{\min}))$ and

$$\alpha_{\max} = ([a_{\max}, b_{\max}, c_{\max}, d_{\max}]; h^{\max}, (1 - h^{\max})).$$

Then

$$\alpha_{\min} \leq H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) \leq \alpha_{\max} \quad (19)$$

Proof Let γ_j be an element that is chosen arbitrarily from h_j . Since $h^{\min} \leq \gamma_j \leq h^{\max}$ for all j , then

$$1 - h^{\min} \geq 1 - \gamma_j \geq 1 - h^{\max} \quad \text{and} \quad 1 + h^{\min} \leq 1 + \gamma_j \leq 1 + h^{\max} \quad \text{for all } j,$$

$$\text{Thus, } \frac{1 - h^{\max}}{1 + h^{\max}} \leq \frac{1 - \gamma_j}{1 + \gamma_j} \leq \frac{1 - h^{\min}}{1 + h^{\min}} \quad \text{for all } j$$

$$\text{i.e., } \left(\frac{1 - h^{\max}}{1 + h^{\max}} \right) \leq \prod_{j=1}^n \left(\frac{1 - \gamma_j}{1 + \gamma_j} \right)^{\frac{T_j}{\sum_{j=1}^n T_j}} \leq \left(\frac{1 - h^{\min}}{1 + h^{\min}} \right), \quad \text{for all } j,$$

$$\text{i.e., } h^{\min} \leq \frac{1 - \prod_{j=1}^n \left(\frac{1 - \gamma_j}{1 + \gamma_j} \right)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{1 + \prod_{j=1}^n \left(\frac{1 - \gamma_j}{1 + \gamma_j} \right)^{\frac{T_j}{\sum_{j=1}^n T_j}}} \leq h^{\max}, \quad \text{for all } j$$

$$\text{i.e., } h^{\min} \leq \frac{\prod_{j=1}^n (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^n (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (1 + \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (1 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}} \leq h^{\max}, \text{ for all } j \quad (20)$$

Since $h^{\min} \leq h_j^+ \leq h^{\max}$, for all j , then in a similar way, it can be proved that

$$1 - h^{\max} \leq \frac{2 \prod_{j=1}^n (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (1 + h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (1 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}} \leq 1 - h^{\min} \text{ for all } j \quad (21)$$

Since, $x_{\min} \leq x_j \leq x_{\max}$ for $x = a, b, c, d$ and for all j ,

$$\text{it can be shown that } x_{\min} \leq \sum_{j=1}^n \left(\frac{T_j}{\sum_{j=1}^n T_j} x_j \right) \leq x_{\max}$$

Now, let $H-ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) = \langle [a, b, c, d]; h, (1 - h^+) \rangle$. Then, $h^{\min} \leq \frac{1}{h} \sum_{\gamma \in h} \gamma \leq h^{\max}$, since $h^{\min} \leq \gamma \leq h^{\max}$, where $\gamma \in h$. On the other hand, since $h^{\min} \leq h^+ \leq h^{\max}$ then $1 - h^{\min} \geq 1 - h^+ \geq 1 - h^{\max}$.

From the above, it can now be easily shown that

$$S(\alpha_{\min}) \leq S(H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n)) \leq S(\alpha_{\max})$$

And hence $\alpha_{\min} \leq H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) \leq \alpha_{\max}$.

Theorem 4 (Additivity) Let $\alpha_j = \langle [a_j, b_j, c_j, d_j]; h_j, (1 - h_j^+) \rangle$ ($j = 1, 2, \dots, n$) be a collection of $H-ITFNs$. Also, let $\alpha = \langle [a, b, c, d]; h, (1 - h^+) \rangle$ be another $H-ITFN$, then

$$\begin{aligned} & H - ITFPEWA(\alpha_1 \oplus_{\varepsilon} \alpha, \alpha_2 \oplus_{\varepsilon} \alpha, \dots, \alpha_n \oplus_{\varepsilon} \alpha) = \\ & H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n) \oplus_{\varepsilon} \alpha \end{aligned}$$

Theorem 5 (Multiplicity) If $\alpha_j = \langle [a_j, b_j, c_j, d_j]; h_j, (1 - h_j^+) \rangle$ ($j = 1, 2, \dots, n$) be a collection of $H-ITFNs$ and $\lambda > 0$, then

$$H - ITFPEWA(\lambda \alpha_1, \lambda \alpha_2, \dots, \lambda \alpha_n) = \lambda H - ITFPEWA(\alpha_1, \alpha_2, \dots, \alpha_n)$$

Theorem 6 (Monotonicity) Let $\alpha_j^1 = \langle [a_j^1, b_j^1, c_j^1, d_j^1]; h_j^1, (1 - h_j^{1+}) \rangle$ and $\alpha_j^2 = \langle [a_j^2, b_j^2, c_j^2, d_j^2]; h_j^2, (1 - h_j^{2+}) \rangle$ be a collection of $H-ITFNs$. Let $a_j^1 \leq a_j^2, b_j^1 \leq b_j^2, c_j^1 \leq c_j^2, d_j^1 \leq d_j^2$ and $\lambda_j^1 \leq \lambda_j^2$ for all j , where $\lambda_j^1 \in h_j^1$ and $\lambda_j^2 \in h_j^2$. Then,

$$H - ITFPEWA(\alpha_1^1, \alpha_2^1, \dots, \alpha_n^1) \leq H - ITFPEWA(\alpha_1^2, \alpha_2^2, \dots, \alpha_n^2)$$

Theorems 4–6 can also easily be proved.

Now, the concept of H-ITFPEWG operator has been introduced.

Definition 13 Let $\alpha_j = \left([a_j, b_j, c_j, d_j]; h_j, \left(1 - h_j^+\right) \right) (j = 1, 2, \dots, n)$ be a collection of H-ITFNs. Also, let $H-ITFPEWG : V^n \rightarrow V$ be a function such that

$$H-ITFPEWG(\alpha_1, \alpha_2, \dots, \alpha_n) = \otimes_{\varepsilon_{j=1}}^n (\alpha_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} \quad (22)$$

Then the $H-ITFPEWG(\alpha_1, \alpha_2, \dots, \alpha_n)$ is called an H-ITFPEWG operator, where $T_j = \prod_{k=1}^{j-1} S(\alpha_k) (j = 2, 3, \dots, n)$ and $T_1 = 1$. Also $S(\alpha_k)$ represents the score value of the H-ITFN α_k .

Theorem 7 Let $\alpha_j (j = 1, 2, \dots, n)$ be a collection of H-ITFNs, then the aggregated value using H-ITFPEWG operator is also an H-ITFN and

$$\begin{aligned} H-ITFPEWG(\alpha_1, \alpha_2, \dots, \alpha_n) &= \otimes_{\varepsilon_{j=1}}^n (\alpha_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} \\ &= \left(\left[\prod_{j=1}^n a_j^{\frac{T_j}{\sum_{j=1}^n T_j}}, \prod_{j=1}^n b_j^{\frac{T_j}{\sum_{j=1}^n T_j}}, \prod_{j=1}^n c_j^{\frac{T_j}{\sum_{j=1}^n T_j}}, \prod_{j=1}^n d_j^{\frac{T_j}{\sum_{j=1}^n T_j}} \right] \right. \\ &\quad ; \bigcup_{\gamma_i \in h_i, i=1,2,\dots,n} \frac{2 \prod_{j=1}^n (\gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (2 - \gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (\gamma_j)^{\frac{T_j}{\sum_{j=1}^n T_j}}} \\ &\quad \left. , \frac{\prod_{j=1}^n (2 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} - \prod_{j=1}^n (h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}}}{\prod_{j=1}^n (2 - h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} + \prod_{j=1}^n (h_j^+)^{\frac{T_j}{\sum_{j=1}^n T_j}} \right) \quad (23) \end{aligned}$$

where $T_j = \prod_{k=1}^{j-1} S(\alpha_k) (j = 2, 3, \dots, n)$, $T_1 = 1$, and $S(\alpha_k)$ is the score value of the H-ITFN α_k .

Proof The proof of this theorem is similar to the proof of Theorem 1.

Theorem 8 (Idempotency) Let $\alpha_j = \left([a_j, b_j, c_j, d_j]; h_j, \left(1 - h_j^+\right) \right)$ be a collection of H-ITFNs. If $\alpha_j = \alpha$ for all j , then

$$H-ITFPEWG(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha \quad (24)$$

Proof Similar to the proof of Theorem 2.

5 An Approach to Multi-criteria Group Decision-Making with H-ITF Information

In this section, H-ITFPEWA and H-ITFPEWG operators are used to solve MCDM problems. A MCDM problem under H-ITF environment is described below.

Let $X = \{x_1, x_2, \dots, x_m\}$ be a set of alternatives. $C = \{C_1, C_2, \dots, C_n\}$ be a collection of criteria and the linear ordering $C_1 > C_2 > C_3 > \dots > C_n$ represents a prioritization between the criteria in such a manner that the criteria C_j have a higher priority than C_i , if $j < i$. And $E = \{e_1, e_2, \dots, e_p\}$ represents a set of decision-makers and the linear ordering $e_1 > e_2 > e_3 > \dots > e_p$ represents prioritization between the decision-makers in such a manner that the decision-maker e_i has a higher priority to take the decision than decision-maker e_j if $i < j$.

Let $R_{m \times n}^{(q)} = (r_{ij}^{(q)})_{m \times n}$ ($q = 1, 2, \dots, p$) be a H-ITF decision matrix, whose elements are represented by H-ITFNs with the form $r_{ij}^{(q)} = ([r_{1ij}^{(q)}, r_{2ij}^{(q)}, r_{3ij}^{(q)}, r_{4ij}^{(q)}]; h_{ij}^{(q)}, 1 - h_{ij}^{(q)+})$. Each $r_{ij}^{(q)}$ designates the value of the alternative $x_i \in X$ on the criteria $C_j \in C$ provided by the decision-maker e_q , where $h_{ij}^{(q)}$ represents the membership degree of the alternative x_i that satisfies the attribute C_j expressed by the decision-maker e_q ; and $(1 - h_{ij}^{(q)+})$ indicates the degree of the alternative x_i that does not satisfies the attribute C_j expressed by the decision-maker e_q , where $h_{ij}^{(q)} \subseteq [0, 1]$ satisfies $0 \leq \max\{h_{ij}^{(q)}\} + (1 - h_{ij}^{(q)+}) \leq 1, i = 1, 2, \dots, m; j = 1, 2, \dots, n; q = 1, 2, \dots, p$.

Then, the developed H-ITFPEWA (and H-ITFPEWG) operators are used to develop an approach for solving multi-criteria decision-making problems in a H-ITF environment. The proposed methodology is described through the following steps:

Step 1. Calculate the value of $T_{ij}^{(q)}$, ($q = 1, 2, \dots, p$) with the following equations.

$$T_{ij}^{(q)} = \prod_{k=1}^{q-1} S(r_{ij}^{(k)}), (q = 1, 2, \dots, p), \text{ and } T_{ij}^{(1)} = 1. \tag{25}$$

Step 2. To aggregate all the individual H-ITF decision matrices $R_{m \times n}^{(q)} = (r_{ij}^{(q)})_{m \times n}$ ($q = 1, 2, \dots, p$) into the collective H-ITF decision matrix $R_{m \times n} = (r_{ij})_{m \times n}$, $i = 1, 2, \dots, m; j = 1, 2, \dots, n$, the following H-ITFPEWA and H-ITFPEWG operators are used.

Thus using the H-ITFPEWA operator

$$r_{ij} = H - ITFPEWA(r_{ij}^{(1)}, r_{ij}^{(2)}, \dots, r_{ij}^{(p)})$$

$$\begin{aligned}
&= \left(\left[\sum_{q=1}^p \left(\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}} r_{1ij}^{(q)} \right), \sum_{q=1}^p \left(\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}} r_{2ij}^{(q)} \right), \right. \right. \\
&\quad \left. \left. \sum_{q=1}^p \left(\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}} r_{3ij}^{(q)} \right), \sum_{q=1}^p \left(\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}} r_{4ij}^{(q)} \right) \right]; \right. \\
&\quad \left. \bigcup_{\gamma_{ij}^{(q)} \in h_{ij}^{(q)}} \frac{\prod_{q=1}^p \left(1 + \gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} - \prod_{q=1}^p \left(1 - \gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}{\prod_{q=1}^p \left(1 + \gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} + \prod_{q=1}^p \left(1 - \gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}}, \right. \\
&\quad \left. \frac{2 \prod_{q=1}^p \left(1 - h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}{\prod_{q=1}^p \left(1 + h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} + \prod_{q=1}^p \left(1 - h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}} \right) \quad (26)
\end{aligned}$$

and using the H-ITFPEWG operator

$$\begin{aligned}
r_{ij} &= H - ITFPEWG \left(r_{ij}^{(1)}, r_{ij}^{(2)}, \dots, r_{ij}^{(p)} \right) \\
&= \left(\left[\prod_{q=1}^p \left(r_{1ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}, \prod_{q=1}^p \left(r_{2ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}, \right. \right. \\
&\quad \left. \left. \prod_{q=1}^p \left(r_{3ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}, \prod_{q=1}^p \left(r_{4ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} \right]; \right. \\
&\quad \left. \bigcup_{\gamma_{ij}^{(q)} \in h_{ij}^{(q)}} \frac{2 \prod_{q=1}^p \left(\gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}{\prod_{q=1}^p \left(2 - \gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} + \prod_{q=1}^p \left(\gamma_{ij}^{(q)} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}}, \right. \\
&\quad \left. \frac{\prod_{q=1}^p \left(2 - h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} - \prod_{q=1}^p \left(h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}}{\prod_{q=1}^p \left(2 - h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}} + \prod_{q=1}^p \left(h_{ij}^{(q)+} \right)^{\frac{T_{ij}^{(q)}}{\sum_{q=1}^p T_{ij}^{(q)}}}} \right) \quad (27)
\end{aligned}$$

Step 3. Calculate the values of T_{ij} ($i = 1, 2, \dots, m$), ($j = 1, 2, \dots, n$) based on the following equations,

$$T_{ij} = \prod_{k=1}^{j-1} S(r_{ik}) \text{ and } T_{i1} = 1, (i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (28)$$

Step 4. Aggregate the H-ITFNs r_{ij} for each alternative x_i using the H-ITFPEWA (and H-ITFPEWG) operator as follows:

$$\begin{aligned} r_i^A &= H - ITFPEWA(r_{i1}, r_{i2}, \dots, r_{in}) \\ &= \left(\left[\sum_{j=1}^n \left(\frac{T_{ij}}{\sum_{j=1}^n T_{ij}} r_{1ij} \right), \sum_{j=1}^n \left(\frac{T_{ij}}{\sum_{j=1}^n T_{ij}} r_{2ij} \right), \sum_{j=1}^n \left(\frac{T_{ij}}{\sum_{j=1}^n T_{ij}} r_{3ij} \right), \right. \right. \\ &\quad \left. \left. \sum_{j=1}^n \left(\frac{T_{ij}}{\sum_{j=1}^n T_{ij}} r_{4ij} \right) \right]; \bigcup_{\gamma_{ij} \in h_{ij}} \frac{\prod_{j=1}^n (1 + \gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} - \prod_{j=1}^n (1 - \gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}{\prod_{j=1}^n (1 + \gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} + \prod_{j=1}^n (1 - \gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}, \right. \\ &\quad \left. \frac{2 \prod_{j=1}^n (1 - h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}{\prod_{j=1}^n (1 + h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} + \prod_{j=1}^n (1 - h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} \right) \end{aligned} \quad (29)$$

and

$$\begin{aligned} r_i^G &= H - ITFPEWG(r_{i1}, r_{i2}, \dots, r_{in}) \\ &= \left(\left[\prod_{j=1}^n (r_{1ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}, \prod_{j=1}^n (r_{2ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}, \prod_{j=1}^n (r_{3ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}, \right. \right. \\ &\quad \left. \prod_{j=1}^n (r_{4ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} \right]; \bigcup_{\gamma_{ij} \in h_{ij}} \frac{2 \prod_{j=1}^n (\gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}{\prod_{j=1}^n (2 - \gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} + \prod_{j=1}^n (\gamma_{ij})^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}, \\ &\quad \left. \frac{\prod_{j=1}^n (2 - h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} - \prod_{j=1}^n (h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}}}{\prod_{j=1}^n (2 - h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} + \prod_{j=1}^n (h_{ij}^+)^{\frac{T_{ij}}{\sum_{j=1}^n T_{ij}}} \right) \\ & i = 1, 2, \dots, m; j = 1, 2, \dots, n. \end{aligned} \quad (30)$$

Step 5. Using the proposed score function as in definition 10 the rank of all alternatives is evaluated.

6 A Numerical Illustration

To establish the application potentiality of the developed methodology, a modified version of a practical problem previously described by Yu [36] is considered.

The school of management in a Chinese university wants to introduce oversea outstanding teachers to strengthen academic education. This introduction has raised great attention from the school; university president (e_1), dean of management school (e_2), and human resource officer (e_3) sets up the panel of decision-makers which will take the whole responsibility for this introduction. They made strict evaluation for five candidates x_i ($i = 1, 2, 3, 4, 5$) from four aspects, namely morality (C_1), research capability (C_2), teaching skills (C_3), educational background (C_4). University president has the absolute priority for making decision, dean of the management school comes next. Besides, this introduction will be strict accordance with the principle of combined ability with political integrity. The prioritization relationship for the criteria is as $C_1 \succ C_2 \succ C_3 \succ C_4$. Three decision-makers evaluated the candidates x_i ($i = 1, 2, 3, 4, 5$) with respect to the attributes C_j ($j = 1, 2, \dots, 4$) and constructed the following three decision matrices whose elements are represented by H-ITF numbers:

$$R_{m \times n}^{(q)} = \left(r_{ij}^{(q)} \right)_{5 \times 4} \quad (q = 1, 2, 3). \quad \text{(Where)}$$

$$R_{5 \times 4}^{(1)} = \left[\begin{array}{l} \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.1, 0.2, 0.3, 0.4]; \\ (0.6), 0.4 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.8, 0.9]; \\ (0.75, 0.85), 0.15 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.9), 0.1 \end{array} \right) \\ \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.7, 0.8]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.75, 0.85), 0.15 \end{array} \right) \\ \left(\begin{array}{l} [0.7, 0.8, 0.9, 1.0]; \\ (0.8, 0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.9]; \\ (0.75, 0.85, 0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.4, 0.6, 0.8, 0.9]; \\ (0.45), 0.55 \end{array} \right) \\ \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.7]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.1, 0.3, 0.5, 0.7]; \\ (0.7, 0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.7, 0.8, 0.9, 1.0]; \\ (0.3), 0.7 \end{array} \right) \\ \left(\begin{array}{l} [0.2, 0.3, 0.4, 0.5]; \\ (0.6, 0.75, 0.85), 0.15 \end{array} \right) \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.6), 0.4 \end{array} \right) \left(\begin{array}{l} [0.2, 0.3, 0.4, 0.5]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.5, 0.6), 0.4 \end{array} \right) \end{array} \right]$$

$$R_{5 \times 4}^{(2)} = \left[\begin{array}{l} \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.75, 0.85), 0.15 \end{array} \right) \left(\begin{array}{l} [0.1, 0.3, 0.4, 0.6]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.7, 0.8]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.3, 0.4), 0.6 \end{array} \right) \\ \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.3, 0.4, 0.6, 0.7]; \\ (0.75, 0.8, 0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.8]; \\ (0.75), 0.25 \end{array} \right) \\ \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.5, 0.7, 0.8, 0.9]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.75), 0.25 \end{array} \right) \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.6, 0.8), 0.2 \end{array} \right) \\ \left(\begin{array}{l} [0.2, 0.3, 0.4, 0.5]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.1, 0.2, 0.3, 0.5]; \\ (0.3, 0.4), 0.6 \end{array} \right) \left(\begin{array}{l} [0.1, 0.2, 0.4, 0.6]; \\ (0.75, 0.85), 0.15 \end{array} \right) \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.6), 0.4 \end{array} \right) \\ \left(\begin{array}{l} [0.1, 0.2, 0.3, 0.4]; \\ (0.45), 0.55 \end{array} \right) \left(\begin{array}{l} [0.2, 0.3, 0.4, 0.5]; \\ (0.6), 0.4 \end{array} \right) \left(\begin{array}{l} [0.1, 0.2, 0.3, 0.4]; \\ (0.9), 0.1 \end{array} \right) \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.7, 0.9), 0.1 \end{array} \right) \end{array} \right]$$

$$R_{5 \times 4}^{(3)} = \left[\begin{array}{cccc} \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.2, 0.3, 0.4, 0.5]; \\ (0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.6, 0.7, 0.9, 1.0]; \\ (0.65, 0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.3), 0.7 \end{array} \right) \\ \left(\begin{array}{l} [0.7, 0.8, 0.9, 1.0]; \\ (0.6), 0.4 \end{array} \right) & \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.75, 0.8, 0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.5, 0.6, 0.8, 0.9]; \\ (0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.6), 0.4 \end{array} \right) \\ \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.5, 0.6, 0.7, 0.8]; \\ (0.6), 0.4 \end{array} \right) & \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.7, 0.8, 0.9, 1.0]; \\ (0.8, 0.9), 0.1 \end{array} \right) \\ \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.6, 0.7, 0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.2, 0.4, 0.5, 0.6]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.2, 0.4, 0.6, 0.8]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.7, 0.8, 0.9, 1.0]; \\ (0.75), 0.25 \end{array} \right) \\ \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.4, 0.5, 0.6, 0.7]; \\ (0.75), 0.25 \end{array} \right) & \left(\begin{array}{l} [0.3, 0.4, 0.5, 0.6]; \\ (0.7, 0.9), 0.1 \end{array} \right) & \left(\begin{array}{l} [0.6, 0.7, 0.8, 0.9]; \\ (0.45), 0.55 \end{array} \right) \end{array} \right]$$

Using H-ITFPEWA operator, the preference ordering of the five candidates is found by performing the following steps:

Step A1: In this step, the value of $T_{ij}^{(1)}, T_{ij}^{(2)}, T_{ij}^{(3)}$ is calculated as described in (25).

$$T_{ij}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, T_{ij}^{(2)} = \begin{bmatrix} 0.6620 & 0.5150 & 0.6877 & 0.6980 \\ 0.7700 & 0.6219 & 0.6125 & 0.6743 \\ 0.7789 & 0.7145 & 0.6219 & 0.4848 \\ 0.5844 & 0.5891 & 0.6190 & 0.4490 \\ 0.5808 & 0.5270 & 0.5656 & 0.5280 \end{bmatrix}$$

$$T_{ij}^{(3)} = \begin{bmatrix} 0.4286 & 0.2913 & 0.4924 & 0.3343 \\ 0.4789 & 0.4341 & 0.4005 & 0.4098 \\ 0.5998 & 0.5437 & 0.3868 & 0.3106 \\ 0.3658 & 0.2870 & 0.3634 & 0.2420 \\ 0.2871 & 0.2746 & 0.3337 & 0.3504 \end{bmatrix}$$

Step A2: The H-ITFPEWA operator is used to aggregate all the individual decision matrices $R^{(q)} = (r_{ij}^{(q)})_{5 \times 4}$, ($q = 1, 2, 3$). The following aggregated decision matrix $R = (r_{ij})_{5 \times 4}$ is found.

$$R_{5 \times 4} = \left[\begin{array}{cc} \left(\begin{array}{l} [0.3932, 0.4932, 0.5932, 0.6932]; \\ (0.8372, 0.8621), 0.1379 \end{array} \right) & \left(\begin{array}{l} [0.1161, 0.2446, 0.3446, 0.4731]; \\ (0.7156), 0.2844 \end{array} \right) \\ \left(\begin{array}{l} [0.5871, 0.6871, 0.7871, 0.8871]; \\ (0.8129), 0.1871 \end{array} \right) & \left(\begin{array}{l} [0.4909, 0.5909, 0.6909, 0.7909]; \\ (0.8089, 0.8179, 0.8424), 0.1576 \end{array} \right) \\ \left(\begin{array}{l} [0.6420, 0.7420, 0.8420, 0.9420]; \\ (0.8656, 0.9000), 0.1000 \end{array} \right) & \left(\begin{array}{l} [0.5000, 0.6316, 0.7316, 0.8759]; \\ (0.7870, 0.8303, 0.8577), 0.1423 \end{array} \right) \\ \left(\begin{array}{l} [0.2888, 0.3888, 0.4888, 0.5888]; \\ (0.7895, 0.8015, 0.8385), 0.1615 \end{array} \right) & \left(\begin{array}{l} [0.2219, 0.3372, 0.4372, 0.6219]; \\ (0.6438, 0.6643), 0.3357 \end{array} \right) \\ \left(\begin{array}{l} [0.1843, 0.2843, 0.3843, 0.4843]; \\ (0.5859, 0.6757, 0.7499), 0.2501 \end{array} \right) & \left(\begin{array}{l} [0.2860, 0.3860, 0.4860, 0.5860]; \\ (0.6266), 0.3734 \end{array} \right) \end{array} \right]$$

$$\left[\begin{array}{l} \left(\begin{array}{l} [0.4910, 0.5910, 0.7910, 0.8910]; \\ (0.7953, 0.8112, 0.8384, 0.8512), 0.1488 \end{array} \right) \\ \left(\begin{array}{l} [0.3895, 0.4895, 0.6895, 0.7895]; \\ (0.7903, 0.8042, 0.8409), 0.1591 \end{array} \right) \\ \left(\begin{array}{l} [0.5193, 0.6193, 0.7193, 0.8193]; \\ (0.7500), 0.2500 \end{array} \right) \\ \left(\begin{array}{l} [0.1183, 0.2871, 0.4871, 0.6871]; \\ (0.7257, 0.7650, 0.8411, 0.8651), 0.1349 \end{array} \right) \\ \left(\begin{array}{l} [0.1878, 0.2878, 0.3878, 0.4878]; \\ (0.8016, 0.8365), 0.1635 \end{array} \right) \\ \left(\begin{array}{l} [0.3821, 0.4821, 0.5821, 0.6821]; \\ (0.7072, 0.7263), 0.2737 \end{array} \right) \\ \left(\begin{array}{l} [0.4873, 0.5873, 0.6873, 0.8197]; \\ (0.7249, 0.7833), 0.2167 \end{array} \right) \\ \left(\begin{array}{l} [0.5059, 0.6616, 0.8173, 0.9173]; \\ (0.5698, 0.6118, 0.6391, 0.6758), 0.3242 \end{array} \right) \\ \left(\begin{array}{l} [0.6469, 0.7469, 0.8469, 0.9469]; \\ (0.4671), 0.5329 \end{array} \right) \\ \left(\begin{array}{l} [0.4905, 0.5905, 0.6905, 0.7905]; \\ (0.5557, 0.6622, 0.6064, 0.7030), 0.2970 \end{array} \right) \end{array} \right]$$

Step A3: Using (28) the value of T_{ij} ($i = 1, 2, 3, 4, 5$), ($j = 1, 2, 3, 4$) are found as follows

$$T_{ij} = \begin{bmatrix} 1 & 0.6655 & 0.3630 & 0.2524 \\ 1 & 0.6875 & 0.4658 & 0.3070 \\ 1 & 0.7763 & 0.5408 & 0.6255 \\ 1 & 0.6172 & 0.3348 & 0.2040 \\ 1 & 0.5499 & 0.2940 & 0.1740 \end{bmatrix}$$

Step A4: Aggregate the H-ITFNs r_{1j} for an alternative x_1 by the H-ITFPEWA operator and get r_1^A .

$$r_1^A = ([0.3267, 0.4350, 0.5509, 0.6592]; \\ (0.7871, 0.7888, 0.7898, 0.7914, 0.7949, 0.7964, \\ 0.7975, 0.7990, 0.8016, 0.8032, 0.8041, 0.8057, 0.8089, \\ 0.8104, 0.8113, 0.8128), 0.1872)$$

and in this similar way we find out r_2^A, r_3^A, r_4^A and r_5^A .

Step A5: The score value of r_i^A ($i = 1, 2, 3, 4, 5$) for each candidate is evaluated using Eq. (13):

$$S(r_1^A) = 0.6218, S(r_2^A) = 0.6734, S(r_3^A) = 0.6901, S(r_4^A) = 0.5919, S(r_5^A) = 0.5575.$$

since $S(r_3^A) > S(r_2^A) > S(r_1^A) > S(r_4^A) > S(r_5^A)$, the ordering of individuals becomes $x_3 > x_2 > x_1 > x_4 > x_5$.

Now, the given problem is solved using H-ITFPEWG operator, for finding the preference ordering of the candidates. The following steps are performed:

Step G1: Same as above step A1.

Step G2: Utilize the H-ITFPEWG operator to aggregate all the individual H-ITF decision matrix $R_{m \times n}^{(q)} = (r_{ij}^{(q)})_{5 \times 4}$, ($q = 1, 2, 3$)

$$R_{5 \times 4} = \left[\begin{array}{cc} \left(\begin{array}{l} [0.3788, 0.4815, 0.5833, 0.6846]; \\ (0.8207, 0.8529), 0.1471 \\ (0.7813), 0.2187 \end{array} \right) & \left(\begin{array}{l} [0.1118, 0.2397, 0.3411, 0.4655]; \\ (0.6873), 0.3127 \end{array} \right) \\ \left(\begin{array}{l} [0.5825, 0.6832, 0.7837, 0.8840]; \\ (0.8576, 0.9000), 0.1000 \end{array} \right) & \left(\begin{array}{l} [0.4857, 0.5866, 0.6872, 0.7877]; \\ (0.7945, 0.8051, 0.8260), 0.1740 \end{array} \right) \\ \left(\begin{array}{l} [0.6402, 0.7404, 0.8406, 0.9408]; \\ (0.7640, 0.7843, 0.8221), 0.1779 \end{array} \right) & \left(\begin{array}{l} [0.5000, 0.6300, 0.7302, 0.8748]; \\ (0.7580, 0.8020, 0.8237), 0.1763 \end{array} \right) \\ \left(\begin{array}{l} [0.2804, 0.3826, 0.4839, 0.5848]; \\ (0.1716, 0.2764, 0.3785, 0.4797]; \\ (0.5713, 0.6475, 0.6973), 0.3027 \end{array} \right) & \left(\begin{array}{l} [0.1997, 0.3218, 0.4259, 0.6151]; \\ (0.5801, 0.6262), 0.3738 \end{array} \right) \\ \left(\begin{array}{l} [0.1716, 0.2764, 0.3785, 0.4797]; \\ (0.5713, 0.6475, 0.6973), 0.3027 \end{array} \right) & \left(\begin{array}{l} [0.2784, 0.3804, 0.4816, 0.5824]; \\ (0.6218), 0.3782 \end{array} \right) \\ \left(\begin{array}{l} [0.4856, 0.5865, 0.7877, 0.8881]; \\ (0.7727, 0.7964, 0.8185, 0.8427), 0.1573 \end{array} \right) & \left(\begin{array}{l} [0.3759, 0.4772, 0.5781, 0.6786]; \\ (0.5459, 0.5946), 0.4054 \end{array} \right) \\ \left(\begin{array}{l} [0.3831, 0.4844, 0.6859, 0.7864]; \\ (0.7791, 0.7944, 0.8245), 0.1755 \end{array} \right) & \left(\begin{array}{l} [0.4822, 0.5830, 0.6837, 0.8187]; \\ (0.7193, 0.7660), 0.2340 \end{array} \right) \\ \left(\begin{array}{l} [0.5179, 0.6181, 0.7182, 0.8184]; \\ (0.7500), 0.2500 \end{array} \right) & \left(\begin{array}{l} [0.4916, 0.6574, 0.8165, 0.9166]; \\ (0.5427, 0.5568, 0.5911, 0.6060), 0.3940 \end{array} \right) \\ \left(\begin{array}{l} [0.1135, 0.2786, 0.4822, 0.6836]; \\ (0.7246, 0.7549, 0.8246, 0.8564), 0.1436 \end{array} \right) & \left(\begin{array}{l} [0.6402, 0.7412, 0.8419, 0.9425]; \\ (0.4204), 0.5796 \end{array} \right) \\ \left(\begin{array}{l} [0.1747, 0.2797, 0.3818, 0.4831]; \\ (0.7847, 0.8200), 0.1800 \end{array} \right) & \left(\begin{array}{l} [0.4858, 0.5867, 0.6872, 0.7876]; \\ (0.5420, 0.5897, 0.5964, 0.6468), 0.3532 \end{array} \right) \end{array} \right]$$

Step G3: Calculate the value of T_{ij} ($i = 1,2,3,4,5$), ($j = 1,2,3,4$)

$$T_{ij} = \begin{bmatrix} 1 & 0.6550 & 0.3519 & 0.2408 \\ 1 & 0.6612 & 0.4398 & 0.2850 \\ 1 & 0.7738 & 0.5192 & 0.3247 \\ 1 & 0.6068 & 0.3201 & 0.1932 \\ 1 & 0.5366 & 0.2858 & 0.1667 \end{bmatrix}$$

Step G4: Aggregate the H-ITFNs r_{ij} for each alternative x_i by the H-ITFPEWG operator and get r_i^G ($i = 1, 2, 3, 4, 5$).

$$r_1^G = ([0.2757, 0.4049, 0.5224, 0.6367]; \\ (0.7423, 0.7483, 0.7460, 0.7520, 0.7493, 0.7553, \\ 0.7530, 0.7590, 0.7561, 0.7621, 0.7598, 0.7658, \\ 0.7632, 0.7692, 0.7669, 0.7729), 0.2271)$$

Consecutively we find out r_2^G, r_3^G, r_4^G and r_5^G .

Step G5: Calculate the score value of r_i^G ($i = 1, 2, 3, 4, 5$) using (13) as follows

$$S(r_1^G) = 0.5934, S(r_2^G) = 0.6549, S(r_3^G) = 0.6725, S(r_4^G) = 0.5654, S(r_5^G) = 0.5426$$

since $S(r_3^G) > S(r_2^G) > S(r_1^G) > S(r_4^G) > S(r_5^G)$ the ordering becomes $x_3 > x_2 > x_1 > x_4 > x_5$.

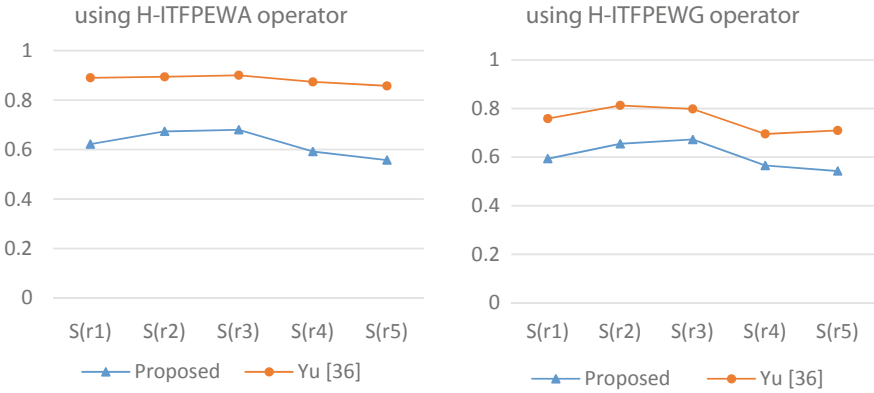


Fig. 2 The comparisons of results are provided through the following figures

It is worthy to mention here that the ranking achieved by Yu [36] is $x_3 > x_2 > x_1 > x_4 > x_5$ and $x_2 > x_3 > x_1 > x_4 > x_5$ using IFPWA and IFPWG operators, respectively. However, using the proposed approach it has been found that the ranking of alternatives remains almost the same using averaging as well as geometric operators. Thus, the proposed method is more consistent than the method developed by Yu [36]. Thus, the proposed method is flexible enough to establish decision-maker’s preferences on the alternatives. The comparison results are graphically shown in Fig. 2.

It is also clear from the figures and achieved results that the difference between any two consecutive score values is increased. Hence, the ordering of the alternatives using this proposed methodology can be performed more efficiently.

7 Conclusions and Scope for Future Studies

The traditional hesitant fuzzy aggregation operators are generally suitable for aggregating the information in the form of fuzzy numbers with some degree of hesitancy; and are unable to deal with hesitant-intuitionistic fuzzy information, which contains some degree of non-membership also. In this paper, H-ITF information aggregation is used to solve MCGDM problems. A new score function is introduced to overcome the difficulties that arise [7] due to higher value of non-membership. A prioritization relationship over the criteria is defined based on Einstein operations. H-ITFPEWA and H-ITFPEWG operators are introduced to aggregate the H-ITFNs using the newly defined score function and Einstein operations for making a reasonable decision in the H-ITF decision-making contexts. Some properties of those operators are discussed in detail. Finally, an illustrative example is considered to demonstrate the proposed methodology. The comparison of the results reflects betterment of the proposed methodology over the existing techniques [36] and provides consistent results

using both the operators. As the scope for future study the proposed method can be extended to solve decision-making problems using dual hesitant fuzzy set [37], hesitant Pythagorean fuzzy set [38], Pythagorean hesitant fuzzy set [39], interval-valued dual hesitant fuzzy [40], and other areas.

However, it is hoped that the proposed methodology may open up new vistas into the way of making a reasonable decision in the hesitant-intuitionistic fuzzy group decision-making environments.

Acknowledgements The authors remain grateful to the reviewers for their valuable comments and suggestions to improve the quality of the manuscript.

References

1. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8:338–353
2. Atanassov KT (1986) Intuitionistic fuzzy sets. *Fuzzy Sets Syst* 20(1):87–96
3. Atanassov KT (1989) More on intuitionistic fuzzy sets. *Fuzzy Sets Syst* 33:37–45
4. Atanassov KT (1999) Intuitionistic fuzzy sets, vol 35. Springer Physica, Heidelberg
5. Zhang Z (2016) Multi-criteria group decision-making methods based on new intuitionistic fuzzy Einstein hybrid weighted aggregation operators. *Neural Comput Appl* 28:3781–3800
6. Sedef C, Balaman SY (2018) A novel outranking based multi criteria group decision making methodology integrating ELECTRE and VIKOR under Intuitionistic fuzzy environment. *Expert Syst Appl* 119(1):36–50
7. Garg H, Arora R (2019) Generalized intuitionistic fuzzy soft power aggregation operator based on t-norm and their application in multicriteria decision-making. *Int J Intell* 34(2):215–246
8. Liu P, Tang G (2018) Some Intuitionistic fuzzy prioritized interactive einstein Choquet operators and their application in decision making. *IEEE Access*, 1. <https://doi.org/10.1109/access.2018.2882071>
9. Dubois D, Prade H (1980) Fuzzy sets and systems: theory and applications. Academic Press, New York
10. Li DF (2008) A note on using intuitionistic fuzzy sets for fault-tree analysis on printed circuit board assembly. *Microelectron Reliab* 48(10):1741
11. Shu MH, Cheng CH, Chang JR (2006) Using intuitionistic fuzzy sets for fault tree analysis on printed circuit board assembly. *Microelectron Reliab* 46(12):2139–2148
12. Li DF (2010) A ratio ranking method of triangular intuitionistic fuzzy numbers and its application to MADM problems. *Comput Math Appl* 60:1557–1570
13. Wan SP, Wang QY, Dong JY (2013) The extended VIKOR method for multi-attribute group decision making with triangular intuitionistic fuzzy numbers. *Knowl Based Syst* 52:65–77
14. Wang JQ (2008) Overview on fuzzy multi-criteria decision-making approach. *Control Decis* 23(6):601–607
15. Jianqiang W, Zhong Z (2009) Aggregation operators on intuitionistic trapezoidal fuzzy number and its application to multi-criteria decision making problems. *J Syst Eng Electron* 20(2):321–326
16. Wei GW (2010) Some arithmetic aggregation operators with intuitionistic trapezoidal fuzzy numbers and their application to group decision making. *J Comput* 5(3):345–351
17. Das S, Guha D (2016) A centroid-based ranking method of trapezoidal Intuitionistic fuzzy numbers and its application to MCDM problems. *Fuzzy Inf Eng* 8(1):41–74
18. Torra V, Narukawa Y (2009) On hesitant fuzzy sets and decision. In: *The 18th IEEE international conference on fuzzy systems Jeju Island Korea*, 1378–1382
19. Torra V (2010) Hesitant fuzzy sets. *Int J Intell Syst* 25:529–539

20. Xia M, Xu Z (2011) Hesitant fuzzy information aggregation in decision-making. *Int J Approx Reason* 52:395–407
21. Fahmi A, Abdullah S, Amin F, Ali A, Ahmad R, Shakeel M (2019) Trapezoidal cubic hesitant fuzzy aggregation operators and their application in group decision-making. *J Intell Fuzzy Syst* 36(4):3619–3635
22. Liao H, Xu Z (2013) A VIKOR-based method for hesitant fuzzy multi-criteria decision making. *Fuzzy Optim Decis Mak* 12(4):373–392
23. Yu D (2015) Hesitant fuzzy multi-criteria decision making methods based on heronian mean. *Technol Econ Dev Econ* 23(2):296–315
24. Wang H, Xu Z, Zeng XJ (2018) Hesitant fuzzy linguistic term sets for linguistic decision making: current developments, issues and challenges. *Inf Fusion* 43:1–12. <https://doi.org/10.1016/j.inffus.2017.11.010>
25. Xu Z, Zhang X (2013) Hesitant fuzzy multi-attribute decision making based on TOPSIS with incomplete weight information. *Knowl Based Syst* 52:53–64
26. Xu Z, Xia M (2011) On distance and correlation measures of hesitant fuzzy information. *Int J Intell Syst* 26:410–425
27. Yu D, Zhang W, Xu Y (2013) Group decision making under hesitant fuzzy environment with application to personal evaluation. *Knowl-Based Syst* 52:1–10
28. Xu Z, Zhang S (2019) An overview on the applications of the hesitant fuzzy sets in group decision-making: Theory, support and methods. *Frontiers Eng Manag* 6(2):163–182
29. Yu D (2014) Some hesitant fuzzy information aggregation operators based on einstein operational laws. *Int J Intell Syst* 29(4):320–340
30. Zhou W, Xu Z, Chen M (2015) Preference relations on hesitant intuitionistic fuzzy information and their application in group decision making. *Comput Ind Eng* 87:163–175
31. Yager RR (2008) Prioritized aggregation operators. *Int J Approx Reason* 48:236–274
32. Wei G (2012) Hesitant fuzzy prioritized operators and their application. *Knowl Based Syst* 31:176–182
33. Yu Q, Hou F, Zhai Y, Du Y (2015) Some hesitant fuzzy Einstein aggregation operators and their application to multiple attribute group decision making. *Int J Intell Syst* 31(7):722–746
34. Xu ZS, Yager RR (2006) Some geometric aggregation operators based on intuitionistic fuzzy sets. *Int J Gen Syst* 35:417–433
35. Klement EP, Mesiar R, Pap E (2004) Triangular norms Position paper I: basic analytical and algebraic properties. *Fuzzy Sets Syst* 143:5–26
36. Yu D (2013) Intuitionistic fuzzy prioritized operators and their application in multi-criteria group decision making. *Technol Econ Dev Econ* 19:1–21. *Economy*
37. Biswas A, Sarkar A (2018) Development of dual hesitant fuzzy prioritized operators based on Einstein operations with their application to multi-criteria group decision making. *Arch Control Sci* 28(4):527–549
38. Liang D, Xu Z (2017) The new extension of TOPSIS method for multiple criteria decision making with hesitant Pythagorean fuzzy sets. *Appl Soft Comput* 60:167–179
39. Sarkar A, Biswas A (2019) Multicriteria decision-making using Archimedean aggregation operators in Pythagorean hesitant fuzzy environment. *Int J Intell Syst* 34:1361–1386
40. Sarkar A, Biswas A (2019) On developing interval-valued dual hesitant fuzzy Bonferroni mean aggregation operator and their application to Multicriteria decision making. *Commun Comput Inf Sci* 1030:27–46

Unsupervised Feature Selection Using Information-Theoretic Graph-Based Approach



Sagarika Saroj Kundu, Abhirup Das and Amit Kumar Das

Abstract Feature selection is a critical part of any machine learning project involving data sets with high dimensionality. Selecting an optimal subset consisting of important features reduces the execution time and increases the predictive ability of the machine learning model. This paper presents a novel graph-based feature selection algorithm for unsupervised learning. Unlike many of the algorithms using correlation as a measure of dependency between features, the proposed algorithm derives feature dependency using information-theoretic approach. The proposed algorithm—Graph-based Information-Theoretic Approach for Unsupervised Feature Selection (GITAUFs) generates multiple minimal vertex covers (MVC) of the feature graph and evaluates them to find the most optimal one in context of the learning task. In our experimental setup comprising 13 benchmark data sets, GITAUFs has shown a 10% increase in the silhouette width value along with a significant feature reduction of 90.62% compared to the next best performing algorithm.

Keywords Feature selection · Mutual information · Minimum vertex cover · Graph-based visualization

1 Introduction

Feature selection is a critical area of research focus, especially in domains having a large number of attributes. Such domains include processing of internet documents [8], customer review analysis [23] and interpretation of data from genomic projects [15, 25] to name a few. It is advantageous as it allows us to design cost-effective

S. S. Kundu (✉) · A. Das · A. K. Das
Institute of Engineering and Management, Kolkata, India
e-mail: tiakundu3@gmail.com

A. Das
e-mail: abhirupdas.iem@gmail.com

A. K. Das
e-mail: amitkrdas.kol@gmail.com

machine learning models as well as reduce model execution time in high-dimensional data sets. An unsupervised machine learning algorithm draws inferences from data without having any known labeled responses. In unsupervised learning, the grouping of unlabeled data instances needs to be done based on some specific set of statistical measures. In this context, feature selection is a combinatorial optimization problem where the objective is to find an optimal feature subset from the entire feature set such that no information is effectively lost from the data in question. The features need to be selected on the basis of how informative they are and contribute to the specific unsupervised learning task. Also, it is important to consider how redundant the features are based on their similarity with other features.

In general, approaches adopted for feature selection are wrapper, filter, embedded, and hybrid. The wrapper approach algorithmically learns and determines the best suited subset of features depending on prediction accuracy. It is very accurate but is prone to overfitting [4, 5, 9]. On the contrary, in the filter approach, statistical measures are used in place of learning algorithms, making it suitable for high-dimensional data sets. The embedded approach chooses the optimal feature subset during training. The hybrid approach exploits the benefits of both filter and wrapper approaches.

In feature selection, it is important to adopt a suitable similarity measure in order to evaluate inter-feature similarity. It is also important to establish feature relevance in order to decide which features should be selected as part of the final subset. A feature is considered irrelevant when it contributes very minimal information and is thus insignificant for tasks such as clustering of given data instances; it can be removed if it does not significantly contribute to the learning task. Feature relevance can be measured using measures like joint mutual information, symmetrical relevance, entropy [16, 18, 26]. Features whose contribution is nearly the same as one or more other features are considered to be potentially redundant. Such similarities can be measured using different measures like Fisher score, Pearson's correlation, mutual information, etc. [12, 13].

In this paper, we have proposed a graph theory-based algorithm to represent the combinatorial relationship between different features of an input data set. This allows visualizing the degree of inter-feature similarity and hence feature redundancy. It also derives a feature subset by using graph theoretic principles of finding subgraphs from graphs [7]. In our algorithm, after rejecting features based on their entropy, the features are represented as graph-based on their mutual information statistics. Then the optimal subset of features is obtained by using the two-approximation algorithm of minimal vertex cover. The use of mutual information to determine the association between features is supported by its ability to measure the general dependence between features and obtain a complete characterization of symbolic as well as numeric sequences and features, as opposed to classical methods like PCA (for dimensionality reduction), KMeans (for clustering) [19, 22] or measures like Pearson's correlation that are able to capture linear relations at best. Furthermore, unlike mutual information, these classical methods are sensitive to scale effects and necessitate the use of preprocessing measures like normalization prior to designing models.

In the following Sect. 2, the existing literature of feature selection algorithms related to the proposed work has been discussed. In Sect. 3, important insights into different statistical concepts for determining relevance and redundancy have been discussed. Our proposal has been described in Sect. 4 with several relevant illustrations in Sect. 5. Section 6 contains an analysis of our experiments with relevant tables and graphs provided. Conclusions are in Sect. 7.

2 Related Work

Feature selection is a topic of great interest in research. Various methods have been adopted by researchers for feature selection. However, very few papers have proposed a graph-based feature selection approach. In graph-based feature selection, features are represented as vertices of a graph and the edges represent the relationship between features like inter-feature similarity, etc. Graph theoretic principle for deriving subgraphs of graphs is used in graph-based feature selection.

In a related work [2], features have been mapped as vertices and weighted edges represent inter-feature mutual information. In this paper, the first subset of features are selected to minimize redundancy by selecting the densest subgraph. Then the final feature subset is obtained by clustering the nonredundant features.

In [20], a community detection algorithm has been proposed. The features represented as vertices in the graph are clustered using community detection algorithm. This reduces the feature subset. Then features from each cluster are selected, in iterations, if they have value more than a threshold. This is done until there is no feature having value more than the threshold.

Lu et al. [17] proposed a feature selection algorithm where a subset of the available features are chosen using the same criteria as PCA and applied it to tasks like face tracking and content-based image retrieval. In another feature selection algorithm [14], a Laplacian score is calculated which determines the locality preserving power. The local structure of data space is the focus rather than the global structure. The nearest neighbor graph is plotted to evaluate the local geometric structure. Feature association mapping has been used as an underlying concept in [6] which applies to both supervised and unsupervised learning. The approach uses maximal independent set and graph theoretic approach of minimal vertex cover to derive the final set of features.

Another work [21] has used an ant colony based search process, a nature inspired optimization algorithm. The approach also uses graph-based modeling of the feature set. Hill-climbing based approach with a graphical representation of the input data set has been shown in [11].

In a recent work [10], a graph encoder based technique for feature selection has been adopted. Different works related to graph-based feature selection involve finding the feature-to-class or feature-to-feature relation.

The information contributed by a feature for the learning task has been a focus in the work by [27]. The approach uses multidimensional interaction information (MII)

as the selection criteria for features into the final feature subset. MII is measured based on the mutual information between the feature subset selected and the target class. Dominant-set clustering is used to group the feature vectors. The final feature subset is made by selecting the features from the dominant-set using the MII criteria. This ensures a final reduced feature set having maximum information about the class.

3 Background Concepts

This section highlights the concepts required for GITAUFS. In the proposed algorithm, there are three stages. First, the irrelevant features are removed from the candidate feature set. Then the association among selected features are measured through the use of well-established algorithms and experimentally chosen threshold values to determine the ones which are redundant and need to be excluded.

3.1 Feature Relevance

The entropy of a feature allows us to quantify the average information contributed by it through measurement of the unpredictability of state; with higher information content being indicated by lower probability. This information I is expressed as:

$$I(x_i) = \log_2 \frac{1}{P(x_i)} = -\log_2 P(x_i) \quad (1)$$

According to Shannon, entropy H of feature $X = \{x_i\}_{i=1\dots n}$ is defined as:

$$H(X) = \sum_{i=1}^n P(x_i) I(x_i) = -\sum_{i=1}^n P(x_i) \log_2 P(x_i) \quad (2)$$

Entropy value is used to remove features that carry little information relevant to learning. Since this information measure is based on the probability distribution of a random variable and does not depend on its actual values, it has been widely used in feature selection [3, 24]. When $P(X)$ is distributed uniformly, the maximum entropy of X is reached. This means that it has the highest level of unpredictability or maximum information content. For that reason, many methods employ some form of entropy in the objective function for clustering.

For each feature x , the information $\overline{I}_{\mathcal{E}-\{x\}}$ contributed by the entire set of features is measured, i.e., \mathcal{E} minus x using Eq. 2. All the features are thus ranked according to the metric \overline{I} and among these ones having higher value are considered to be potentially irrelevant.

3.2 Feature Redundancy

GITAUFS uses mutual information of candidate features as the starting component of a pruning algorithm to strip away redundant features from the initial subset of available ones. While measures like correlation allow us to understand the association between different features and dimensionality reduction techniques like PCA that are sufficient for patterns distributions across different classes, feature selection using these fail in the case of classification or clustering tasks with complex decision boundaries [3], since they consider only linear relations between features. On the other hand, mutual information can measure arbitrary relations between variables and is independent of the transformation done on them and is suitable for assessing their “information content” for a robust estimation of redundancy. Hence, if two features f_1 and f_2 are strongly similar, or one contributes significant amount of information about the other, then their MI is large. If both f_1 and f_2 retained in the feature subset for clustering, then the results obtained will be similar to the ones obtained when any one of the features is used.

The mutual information M for a pair of features (X, Y) , is defined as

$$M(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \quad (3)$$

where $P(x, y)$ represents the joint probability. In particular, Eq. 3 measures how much information is communicated, on average, in one random variable about another. Features having mutual information of 0 are considered statistically independent. Based on this, matrix MI is constructed, as described in Sect. 5.

4 Proposed Approach

In the approach proposed, data set features represent the vertices in the graph. The main stages of GITAUFS algorithm include selecting the relevant features, identifying the potentially redundant features, and selecting the most optimal feature subset based on silhouette width value after evaluating γ minimal vertex covers derived from the potentially redundant features.

- Step 1: Highlighting the irrelevant (or least informative) features.
Entropy can determine the information contribution of a feature in the data set for a learning task. High entropy for a particular feature signifies high information contribution by the feature for the clustering of data. Thus, features with high entropy need to be considered for further evaluation. At this stage, high entropy features are marked “green” and the rest are marked “red” as they are irrelevant (due to low information contribution). The irrelevant features are thus highlighted in this stage.

- Step 2: Highlighting potentially redundant features.
 The features marked “green” in the previous stage are evaluated in this stage for potential similarity between them. Mutual Information between features is calculated. The similarity matrix (MI_{mat}) generated holds each cell corresponding to the Mutual Information value between features respective to the column and row for that cell. The mean of the similarity matrix is used as the threshold to indicate how high or low is the similarity between two features. Feature pairs having mutual information value greater than the mean are considered similar and below or equal to the mean are considered as dissimilar.
 The adjacency matrix is derived using mutual information between features and the threshold value (mean of the matrix). The cells having mutual information lesser than or equal to the mean are made 0, and rest are made 1. The leading diagonal of the matrix is made 0 as it represents the mutual information of a feature to itself. The similar features are represented in the graph using the adjacency matrix. These features are colored “blue” in the feature graph and they contain the potentially redundant features. The rest of the features initially marked “green” are candidates for the final subset of features without further evaluation because they have no similarity with any other feature as well as high information contribution (due to high entropy). These features are hence kept “green” in color. This stage thus marks similar features with edges.
- Step 3: Highlighting the final feature subset.
 At this stage, a feature subset is selected from the connected features marked “blue” as a representative of the whole set. This ensures that the information content of the features remains the same even though the number of features reduces. Minimal Vertex Cover (MVC) algorithm has been used for this purpose. A subset of vertices having edges incident to at least one of the vertices is identified. However, the minimum subset is not always determined by the algorithm. Finding the minimum feature subset is a NP hard problem. The proposed approach, however, tries to find the best subset among multiple minimal vertex covers given by MVC algorithm. The subsets are ranked based on the entropy (or information contribution) of the features. The top γ subsets having higher entropy value are selected. These subsets are then evaluated based on the silhouette width value. The subset giving the highest silhouette value is selected as the final subset. The features in this subset are then marked “green” and the rest of the “blue” vertices are colored “red”.

By the conclusion of Step 3, the final set of features GITAUFS algorithm for the data set is ready. The features represented by “red” colored vertices are the rejected features and the final subset of selected features is represented in “green”. All the three stages of GITAUFS generate graph which conveys important information about the features in the respective data set (Figs. 1 and 2).

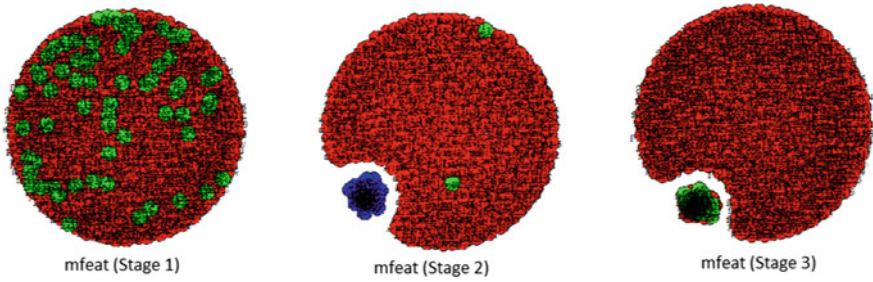


Fig. 1 Illustration based on ‘mfeat’ data set

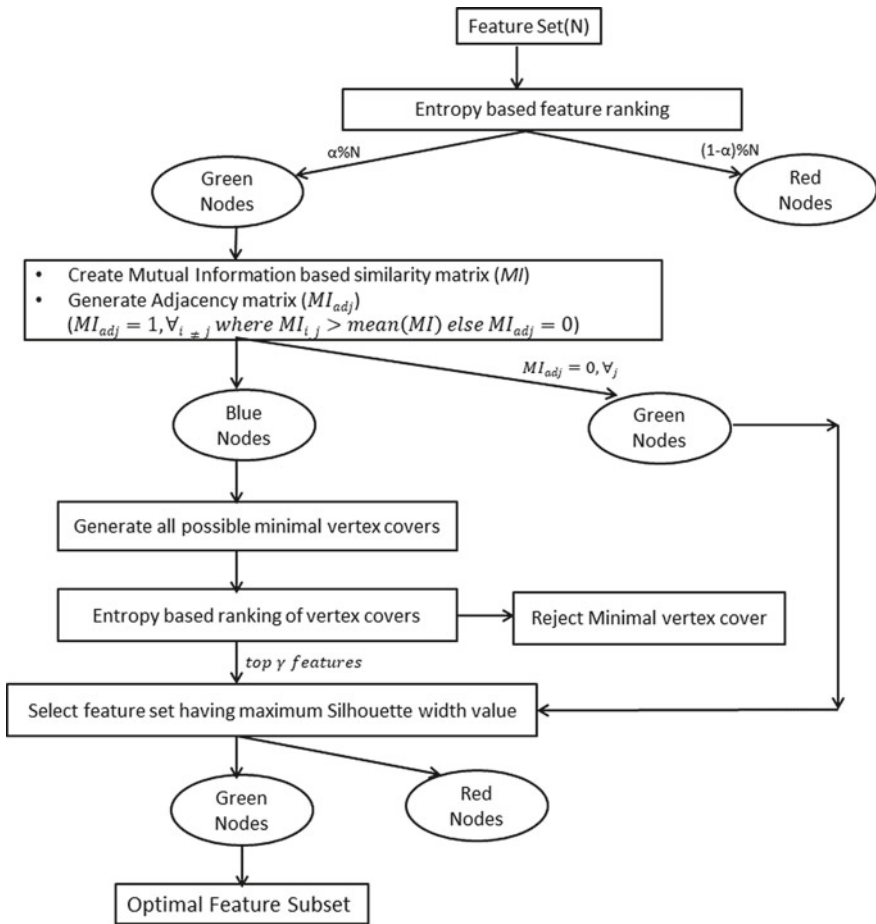


Fig. 2 Flow of GITAUFs algorithm

Algorithm: Graph-based Information-Theoretic Approach for Unsupervised Feature Selection (GITAUFs)

Input: N-dimensional data set D_N having original feature set $F = f_1, f_2, \dots, f_N$,

α - Relevance threshold

γ - Number of MVCs to be compared for final feature set

Output: Optimal feature subset F_{opt} .

Begin

/ Stage 1: Calculate feature entropy and color the top contributing features as green.*/*

1: For $i = 1$ to N

2: $E_i = \text{ENTROPY}(f_i)$

3: Next

4: SORT(E)

5: For $i = 1$ to $(\alpha\%N)$

6: $\text{color}(f_i) = \text{"green"}$

7: Next

8: For $i = ((\alpha\%N)+1)$ to N

9: $\text{color}(f_i) = \text{"red"}$

10: Next

11: $g_1 = \text{generateGITAUFs}(F)$

/ Stage 2: Color the similar features, among the possible optimal feature set, as "blue".*/*

12: $F' = \{x: x \subseteq F \text{ and } \text{color}(x) = \text{"green"}\}$

13: $MI_{mat} = \text{mutual information}(D_N[F'])$

14: For $i = 1$ to $|F'|$

15: For $j = 1$ to $|F'|$

16: If $((MI_{mat} > \text{mean}(MI_{mat})) \ \& \ (i \neq j))$ then

17: add edge(F_i, F_j, g_1)

18: $\text{color}(F_i) = \text{"blue"}$

19: $\text{color}(F_j) = \text{"blue"}$

20: End If

21: Next

22: Next

*/*Stage 3: The MVC algorithm generates possible minimal set of features from "blue" marked features.*

Top γ subsets based on entropy ranking are further evaluated for silhouette width value and the subset with highest silhouette width value is declared as the final feature subset by GITAUFs and is marked by "green" color./*

23: $F'' = \{x: x \subseteq F \text{ and } \text{color}(x) = \text{"blue"}\}$

24: $V = \{x: x \in \text{Minimal-Vertex-Covers}(F'')\}$

25: For $i = 1$ to $\text{length}(V)$

26: $S_i = \sum (\text{ENTROPY}(f_j)), \{f_j: f_j \in V_i, V_i = \text{Minimal-vertex-covers}_i(F'')\}$

27: Next

```

28: SORT(S)
29: For i = 1 to  $\gamma$ 
30:   opt = max(silhouette width-value( $S_i$ ))
31: color( $V_{opt}$ ) = "green"
32: color( $F'' - V_{opt}$ ) = "red"
33:  $F_{opt} = \{x: x \subseteq F \text{ and color}(x) = \text{"green"}\}$ 
End

```

5 Illustration

In this section, the three stages of GITAUFs approach have been illustrated with the help of generated graphs with color codes representing the selected and rejected vertices after each pruning stage.

- Stage-1: The initial filtering is carried out according to the entropy E of each attribute using Eq. 2 as described in Sect. 4. The top $\alpha\%$ of attributes are chosen for the next stages (they are colored "green") and the rest are discarded (colored "red"). In Fig. 3a, "At2" and "At4" are filtered out from the list of seven attributes.
- Stage-2: The mutual information (MI) of each of the "green" attributes is calculated using Eq. 3 mentioned earlier. The mean \overline{MI} of the resulting similarity matrix is used as threshold to distinguish potentially redundant features from others. An adjacency matrix MI_{adj} is created having $MI_{adj} = 1, \forall_{i \neq j}$ if $MI_{i,j} > \overline{MI}$ and $MI_{adj} = 0, \forall_{i=j}$. The features having a higher than average mutual information are grouped into a subset (colored "blue"). "At5" is not a redundant attribute and is selected as a final attribute (colored "green") for Stage 3, as shown in Fig. 3b. After this, the attributes having mutual information $MI_{i,j} > \overline{MI}$ are connected by edges to form the graph also depicted in the figure.

$$MI = \begin{pmatrix} & At_1 & At_3 & At_5 & At_6 & At_7 \\ At_1 & 1.34 & 0.12 & 0.24 & 0.69 & 0.56 \\ At_3 & 0.12 & 1.37 & 0.15 & 0.54 & 0.58 \\ At_5 & 0.24 & 0.15 & 1.38 & 0.32 & 0.17 \\ At_6 & 0.69 & 0.54 & 0.32 & 1.38 & 0.68 \\ At_7 & 0.56 & 0.58 & 0.17 & 0.68 & 1.37 \end{pmatrix}, \quad \overline{MI} = 0.46$$

$$MI_{adj} = \begin{pmatrix} & At_1 & At_3 & At_5 & At_6 & At_7 \\ At_1 & 0 & 0 & 0 & 1 & 1 \\ At_3 & 0 & 0 & 0 & 1 & 1 \\ At_5 & 0 & 0 & 0 & 0 & 0 \\ At_6 & 1 & 1 & 0 & 0 & 1 \\ At_7 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

- Stage-3 : The final selection of vertices is done from the "blue" vertices. The vertices/attributes marked "green" in Stage 2 are directly selected into the final

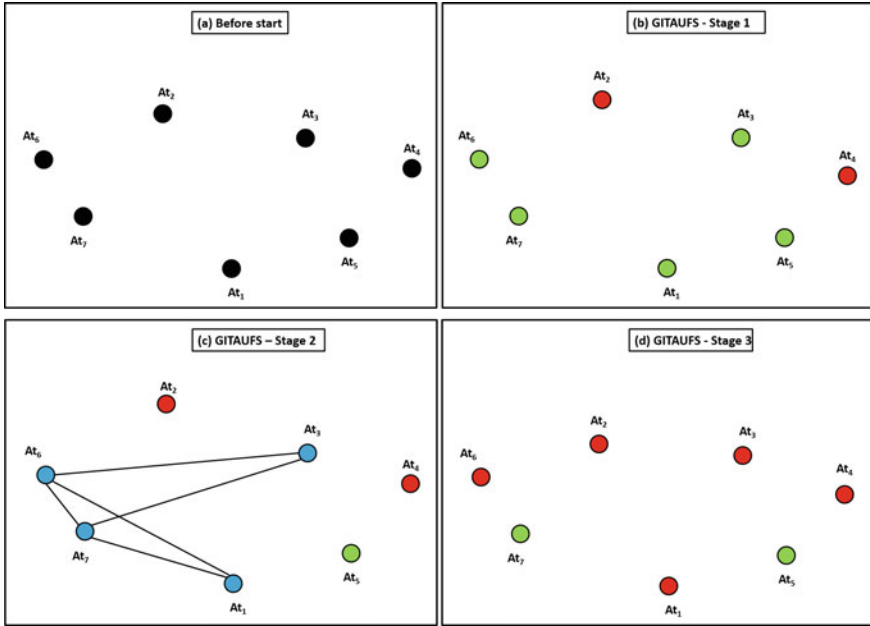


Fig. 3 Illustration for GITAUPS

feature subset. Minimal vertex cover (MVC) is run on features marked ‘blue’ in Stage 2. The minimal vertex algorithm returns all possible minimal vertex covers, of which the top γ minimal vertex covers having high entropy are further evaluated along with features marked ‘green’ in Stage 2 on the basis of silhouette value. The subset having highest silhouette width value is the most final feature subset derived by GITAUPS. These features are marked ‘green’ such as “At7”, is chosen (colored “green”) while the others, such as “At1”, “At3”, and “At6” are rejected (colored “red”), as shown in Fig. 3c.

6 Experiments and Outcome

Our approach has been benchmarked on data sets obtained from the UCI Machine Learning repository [1]. Our graphs have been generated using the ‘matplotlib’ library in Python. Table 1 presents the data sets used for experimenting. The value of α is 10% in this experiment. GITAUPS has been compared with two graph-based feature selection algorithms—DSUB [2] and UFAM [6] and two benchmark feature selection algorithms—Laplacian [14] and Principal Feature Analysis (PFA) [17].

Table 1 Description of UCI data sets

Data set	# of features	# of instances
apndcts	7	106
btissue	9	106
cleave	13	297
ecoli	7	336'
glass	9	214
ILPD	10	579
mfeat	649	2000
pima	8	768
sonar	60	208
vehicle	18	846
wbdc	30	569
wine	13	178
wiscon	9	682

6.1 Summary of Outcome

The proposed algorithm (GITAUFS) has been evaluated for its performance by comparing it with other competing algorithms based on three main aspects—silhouette width value, percentage of feature reduction, and execution time. The following sub sections describe the comparative results obtained in each of these aspects.

Table 2 Performance of silhouette width

Data set	GITAUFS	UFAM	ALL	LAPLACIAN	PFA	DSUB
apndcts	0.66	0.6	0.46	0.44	0.48	0.43
btissue	0.65	0.62	0.58	0.56	0.53	0.58
cleave	0.55	0.31	0.27	0.57	0.58	0.27
ecoli	0.54	0.54	0.44	0.56	0.47	0.44
glass	0.59	0.61	0.52	0.54	0.53	0.52
ILPD	0.86	0.71	0.49	0.72	0.72	0.49
mfeat	0.20	0.2	0.24	0.18	0.14	0.19
pima	0.56	0.27	0.51	0.27	0.27	0.24
sonar	0.41	0.17	0.47	0.19	0.12	0.39
vehicle	0.62	0.52	0.32	0.43	0.38	0.47
wbdc	0.55	0.44	0.65	0.49	0.29	0.4
wine	0.57	0.4	0.002	0.35	0.31	0.32
wiscon	0.63	0.69	0.19	0.66	0.65	0.65

6.1.1 Comparison of Silhouette Width Value

The silhouette width values for the proposed algorithm GITAUFs along with other competing algorithms UFAM, LAPLACIAN, PFA, and DSUB have been recorded in Table 2. The column corresponding to ALL contains silhouette width values for all features of the dataset.

A graphical comparison of the performance of silhouette width value has been presented in Fig. 4. From Table 2 and Fig. 4 it is evident that GITAUFs has outperformed the benchmark algorithms. GITAUFs has recorded the highest or the near highest silhouette width value for all the data sets used. The summary of the comparison based on silhouette width value is given below.

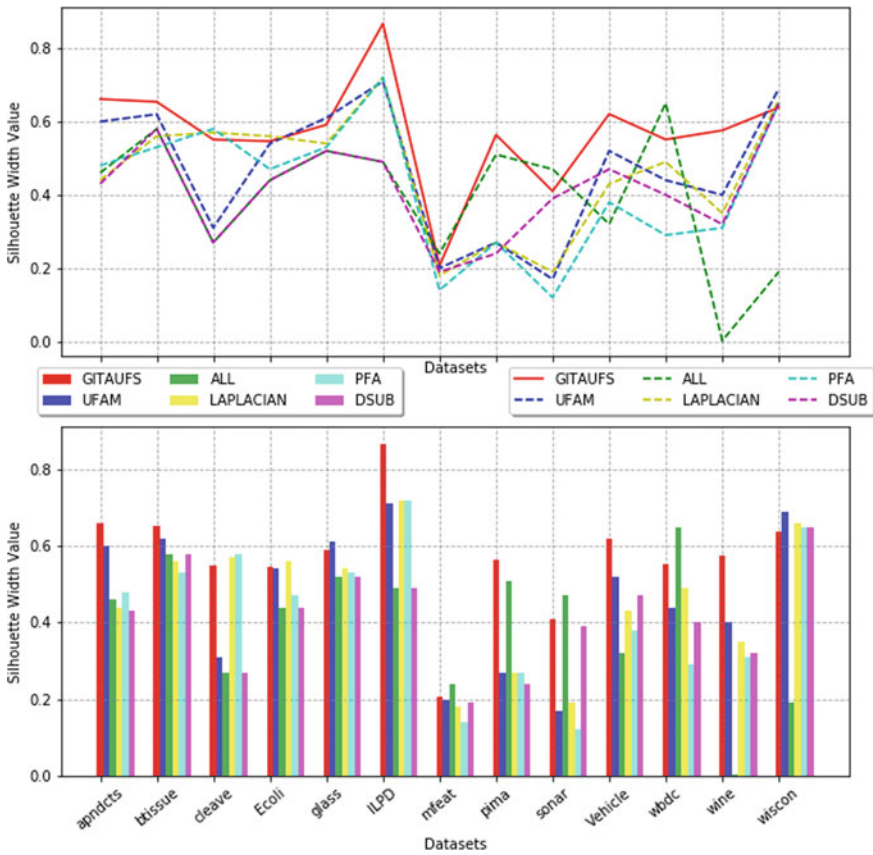


Fig. 4 Performance of silhouette width

- GITAUFs has the highest silhouette width value for 6 out of 13 data sets used in the experiment.
- For the other data sets where GITAUFs does not have highest silhouette width value, it is giving value very close to the best value
- GITAUFs has outperformed both the benchmark algorithms—Laplacian and PFA with respect to the silhouette width value

6.1.2 Comparison of Feature Reduction

The percentage of feature reduction for GITAUFs and other competing algorithms has been presented in Table 3. Graphical representation of the comparison has been shown in Fig. 5.

GITAUFs has shown a very high percentage of feature reduction for all the data sets when compared to the benchmark algorithms. An overview of the comparison for feature reduction is given below.

- GITAUFs has shown the highest reduction in the number of features compared to the competing algorithms for all the data sets except ‘mfeat’ where DSUB has a higher reduction in the number of features. However, GITAUFs has a better silhouette width value than DSUB for ‘mfeat’ justifying its efficiency
- GITAUFs has the highest feature reduction as high as 96.67% and the lowest being 85.71% which is also very high compared to the other competing algorithms
- GITAUFs has performed extremely good for high-dimensional data set giving a reduction in features exceeding 91%.

Table 3 Percentage feature reduction

Dataset	GITAUFs	UFAM	PFA	LAPLACIAN	DSUB
apndcts	85.71	50	57.14	57.14	71.43
btissue	88.89	62.5	55.56	55.56	66.67
cleave	92.31	25	23.08	23.08	76.92
Ecoli	85.71	33.33	28.57	28.57	71.43
glass	88.89	37.5	33.33	33.33	77.78
ILPD	90	44	30	30	70
mfeat	91.68	29.78	85.82	85.82	99.69
pima	87.5	14.29	12.5	12.5	75
sonar	95	44.07	63.33	63.33	68.33
vehicle	94.44	35.29	72.22	72.22	88.89
wbdc	96.67	31.03	76.67	76.67	93.33
wine	92.31	25	38.46	38.46	84.62
wiscon	88.89	37.5	33.33	33.33	77.78

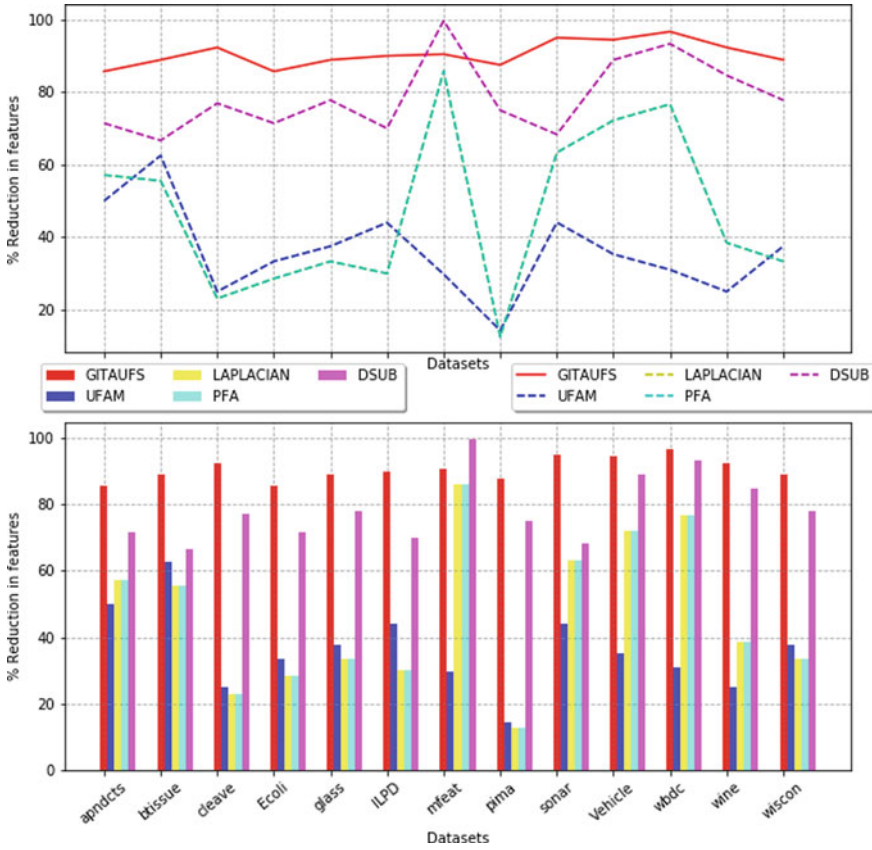


Fig. 5 Feature reduction

6.1.3 Comparison of Execution Time

The values representing the execution time for GITAUFs and other competing algorithm is shown in Table 4. Execution time for GITAUFs is very close to benchmark algorithms. A summary of the conclusions drawn on comparing the execution time is given below.

- The execution time for GITAUFs is almost similar to that of benchmark algorithms
- For high-dimensional data sets like ‘mfeat’, GITAUFs has a fairly acceptable execution time. DSUB has almost double execution time for ‘mfeat’ and the execution time of PFA is almost 48 times higher than GITAUFs.

6.2 Overall Comparison of Performance

An overall comparison of GITAUFs has been done in Table 5 with benchmark algorithms. Summary of the silhouette width performance is given in Table 6. A graphical representation of mean rank for various competing algorithms based on silhouette width value is shown in Fig. 6.

Observations made from the Tables 5, 6 and Fig. 6 are described below.

- GITAUFs has higher silhouette width value than all the other competing algorithms. The mean rank for GITAUFs is the lowest which indicates that GITAUFs has the best silhouette value for most of the data sets used. GITAUFs has the highest number of wins based on its performance of silhouette width.
- Feature reduction is the highest for GITAUFs. GITAUFs has a mean feature reduction of 90.62% which is very high in general and especially high when compared with other competing algorithms.

Table 4 Execution time (in seconds)

Dataset	GITAUFs	UFAM	PFA	LAPLACIAN	DSUB
apndcts	0.17	0.39	0.15	0.12	0.58
btissue	0.25	0.4	0.18	0.15	0.09
cleave	0.90	0.4	1.74	0.17	0.16
ecoli	1.37	0.41	1.27	0.1	0.09
glass	0.61	0.41	0.46	0.12	0.11
ILPD	1.09	0.39	9.58	0.2	0.11
mfeat	854.13	2.63	41,034.06	25.99	1,531.69
pima	1.17	0.39	23.9	0.14	0.1
sonar	0.96	0.02	2.63	0.75	2.63
vehicle	2.74	0.47	65.5	0.14	0.3
wbdc	1.03	0.01	32.4	0.27	0.52
wine	0.34	0.38	0.62	0.16	0.14
wiscon	0.67	0.4	17.78	0.13	0.11

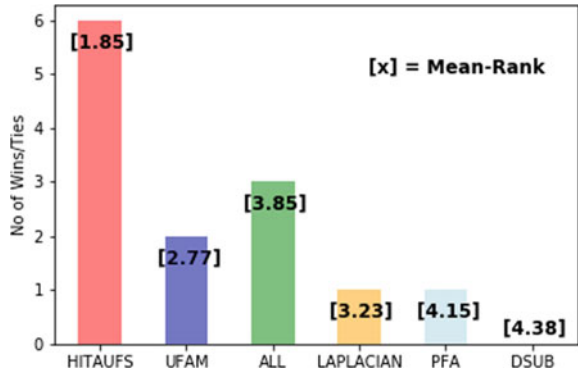
Table 5 Comparison of different algorithms

Algorithm	Mean silhouette width	Mean execution time (s)	Mean feature reduction (%)
GITAUFs	0.57	66.58	90.62
UFAM	0.47	0.52	36.1
LAPLACIAN	0.46	2.19	46.92
PFA	0.42	3168.48	46.92
DSUB	0.41	118.2	78.61

Table 6 Summary of performance (Silhouette Width)

Algorithm	Mean rank	Number of Wins/Ties
HITAUFS	1.85	6
UFAM	2.77	2
ALL	3.85	3
LAPLACIAN	3.23	1
PFA	4.15	1
DSUB	4.38	0

Fig. 6 Wins/Ties for silhouette width value



- The execution time for GITAUFS is similar to other competing algorithms as seen in Table 4. For high-dimensional data sets, UFAM has best execution time. However, GITAUFS also has better than average execution time for high-dimensional data sets.

7 Conclusion

The proposed approach, GITAUFS, has shown significant feature reduction of 90.62% which is 12% higher than the next best performing algorithm. GITAUFS also gives a high silhouette width value of 57% with the lowest mean rank among other competing algorithms and stands out as the best performer. GITAUFS is an information-theoretic approach which captures the general dependency between features. GITAUFS also addresses the challenge of finding the minimal vertex covers being an NP hard problem by evaluating multiple minimum vertex cover. GITAUFS has an added advantage of being a graph-based approach which provides the visualization of all the stages of the algorithm. The graphical representation of features as vertices of a graph gives a visual understanding of the relevance of features and similarity between features. GITAUFS has the overall best performance when compared to the benchmark algorithms for the experimental setup.

References

1. Bache K, Lichman M (2013) Uci machine learning repository <http://archive.ics.uci.edu/ml>. irvine, ca: University of California. School Inf Comput Sci **28**
2. Bandyopadhyay S, Bhadra T, Mitra P, Maulik U (2014) Integration of dense subgraph finding with feature clustering for unsupervised feature selection. *Pattern Recognit Lett* 40:104–112
3. Battiti R (1994) Using mutual information for selecting features in supervised neural net learning. *IEEE Trans Neural Netw* 5(4):537–550
4. Bennasar M, Hicks Y, Setchi R (2015) Feature selection using joint mutual information maximisation. *Expert Syst Appl* 42(22):8520–8532
5. Brown G, Pocock A, Zhao MJ, Luján M (2012) Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *J Mach Learn Res* 13:27–66
6. Das AK, Goswami S, Chakrabarti A, Chakraborty B (2017) A new hybrid feature selection approach using feature association map for supervised and unsupervised classification. *Expert Syst Appl* 88:81–94
7. Das AK, Goswami S, Chakraborty B, Chakrabarti A (2017) A graph-theoretic approach for visualization of data set feature association. *Adv Comput Syst Secur*, 109–124. Springer
8. Dey Sarkar S, Goswami S, Agarwal A, Aktar J (2014) A novel feature selection technique for text classification using naive bayes. *Int Sch Res Notices*
9. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. *J Bioinform Comput Biol* 3(02):185–205
10. Feng S, Duarte MF (2018) Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation. *Neurocomputing* 312:310–323
11. Goswami S, Das AK, Guha P, Tarafdar A, Chakraborty S, Chakrabarti A, Chakraborty B (2017) An approach of feature selection using graph-theoretic heuristic and hill climbing. *Pattern Anal Appl*, 1–17 (2017)
12. Gu Q, Li Z, Han J (2012) Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* (2012)
13. Hall MA (1999) Correlation-based feature selection for machine learning
14. He X, Cai D, Niyogi P (2006) Laplacian score for feature selection. In: *Advances in neural information processing systems*, 507–514
15. Hua J, Tembe WD, Dougherty ER (2009) Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recogn* 42(3):409–424
16. Lewis DD (1992) Feature selection and feature extraction for text categorization. In: *Proceedings of the workshop on speech and natural language*, pp 212–217. Association for Computational Linguistics (1992)
17. Lu Y, Cohen I, Zhou XS, Tian Q (2007) Feature selection using principal feature analysis. In: *Proceedings of the 15th ACM international conference on Multimedia*, pp 301–304. ACM
18. Meyer PE, Schretter C, Bontempi G (2008) Information-theoretic feature selection in microarray data using variable complementarity. *IEEE J Sel Topics Signal Process* 2(3):261–274
19. Moghaddam B, Pentland A (1995) Probabilistic visual learning for object detection. In: *Proceedings of IEEE international conference on computer vision*, pp 786–793. IEEE (1995)
20. Moradi P, Rostami M (2015) A graph theoretic approach for unsupervised feature selection. *Eng Appl Artif Intell* 44:33–45
21. Moradi P, Rostami M (2015) Integration of graph clustering with ant colony optimization for feature selection. *Knowl Based Syst* 84:144–161
22. Murphy K, Torralba A, Eaton D, Freeman W (2006) Object detection and localization using local and global features. In: *Toward category-level object recognition*, pp 382–400. Springer
23. Ng K, Liu H (2000) Customer retention via data mining. *Artif Intell Rev* 14(6):569–590
24. Quinlan JR (2014) *C4. 5: programs for machine learning*. Elsevier

25. Xing EP, Jordan MI, Karp RM (2001) Feature selection for high-dimensional genomic microarray data. In: ICML, vol. 1, pp. 601–608. Citeseer (2001)
26. Yang HH, Moody J (2000) Data visualization and feature selection: New algorithms for non-gaussian data. In: Advances in neural information processing systems, pp 687–693 (2000)
27. Zhang Z, Hancock ER (2011) A graph-based approach to feature selection. In: International workshop on graph-based representations in pattern recognition, pp 205–214. Springer (2011)

Fact-Based Expert System for Supplier Selection with ERP Data



Kartick Chandra Mondal, Biswadeep Deb Nandy and Arunima Baidya

Abstract For any business enterprise, supply chain management (SCM) plays an important role in an organization's decision- and profit-making process. A very crucial step in SCM is supplier selection. It is such a pivotal step because it deploys a large amount of a firm's financial resources. In return, the firms expect significant interest from contracting with suppliers offering higher value. Any discrepancy in this process can lead to low SCM performance which in turn may cause financial losses as well as bring about a decline in the firm's market performance. This paper deals with the development of a strictly fact-based expert system for appropriate supplier selection and shows how rules can be broken down into atomic clauses.

Keywords Supply chain management · Supplier selection · Knowledge based · Fact based · Expert system

1 Introduction

A supply chain is basically a connected network of individuals, resources, activities, and technologies involved in the manufacture and sale of a product which starts with the delivery of raw materials from a supplier to a manufacturer and ends with the delivery of the finished product to the consumer. Management of this network plays a very important role in the firm's financial benefit. Supply chain management revolves around the flow of goods and comprises all the processes that transform raw materials into final products [4, 13, 15]. Supplier Selection is defined in Westburn Dictionary as, "The stage in the buying process where the intending buyer chooses

K. C. Mondal (✉) · B. D. Nandy · A. Baidya
Department of Information Technology, Jadavpur University, Kolkata, India
e-mail: kartickjgcec@gmail.com

B. D. Nandy
e-mail: biswadeep1997@gmail.com

A. Baidya
e-mail: arunimabaidya@gmail.com

the preferred supplier or suppliers from those qualified as suitable.” Supplying the correct goods specified by the firm is a key requirement for the management system. To ensure this, selection of the most coherent supplier, based on different criteria like time for delivery, cost of resource, and reliability, is very crucial [1, 15].

In this modern era of automation and rapid growth, organizations are seeking a reasonable, fast, and robust way of decision-making which would uplift main performance criteria such as quality, service, speed, and cost. This decision-making capability of such a system should also be as accurate as possible and comparable with human expert [3–5]. In this paper, we have tried to develop an expert system made in accordance with the past purchases of a firm. In the real world, every firm has huge data from previous suppliers and the products supplied. We have tried to find the relationship between the various databases in the dataset we have. This played a key role in helping us generate facts which were properly processed to form accurate Prolog clauses which have been used to build an appropriate knowledge base and thus provide the most efficient supplier based on different situations as queried or required by the user of this system.

In this paper, we have tried to develop a strictly fact-based expert system for simplifying the process of supplier selection, thus, speeding up the entire supply chain management as well as increasing its efficiency. According to our study, different types of expert systems have been developed for supplier selection such as rule-based, fuzzy rule-based, and fuzzy systems but an expert system only containing facts as clauses in a knowledge base generated from a firm’s previous data has not been proposed as per our study goes. In addition to this, we have shown how a single rule can be separated to form multiple facts, which are atomic in nature, but keeps the overall knowledge intact without causing any changes. This is also another novelty of our work presented in this paper.

Without the presence of a rule base or a rule generation engine, a significant amount of computation does not take place. For this, our proposed model consists of three basic components which are knowledge base, inference engine, and user interface. The jump comes in the way of accurate deductions in our model. Consider a situation with a single erroneous rule in the rule base. It would provide multiple erroneous results for multiple queries. But, facts are computed from historical and present data. One erroneously generated fact will only affect one result of a particular query out of a hundred possible queries that could be supplied to the expert system.

The data collected from an organization has been preprocessed at the beginning which includes steps like avoiding incomplete data, removal of duplicate data, and identifying proper primary and foreign keys. After this, facts have been developed based on this processed data which forms the knowledge base of our expert system. Prolog has been used for the knowledge base implementation while Python is used for making the user interface, which helps in communicating with the end user. A bridging package has been used called PySWIP which helps in the logical transfer of input and output values between the two platforms.

Moreover, removing the heavy computational portion (rule based) from the general CLIPS Expert System architecture makes our system way more lightweight in

terms of memory consumption and processing speed when it comes to handling multiple dense datasets such as ours, thereby making the system more scalable as well; we are not required to generate rules at all, even at the addition of new datasets.

The remaining of the paper is organized as follows: in Sect. 2, a brief overview of previous studies in which different types of expert systems are used to solve supplier selection problems are presented. Structure of our proposed expert system is explained in detail in Sect. 3. Section 4 showcases the dataset we worked with and it is analyzed in detail in accordance with a star schema for a better understanding of the simulation. In Sect. 5, the experiment, analysis, and results are discussed. Finally, concluding remarks are presented in Sect. 6.

2 Related Work

Article [7] put forward the idea of using expert systems which can be beneficial in determining suppliers for a single product. Their proposed model of an expert system took into consideration different attribute types like price, due date, discount, and so on. These attribute values increased the model's reliability as well as complexity as shown in [7]. Similarly, [8] presented a concept of the use of rule-based reasoning systems for evaluation and classification of suppliers. Here, authors showed how an expressive system of rule management can be used as an effective tool for supplier evaluation. Experiment and analysis were done on Rebit system and a demonstration has been given in article [8] as to how an individual evaluation criteria can be grouped into sets of independent rules and how one may use tools to enhance knowledge acquisition.

Authors in [3] claim that the success of a supply chain management depends on various decisive factors and can be achieved by managing several components efficiently. In article [3], authors continue their study on one such significant decisive factor to design an expert system using Prolog which takes rational decisions for vendor selection in the shoe industry.

Yunusoglu and Selim [18] developed a fuzzy rule-based expert system to support portfolio managers in their investment decisions. The proposed expert system has been validated by using the data of the Istanbul Stock Exchange (ISE) National 100 Index (XU100). It is also stated in the study that the performance of the proposed expert system is relatively higher in risk-averse investor and middle-term investment period cases. Again, Pitchipoo in his paper [11] showcases that supplier selection problem is a multi-criteria decision problem in which both qualitative and quantitative factors are involved. A fuzzy logic expert system was developed for supplier selection in the chemical industry and the results obtained in [11] portrayed the efficiency and effectiveness of an expert system in such decision-making problems.

Authors of article [12], have adopted a rather holistic and comprehensive view of some of the organizational profiles and attempted to point out the relationships between some organizational profiles and ERP System success. Their research attempted to link the organizational profiles to the ERP success level by using the

capability of artificial neural networks in articulating such relationships. Further, in [14], authors have defined Enterprise resource planning as one of the major sets of modules of enterprise systems being implemented in various organizations across the globe. They programmed the knowledge base as an if ...then logical structure. They further explained how an expert system usually contains two components, a knowledge base, and an inference engine program, enabling it to suggest conclusions.

The research conducted in [2] has contributed to designing a system that is the combination of an expert system and ANN. The customers can interact with the interface of the expert system to ask and get advice from the system. In addition to this, the knowledge received from the data analysis is used to identify a specific customer's behavior. The study conducted by Hokey Min in [10] gives us an insight into the application of Artificial Intelligence in the field of supply chain process. The paper [10] gives us a detailed view of the different applications conducted in this field to automate the supply chain management process. Authors in [13] identified the criteria for evaluation of two types of suppliers in the construction industry and their fuzzy membership functions, distinctly, through literature review, questionnaire survey, and statistical analysis of expert judgment. They utilized Mamdani's inference mechanism to develop a new methodology of the fuzzy expert system.

3 Development of Expert System

Expert systems (ES) or knowledge-based systems [6] solve problems and take decisions concerning a specific field by using expert's knowledge stored in a knowledge base. They closely and effectively mimic human decision-makers or experts. ES is interactive and is designed for solving complex problems by reasoning through bodies of knowledge [17]. It utilizes data, provides an easy user interface, and it also allows for the decision maker's own insights. For showing the practical application, the study of ES for supplier selection has been done. The expert system is built to develop or to reduce the manual calculation risk faced by a firm [5, 15, 16]. Figure 1 demonstrates the model of an expert system in general. It consists of the five components as explained here.

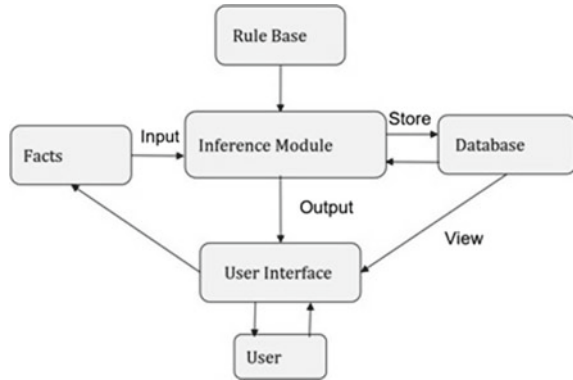
Facts: They are used for the representation of expert's knowledge [6, 9].

Rule: It is referred to as an IF-THEN structure that relates the given facts in the IF part, called the antecedent, with a particular action in the THEN part, called the consequent [6, 9].

Database: It is a collection of data where data is organized into rows, columns, and tables, and it is indexed to make it easier to find relevant information.

Inference engine: Given the knowledge, the inference engine tries to derive a suitable result. It precedes using two techniques, namely, forward chaining and backward chaining. Forward chaining starts with the given data and uses the inference engine to extract more data until a possible solution or goal is reached. Backward

Fig. 1 The outline of an expert system



chaining, on the other hand, is an inference method which starts from the goal and ends as per user required solution [6].

User Interface: It helps the end users to interact with the inference engine, ask relevant questions and thus, get the correct decision as output.

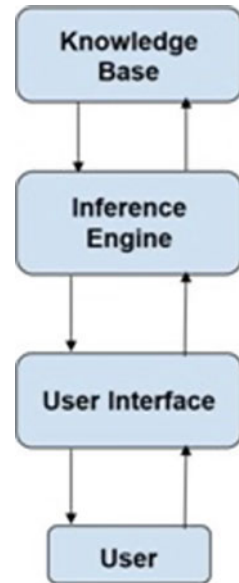
3.1 Modeling Our Proposed Expert System

- Given a set of queries, the user enters appropriate responses based on the program requirements. Queries appear to be in the form of regular questions such as “Name of the firm”, “Item requirement”, “location of the firm”, “Date of requirement”, and “Payment methods looking out for”.
- The program searches for the best possible outcome based on the input parameters, considering one or more than one simultaneously. Backward chaining is used to find all possible solutions for each parameter.
- The user has the opportunity to tune the results based on multiple parameters like quality, cost, reputation, speed of delivery, type of payment accepted, and location or any number of parameters.

3.2 Components of Our Proposed Expert System

Based on the above modeling, our proposed ES framework contains three components, namely, Knowledge base, Inference engine, and User interface (UI) where all the flows are bidirectional in nature. Pictorial representation of the proposed system is shown in Fig. 2.

Fig. 2 The model of our proposed expert system



In our implementation as represented in Fig. 2, the knowledge base consists strictly of facts as clauses. In fact, the presence of rules as clauses are not required if every single rule is broken down into multiple facts of their specific relation. For example, Rule 1: “All humans like food” can be patterned into facts as `human(alice)`, `human(bob)`, `likes(alice,food)`, `likes(bob,food)`. More specific rules can also be easily removed to imply the same using multiple facts which bring about the same scenario without causing any change in the overall knowledge. For example, Rule 2: “Few mammals eat meat, have stripes, and can swim” can be used in a similar manner if facts like these are present—`mammals(leopard)`, `mammals(cheetah)`, `mammals(cow)`, `eats(leopard, meat)`, `eats(cheetah, meat)`, `eats(cow, grass)`, `have_stripes(cheetah)`, `have_stripes(leopard)`, `can_swim(cheetah)`.

An inference engine helps to deduce, or infer new knowledge from a set of given logical deductions, facts or rules. The inference method used in our case is backward chaining where given a goal, it works upward toward facts to derive multiple solutions, with the goal here being the input given by the user as the firm’s name, material required, location, and other data. Thus, working up the inference engine finds the most appropriate supplier given in a particular scenario [9]. For this work, Prolog has been used for the knowledge base implementation and Python for the making of the user interface. A Python–Prolog bridging package has been used named PySWIP for the logical transfer of input and output values between the two platforms.

4 Dataset

4.1 Database Naming

Six databases have been used for our study from which the facts have been generated to form the knowledge base. Each record caters to some specific information for a particular supplier. Every database contains different information about different sections present in the supplier management process. Given below is an elaborate description of all six databases and what each of them contains.

Database 1: Supply order cash purchase: Contains data related to suppliers pertaining to only cash related purchases.

Database 2: Supply order contract based purchase: Contains data related to only suppliers who deal with contract-based payment schemes.

Database 3: Supply order payments: Contains information about the type of payment and document.

Database 4: Planning reports: Contains the supplier name and their item descriptions as well as the firms that previously bought products from them.

Database 5: Bills cleared and bills pending: Contains all the previously paid bills as well as bills yet to be cleared including every item price to tax value.

Database 6: Vendor registration details: Keeps a record of all the supplier information.

4.2 Dataset Preparation

All the datasets had to be preprocessed before the generation of the facts from them. The different databases went through the following preprocesses:

1. Analyzing the data thoroughly: This process helps in understanding the relationship of one database with another to recognize proper primary and foreign keys.
2. Preprocessing: This includes the following steps:
 - Empty fields are hashed to remove noisy or incomplete data.
 - Delimiters are removed for the proper creation of predicates.
 - Duplicate data is removed to avoid the generation of redundant facts.
 - Proper letter case orientation is applied.
 - Found primary and foreign keys are used for forming appropriate relations between the databases.

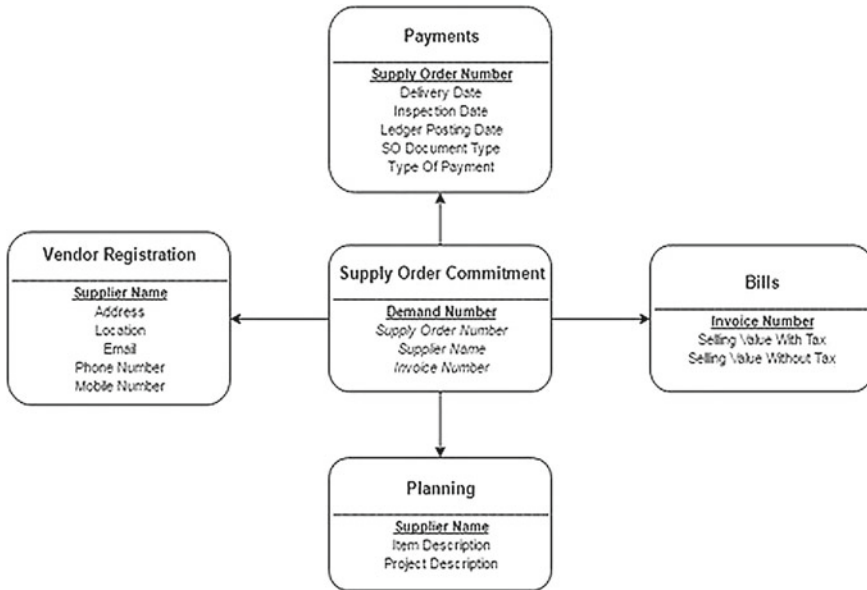


Fig. 3 The star schema of the databases used in the development process

Figure 3 showcases the star schema diagram of the databases. We have combined the first two databases, namely, Database 1 and Database 2 containing Demand number as the primary key and Supply order number, Supplier name, and Invoice number as foreign keys. Database 3 contains information about the payment corresponding to a single supply order number. The primary key is the supply order number and the elements are the delivery date, inspection date, and ledger posting date, SO Document type and payment type. Database 4 contains the planning reports with the primary key as the supplier name and elements like item description and project description. Database 5 accommodates information about the bills, containing invoice number as the primary key and selling value as elements. Finally, Database 6 contains details of the vendor registration. Supplier name is the primary key of Database 6 and it contains elements like address, location, email id, phone number, and mobile number.

Table 1 displays a vivid analysis of all the databases present in the dataset. The maximum number of rows is considered to give an estimation of the actual length of each database. Foreign keys are considered to find a relation between each database, understanding of this relationship plays an important role in the generation of facts. Primary keys of each database help to specify a unique identifier for each record.

Table 1 Details of used databases

Database No.	Max. rows	Primary keys	Foreign keys
Database 1	2170	Demand No.	Supply Order Number, Supplier Name, Invoice Number
Database 2	409	Demand No.	Supply Order Number, Supplier Name, Invoice Number
Database 3	10492	Supplier Order No.	None
Database 4	3308	Supplier Name	None
Database 5	1603	Invoice No.	None
Database 6	4126	Supplier Name	None

5 Experiment and Analysis

5.1 Experiment

5.1.1 Experimental Setup

We have used a single machine for carrying out our experiments, whose configuration is stated as follows:

- Processor: 6th Generation Intel Core i5-6 300HQ Quad Core (6M Cache, up to 3.2 GHz),
- Memory: 8GB 1 DIMM (1 × 8GB) DDR3L 1600M hz,
- Disk/Hard Drive: 1TB (5400rpm) Hybrid HDD with 8GB Flash,
- System Type: 64 bit Operating System, x64-based Processor, and
- Operating System: Windows 10 Home Single Language.

5.1.2 Implementation Setup

- Database processing and User interface(UI): Python,
- Knowledge base: Prolog, and
- Logical Interface: PySWIP.

5.2 Analysis

Working on six databases where the rows vary from 409 to 10,942, the total facts constituting the knowledge base is over 2,00,000. The user interface has been made such that it interacts with the user, asks intelligent questions, and provides the name of the supplier based on various scenarios in an efficient manner, much like a human expert but, at a higher speed and devoid of human errors.

Below, an outlook of the overall system working is demonstrated in the form of tables. Questions asked by the user interface of the expert system to the user are used to get appropriate goal knowledge which in turn is used to generate appropriate Prolog queries for running on the inference engine to give appropriate solutions. Listing 1 shows the user interface where intelligent questions are asked to the user. These questions are well formed such that there is limited and restricted searching done on the knowledge base.

Listing 3.1 The User Interface and the intelligent questions asked from the user of this system.

Program: Name of the firm?
User: IIR Seekers Directorate .

Program: Item required?
User: Scopecorder .

Program: Document or purchase type required (Cash or Contract)?
User: Cash Type.

Program: Within how many days do you want the item to be delivered?
User: 14 days.

Program: Any particular attribute you would like to select on? (speed/cost/location)
User: No.

These answers acquired from the user of the expert system are used to form appropriate Prolog queries through the Python interface to run on the Prolog knowledge base. This forming of Prolog queries from user answers are showcased in Table 2. For example, in “Name of the firm?” answered by the user to be “IIR Seekers Directorate”, the query in the background generated by the Python interface for running on the knowledge base is “suppliername (X, iirseekersdirectorate)” as shown in Result 1 of Table 2. Based on this, the result generated is put up in the next column of that table. Now, similarly moving on to a more complex scenario, based on two answers given by the user, another query is generated as shown in Result 2 present in Table 2 where the cash purchase type and item requirement is taken into consideration as given by the user in Listing 1 to generate the Prolog query “sodocumenttype (cash-purchase, X), itemsupplier(scopecorder, X)”. The appropriate result is put forward in the next column as well and explained. Similarly, Result 3 and Result 4 presented in Table 2 show us complex query generation based on the given firm name, item requirement, and various purchase types. Moving forward, such user answers can be replicated in the form of a useful Prolog query to generate a list of results.

Table 2 The conversion of user input to a working Prolog query and the result generated by the expert system for it

Result no.	English sentence	Query	Generated output	Explanation of the output
Result 1	Gives results for all the suppliers that can or have supplied items to the organization IIR Seekers Directorate	Suppliername(X, iirseekersdirectorate)	X = accesstechnologies; X = adrystechnologies ; X = agmatelindiapvtltd; X = cisystemltd; X = cmenvirossystemspvtltd; etc.	List of all the suppliers by name which can supply items to this firm. A long list of results since it is a single query and as a more specific question has not been asked by the user
Result 2	Gives the names of suppliers which provide scopecorder as well as accepts cash purchase	Sodocumentype (cashpurchase, X), itemsupplier (scopecorder, X)	X = accesstechnologies	Resulting supplier is the only supplier which sells scopecorder as well as accepts cash payments, according to the knowledge base prepared
Result 3	Gives the name of an organization that requires accelerometers as well as cash transactions	Suppliername (X,entestdirectorate), sodocumentype (cashpurchase, X), itemsupplier (accelerometers, X)	X = southernssystem	The resulting supplier name is the only supplier which provides accelerometers and accepts cash payments
Result 4	Gives the name of an organization that requires accelerometers as well as cash transactions and local purchase schemes	Suppliername (X, entestdirectorate), sodocumentype (cashpurchase, X), sodocumentype (localpurchase, X), itemsupplier (accelerometers, X)	X = southernssystem	The resulting supplier name is the only supplier which provides accelerometers and accepts cash payments as well as local purchase schemes

This list of results can be organized and given as output by the user interface of the expert system, based on proper attribute selection criterion given by the user as well, as shown in Table 3.

Table 3 The final result of the expert system

Appropriate supplier	Address	Location	Email ID	Contact details	Scenario
Supplier 1	<Address of Supplier 1>	<Location of Supplier 1>	<email id of Supplier 1>	<Phone no. of Supplier 1>	Speed of delivery
Supplier 2	<Address of Supplier 2>	<Location of Supplier 2>	<email id of Supplier 2>	<Phone no. of Supplier 2>	Quality of product
Supplier 3	<Address of Supplier 3>	<Location of Supplier 3>	<email id of Supplier 3>	<Phone no. of Supplier 3>	Reputation
Supplier 4	<Address of Supplier 4>	<Location of Supplier 4>	<email id of Supplier 4>	<Phone no. of Supplier 4>	Cost of item
Supplier 5	<Address of Supplier 5>	<Location of Supplier 5>	<email id of Supplier 5>	<Phone no. of Supplier 5>	Location
Supplier 6	<Address of Supplier 6>	<Location of Supplier 6>	<email id of Supplier 6>	<Phone no. of Supplier 6>	Inspection/ Warranty date

6 Conclusion

In this paper, a strictly fact-based multi-criteria expert system was developed for selecting the most efficient supplier which takes various scenarios into consideration. The aim of this paper is to find an appropriate supplier for a particular product or service using the previous data present at the firm's end. Since this study also looks into the various criteria for supplier selection like time of delivery, cost of good, and reliability, it will help in providing the firm a competitive edge, thus, in turn, increasing its profit at a significant scale. This expert system reduces human effort to a large extent by automating the entire process of supplier selection which further helps in reducing the time and human error. In addition to this, we also conclude that in a rule-based expert system, if a rule is wrong by any means then, it will, in turn, give an inefficient result for multiple scenarios that get covered by that rule in particular. Whereas in our implementation, where knowledge base consists only of facts, an erroneous fact may lead to only one such circumstance for an inefficient result, the possibility of which is very low.

The study conducted opens up a new way of developing expert systems, which can be used for further studies or implementations in other fields later on. The idea of replacing a single rule of any type with many relational facts without altering the overall knowledge present also opens up a new way of designing all relative systems or algorithms that work only with clauses. Further research may help in deducing knowledge which can be produced from rules or facts alternatively.

References

1. Boer LD, Labro E, Morlacchi P (2001) A review of methods supporting supplier selection. *Eur J Purch Supply Manag* 7(2):75–89
2. Devi PI, Rajagopalan SP (2011) The expert system designed to improve customer satisfaction. *Adv Comput: Int J* 2(6)
3. Ersoz S, Yuzukirmizi M, Turker A, Birgoren B (2009) Vendor selection in supply chain management by expert systems and a case study. *Int J Eng Res Dev* 1(2):61–66
4. Ghodspour SH, O'Brien C (1998) A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming. *Int J Prod Econ* 56:199–212
5. Jalao ERL, Martinez IAG (2009) The contract expert system: a proposal for long-term supplier evaluation, selection and performance monitoring, vol 1. CRC Press
6. Lucas P, Gaag LVD (1991) Principles of expert systems. Addison-Wesley, Wokingham
7. Luji R, İLari T, Heffer G (2009) Application of expert system for determination of the most beneficial suppliers in single production. *Tehničkivjesnik* 16(4):81–86
8. Macio A, Jedrusik S, Rebiasz B (2013) Rule-based approach for supplier evaluation. *Federated Conference on Computer Science and Information Systems*, pp 1207–1214
9. Merritt D (2012) Building expert systems in prolog. Springer Science and Business Media
10. Min H (2011) Artificial intelligence in supply chain management: theory and applications. *Int J Logist Res Appl*
11. Pitchipoo P, Venkumar P, Rajakarunakaran S (2012) Development of fuzzy expert system for supplier evaluation and selection. In: *IEEE-International Conference On Advances In Engineering, Science And Management*, pp 1–6
12. Rouhani S, Ravasan Z (2013) Erp success prediction: an artificial neural network approach. *Scientia Iranica*
13. Shahvand E, Sebt MH, Banki MT (2016) Developing fuzzy expert system for supplier and subcontractor evaluation in construction industry. *ScientiaIranica Trans A, Civil Eng* 23(3):842
14. Sharma A, Sharma MC (2013) Enterprise resource planning and expert systems. *Int J Res Appl Sci Engg Tech* 1(4):43–46
15. Vokurka RJ, Choobineh J, Vadi L (1996) A prototype expert system for the evaluation and selection of potential suppliers. *Int J Oper Prod Manag* 16(12):106–127
16. Yigin IH, Takin H, Cedimoglu IH, Topal B (2007) Supplier selection: an expert system approach. *Prod Plann Control* 18(1):16–24
17. Yucel A, Guneri AF (2011) A weighted additive fuzzy programming approach for multi-criteria supplier selection. *Expert Syst Appl* 38(5):6281–6286
18. Yunusoglu MG, Selim H (2013) A fuzzy rule based expert system for stock evaluation and portfolio construction: an application to istanbul stock exchange. *Int J Expert Syst Appl* 40(3):908–920

Handling Seasonal Pattern and Prediction Using Fuzzy Time Series Model



Mahua Bose and Kalyani Mali

Abstract Seasonal variation is one of the important components of the time series. There are many techniques available in the literature to deal with the problem of seasonality. A few hybrid fuzzy time series models investigated the problem of forecasting in the presence of seasonal variation. But these techniques follow complex computational procedures. The aim of this present study is to develop a new fuzzy time series forecasting model that can process seasonal patterns present in the data directly without any seasonal adjustment by applying certain mathematical techniques. The proposed Neuro-uzzy model is capable of extracting the seasonal pattern from the training set and forecasting the future pattern. This model makes use of Self-organizing map (SOM) for clustering similar patterns. Performance of the model is evaluated using Rainfall data and Milk Production data.

Keywords Cluster · Fuzzy · Pattern · Relationship · Seasonal

1 Introduction

Seasonal variation is periodic in nature. It occurs at regular intervals of time (within a week, month, or quarterly, etc.) [1]. For example, India receives heavy rainfall during the monsoon season. At the beginning of the year (in January), rainfall is low. Then it starts rising. During rainy season, it reaches the maximum and then it decreases gradually. The sale of certain products is very high during festivals (in the month of October–November) in India. This is another example of seasonal data.

In fuzzy time series models [2–4], observed values are represented by a sequence of fuzzy sets [5]. These models can process both fuzzy data and crisp data which a limitation of traditional time series. Its another advantage is that it does not require

M. Bose (✉) · K. Mali

Department of Computer Science & Engineering, University of Kalyani, Nadia, West Bengal, India

e-mail: e_cithi@yahoo.com

K. Mali

e-mail: kalyanimali1992@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,

Studies in Computational Intelligence 870,

https://doi.org/10.1007/978-981-15-1041-0_4

large training samples as in statistical models. In these models, a relationship exists between the present state and one or more previous states. If there is only a single previous state in which the present state depends, then it is known as the first-order relationship. In case of multiple previous states, the relationship is considered as high order [2–4, 6, 7].

Generally, two different approaches [8] can be adopted for forecasting seasonal time series: (1) remove the seasonal factor from the time series by a deseasonalization technique and then apply any forecasting method and (2) generate forecast for all the seasons directly.

The present study employs the second approach. In this paper, our focus is on the issues related to handling and forecasting seasonal patterns present in the data without applying any seasonal adjustment techniques. The objective of this study is to develop a fuzzy time series forecasting model for this purpose. The proposed Neuro-fuzzy technique is capable of generating the predicted values for all of the seasons directly at a time. There is no need for seasonal adjustment.

This paper is divided into the following sections: Sect. 2 summarizes the previous studies in this direction. Description of the proposed forecasting model is given in Sect. 3. In Sect. 4, the performance of the proposed model is analyzed. The conclusion is presented in Sect. 5.

2 Related Works

Fuzzy time series forecasting models (FTS) have been applied in various areas such as Finance [9–18], Climatology [19–23], Enrollments [3, 4, 6, 13, 23], Electricity [24, 25], Environment [26], and Unemployment [14, 18]. Accuracy of the forecast of these models is dependent on three issues: (1) partitioning of data, (2) formulation of Fuzzy logical relationship (FLR), and (3) defuzzification.

There is no forecasting approach that can forecast all types of time series data. Almost all of the FTS models can work with the nonseasonal data only. But in presence of seasonal patterns (Fig. 1 a, b), these models cannot be applied. They are suitable for handling the stationary or trend time series. Otherwise, they produce large forecasting errors. To get accurate forecasts for different seasons, these nonseasonal models need to be applied for each season separately.

A few seasonal forecasting models using fuzzy time series are available in the literature also. The seasonal model by Song [27] is an extension of the basic model by Song and Chissom [2, 3]. Fuzzy trend and seasonality are also analyzed in the context of fuzzy regression [28]. “FSARIMA” model [29] is an improved version of the Seasonal Autoregressive Integrated Moving Average (SARIMA) model. This model integrates the SARIMA model with Tanaka’s fuzzy regression model. It is applied to the prediction of production values of Taiwan’s machinery industry and the sales volume of soft drinks. An FTS model using max-min composition has been presented to handle seasonal patterns in the presence of trend [8]. This study calculates the seasonal indices using ratio-to-moving-average method [30] and then removes

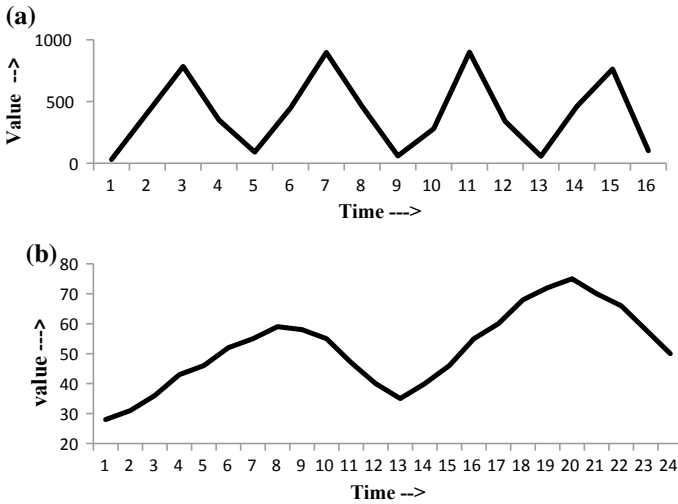


Fig. 1 **a** Seasonal pattern (Season = 4) without trend. **b** Seasonal pattern (Season = 12) with trend

seasonal variation from the time series by deseasonalization. To deal with nonlinear forecasting problems (like electricity load forecasting), a “SARFIMA” model based on seasonal long memory time series is developed [25]. A bivariate SARIMA model [31] integrated with fuzzy time series is successfully implemented. Another bivariate model called Seasonal fuzzy integrated logical forecasting (SFILF) model is proposed [32] also. Recently, a novel fuzzy forecasting model [33] utilizing the concept of decomposition model [34] is presented. In this model, each of the components of time series, i.e., trend, seasonality, and irregular fluctuations are modeled separately using the fuzzy technique and finally the results are combined.

Previous works on seasonal forecasting models integrating with fuzzy techniques involve complex mathematical calculation due to parameter estimations or seasonal factor elimination. The proposed Neuro-fuzzy technique is capable of handling seasonal patterns present in the data and generating predicted values for all of the seasons directly at a time. No seasonal adjustment is required.

3 Proposed Fuzzy Time Series Model

Step 1. Define Data Domain D in the following way:

$$D = [(D_val_{min} - Z_1) - (D_val_{max} - Z_2)]$$

Table 1 Rainfall data India (in cm.) from 1871–2014

Year	Jan.	Feb.	March	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
1871	19.6	10.7	14.4	33.9	63.6	208	277.8	179.4	183.5	36.8	32.3	6.7
1872	7.6	7.5	7.3	24	43.8	189.2	291.3	245.1	187.9	78.5	27.6	19.1
1873	3.6	13.5	15	24.3	42.8	113	264.4	214.2	165.6	60.7	11.5	8.9
1874	8.6	15.8	10.6	16.9	68.3	227.8	306.9	233.4	206.2	93.2	18.7	4
–	–	–	–	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	–	–	–	–
2012	17.4	3.1	3.3	34.7	23.5	112.3	243.8	241	183.6	57.5	38.1	5.5
2013	4.7	30.8	8.3	26.7	47.8	227	307.7	235.1	153.1	133.9	13.7	4.2
2014	13.3	23.3	21.1	13.4	64.8	90.8	256.4	219	170.2	61.8	15	9.9

where Lower Bound(LB) = $(D_val_{min} - Z_1)$ and Upper Bound(UB) = $(D_val_{max} - Z_2)$.

Z_1 and Z_2 are two positive values chosen arbitrarily.

Table 1 shows the monthly rainfall data of India. The graphical display is given in Fig. 2. The dataset has seasonal pattern but there is no trend. It is clearly seen that the amount of rainfall is high in June–September and it is very low during winter.

Step 2. Fuzzification of data values: let us consider n intervals (I_1, I_2, \dots, I_n) in which dataset is to be partitioned. Fuzzy sets Z_1, Z_2, \dots, Z_n are defined as follows:

$$Z_1 = 1/I_1 + 0.5/I_2 + \dots + 0/I_{n-1} + 0/I_n$$

$$Z_2 = 0.5/I_1 + 1/I_2 + \dots + 0/I_{n-1} + 0/I_n$$

.....

$$Z_{n-1} = 0/I_1 + 0/I_2 + \dots + 1/I_{n-1} + 0.5/I_n$$

$$Z_n = 0/I_1 + 0/I_2 + \dots + 0.5/I_{n-1} + 1/I_n$$

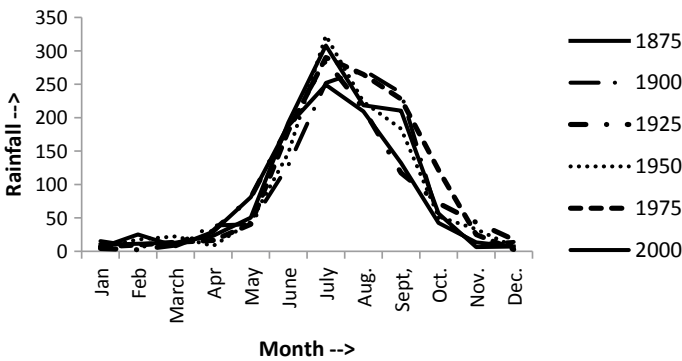


Fig. 2 Rainfall pattern in India

Table 2 Fuzzification of rainfall data

Year	Season											
	1	2	3	4	5	6	7	8	9	10	11	12
1871	Z ₁	Z ₁	Z ₁	Z ₂	-	-	-	-	-	-	Z ₂	Z ₁
1872	Z ₁	Z ₁	Z ₁	Z ₁	-	-	-	-	-	-	Z ₂	Z ₁
1873	Z ₁	Z ₁	Z ₁	Z ₁	-	-	-	-	-	-	Z ₁	Z ₁
-	-	-	-	-	-	-	-	-	-	-	-	-
2013	Z ₁	Z ₂	Z ₁	Z ₂	-	-	-	-	-	-	Z ₁	Z ₁
2014	Z ₁	Z ₁	Z ₁	Z ₁	-	-	-	-	-	-	Z ₁	Z ₁

In this example (Table 2), equal-sized intervals are created (interval length = 25). Data values are fuzzified into an interval with the highest membership value.

Step 3. Defining Fuzzy Logical relationship

According to the basic model of FTS [2, 3], an FLR of first order between the present state F_val (t + 1) and past state F_val (t) is represented as

$$F_val(t) \rightarrow F_val(t + 1)$$

Following the concept of seasonal version of fuzzy time series model [27], it can be written as, F_val (t) -> F_val (t + s)

where s is the number of seasons.

It means that each of the observations represents a time period and the relationship (between two fuzzified values) is defined for each quarter separately. For example, from the entries of the years 1871 and 1872 (Table 2), the following relationships can be obtained.

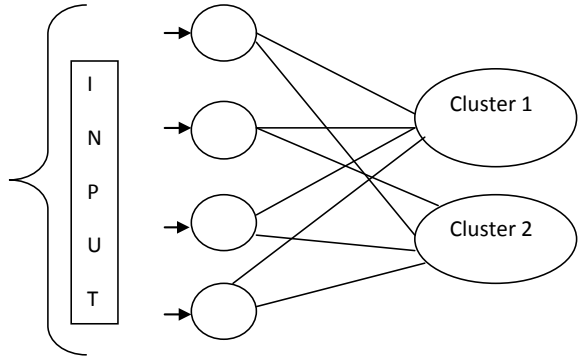
- Z₁ -> Z₁ (Season 1)
- Z₁ -> Z₁ (Season 3)
- Z₁ -> Z₁ (Season 3)
- Z₂ -> Z₁ (Season 4)
-
- Z₂ -> Z₂ (Season 11)
- Z₁ -> Z₁ (Season 12)

Novelty of this research work is that, it groups fuzzified values of all the seasons into a pattern where each pattern represents a time period. First-order relationship (FLR) between two patterns (for the year 1871 and 1872) is shown as follows:

$$Z_1, Z_1, Z_1, Z_2, \dots, Z_2, Z_1(t - 1) \rightarrow Z_1, Z_1, Z_1, Z_1, \dots, Z_2, Z_1(t)$$

Step 4. Clustering patterns

Fig. 3 Self-organizing feature map with two clusters



Self-organizing feature map (SOFM) or Self-organizing map (SOM) uses unsupervised learning for clustering similar patterns. It is also called or Kohonen network [35]. Let the number of training patterns be n . These patterns will be grouped into c clusters (Fig. 3).

In this paper, neighborhood = 0. The number of input nodes = s (season). In this case, index values of the fuzzy sets in a pattern will be input to the system. From the above example, the first input will be 1, 1 1, 2, ..., 2, 1.

Step 1. Initialize the weight vector randomly and the learning rate α . Set the maximum iteration number.

Step 2. Set iteration = 0. Set neighborhood = 0.

Step 3. Repeat Step 4–6 for every input pattern p .

Step 4. Calculate the Distance from pattern p_i to each output node c_j .

$$\text{Dist}(j) = \sum_{i=1}^n \sum_{j=1}^c (p_i - w_{ij})^2$$

Step 5. Node corresponding to minimum $\text{Dist}(j)$ is the winning unit.

Step 6. Update weight of the winning unit as follows:

$$W[ij] (\text{new}) = W[ij] (\text{old}) + \alpha(p[i] - w[ij](\text{old}))$$

Step 7. Update the learning rate α .

Step 8. Increment the iteration number.

Step 9. If maximum iteration occurs, stop else goto Step 3.

Step 4. Forecasting future pattern

Step 4.1. Assign each pattern a group/cluster id (Table 3). From the sequence of group-id, obtain the trend of occurrence of patterns and count the number of occurrences in $c \times c$ pattern_transition matrix (Fig. 4). So, for five clusters, $c = 5$.

Table 5 Middle values

13	13	13	13	38	188	288	213	163	63	13	13
13	13	13	13	38	163	263	263	138	38	38	13
13	13	13	13	38	113	263	213	188	63	13	13
13	13	13	13	38	163	213	188	113	63	13	13
13	13	13	13	63	163	263	263	188	113	13	13

Table 6 Season with predicted rainfall

1	2	3	4	5	6	7	8	9	10	11	12
13	13	13	13	40.9	161.1	269	226.5	163	63.96	17.8	13

$$\text{Defuzz}[j] = \frac{\sum_{i=1}^c ((mid[c][j]) * (\text{pattern_trend}[k][c]))}{\sum_{i=1}^c (\text{pattern_trend}[k][c])} \quad (2)$$

where k is the group-id of the previous pattern. Each season is represented by j ($j = 1, 2, \dots, s$).

Seasonal pattern for the year 2013 is in cluster1. From Fig. 4, numbers of occurrences of the next possible clusters are obtained. Using Eq. 2, predicted rainfall values for the next 12 seasons (in 2014) are shown in Table 6.

4 Performance Evaluation

4.1 Data

We have collected the following datasets: (1) all India monthly rainfall data (1871–2014), from the website of the Indian Institute of Tropical Meteorology (<http://www.tropmet.res.in>). Rainfall data for the years 1871–1960 are used as training samples. Remaining data are used for evaluation performance of the proposed model; (2) Milk Production data (monthly milk production: pounds per cow. January 62–December 75) [36].

4.2 Results and Discussion

Initially, the data values are partitioned into equal-length intervals and then fuzzy patterns are generated. Then using SOM, training patterns are grouped into clusters. For the experiment, 4–6 clusters are generated.

Performance of the proposed model is compared with a recently presented FTS model [14]. This model [14] is developed for nonseasonal data only. So, the model is applied separately for each quarterly dataset. The proposed model generates a forecast for all seasons in a single run. It is seen that average RMSE value is lower than that of nonseasonal model [14]. For the estimation of forecast error, Root Mean Square Error (RMSE) is chosen. Results are displayed in the tables [9, 10].

$$RMSE = \sqrt{\sum_{i=1}^n ((Actual - predicted) * (Actual - predicted))/n} \tag{3}$$

where n represents the number of test samples.

4.2.1 Rainfall Prediction

We have applied this model to rainfall data of India (1871–2014). The amount of rainfall India receives is mainly dependent on South-West monsoon. About 70% of the population is dependent on agriculture. Rainfall prediction is a matter of great importance for economic development of the country and also for disaster management activities.

RMSE values for s = 12 (with interval length 25) are shown in Table 7. It is seen that average RMSE values using 4, 5, and 6 clusters are 21.97, 21.99, and 21.90, respectively.

In India, we are concerned with monsoon rainfall only. We have also organized the data into four quarters and predicted quarterly total rainfall also (Table 8). The first quarter represents the total rainfall of the months (January–March). Similarly, total rainfalls for other three quarters are calculated.

In Table 8, the Minimum and Maximum values are 12.7 and 845.7, respectively.

Here, LB = 0 and UB = 850. We have partitioned the dataset into 17 equal length segments. The length of each interval is 50.

Data values are fuzzified into an interval with the highest membership value. For example, rainfall value (Table 8) in the first quarter of 1971 is within the range of the first interval (0–50). So, it is fuzzified into interval 1 and its fuzzified value is Z₁.

It is evident from Table 9 that the average forecast accuracy of the proposed seasonal model is better than that of the nonseasonal model [14]. Nonseasonal model

Table 7 RMSE values (s = 12) for rainfall data

Clusters	Season											
	1	2	3	4	5	6	7	8	9	10	11	12
4	6.85	8.47	9.07	16.43	18.8	37.7	36.5	33.77	32.29	30.94	22.49	10.34
5	6.85	8.47	9.07	16.43	19.61	35.76	36.9	34.55	33.26	30.8	21.85	10.34
6	6.85	8.47	9.07	16.43	20.74	38.05	37	34.45	32.29	28.4	20.79	10.34

Table 8 Yearly Rainfall data of India (cm.) in four quarters

Year	Q1 (Jan.–Mar.)	Q2 (Apr.–June)	Q3 (July–Sept.)	Q4 (Oct.–Dec.)
1871	44.7	305.5	640.7	75.8
1872	22.4	257	724.3	125.2
1873	32.1	180.1	644.2	81.1
1874	35	313	746.5	115.9
–	–	–	–	–
–	–	–	–	–
2012	23.8	170.5	668.4	101.1
2013	43.8	301.5	695.9	151.8
2014	57.7	169	645.6	86.7

Table 9 RMSE values (using $s = 4$)

Season	1	2	3	4	Average
Nonseasonal model [14]	19.5	64.65	85.55	38.36	52.015
Proposed Seasonal model (using 6 clusters)	18.45	54.454	74.727	36.10	45.93
Proposed Seasonal model (using 5 clusters)	18.45	50.77	78.117	49.86	49.3
Proposed Seasonal model (using 4 clusters)	18.45	48.97	72.44	41.94	45.45

treats data for each season, a separate series and calculates predicted values. But the proposed seasonal model generates the forecast for each season at a time.

4.2.2 Prediction of Milk Production

Milk Production data (Fig. 5a, b), has both trend and seasonality. So, first of all trend is removed by first-order differencing (by computing the differences between each two consecutive observations at time t and $t-1$). Then a positive number greater than the highest negative value is added to each to the difference values. After that, the proposed algorithm is applied. In the end, a positive value is subtracted from each predicted value. This is the adjusted predicted value.

$$\text{Forecast}(t + 1) = \text{Actual value}(t) + \text{Adjusted prediction} \quad (4)$$

Here, the first 12 years' data is used for training and the last 2 years' data is identified as a test set. RMSE values (using four clusters) are displayed in Table 10.

Fig. 5 **a** Milk Production data with trend and seasonal component. **b** Milk Production data after removal of trend component

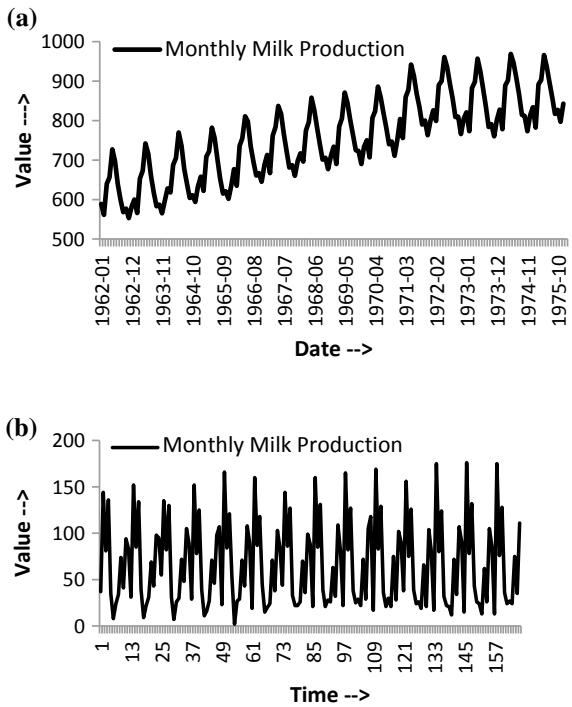


Table 10 RMSE values: Seasonal model ($s = 12$)

Interval length	Season											
	1	2	3	4	5	6	7	8	9	10	11	12
30	2.92	4.1	25.5	7.07	2	10.12	5.1	4.7	10.12	6.67	4.53	3.61
25	7.62	4.5	13.95	2.24	12.02	6.519	10	11	6.519	9.19	14.6	4.24

5 Conclusion and Future Work

This paper presents a novel technique for forecasting fuzzy time series in the presence of seasonal variation. The objective of this study is to forecast the entire pattern without adjustment of the seasonal factor. Average forecast accuracy of the proposed seasonal model is better than that of the recently developed fuzzy time series model (nonseasonal model).

The proposed model can be applied to any kind of seasonal data series. Currently, it deals with seasonal patterns only. It can handle seasonal patterns with trends also. The problem of complex seasonal patterns or multiple seasonal patterns is to be investigated in future.

References

1. Hylleberg S (1992) Modeling seasonality. Oxford University Press, Oxford
2. Song Q, Chissom B (1993) Fuzzy time series and its models. *Fuzzy Sets Syst* 54:269–277
3. Song Q, Chissom BS (1993) Forecasting enrollments with fuzzy time series—Part I. *Fuzzy Sets Syst* 54:1–9
4. Song Q, Chissom (1994) Forecasting enrollments with fuzzy time series—Part II. *Fuzzy Sets Syst* 64:1–8
5. Zadeh LA (1965) Fuzzy set. *Inf Control* 8:338–353
6. Chen SM (1996) Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst* 81:311–319
7. Lee LW, Wang LH, Chen SM, Leu YH (2006) Handling forecasting problems based on two-factors high-order fuzzy time series. *IEEE Trans Fuzzy Syst* 14(3):468–477
8. Liu H-T, Wei M-L (2010) An improved fuzzy forecasting method for seasonal time series. *Expert Syst Appl* 37(9):6310–6318
9. Chen S-M, Phuong BDH (2017) Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors. *Knowl-Based Syst* 118:204–216
10. Chen S-M, Jian W-S (2017) Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups, similarity measures and PSO techniques. *Inf Sci* 391–392:65–79
11. Cheng S-H, Chen S-M, Jian W-S (2016) Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures. *Inf Sci* 327:272–287
12. Cai Q, Zhang D, Zheng W, Leung SCH (2015) A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowl-Based Syst* 74:61–68
13. Chen MY (2014) A high-order fuzzy time series forecasting model for internet stock trading. *Future Gener Comput System* 37:461–467
14. Bose M, Mali K (2018) An improved technique for modeling fuzzy time series. In: the Proceedings of 2nd International Conference on Computational Intelligence, Communications, and Business Analytics, Kalyani Govt. Engg. College, West Bengal (Communications in Computer and Information Science, Vol. 1030, Springer, 2019)
15. Bisht K, Kumar S (2016) Fuzzy time series forecasting method based on hesitant fuzzy sets. *Expert Syst Appl* 64:557–568
16. Bose M, Mali K (2018) A novel data partitioning and rule selection technique for modeling high-order fuzzy time series. *Appl Soft Comput* 63:87–96
17. Rubio A, Bermúdez J, Vercher E (2017) Improving stock index forecasts by using a new weighted fuzzy-trend time series method. *Expert Syst Appl* 76:12–20
18. Deng W, Wang G, Zhang X, Xu J, Li G (2016) A multi-granularity combined prediction model based on fuzzy trend forecasting and particle swarm techniques. *Neurocomputing* 173:1671–1682
19. Wang W, Pedrycz W, Liu X (2015) Time series long-term forecasting model based on information granules and fuzzy clustering. *Eng Appl Artif Intell* 41:17–24
20. Singh P (2016) Rainfall and financial forecasting using fuzzy time series and neural networks based model. *Int J Mach Learn Cyber* <https://doi.org/10.1007/s13042-016-0548-5>
21. Hsu L-Y, Horng S-J, Kao T-W, Chen Y-H, Run R-S, Chen R-J, Lai J-L, Kuo I-H (2010) Temperature prediction and TAIFEX forecasting based on fuzzy relationships and MTPSO techniques. *Expert Syst Appl* 37:2756–2770
22. Singh P, Borah B (2013) High-order fuzzy-neuro expert system for daily temperature forecasting. *Knowl-Based Syst* 46:12–21
23. Bose M, Mali, K (2017) Fuzzy time series forecasting model using particle swarm optimization and neural network. In: the Proceedings of 7th International Conference. *Soft Computing for Problem Solving*, IIT, Bhubaneswar, Odisha (Advances in Intelligent Systems and Computing, Vol. 816, Springer, 2019)
24. Efendi R, Ismail Z, Deris MM (2015) A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand. *Appl Soft Comput J* 28:422–430

25. Sadaei HJ, Guimarães FG, da Silva CJ, Lee MH, Eslami T (2017) Short-term load forecasting method based on fuzzy time series, seasonality and long memory process. *Int J Approx Reason* 83:196–217
26. Domanska D, Wojtylak M (2012) Application of fuzzy time series models for forecasting pollution concentrations. *Expert Syst Appl* 39(9):7673–7679
27. Song Q (1999) Seasonal forecasting in fuzzy time series. *Fuzzy Sets Syst* 107:235–236
28. Chang P-T (1997) Fuzzy seasonality forecasting. *Fuzzy Sets Syst* 90:1–10
29. Tseng F-M, Tzeng G-H (2002) A fuzzy seasonal ARIMA model for forecasting. *Fuzzy Sets Syst* 126:367–376
30. Mansfield E (1994) *Statistics for business and economics: Methods and application*. NY: W.W. Norton and Company
31. Egrioglu E, Aladag CH, Yolcu U, Basaran MA, Uslu VR (2009) A new hybrid approach based on SARIMA and partial high order bivariate fuzzy time series forecasting model. *Expert Syst Appl* 36:7424–7434
32. Bulut E (2014) Modeling seasonality using the fuzzy integrated logical forecasting (FILF) approach. *Expert Syst Appl* 41(4 PART 2):1806–1812
33. Nguyen L, Novák V (2019) Forecasting seasonal time series based on fuzzy techniques. *Fuzzy Sets Syst* 361:114–129
34. Box GEP, Jenkins GM (1976) *Time series analysis: forecasting and control*. Oakland, CA
35. Kohonen Teuvo (1982) Self-organized formation of topologically correct feature maps. *Biol Cybern* 43(1):59–69. <https://doi.org/10.1007/bf00337288>
36. Yang X, Yu F, Pedrycz W (2017) Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system. *Int J Approx Reason* 81:1–27

Automatic Classification of Fruits and Vegetables: A Texture-Based Approach



Susovan Jana, Ranjan Parekh and Bijan Sarkar

Abstract Fruits and Vegetables are very important food product for the daily life of the humans. Classification of fruits and vegetable is needed for every aspect of the agricultural industry. It is quite challenging to automatically classify fruits and vegetables from digital images. The task of automatic classification becomes more difficult when the image is captured from a different viewing angle. This paper proposes a complete texture-based approach for addressing the effect of viewing angle change to classify fruits and vegetables automatically. At first, a grayscale image is generated from the input color image. The grayscale version of the input image is used to extract multiple threshold values using the multilevel Otsu thresholding technique. Those threshold values are used to generate a set of binary images. The binary images pass through a border extraction process to generate the border image of every binary image. Finally, the border image is processed to calculate the fractal dimension. In parallel flow, the same grayscale image is processed to compute gray-level co-occurrence matrix based features. The fractal dimension and gray-level co-occurrence matrix based features are combined to make a feature vector for classifying the fruit and vegetable classes. Images are collected by covering the entire range of 0° – 360° angle for each class in our dataset. In total, 1656 images of 23 classes of fruits and vegetables are used for experimentation. The maximum accuracy of the system is 98.33% with Naive Bayes classifier.

Keywords Multilevel thresholding · Texture · Fractal · GLCM · Classification

S. Jana (✉) · B. Sarkar

Department of Production Engineering, Jadavpur University, Kolkata, India
e-mail: jana.susovan2@gmail.com

B. Sarkar

e-mail: bijan.sarkar@jadavpuruniversity.in

R. Parekh

School of Education Technology, Jadavpur University, Kolkata, India
e-mail: rparekh.edutech@jadavpuruniversity.in

© Springer Nature Singapore Pte Ltd. 2020
J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,
Studies in Computational Intelligence 870,
https://doi.org/10.1007/978-981-15-1041-0_5

1 Introduction

Fruits and vegetables are very essential in our daily diet. It contains most of the important vitamins, minerals, and antioxidants. A large number of fruit and vegetable species exist [24] in the world. All the species are not edible. We are focusing only on edible fruits and vegetables here. Although fruits and vegetables have different tastes, many of them have similar appearances. The edible fruits and vegetables are harvested, sorted, and packed for delivery to the customers. It needs a large number of expert resources and a long time to process the fruits from the agricultural field to the supermarket. Automation in the agricultural field and supermarket is a must to reduce the time as well as the dependency on the manual resource. Classification among different fruit and vegetable types is one of the major tasks to migrate to automation. Automatic classification of fruit and vegetable types is a very challenging work using the visual features [4] from an image. The challenges are addressed using image processing techniques in many existing works. The color, shape, texture, and size [19] features were extracted from the image and utilized for the classification of fruits and vegetables. Fruit and vegetable identification in the mobile environment [15, 32] is also helpful for a visually impaired person as well as one who is not aware of the outlook of a fruit and vegetable species. In a conveyor belt, fruits and vegetables may appear in a random orientation. It may face the camera from any side of the surface as the camera is fixed in a position. This problem increases the challenges of the classification task. This paper attempts to propose a solution of fruit and vegetable classification addressing viewpoint changing problem. This work is a texture-based classification of fruits and vegetables. The selection and processing of features have been done in such a way that the classification result and performance will not be affected by the change of viewing position for particular fruits and vegetables. The proposed work has been designed to enable the classification of fruits and vegetables from all viewing positions. The dataset also contains the images from 72 viewing positions for a particular class to validate the proposed approach.

Section 2 describes some of the previous works on fruit and vegetable classification using image processing. Section 3 contains the materials and the proposed technique in details. Section 4 brings up the results of the experimentations and the discussion on the work. Section 5 draws a conclusion of the work with the future direction of work.

2 Previous Works

Classification is a very challenging work based on image analysis and machine learning. A number of previous works attempt a classification of fruits and vegetables by utilizing differences in their visual appearances, namely size, color, shape, and texture. A fruits and vegetables classification system was proposed by Cornejo et al. [6]. Features extracted from hue and saturation histograms have been utilized. They

experimented with 15 different fruit and vegetable categories and achieved the best accuracy with the SVM classifier. Rachmawati et al. [26] developed a color palette from RGB fruit images using the k-means clustering technique. The color descriptor of the fruit is selected from the clusters. Further, it was used for the classification of fruit class. Quantization of the color histogram, which was extracted from the RGB image, was used to differentiate between 32 classes [25] of fruits. The chi-square method was used to select the most discriminating feature. Improved Sum and Difference Histogram [8] features, which were extracted from a color image, were applied for the classification of fruits. The SVM classifier showed a good classification accuracy for 15 classes with these features. Mango fruit recognition was proposed using shape analysis and backpropagation neural network [20].

One type of feature is not sufficient when the number of classes is large. It requires lots of training data. To address this problem, a multiple feature and classifier fusion [27] based approach was introduced by Rocha et al.; global color histogram, Unser's descriptor, color coherence vector, border/interior pixel classification, and appearance descriptor were combined for classification. Naskar et al. proposed a multiple-feature-based classification using a neural network [21]. They extracted texture features after passing the image through log Gabor filter then mean of hue used as a color feature, area, and perimeter used as a shape feature. Another mixed approach [30] was given for fruit classification based on shape and color. The feature vector is formed with perimeter, area, roundness, and color means of each of the RGB channels and obtained 90% accuracy with the k-NN classifier. Ninawe et al. [22] added entropy features with Seng et al.'s approach and demonstrated 5% improvement of accuracy. A technique for the classification of three fruit classes was proposed by Zawbaa et al. [35] using two sets of features. The first set contains statistical features like mean, variance and skewness, and kurtosis extracted from color channels were combined with shape-based features like centroid, Euler number, and eccentricity. SIFT was the only feature for the second set. A number of features representing color, texture, and shape were first extracted and then PCA was used to reduce the feature vector dimension by Zhang et al. [36]. Fruit classification shows the best accuracy using multiclass SVM. Another mixed approach was used to recognize vegetables in the supermarket and grocery store [2]. Color (HSV), texture (LBP), shape (circumference, area, and roundness) features, and backpropagation neural network were utilized for the classification of four classes of fruits [34]. A combined approach [1] of color (standard deviation, mean, kurtosis, and skewness are extracted from Hue and Saturation channels) and texture feature (energy, contrast, local homogeneity, cluster prominence, and cluster shade are extracted from gray-level co-occurrence matrix) were used to classify 15 classes of fruit. A multi-feature fusion was used for classification of five fruit classes [16] and achieved a good classification result with histogram oriented gradient (HOG), local binary pattern (LBP), global color histogram, and Gabour local binary pattern of multiple color channel and LIBSVM. A visual feature based vegetable classification system was proposed for blind people [5]. The system captures an image and informs the person about the vegetable.

The classification of the subtypes becomes more difficult for different fruit classes. The features which were used for the classification of different fruit classes may not be effective for the classification of subtypes of a fruit class. A shape-based approach [29] was demonstrated to classify three subtypes of mango using the Naive Bayes classifier. Another apple subtype classification method was proposed using color and shape-based features [28]. Naive Bayes classifier recognizes most accurately among the three types of apple in Ronald's work. An improved intra-class [14] fruit classification technique was proposed using statistical texture features, i.e., contrast, correlation, energy, homogeneity, and standard deviation of the histogram for each of the three color channels of the RGB image. It was observed that those features and the neural network classifier are appropriate for subtype recognition of any fruit class. A date fruit classification was introduced using a combination of color, texture, shape, and size-related features [12]. They achieved the best accuracy of classification among seven classes using the neural network classifier.

The classification of fruits and vegetables has already been explored by many researchers. They have extracted the different types of features and proposed suitable classifiers with those features. But, nobody considered the effect of viewing the position change in the classification of fruits and vegetables. We observed that the classification result degrades when position of viewing was changed. The reason for this degradation is the variation of the feature pattern for a fruit or vegetable image with viewing position change. The need is to propose a solution of viewpoint-independent fruit and vegetable classification.

3 Materials and Methods

This paper addresses the problem of change in viewpoint for classification. It finds a set of classification features to discriminate fruits and vegetables into multiple classes even when their images have been captured from different viewpoints. The details of this dataset and technique are elaborated in different subsections.

3.1 Dataset

The dataset was collected from the Amsterdam Library of Object Images (ALOI) [11]. The fruits and vegetables are placed on a rotating wheel and the images are captured from a fixed position. Each object is rotated by 360° in steps of 5° and 72 different images were captured. Figure 1 shows the appearance of cauliflower for 30° variation in each snapshot. 23 classes of fruits and vegetables are selected to validate the system. In total, there are 1656 images in this dataset.

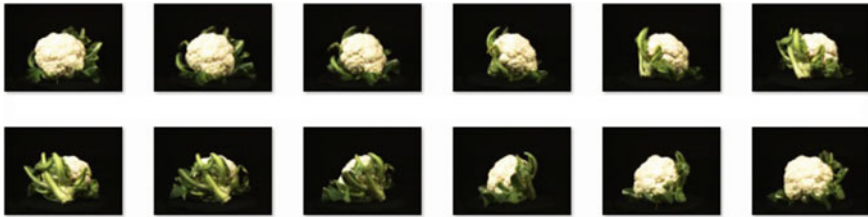


Fig. 1 Snapshots of a cauliflower from different viewpoints

3.2 System Overview

This subsection demonstrates the complete system overview. Figure 2 indicates the overall data flow of the proposed system. The dataset is separated into two parts, i.e., training images and testing images. Both training and testing images are processed for feature extraction. Fractal dimension and GLCM-based features are merged to generate a single vector. The training features are used to train the classifier. Finally, test samples are fed to the trained classifier to get the classification result of fruits and vegetables.

3.3 Conversion to Grayscale

RGB color image (I) has three channels, i.e., IR , IG , and IB . Three channels are merged to generate a grayscale image (I_g) using Eq. (1). This conversion is mandatory for both fractal analysis and GLCM analysis.

$$I_g = 0.299 \times IR + 0.587 \times IG + 0.114 \times IB \quad (1)$$

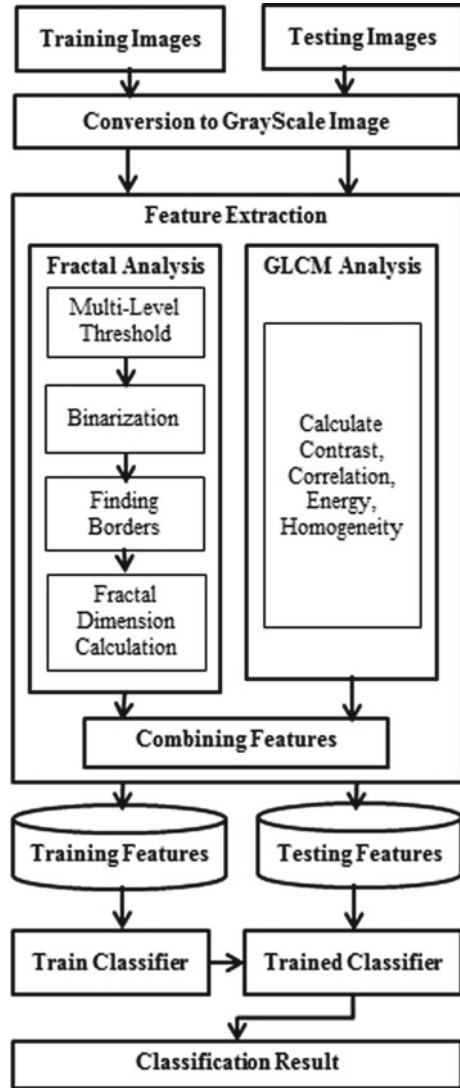
3.4 Fractal Analysis

3.4.1 Multilevel Thresholding

Multilevel thresholding technique [17] is the advanced version of basic Otsu thresholding [23]. This is a very popular algorithm for its faster processing and low storage requirement. It has been applied on the grayscale image (I_g) to extract the threshold values. Assume that I_g has M number of pixels, which has intensity level from 1 to L . f_i denotes the total count of pixel for gray level i . p_i is the probability of intensity level i in the grayscale image. Equation (2) depicts this probability.

$$p_i = \frac{f_i}{M} \quad (2)$$

Fig. 2 Overview of the proposed system



The number of threshold levels (N_t) is given to the system as an input. The set of N_t threshold values $\{T_1, T_2, T_3, \dots, T_{N_t}\}$ is the output of this step. The output depends on the distribution of gray-level histogram. The image has been divided into $K = N_t + 1$ classes. The ω_k in Eq. (3) depicts the cumulative probability of k th class and μ_k in Eq. (4) represents the average intensity of k th class. μ_T represents the average intensity of the entire image and σ_b^2 represents the between class variance. Equations (5) and (6) depict the formulations of μ_T and σ_b^2 , respectively.

$$\omega_k = \sum_{i \in c_k} p_i \quad (3)$$

$$\mu_k = \sum_{i \in c_k} \frac{i \cdot p_i}{\omega_k} \quad (4)$$

$$\mu_T = \sum_{i=1}^L i \cdot p_i = \sum_{k=1}^K \mu_k \cdot \omega_k \quad (5)$$

$$\sigma_b^2 = \sum_{k=1}^K \omega_k (\mu_k - \mu_T)^2 \quad (6)$$

The optimum threshold values $\{T_1^*, T_2^*, T_3^*, \dots, T_{N_t}^*\}$ are selected by maximizing the value of σ_b^2 . Refer to Eq. (7).

$$\{T_1^*, T_2^*, T_3^*, \dots, T_{N_t}^*\} = \text{Arg Max}\{\sigma_b^2(T_1, T_2, T_3, \dots, T_{N_t})\} \quad (7)$$

3.4.2 Binarization

I_g is converted to a binary image using Eq. (8) for each of the values from set T , which was extracted using multilevel thresholding techniques. It creates N_t number of binary images. The binary images using Eq. (8) of a cauliflower sample is represented by Fig. 3, where N_t is 4.

$$I_{bw}(x, y) = \begin{cases} 1, & \text{if } I_g(x, y) > T \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Again, the grayscale image is converted to a binary image using Eq. (9) for each threshold range. The lower threshold value is represented by T_l and the higher threshold value is represented by T_h . The count of threshold range will be $N_t - 1$ when N_t is number of threshold level. A set of binary images are generated from an equal number of threshold range. Figure 4 depicts the binary images with ranged threshold value for cauliflower where N_t is 4.

$$I_{bw}(x, y) = \begin{cases} 1, & \text{if } I_g(x, y) > T_l \text{ and } I_g(x, y) < T_h \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

N_{bw} in Eq. (10) denotes the total number of binary images generated from the binarization. The reason for generating multiple binary images from a single input image is to represent different levels of detailing of the surface texture.

$$N_{bw} = 2N_t - 1 \quad (10)$$



Fig. 3 Demonstration of the generated binary images by Eq. (8), where $N_t = 4$



Fig. 4 Demonstration of the generated binary images by Eq. (9), where $N_t = 4$

3.4.3 Finding Borders

A pattern is hidden inside the segmented image for each class. The identification of border pixels is necessary to find this pattern. The border pixel is selected by analyzing the neighboring pixels. A pixel in a segmented image will be chosen as the border pixel if a pixel ($I_{bw}(x,y)$) with value 1 is surrounded along any of its 8 sides ($N_8[x,y]$) by pixels with value 0. The remaining pixels, which do not satisfy the condition, are treated as non-border pixels. Border pixels are represented in white while all other non-border pixels are converted to black. Equation (11) represents the border image (I_b) generation process from the binary image (I_{bw}). Figures 5 and 6 depict the border images from Figs. 3 and 4, respectively.

$$I_b(x, y) = \begin{cases} 1, & \text{if } I_{bw}(x, y) = 1 \text{ and } I_{bw}(x', y') = 0 \\ & \text{where } (x', y') \in N_8[x, y] \\ 0, & \text{otherwise} \end{cases} \quad (11)$$



Fig. 5 Border image representation of Fig. 3

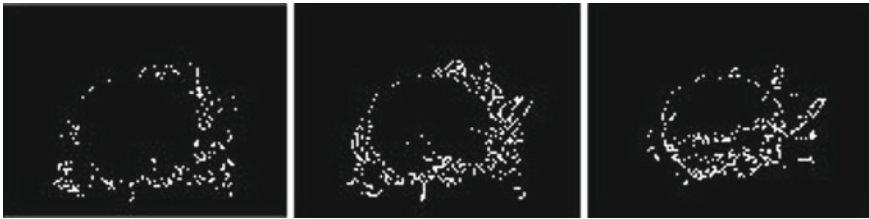


Fig. 6 Border image representation of Fig. 4

3.4.4 Fractal Dimension Calculation

Fractal is a never-ending pattern. It was discovered by Mandelbrot [18]. The fractal dimension (D) computation is done by Eq. (12). In this equation, N is the smaller fragments used to replace a larger piece and F is the size of smaller fragments compared with a larger piece.

$$D = \frac{\log N}{\log \frac{1}{F}} \tag{12}$$

Snapshots of the first four stages of Koch snowflake are shown in Fig. 7. Koch snowflake [33] is an example of a fractal pattern. The Koch curve is initiated as a triangle after which each side of the triangle is replaced by four line segments, each having one-third its original length. Hence the computed dimension of Koch snowflake is 1.231. This fractal dimension can never be in integer form. It always comes in a fractional form.

In this work, fractal dimensions [31] are computed using Hausdorff’s box-counting method [7, 9]. Initial dimension of the border image (I_b) is $(H \times W)$. Equation (13) represents the calculation of the box size (S) at the beginning.

$$S = 2^{\lceil \log_2(\max(H,W)) \rceil} \tag{13}$$

The $I_b[H \times W]$ is padded by zero for $(S - H)$ rows and $(S - W)$ columns to make $I_b[S \times S]$. It is split into square grids of size $\varepsilon \times \varepsilon$. The initial size of the grid is $S \times S$. The number of grids or boxes ($N(\varepsilon)$) with minimum one pixel of the object is counted.

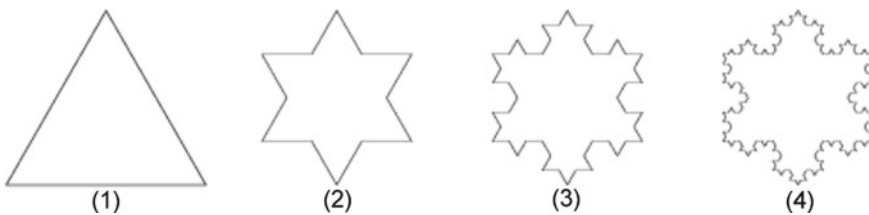


Fig. 7 Snaps of the first four stages of Koch snowflake

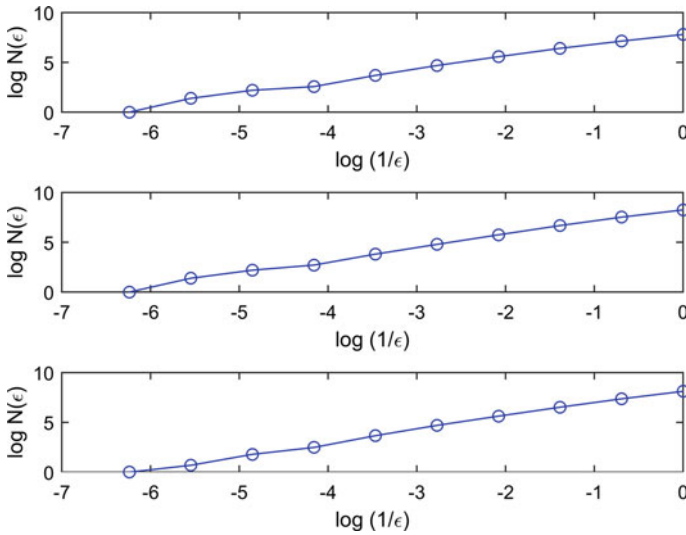


Fig. 8 $\log(1/\varepsilon)$ versus $\log N(\varepsilon)$ plot for the images in Fig. 6

In the next stage, the grid size is made half of the prior stage in both directions. Again the box count ($\bar{N}(\varepsilon)$) is done with a similar criteria. The tasks will be repeated while $\varepsilon > 1$. It will store $\log(1/\varepsilon)$ and $\log N(\varepsilon)$ as x and y coordinates, respectively, in every stage. Figure 8 shows the sample plot of $\log(1/\varepsilon)$ versus $\log N(\varepsilon)$ for the images in Fig. 6. The coordinates are fitted into a straight line with the least square technique. The slope of the approximated line is considered as the fractal dimension (D). Refer to Eq. (14).

$$D = \frac{\Delta\{\log N(\varepsilon)\}}{\Delta\{\log \frac{1}{\varepsilon}\}} \quad (14)$$

In Table 1, a sample study was done to see the effect of changing the viewing position on box-counting based fractal dimension of a fruit or vegetable surface. Fractal dimension was computed and recorded for a sample changing 30° viewing angle every time, where N_t equals 7. The study considered the lowest and highest threshold values as well as the lowest and highest threshold range. The standard deviation nearly equals 0.02 while the viewing angle changes. It states that the fractal dimension is not varying too much when changing the viewing angle. A little variation of the fractal dimension is also observed for a sample image when threshold value or threshold range changes for the same viewing angle. This motivates us to compute fractal dimension for multiple threshold values.

Table 1 Sample study on fractal dimension of mango fruit ($N_t = 7$)

Viewing angle	Lowest threshold value	Highest threshold value	Lowest range threshold	Highest range threshold
0°	1.2664	1.3145	1.2916	1.4810
30°	1.3504	1.3066	1.3683	1.4226
60°	1.2880	1.2718	1.3107	1.4184
90°	1.3099	1.2827	1.3370	1.4118
120°	1.2745	1.2973	1.3127	1.4436
150°	1.2563	1.3174	1.2968	1.4389
180°	1.2393	1.4052	1.2740	1.4506
210°	1.2668	1.3168	1.2882	1.4402
240°	1.2559	1.3413	1.2843	1.4269
270°	1.2697	1.4315	1.2913	1.4678
300°	1.2639	1.2507	1.2891	1.4201
330°	1.2809	1.3030	1.3053	1.4085
Standard deviation	0.0291	0.0519	0.0261	0.0224

Table 2 Algorithm: fractal dimension computation

Input:	A GrayScale image (I_g) and the number of threshold levels (N_t)
Output:	Fractal dimensions (D)
1.	Calculate N_t number of optimum threshold values using the multilevel Otsu thresholding technique and store them in set T in ascending order
2a.	<i>for</i> $i = 1$ to N_t Generate binary image (I_{bwi}) for the threshold value T_i using Eq. (8) <i>end for</i>
2b.	<i>for</i> $j = 1$ to $(N_t - 1)$ Generate binary image ($I_{bw(N_t+j)}$) for the threshold range T_j to T_{j+1} using Eq. (9) <i>end for</i>
3.	<i>for</i> $k = 1$ to $(2N_t - 1)$ Generate border image (I_{bk}) from each of the binary images (I_{bwk}) using Eq. (11) <i>end for</i>
4.	<i>for</i> $k = 1$ to $(2N_t - 1)$ Compute fractal dimension (D_k) for the border image (I_{bk}) using Eq. (14) <i>end for</i>

3.4.5 Feature Vector from Fractal Analysis

Multiple fractal dimensions are extracted from the single input image. Combination of fractal dimension directs toward perfect classification. A feature vector (FF) is formed with the extracted fractal dimensions. FF has $2(N_t) - 1$ number of features as the number of segmented images for each input image. Refer to Eq. (15) (Table 2).

$$FF = \{D_1, D_2, D_3, \dots, D_{2N_t-1}\} \quad (15)$$

3.5 Gray-Level Co-occurrence Matrix (GLCM) Analysis

A gray-level co-occurrence matrix [3, 13] represents the spatial relationships among the pixels. The probability of an intensity value i appears as the neighbor of another intensity value j at a fixed distance d and fixed angle θ , considering that L (total intensity levels in grayscale) is known. Refer to Eq. (16). The dimension of G is decided by the number of gray levels (e.g., 256×256).

$$G = Pr(i, j|d, \theta, L) \quad (16)$$

Directions can be along 0° , 45° , 90° , and 135° . A GLCM is converted to symmetrical form by adding the transpose to the original. Then the normalization of the GLCM is done by dividing all the elements by the sum of the entire matrix.

3.5.1 Contrast, Correlation, Energy, and Homogeneity Calculation from GLCM

This gray-level co-occurrence matrix cannot be directly used as a classification feature. Rather, some statics are computed from the GLCM to provide texture properties of an image. This scalar value of statics can be used as a feature for recognizing a particular pattern. The scalar features [10] are used here, namely Contrast (Ct), Correlation (Cn), Energy (Ey), and Homogeneity (Hy) as defined in Eqs. (17)–(20). $S(i, j)$ is the value of (i, j) th position in the symmetrical normalized directional GLCM, and the range of i, j is from 1 to L .

$$Ct = \sum_{i,j=1}^L |i - j|^2 S(i, j) \quad (17)$$

$$Cn = \sum_{i,j=1}^L \frac{(i - \mu_i)(j - \mu_j)S(i, j)}{\sigma_i \sigma_j} \quad (18)$$

$$E_y = \sum_{i,j=1}^L S(i, j)^2 \quad (19)$$

$$H_y = \sum_{i,j=1}^L \frac{S(i, j)}{1 + |i - j|} \quad (20)$$

3.5.2 Feature Vector from GLCM Analysis

Each of the features is calculated with 1-pixel distance in four directions, i.e., 0° , 45° , 90° , and 135° . It makes a feature vector (FG) of length 16. Refer to Eq. (21) for the structure of the feature vector.

$$FG = \{Ct_{0^\circ}, Ct_{45^\circ}, Ct_{90^\circ}, Ct_{135^\circ}, Cn_{0^\circ}, Cn_{45^\circ}, Cn_{90^\circ}, \\ Cn_{135^\circ}, Ey_{0^\circ}, Ey_{45^\circ}, Ey_{90^\circ}, Ey_{135^\circ}, \\ Hy_{0^\circ}, Hy_{45^\circ}, Hy_{90^\circ}, Hy_{135^\circ}\} \quad (21)$$

3.6 Combined Feature Vector

A feature vector is usually represented by an n-dimensional vector of numerical values and plays an important role in pattern recognition problems. Multiple features are combined to form one feature vector and achieve better accuracy. In this work, it is observed that accuracy gets increased when GLCM-based features are combined with the fractal dimension. Here, the final feature vector (FT) is formed by merging FF and FG . Refer to Eq. (22)

$$FT = \{FF, FG\} \quad (22)$$

3.7 Classification

Classification of objects from images is mostly done by a set of classification features and a classification algorithm. A perfect classification is not possible in practice because of intra-class differences, inter-class similarities and undesirable noise introduced in the image. The choice of the classifier and the decision rule play an important role in accurate classification. The process of selection and extraction of classification features should be decided very carefully by analyzing the problem domain. The set of features is extracted from the training images and the corresponding class labels are given to a classification algorithm to train the prediction model. The same set of features as given during training are extracted from the test image and passed to the trained prediction model. This model returns the class label of an unknown test sample based on the matching probability with a particular class.

4 Experimentation, Result, and Discussion

The complete system was developed in MATLAB R2017a 64-bit version. The entire experimentations are performed in the Intel Core i5 processor with 3.00GHz speed and 4GB Random Access Memory with Windows 7 64-bit operating system. Every fruit and vegetable class has 72 images in the dataset. Twelve images are selected for training out of 72 for each class. The viewing angles of the training set are 0° , 30° , 60° , 90° , 120° , 150° , 180° , 210° , 240° , 270° , 300° , and 330° . The remaining 60 images from other directions are used for testing. In total, the training dataset contains 276 images and the testing dataset contains 1380 images. The number of testing images is five times that of the number of training images. The number of images for training and testing can be selected in random order.

At first, experimentations are done to see the classification ability of the fractal dimension. The classification algorithms are used for experimentation, namely Naive Bayes (NB), Discriminant Analysis (DA), k-Nearest Neighbor (k-NN), and Support Vector Machine (SVM). Table 3 shows the outcome of the complete system using those classification algorithms and a different value of N_f . It shows that the k-NN classifier is performing consistently good for every level of threshold. The maximum accuracy using the k-NN classifier is 96.01%. The efficiency is the problem with the k-NN classifier. k-NN does not generate a trained model, rather it does the prediction

Table 3 Overall classification accuracy using fractal analysis and various classifiers

N_f	SVM	NB	DA	k-NN
4	61.67	79.49	80.22	92.54
5	62.83	83.48	86.38	93.84
6	66.09	85.72	89.49	94.78
7	75.43	89.06	91.67	96.01

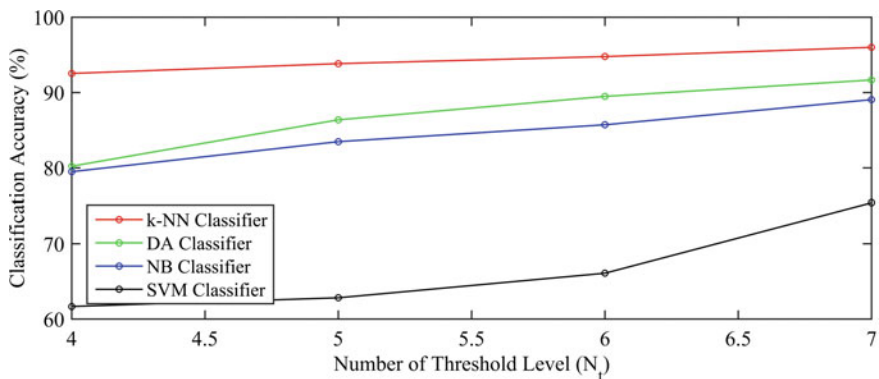


Fig. 9 Accuracy variation with the number of threshold levels (N_f)

at the time of execution with the involvement of the complete training dataset. This issue is addressed by selecting very less data for training compared with testing data. Naive Bayes classifier and Discriminant Analysis classifier also show good results. Plots in Fig. 9 confirm that the overall system accuracy is improving with the increment of N_t for all the classifiers, which have been experimented here. Accuracy degrades after a certain number of threshold levels because the grayscale image has been oversegmented. Sometimes the segmented images are completely black without any object pixel when the lower range of threshold is used for binarization. Particularly for this dataset, the accuracy of classification starts degrading when N_t increases above 7. The same dataset is tested with GLCM-based features and the same classifiers. The accuracy outcome is shown in Table 4. Here also, the overall classification result is good with k-NN, DA, and NB classifier.

The improvement of accuracy is observed when the features from fractal analysis and GLCM analysis are merged. Table 5 shows the overall classification accuracy with a combined feature vector and different classifier. Figure 10 depicts the accuracy comparison of 23 classes using the combined feature vector and the number of

Table 4 Overall classification accuracy using GLCM analysis and various classifiers

SVM	NB	DA	k-NN
40.29	92.54	94.06	93.91

Table 5 Overall classification accuracy using combined feature vector and various classifiers

N_t	SVM	NB	DA	k-NN
4	74.64	97.83	97.03	97.46
5	80.14	98.33	97.32	97.75
6	82.54	97.90	97.83	97.17
7	86.45	98.26	97.97	97.83

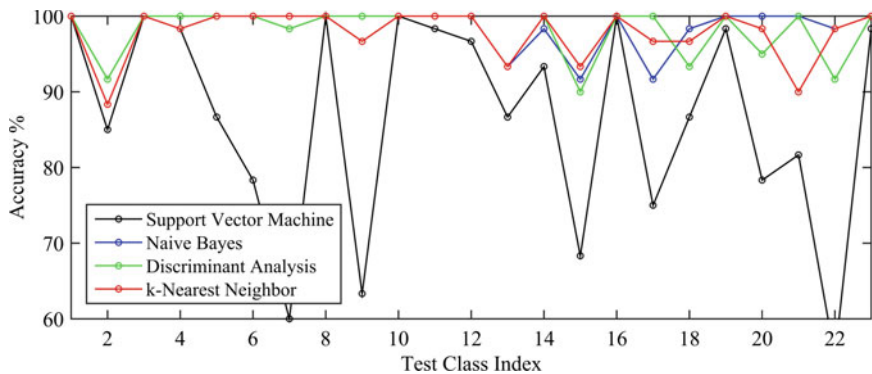


Fig. 10 Class-wise accuracy of the proposed approach using different classifiers when $N_t = 7$

Table 6 Classification time variation using different classifiers and different threshold levels. Image Size: Width-384, Height-288

N_t	SVM	NB	DA	k-NN
4	0.6274	0.6271	0.6271	0.6272
5	0.7873	0.7870	0.7870	0.7871
6	0.9553	0.9551	0.9551	0.9552
7	1.1761	1.1759	1.1759	1.1762

threshold levels (N_t) as 7. The classification outcome is not good with the SVM classifier for most of the classes irrespective of the threshold level. Since SVM is mainly designed for binary linear classification, it is not preferred due to poor performance for large datasets and a large number of classes though it has been extended for multi-class nonlinear problems by using kernel trick. The system accuracy ranged between 97.03 and 98.33% except for the SVM classifier's accuracy. So, we can say that accuracy is stable and not varying too much with the change of classifier or the number of threshold levels. The stability of accuracy is another reason for combining features from both the fractal analysis and GLCM analysis. Table 6 shows the average time taken to classify a test fruit or vegetable sample with combined features. The system takes approximately 1 sec to classify one sample, which is acceptable for fruit and vegetable classification in real time. It is observed that the required classification time for each sample is increased with the increment of N_t . The increment of N_t generates more binary images, which leads to more computation and more processing time.

Table 7 Comparison of technique and the overall accuracy among different approaches

Sl no.	Author and Approach	Techniques	Overall percentage of accuracy
[1]	Roomi et al. [29] (Shape Based)	Circulatory ratio, major axis and minor axis ratio, and eccentricity features with Naive Bayes classifier	40.22
[2]	Ninawe et al. [22] (Mix Feature Based)	Perimeter, area, roundness, mean value of RGB channels, and grayscale entropy features with k-NN classifier	54.49
[3]	Cornejo et al. [6] (Color Based)	Histogram features of hue, saturation channel and census transformed grayscale image with SVM classifier	90
[4]	Proposed Approach	Combination of fractal dimensions and contrast, correlation, energy, homogeneity from GLCM with k-NN classifier, discriminant analysis classifier, and Naive Bayes classifier	98.33

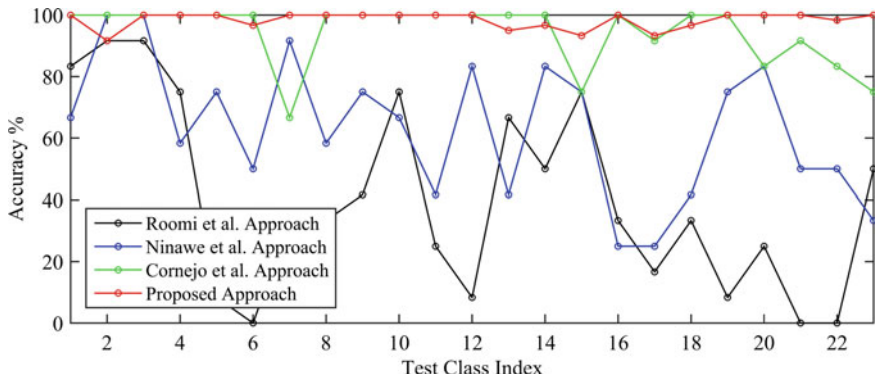


Fig. 11 Comparison of accuracy for each class using previous approaches and the proposed approach

Table 7 compares the proposed approach vis-a-vis some previous approaches. Three papers, which are based on color, shape, and mixture of different types of visual features, are selected for comparing with the proposed approach. Approach 1 [29] deals entirely with shape-based features, however, in most cases, contour shapes of fruits and vegetables tend to change with change in viewing angles. This problem leads to very bad performance with Approach 1. Approach 2 [22] uses a combination of color, texture, and shape features, which however failed to capture the variations arising from a large number of classes. Shape descriptors used by them vary with the change of fruit size or camera distance, i.e., area and perimeter. Approach 3 [6] uses a combination of color and intensity histograms which however works well only when the number of colors is limited. Figure 11 depicts the comparison of accuracy for each class using previous approaches and the proposed approach. Redline in the plot proves the effectiveness of the proposed approach for every class over the previous approaches.

5 Conclusion

A technique for viewpoint-independent classification of fruit and vegetable is proposed in this paper. The input image is transformed into a grayscale image. The grayscale image is split into a number of binary images based on the threshold values generated during the segmentation phase. The border image is generated by excluding the non-border pixels from the binary image. Each of the border images passes through a fractal dimension calculation process using the box-counting method. The multiple dimension values are treated as classification features. In a parallel way, some statistical texture features are computed from GLCM, which was generated from the grayscale version of the input image. The features from the fractal analysis and GLCM analysis are merged to get a stable and robust feature vector for classifica-

tion. The algorithms that can be used for classification are k-Nearest Neighbor, Discriminant Analysis, and Naive Bayes. The overall classification accuracy is between 97.03 and 98.33%. The novelty of this work is the identification and processing of viewpoint-independent texture features and selection of proper machine learning algorithms to make fruits and vegetables classification system robust by changing the viewing position. The classification time is around 1 sec, which is acceptable for an automated system. Another issue of fruit and vegetable classification is the variation of illumination condition. This problem can be a direction toward future research.

References

1. Arivazhagan S, Shebiah RN, Nidhyandhan SS, Ganesan L (2010) Fruit recognition using color and texture features. *J Emerg Trends Comput Inf Sci* 1(2):90–94
2. Biswas H, Hossain F (2013) Automatic vegetable recognition system. *Int J Eng Sci Invent* 2(4):37–41
3. Capizzi G, Sciuto GL, Napoli C, Tramontana E, Woźniak M (2015) Automatic classification of fruit defects based on co-occurrence matrix and neural networks. In: 2015 federated conference on computer science and information systems (FedCSIS). IEEE, pp 861–867
4. Chaidee S, Sugihara K (2017) Approximation of fruit skin patterns using spherical voronoi diagrams. *Pattern Anal Appl* 20(3):783–795
5. Chowdhury MT, Alam MS, Hasan MA, Khan MI (2013) Vegetables detection from the glossary shop for the blind. *IOSR J Electr Electron* 8(3):43–53
6. Comejo JYR, Pedrini H (2016) Automatic fruit and vegetable recognition based on centroid and color representation. In: Iberoamerican congress on pattern recognition. Springer, pp 76–83
7. Costa AF, Humpire-Mamani G, Traina AJM (2012) An efficient algorithm for fractal analysis of textures. In: Proceedings of the 2012 25th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI). IEEE, pp 39–46
8. Dubey SR, Jalal A (2012) Robust approach for fruit and vegetable classification. *Procedia Eng* 38:3449–3453
9. Feng J, Lin WC, Chen CT (1996) Fractional box-counting approach to fractal dimension estimation. In: Proceedings of the 13th International Conference on Pattern Recognition, vol 2. IEEE, pp 854–858
10. Gambardella A, Giacinto G, Migliaccio M, Montali A (2010) One-class classification for oil spill detection. *Pattern Anal Appl* 13(3):349–366
11. Geusebroek JM, Burghouts GJ, Smeulders AW (2005) The amsterdam library of object images. *Int J Comput Vis* 61(1):103–112
12. Haidar A, Dong H, Mavridis N (2012) Image-based date fruit classification. In: Proceedings of the 2012 4th international congress on ultra modern telecommunications and control systems and workshops (ICUMT). IEEE, pp 357–363
13. Haralick RM, Shanmugam K et al (1973) Textural features for image classification. *IEEE Trans Syst Man Cybern* 6:610–621
14. Jana S, Parekh R (2016) Intra-class recognition of fruits using color and texture features with neural classifiers. *Int J Comput Appl* 148(11)
15. Kim Jy, Vogl M, Kim SD (2014) A code based fruit recognition method via image conversion using multiple features. In: Proceedings of the 2014 international conference on IT convergence and security (ICITCS). IEEE, pp 1–4
16. Kuang HL, Chan LLH, Yan H (2015) Multi-class fruit detection based on multiple color channels. In: Proceedings of the 2015 international conference on wavelet analysis and pattern recognition (ICWAPR). IEEE, pp 1–7

17. Liao PS, Chen TS, Chung PC et al (2001) A fast algorithm for multilevel thresholding. *J Inf Sci Eng* 17(5):713–727
18. Mandelbrot B (1967) How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science* 156(3775):636–638
19. Meruliya T, Dhameliya P, Patel J, Panchal D, Kadam P, Naik S (2015) Image processing for fruit shape and texture feature extraction-review. *Int J Comput Appl* 129(8):30–33
20. Nanaa K, Rizon M, Rahman MNA, Ibrahim Y, Aziz AZA (2014) Detecting mango fruits by using randomized hough transform and backpropagation neural network. In: *Proceedings of the 2014 18th international conference on information visualisation (IV)*. IEEE, pp 388–391
21. Naskar S, Bhattacharya T (2015) A fruit recognition technique using multiple features and artificial neural network. *Int J Comput Appl* 116(20)
22. Ninawe P, Pandey MS (2014) A completion on fruit recognition system using k-nearest neighbors algorithm. *Int J Adv Res Comput Eng Technol (IJARCET)* 3
23. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern* 9(1):62–66
24. Pennington JA, Fisher RA (2009) Classification of fruits and vegetables. *J Food Compos Anal* 22:S23–S31
25. Rachmawati E, Khodra ML, Supriana I (2015) Histogram based color pattern identification of multiclass fruit using feature selection. In: *Proceedings of the 2015 international conference on electrical engineering and informatics (ICEEI)*. IEEE, pp 43–48
26. Rachmawati E, Khodra ML, Supriana I (2015) Toward new fruit color descriptor based on color palette. In: *Proceedings of the 2015 international conference on science in information technology (ICSITech)*. IEEE, pp 315–320
27. Rocha A, Hauagge DC, Wainer J, Goldenstein S (2010) Automatic fruit and vegetable classification from images. *Comput Electron Agric* 70(1):96–104
28. Ronald M, Evans M (2016) Classification of selected apple fruit varieties using naive bayes. *Indian J Comput Sci Eng (IJCSSE)* 7(1):13–19
29. Roomi SMM, Priya RJ, Bhumesh S, Monisha P (2012) Classification of mangoes by object features and contour modeling. In: *Proceedings of the 2012 international conference on machine vision and image processing (MVIP)*. IEEE, pp 165–168
30. Seng WC, Mirisae SH (2009) A new method for fruits recognition system. In: *Proceedings of the international conference on electrical engineering and informatics, 2009. ICEEI'09, vol 1*. IEEE, pp 130–134
31. Siddiqi I, Djeddi C, Raza A, Souici-Meslati L (2015) Automatic analysis of handwriting for gender classification. *Pattern Anal Appl* 18(4):887–899
32. Vogl M, Kim JY, Kim SD (2014) A fruit recognition method via image conversion optimized through evolution strategy. In: *Proceedings of the 2014 IEEE 17th international conference on computational science and engineering (CSE)*. IEEE, pp 1497–1502
33. Von Koch H (1904) Sur une courbe continue sans tangente obtenue par une construction géométrique élémentaire. *Norstedt & soner*
34. Wang X, Huang W, Jin C, Hu M, Ren F (2014) Fruit recognition based on multi-feature and multi-decision. In: *Proceedings of the 2014 IEEE 3rd international conference on cloud computing and intelligence systems (CCIS)*. IEEE, pp 113–117
35. Zawbaa HM, Hazman M, Abbass M, Hassanien AE (2014) Automatic fruit classification using random forest algorithm. In: *Proceedings of the 2014 14th international conference on hybrid intelligent systems (HIS)*. IEEE, pp 164–168
36. Zhang Y, Wu L (2012) Classification of fruits using computer vision and a multiclass support vector machine. *Sensors* 12(9):12,489–12,505

Deep Learning-Based Early Sign Detection Model for Proliferative Diabetic Retinopathy in Neovascularization at the Disc



Nilanjana Dutta Roy and Arindam Biswas

Abstract The patients with diabetes for many years are prone to have Diabetic Retinopathy (DR) which is one of the leading causes of blindness. Proliferative Diabetic Retinopathy (PDR) is the advanced stage out of four major progressive stages of DR, where high risk of visual impairments occur. This work shows a deep learning-based automated method for detection of early signs of Proliferative Diabetic Retinopathy at the optic disc area in human retina. Here, we propose the design and implementation of a deep neural network model in replacement of the semi-automated and automated retinal vascular feature extraction methods. Finding the optic disc (OD) center, followed by artery and vein classification from segmented images are essentially important to focus on Neovascularization at the Disc (NVD). A count on the number of major vessels and their width measurement around the OD center are the two indicative parameters for disease diagnosis. Finally, the major vessels are classified as artery and vein sets to differentiate from the newly generated and unwanted blood vessels. This network was trained with the training and testing images of DRIAVE/RITE database for segmentation and artery vein classification. Also, some of our previously published result sets on automated center of optic disc detection on DRIVE dataset have been used to train the model on NVIDIA Titan Xp 8 GB GPU. Finally, the images from MESSIDOR and DIARETDB0 databases were used for testing in detection of Neovascularization at the Disc.

Keywords Deep learning · Convolutional neural network · Proliferative diabetic retinopathy · Early sign detection · Neovascularization at the disc · Image processing

N. D. Roy (✉)

Department of Computer Science & Engineering, Institute of Engineering & Management, Kolkata, India

e-mail: nilanjanaduttaroy@gmail.com

A. Biswas

Department of Information Technology, Indian Institute of Engineering Science & Technology, Shibpur, Howrah, India

e-mail: barindam@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,

Studies in Computational Intelligence 870,

https://doi.org/10.1007/978-981-15-1041-0_6

1 Introduction

The patients with diabetes for many years are prone to have Diabetic Retinopathy (DR), which is one of the leading causes of blindness. Proliferative Diabetic Retinopathy (PDR) is the advanced stage out of four major progressive stages of DR where high risk of visual impairments occur. Proliferation of tiny and abnormal blood vessels at this stage makes the situation more complex. And it needs a fast and robust automated system to extract the quantitative measures for analysis of large amount of data from different sources. Deficiency of oxygen supply in blood vessels is one of the major reasons for proliferation of abnormal and fragile blood vessels. Neovascularization at the Disc (NVD) is a possible case in this situation, where abnormal vessels often grow and rupture. These tiny vessels, which grow abruptly around few diameters of the optic disc (OD) region, have a tendency to leak and bleed. This stage of proliferation is classified as Neovascularization at the Disc (NVD). Neovascularization Elsewhere (NVE) is another form of the disease where they grow anywhere in the vascular structure. Both the classes are equally responsible for vessel proliferation and vitreous hemorrhage, which further trigger chances of vision loss. There are many algorithms exist in the literature for PDR detection in retina [1–4] etc.,. But, in comparison to Non-proliferative Diabetic Retinopathy (NPDR), the number is significantly less. A method for vessel pattern and OD region extraction for NVD/NVE classification has been shown in [5] by multilayered threshold. Some structural and intensity-based features have been used in this method for classification. Another method for monitoring major temporal arcade openness was shown in [6]. For normal and PDR images, they have used single and dual parabolic methods. For stage classification of Neovascularization, Gaussian filtering and morphological operations were followed in [7]. But the problem with these methods is that they are prone to show high false positive rates. This happens due to proper analysis of feature classification for NVD/NVE separately. Also, NPDR lesions detection was performed by the authors of [8] using Hessian-based filtering. But there is a lack of analysis on setting the value of the filters. Vascular feature selection bears significance in diagnosis of early signs of Neovascularization at the Disc (NVD) and other ophthalmic diseases also. Accurate and robust automated system with deep neural network boosts the diagnosis process than the manual and other automated methods. But it needs large amount of data from different sources to extract the quantitative measures for analysis. Therefore, detection of early signs of NVD with deep learning by using convolutional neural network shows an evolutionary step toward fast and accurate ophthalmic disease diagnosis.

Last couple of years has seen a tremendous surge in transferring various fields toward deep learning. Human brain's biological structure and function are the source of inspiration for deep learning. Deep learning contains various features of machine learning and artificial intelligence [9]. The term "deep" in deep learning comes from the fact that the models used in deep learning are constructed of several layers, where the number of layers coins the above term. Though large datasets are required to train the neural network models, they have pretty fast computational speed. The

neural network models which are large, require more data to show better results and alongside, need more number of computations to train itself. Raw data provide several features which contribute to feature learning quality of deep learning models. The condition to be met is that the inputs are fed to the layers need to be labeled. To make the systems capable of learning complex functions at any level, we require unsupervised learning techniques.

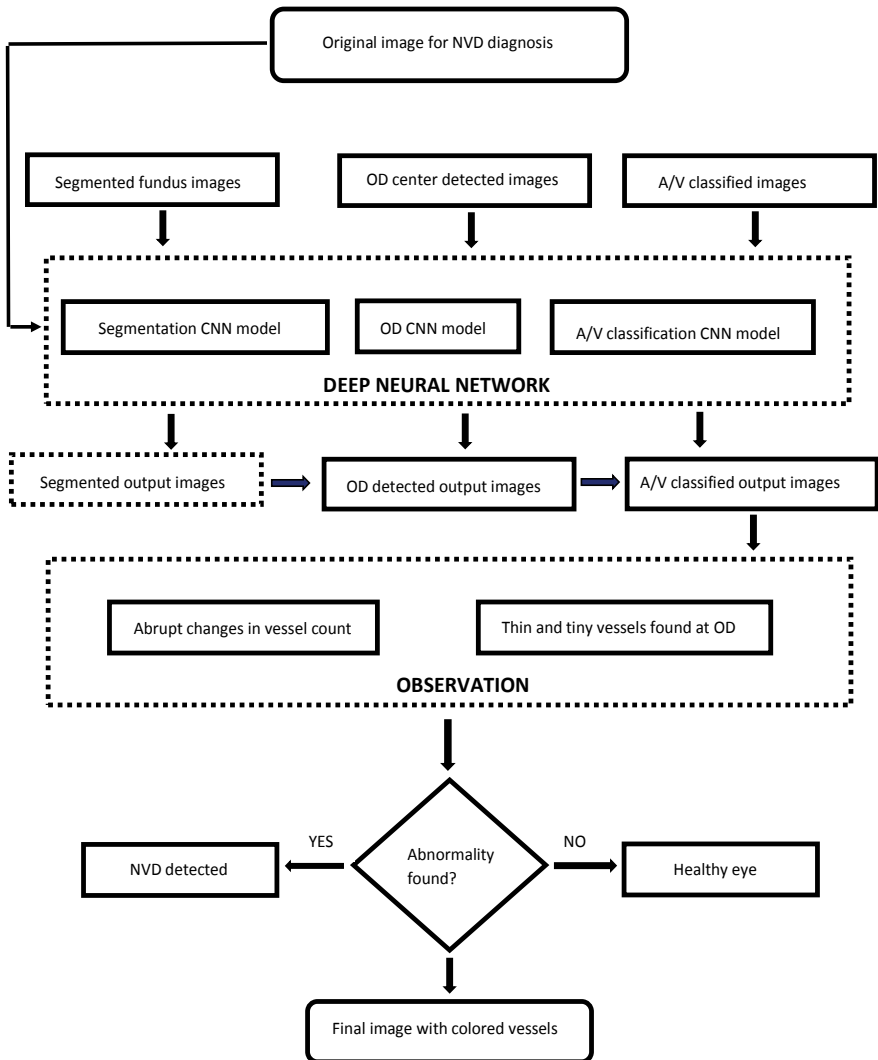


Fig. 1 Flow diagram of the NVD diagnosis method

In this work, we have shown the design and implementation of a convolutional neural network (CNN) model to replace the manual, semiautomated and automated ophthalmic disease diagnosis. This work shows a deep learning-based approach to diagnose the early signs of PDR at the disc which occurs around OD in retina. For accurate, fast and robust NVD diagnosis, the model has been trained accordingly. The architecture of the CNN model always assumes images as the inputs. To avoid the use of handcrafted methods, the model has been trained with some of the labeled datasets from RITE database on a NVIDIA Titan Xp 8 GB GPU with 100 epochs, with

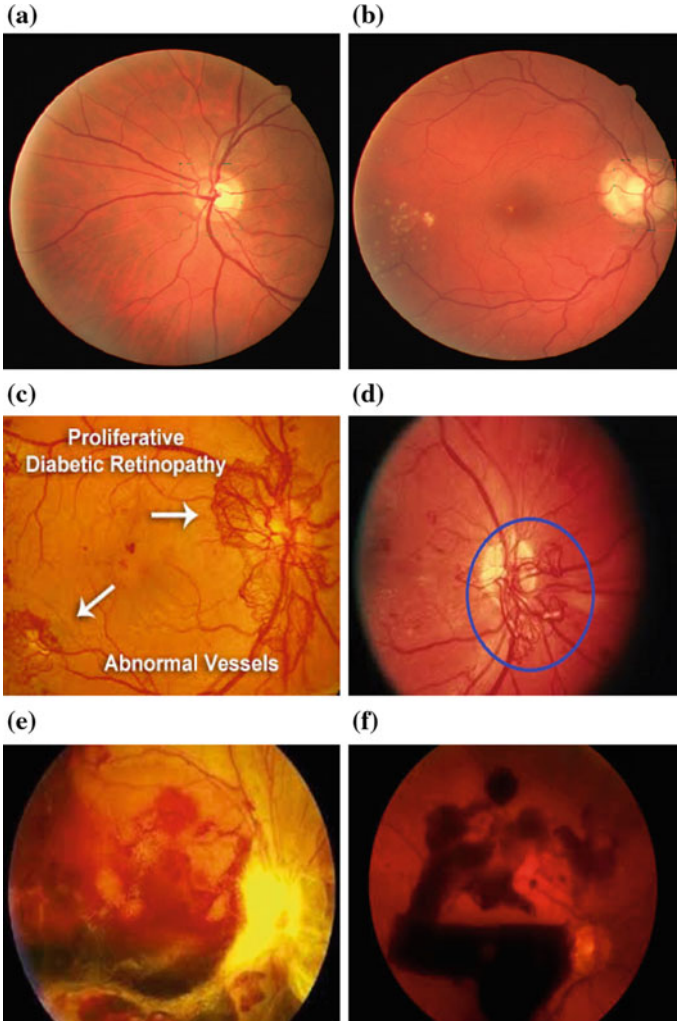


Fig. 2 Normal and PDR affected eye images at various stages of NVD (a), (b) healthy eye images (c), (d) proliferation of fragile vessels around OD and (e), (f) after rupture of the fragile vessels

a training time for 24 h. The final results for segmentation after 1 day of training have been shown in Fig. 11. This work could be helpful for the ophthalmologists to assess the symptoms and take preventive measures accordingly. An overall diagrammatic representation of the complete method is shown in Fig. 1. Healthy eye images and PDR affected images are shown in Fig. 2 where fragile vessels (Fig. 2c, d) and eye health condition after rupture (Fig. 2e, f) are also pointed out.

Here, Sect. 2 describes the working methodology which includes overview, data preparation, segmentation, feature identification, network architecture, and observation. The experimental results and comparison with existing methods are shown in Sect. 3. Finally, an overall conclusion is drawn at Sect. 4.

2 Method

2.1 Overview

The task for NVD-PDR diagnosis is accomplished by segmentation, followed by OD center detection, artery–vein (A/V) classification, count on arterial–venous vessels around OD and width measurement techniques. The vasculature of the images is enhanced by preprocessing and segmentation steps to ensure the quality of the images. Within few diameter areas around OD, regular analysis of diabetic patient’s eye is indicative of the abrupt and unusual changes. The purpose of this work is to diagnose early signs of Neovascularization at the Disc (NVD). So the center of OD, from where the vessels emanate, has been used as the major reference point. A/V classification further helps in finding major vessels of vasculature. Repetitive count on artery and veins at OD and their width measurement are helpful as significant indications toward the abnormal situation. Abrupt changes in vessel count are indicative to unusual changes whereas latter help in finding newly generated thin and unwanted fragile vessels. The framework finally decides an abnormal situation based on observing the number and width of blood vessels which have generated abnormally at the disc area.

The vessel structure from the retinal image is extracted and the vascular skeleton, consisting of centreline pixels, is generated by removing tiny and disconnected vessels. The purpose of providing this as an input is to assign different feature labels to the target pixel. The whole vessel segment is then marked with different features by the labels of the centreline pixels.

The initial framework for the proposed methodology has been described in [10, 11]. So, the Sects. 2.3 and 2.4 have been kept brief. Additional detailing has been shown in the Sects. 2.2, 2.5 and 2.6 for CNN to replace the feature extraction methods by a supervised classifier model. An overall impression of the paper is drawn in Algorithm 1.

Algorithm 1 Overview of the NVD detection method

```

1: Result: Artery–vein classified images from CNN
2: Set accuracy (Acc), Threshold (Thr)
3: for All A/V classified images from database as input do
4:   Train the CNN model by each image
5: end for
6: Count the average width and number of arteries and veins
7: Calculate the accuracy of the trained CNN
8: if  $Acc \geq Thr$  then
9:   Use the same model for testing
10:  Use testing datasets for validation of the CNN model
11:  Note down the results with abnormality after processing
12: end if

```

2.2 Data Preparation

The classifier proposed in this paper has been designed with the aim to label every vessel centerline pixel in accordance with various features of retinal vascular structure. Huge amount of data needs to be provided to CNNs for training of the model. From DRIAVE dataset which include both training and testing sets, 40 retinal images have been used to train the model for the purpose of segmentation. Also, the hand driven A/V classified images from DRIAVE/RITE (Retinal Images vessel Tree Extraction) [12] database have been used to train the proposed CNN model to differentiate between arteries and veins. RITE is a database, which shows A/V classified images inherited from DRIVE. Some of the training images from DRIAVE/RITE database are shown in Fig. 4. Overfitting situations may occur in deep learning due to size of the network. This issue can be resolved with data augmentation where the training dataset is enlarged artificially. A training set of 7200 images has been chosen to avoid overfitting which has been obtained by cropping arbitrary patches of 64×64 from a dataset of 20 records of the DRIAVE/RITE datasets. Our formerly published work (displayed in Fig. 5) for center of optic disc detection [10] has been reused for our proposed classifier (CNN) as a training dataset. For detection of Neovascularization at the Disc, the images from the other two databases, MESSIDOR [13] and DIARETDB0 [14], have also been used here. A glimpse of the MESSIDOR images is shown in Fig. 3. All the 1200 fundus images of MESSIDOR were captured in a 8-bit per color plane at 1440×960 , 2240×1488 or 2304×1536 pixels [15]. DIARETDB0 database contains 130 images altogether with 20 normal and 110 images with signs of diabetic retinopathy (hard and soft exudates, hemorrhages, microaneurysms, and neovascularization). 10% images from all the training sets have been used for creating a test set that has been used for validation. It has taken 1 h on a NVIDIA Titan Xp 8 GB GPU for initial training of the model that has been shown in Fig. 7. Computation has been done in 100 epochs. Figure 11 shows the final segmentation result that has been computed from the model after a day training.

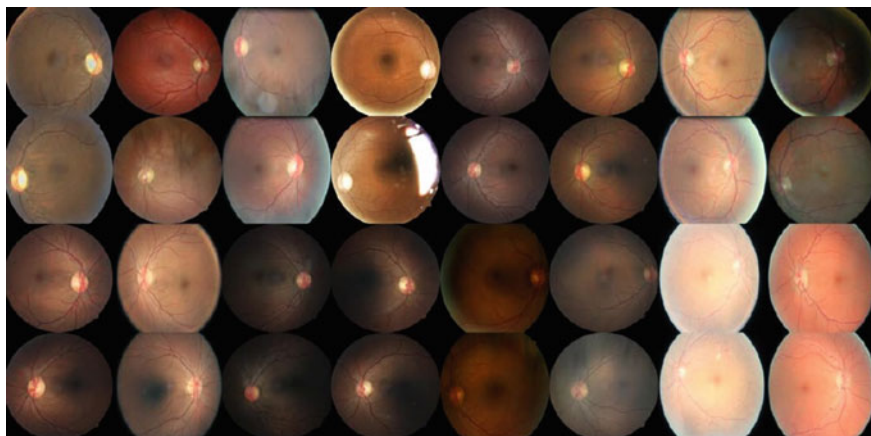


Fig. 3 Fundus images from MESSIDOR database

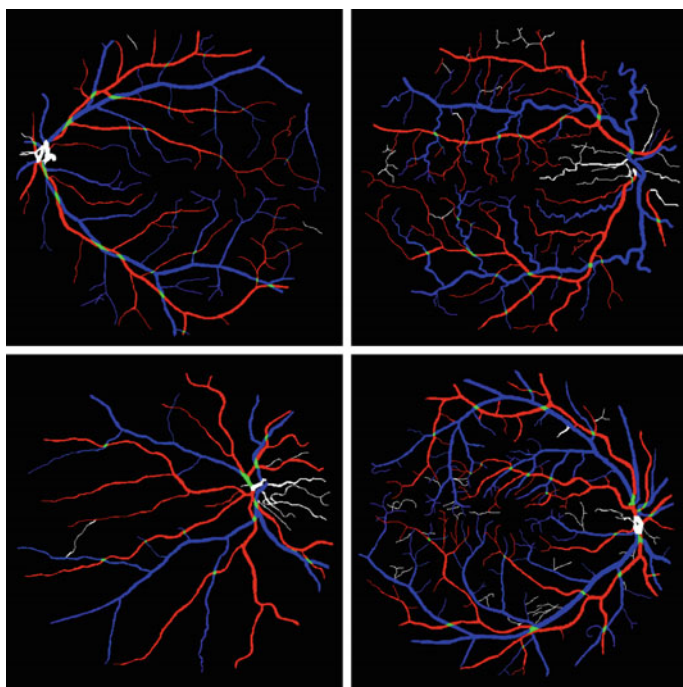


Fig. 4 Sample training A/V classified images from DRIAVE/RITE database

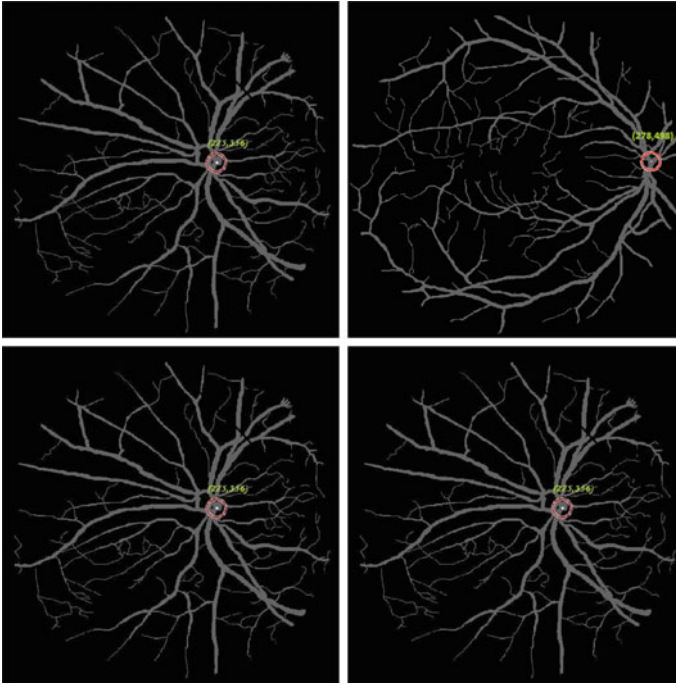


Fig. 5 Sample training OD center located images from our previous work [10] were fed as input to the CNN classifier

2.3 Preprocessing and Vessel Segmentation

To prepare RGB images for segmentation (Fig. 6), a small number of preprocessing steps are implemented on images here which are described in [16]. To make image's content rich, one may use grayscale conversion, normalization of dataset, (using zero mean and unit variance ($x' = \frac{x - \bar{x}}{\sigma}$) and CLAHE (Contrast Limited Adaptive Histogram Equalization) for sharpening. These methods will be followed by Gamma correction ($O = \frac{I^y}{255} \times 255$). The abovementioned preprocessing techniques have been used in the following images which comprise blood vessels that can be clearly seen. Otsu's thresholding [17] is useful for initial stage segmentation. 2D median filter following binarization leads to denoising and binary conversion. The final result comprises a well-shaped image coupled with smooth texture [18].

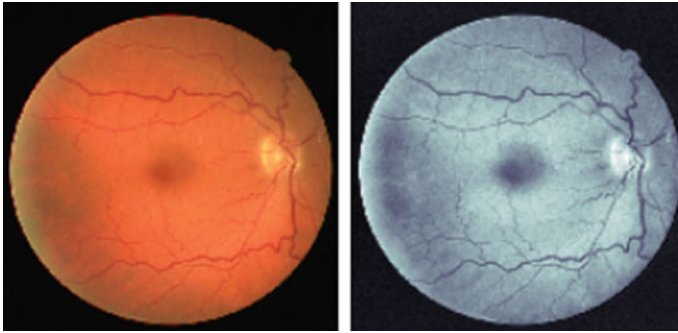


Fig. 6 Initial stage of segmentation shown where original data (Left) with three channels converted into one channel by data preprocessing techniques (Right)

2.4 Feature Identification for NVD Diagnosis

Finding right feature at right time accelerates the performance of any algorithm. Here, some of the crucial features present on retinal vascular network have been identified as reliable parameters toward diagnosis of NVD at early stage. Center of optic disc (OD) [10], classification into artery and vein sets, average width calculation and count on artery and vein vessels have been chosen as working parameters. Our previously published results on optic disc center localization (few samples shown in Fig. 12) are comparable with other well-known optic disc center localization methods [19–21]. For artery and vein classification, methods described in [22] have been used as initial working platform. Then the training images from RITE database are fed into the system for model learning. Finally, observation on average width and count on artery and vein vessels have been used to diagnose the NVD at its early stage. All these features are served as labels for our neural network models.

2.5 Network Architecture for NVD Diagnosis

Weights and biases are learnable parameters of neurons in Convolution neural networks. They look similar to biological neural network. Layers are formed of neurons where they accept input followed by dot product computation and in some cases, follow the same with nonlinearity. For CNN architecture explicitly, an assumption is made that the inputs are in the form of digital image. The connection between every neuron is formed in a local region. This local region camouflages itself as the receptor field of the neuron. The weights that the receptors provide are taken as inputs by the neuron. This is done to lessen the burden in the network on parameters. Weight, height, and depth are the three parameters as per which a neuron is presented in three dimensions. The width and height of a neuron's receptive field

are pretty small. But the model’s former layer has complete extension of the neuron in its depth. Every neuron at the depth slice (single depth) share weight (parameter sharing). Therefore, at single depth for every layer, the convolution of the weight of the neuron and the volume of the input gives the output. Hence, as filter in every layer, the number of weights coupled with the inputs are used. Number of filters that have been used determines the depth of every layer. There are various layers like softmax, dropout, fully connected, ReLU (rectified linear units), pooling, and convolutional cross-channel normalization. These layers can be used to construct a CNN architecture [23]. Compared to tanh and sigmoid function, the ReLU activation function is preferred because of the computational efficiency that this layer provides with training dataset. To prevent overfitting, pooling coupled with cross-channel normalization and dropout is used. Pooling helps in reducing the number of parameters in a network. The fully connected final layer produces probability distribution on the output class labels where this layer is followed by softmax layer.

In this paper, we propose three distinct CNN models for segmentation, center of OD detection and A/V classification, respectively. The CNN model architecture used in this paper for segmentation, serves as the building block for the rest of the two models. In the segmentation model (Fig. 7), for downsampling three convolutional blocks are used and for upsampling three transpose convolutional(deconvolutional) blocks are used. In comparison to the segmentation model, the A/V classification model (shown in Fig. 8) consists of one more convolutional and deconvolutional block. The difference between the model for OD detection and first convolutional block and second last deconvolutional block from the segmentation model is only

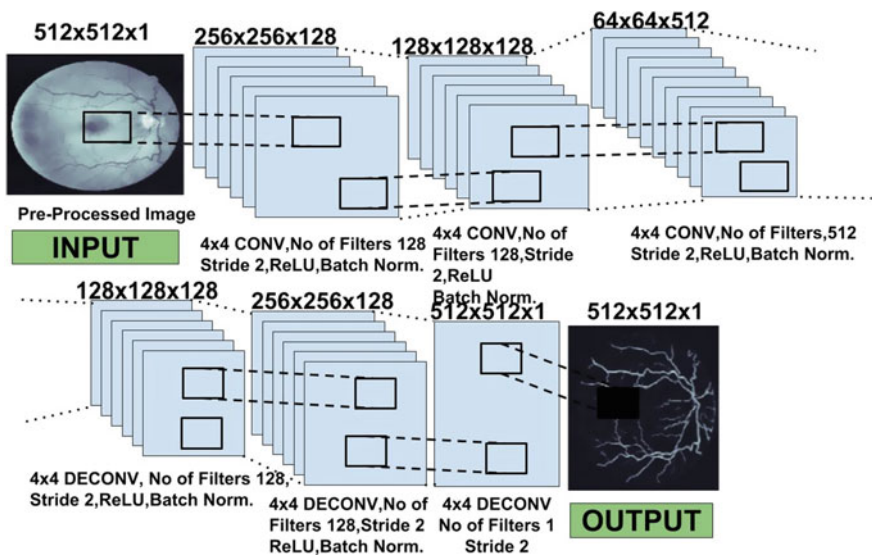


Fig. 7 Our Model Architecture for Segmentation with input channel size 1 and results in output of channel size 1

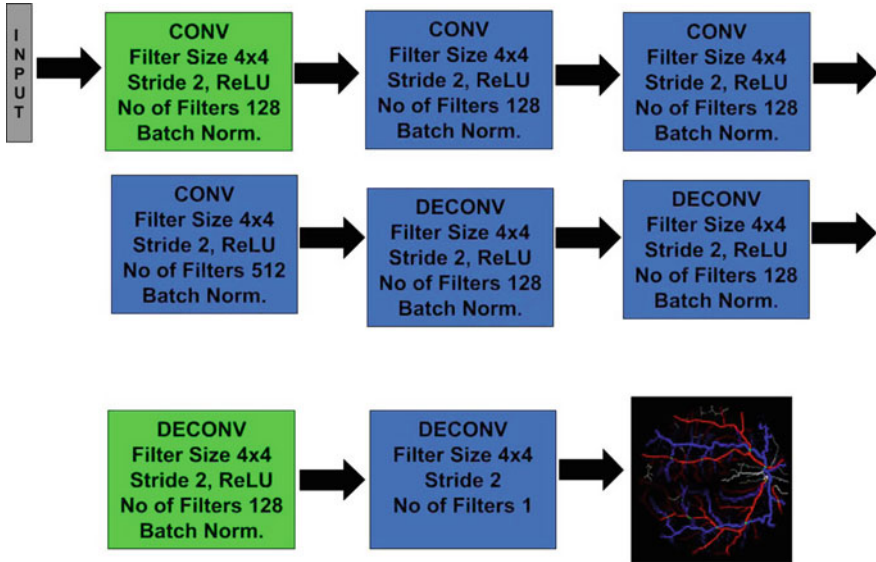


Fig. 8 Our A/V classification Model Architecture. Green color block are added on our Segmentation Model

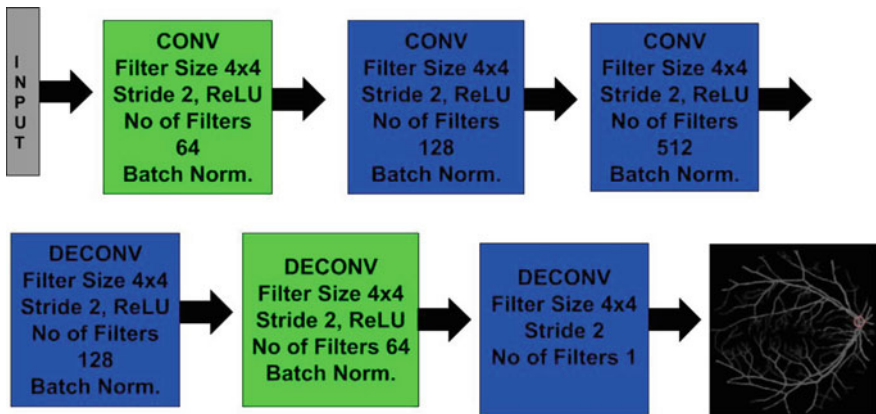


Fig. 9 Our OD detection Model Architecture. Green color block shows the changes from Segmentation Model

in the number of filters (Fig. 9). A filter size of 4×4 and stride of 2 is chosen for convolutional blocks and deconvolutional blocks. Except the last layer, batch normalization and ReLU activation are used at every layer. To stabilize the performance, training batch normalization is used. This leads to faster convergence and other helpful properties. The optimization algorithm that has been used is Adam and Mean Squared Error is used to train the network for loss.

2.6 Observation for NVD Diagnosis

We notice that the CNN classifier accomplishes the NDV detection with an accuracy of 0.96% and achieves low false positive rates. After successful detection of OD center by the CNN model for fixing range of diameters, the model classifies the arteries and veins as well in red and blue colors, respectively. At this stage, an automated verification for tracking the count and width of arteries and veins are done. Abrupt changes in number of vessels and less than a fixed threshold width are indicative toward the abnormal situation. Thus, diagnosis of Proliferative Diabetic Retinopathy in Neovascularization at the Disc becomes easier at its early stage in regular analysis of a patient's eye.

The results from DIARETDB0 and MESSIDOR databases are shown in Table 1 in average case and Table 2 depicts a clear figure of comparison with existing works in terms of its accuracy, sensitivity, and specificity. We found the Fig. 10b as an abnormal one in finding abrupt changes in vessels count, drastic width mismatch and presence of some tiny vessels (shown in yellow color) around the optic disc center area. We, therefore, suggest to provide medical attention to this PDR image where proliferation has already started.

Table 1 Results of Neovascularization at the Disc detection on retinal images

Database	No. of images	Accuracy	Sensitivity	Specificity
DIARETDB0	130	0.96	0.94	0.95
MESSIDOR	1200	0.97	0.94	0.96

Table 2 Comparison with other PDR detection methods

Methods	Accuracy	Sensitivity	Specificity	Level
Jack [15]	0.985	0.963	0.991	–
Agurto [25]	0.94	0.96	0.83	Image
Saranya [1]	0.966	0.962	0.896	–
Goatman [26]	0.91	0.84	0.85	Image
Akram [5]	0.983	0.963	0.989	–
Hassan [7]	0.70	0.63	0.89	Pixel
Jelinek [3]	0.90	0.94	0.82	Image
Nithyaa [2]	–	0.92	–	–
Welikala [4]	0.97	1	0.90	Image
Proposed method	0.96	0.94	0.95	Image

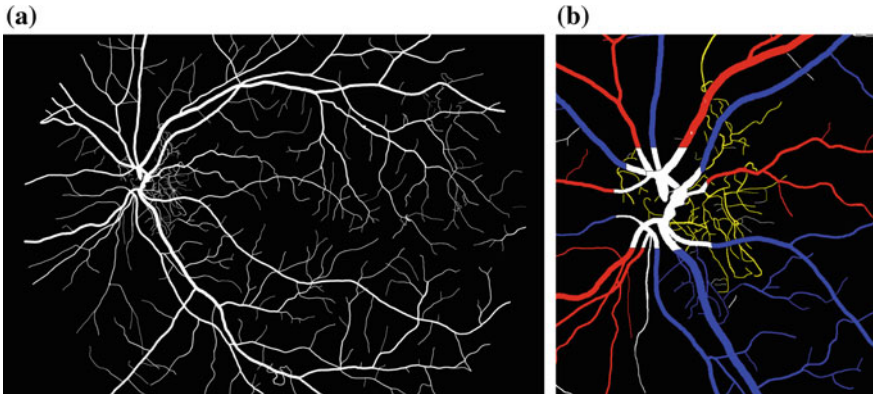


Fig. 10 Showing proliferation of fragile vessels **a** test image for diagnosis **b** proliferation found (yellow colored thin vessels) around optic disc center

3 Result and Discussion

The designed model's performance shows a commendable accuracy improvement in detecting NVD signs. These features are robust enough against rotation, scaling, and poor illumination as the focus is given on major blood vessels of the vascular

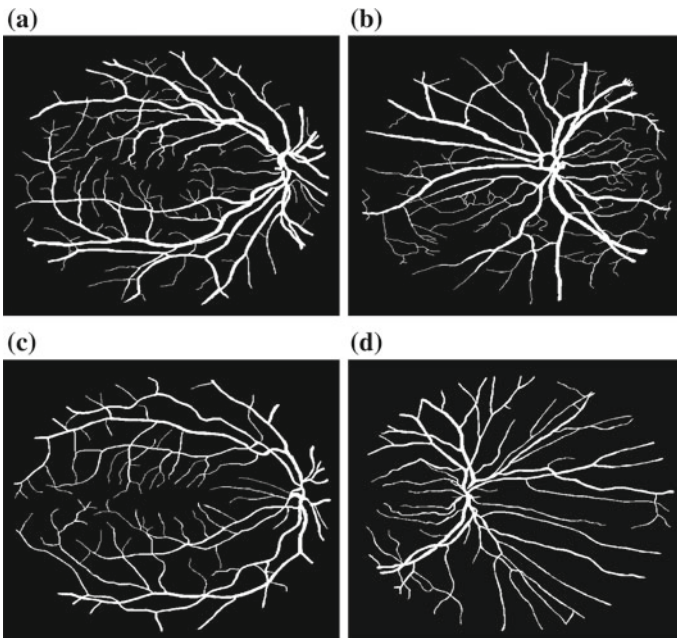


Fig. 11 Segmented results from the classifier

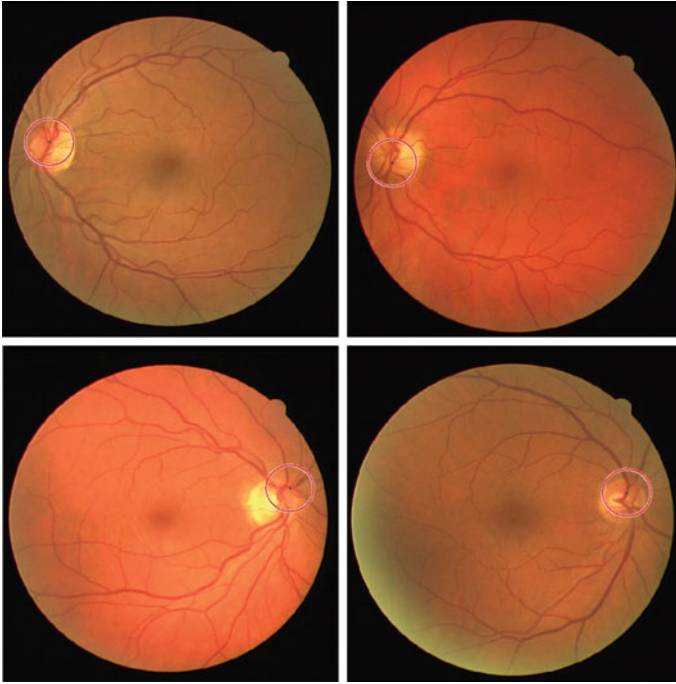


Fig. 12 OD center detected images from DRIVE [24] database

network. Our proposed model has been evaluated on performance and quantitative parameters like accuracy, sensitivity, and specificity with several existing state-of-the-art techniques on MESSIDOR and DIARETDB0 databases. Our method has significant improvement over other existing techniques (Table 2).

In the first step, the methodology was evaluated for the training set of 720,000 images, made by cropping a random patch of 64×64 from the 20 training images of the DRIAVE/RITE dataset. A test set for validation is created by taking 10% images of the training sets. It shows a direct sight of the classifier's performance. The system's performance in terms of accuracy and loss is presented in Fig. 14. More detailing of the model has been shown in Sect. 2.2. ReLU activation and batch normalization are used at all layers except the last layer. Batch normalization stabilizes training, leads to faster convergence and other useful properties. The loss used to train the network is Mean Squared Error and the optimization algorithm used is Adam. The model was trained for 100 epochs with batch size of 1200. The model was trained with a starting learning rate of $1e - 4$ and learning rate decay was used which monitored the test loss. The segmentation model converged with a training loss of 0.0248 and test loss of 0.0329 (Fig. 14a) and training accuracy of 96.89% and test accuracy of 95.63% (Fig. 14b). After the segmentation model was trained, the 20 test images from DRIAVE folder were taken and resized to 512×512 resolution which were then passed to the classifier to get the desirable outputs (Fig. 11). The

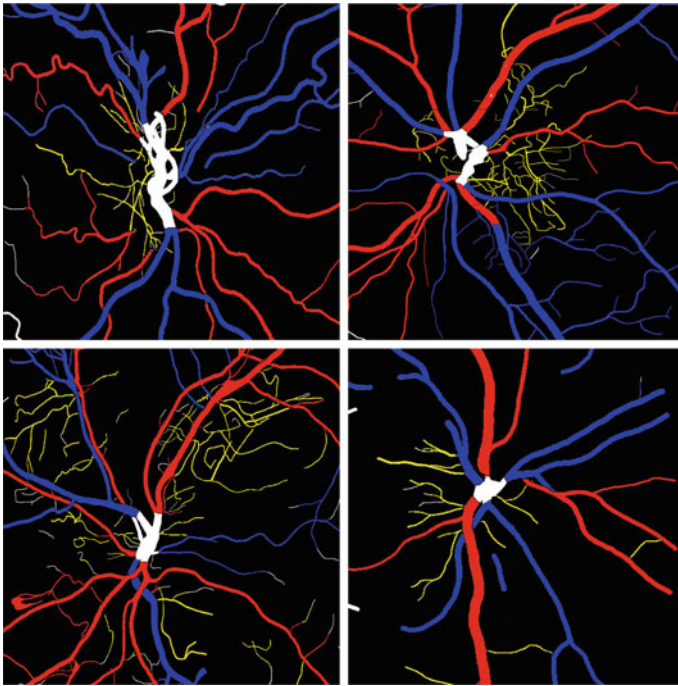


Fig. 13 Showing proliferation of fragile vessels around optic disc center, marked with yellow color in artery vein separated images

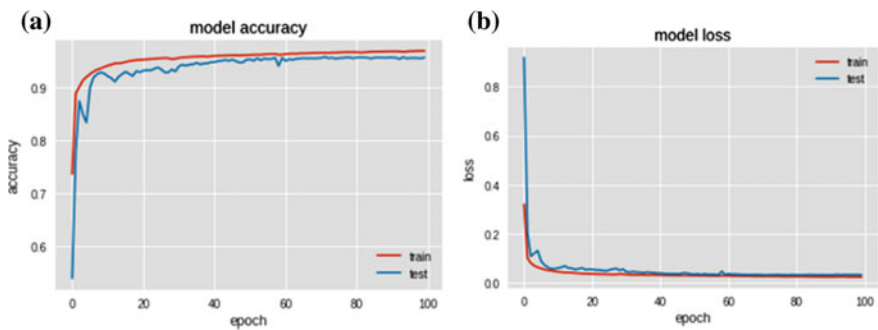


Fig. 14 System’s performance in terms of **a** accuracy **b** loss

A/V classification model and the OD detection models were trained with the same number of epochs, batch size, and learning rate. The performance of the CNN classifiers on DIARETDB0 and MESSIDOR databases has been shown in Table 1. After training, desirable outputs from the A/V classification model (Fig. 13) and the OD center detection model (Fig. 12) were derived. Additionally, the training and testing loss and accuracy for all the models have been reported in Table 3.

Table 3 Loss and accuracy of all the Models

Loss and accuracy				
Network	Train loss	Test loss	Train accuracy (%)	Test accuracy (%)
Segmentation model	0.0357	0.0414	95.23	94.47
OD detection model	0.0217	0.0304	97.03	95.93
AV classification model	0.0215	0.0310	97.01	95.03

4 Conclusion

Proliferative Diabetic Retinopathy is the ultimate stage of Diabetic Retinopathy where proliferation of unwanted blood vessels obstructs vision and leads to permanent vision loss. In this work, we focus on early diagnosis of the signs of NVD where thin and unwanted vessels develop abruptly at the center of optic disc. This is undeniable that the percentage of cure from Diabetic Retinopathy is very less. Photocoagulation is actually a laser analysis which is an effective treatment if it is started before the disease aggravates. Vitrectomy, which is the surgical elimination of vitreous gel, is also followed to improve vision. But this method fails when the retina has already been affected by massive damage. An anti-inflammatory medicine for new blood vessel contraction process, or antivascular endothelial growth factor medication injection is also useful. But, as a preventive measure, this work helps in diagnosis of the early signs Neovascularization at the Disc (NVD) by detecting relevant features present in human retina. Also, for fast and accurate disease diagnosis work on huge data has been accomplished by deep neural network. Ophthalmologists may get immense benefit out of it as this automated method is completely a fast, reliable, and robust one. Also, in future, this will show a ray of hope to many patients with diabetes. Since there is no symptom noticed for PDR until the disease turns severe, periodic analysis on patient's eye is absolutely essential to prevent vision loss. An application development of this framework would be worthwhile in detecting early signs of PDR in future.

Acknowledgements The authors would like to gratefully acknowledge the support of NVIDIA Corporation for donating Titan Xp GPU used in this research.

References

1. Saranya KB, Mohideen SK (2012) A novel approach for the detection of new vessels in the retinal images for screening Diabetic Retinopathy. In: Proceedings of the 2012 international conference on communications and signal processing (ICCSP). IEEE, pp 57–61

2. Nithyaa S (2012) Karthikeyan S identification of the Diabetic Retinopathy By detecting new retinal vessel using fundus image. *IJART* 2(4):55–59
3. Jelinek HF, Cree MJ, Leandro JJ, Soares JV, Cesar RM Jr, Luckie A (2007) Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy. *J Opt Soc Am Opt Image Sci Vis* 1448–1456
4. Welikala R, Dehmeshki J (2014) Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification. *Comput Methods Programs Biomed* 114(3):247–261
5. Akram MU, Khalid S, Tariq A, Javed MY (2013) Detection of neovascularization in retinal images using multivariate m-mediods based classifier. *Comput Med Imaging Graph* 37(5):346–357
6. Oloumi F, Rangayyan RM, Ells AL (2012) Computer-aided diagnosis of proliferative diabetic retinopathy. In: *Proceedings of the international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp 1438–1441
7. Hassan SSA, Bong DB, Premsenthil M (2012) Detection of neovascularization in diabetic retinopathy. *J Digit Imaging* 25(3):437–444
8. Srivastava R, Wong DW, Duan L, Liu J, Wong TY. Red lesion detection in retinal fundus images using frangi-based filters. In: *Proceedings of the IEEE engineering in medicine and biology society (EMBC)*, pp 5663–5666
9. Lan K, Wang D, Fong S, Liu L, Wong KL, Dey N (2018) Survey of data mining and deep learning in bioinformatics. *J Med Syst* 42(8):139
10. Dutta Roy N, Biswas A (2018) Finding center of optic disc from fundus images for image characterization and analysis. In: *Proceedings of ISMAC-CVB, 2018, Springer’s lecture notes in computational vision and biomechanics*, vol 30, pp 1451–1460
11. Dutta Roy N, Biswas A (2018) Early detection of proliferative diabetic retinopathy in neovascularization at the disc by observing retinal vascular structure. In: *Proceedings of ISMAC-CVB, 2018, Springer’s lecture notes in computational vision and biomechanics*, vol 30, pp 1441–1450
12. Hu Q, Garvin MK, Abramoff MD (2015) Rite dataset. <http://www.medicine.uiowa.edu/eye/RITE/>
13. MESSIDOR database (2011). <http://messidor.crihan.fr>. Accessed 28 Feb 2011
14. DIARETDB0 database (2011). <http://www2.it.lut.fi/project/imageret/diaretdb0/index.html>. Accessed 29 Feb 2011
15. Lee J, Chung YZB, Li Q (2013) Detection of Neovascularization based on fractal and texture analysis with interaction effects in Diabetic Retinopathy. *Plos One* 8(12):1–9
16. Chaki J, Dey N (2018) A beginners guide to image preprocessing techniques. ISBN 9781138339316
17. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
18. Dutta Roy N, Goswami S, Goswami S, Biswas A (2018) Biometric template generation framework using retinal vascular structure. In: *Proceedings of CICBA 2018, Springer’s CCIS book series*, vol 1031, pp 245–256
19. Wang R, Zheng L, Xiong C, Qiu C, Li H, Hound X, Li P, Wu Q (2017) Retinal optic disc localization using convergence tracking of blood vessels. *Multimed Tools Appl* 76(22):23309–23331
20. Muhammed LA (2018) Localizing optic disc in retinal image automatically with entropy based algorithm. *Int J Biomed Imaging* Article ID 2815163, 7
21. Abdullah MA, Barman SA (2016) Localization and segmentation of optic disc in retinal images using circular hough transform and grow-cut algorithm. *Peer J*. 4
22. Joshi VS, Reinhardt JM, Garvin MK, Abramoff MD (2014) Automated method for identification and artery-venous classification of vessel trees in retinal vessel networks. *PLoS One* 9(2)
23. Welikala RA, Foster P, Whincup PH, Owen CG, Strachan DP, Barman SA (2017) Automated arteriole and venule classification using deep learning for retinal images from the uk biobank cohort. *Comput Biol Med* 90:23–32

24. The DRIVE database (2017) Image sciences institute, university medical center utrecht. The Netherlands. <http://www.isi.uu.nl/Research/Databases/DRIVE/>. Accessed 7th July 2007
25. Agurto C, Honggang Y, Murray V, Pattichis MS, Barriga S, Bauman W (2012) Detection of neovascularization in the optic disc using an AMFM representation, granulometry, and vessel segmentation. In: Annual international conference of the IEEE, engineering in medicine and biology society (EMBC), pp 4946–4949
26. Goatman KA, Fleming AD, Philip S, Williams GJ, Olson JA et al (2011) Detection of New Vessels on the optic disc using retinal photographs. *IEEE Trans Med Imaging* 30(4):972

A Linear Regression-Based Resource Utilization Prediction Policy for Live Migration in Cloud Computing



Gopa Mandal, Santanu Dam, Kousik Dasgupta and Paramartha Dutta

Abstract A new emerging state-of-the-art challenging research area has been found in cloud computing. Cloud Computing is an idea, rely on service and delivery, it is distributed over the Internet and governed by appropriate set of protocol. In last few decades, Internet is growing rapidly as a result cloud computing and also expanded exponentially. Cloud computing is said to provide resources such as Software, Platform, and Infrastructure as services, namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Cloud profaned the infrastructure resources like CPU, bandwidth, and memory to its end users as a part of its IaaS service. To meet the end users' heterogeneous needs for resources it profaned and unprofane the resources dynamically, with minimal management effort of the service providers over the Internet. Thus, eliminating the need to manage the expensive hardware resources by companies and institutes. However, to satisfy the need for resources of the users on time, Cloud Service Provider (CSP) must have to maintain the Quality of Service (QoS). Service Level Agreement (SLA) is done between the Datacenters and its end users. Minimization of the violation of the SLA ensures better QoS. Research fraternity has proposed that one of the main reasons for violation of SLA is inefficient load balancing approaches in hosts that fail to ensure QoS, without missing the deadline by the distribution of dynamic workload evenly. In this paper, we propose to extend our previous work of simulated annealing-based optimized load balancing [1] by adding VM migration policy from one host to another on the basis of linear regression-based prediction policy for futuristic resource utilization.

G. Mandal (✉) · K. Dasgupta
Kalyani Government Engineering College, Kalyani, India
e-mail: gopa.mandal@gmail.com

K. Dasgupta
e-mail: kousik.dasgupta@gmail.com

S. Dam
Future Institute of Engineering and Management, Kolkata, India
e-mail: sntndm@gmail.com

P. Dutta
Visva-Bharati, Santiniketan, India
e-mail: paramartha.dutta@gmail.com

In our approach, we are going to predict short-time future resource utilization using linear regression based on the history of the previous utilization of resources by each host. We further use it in migration process to predict the overloaded hosts to underloaded ones. Experiments were simulated in CloudAnalyst and the results are quite encouraging and outperform some previous existing strategies of load balancing for ensuring QoS.

1 Introduction

As the computational world is growing aggressively, so to fulfill the need for resources today, huge amount of computational resources is highly required. Cloud computing nowadays is becoming very popular for its extensive use to provide resources to its external users. As a result, it emerges as the most promising technology in industry as well as academia. It is a model, backboneed over the distributed system use to disperse user's task from resident computer to distant virtualized performance-isolated computer clusters known as virtual machines or VMs. To envision the VMs from the host and to provision to its users it needs high-speed Internet. Cloud also de-provisions the resources whenever the need is over. Cloud provides its services in very flexible manner which enables elasticity and scalability of resources by following the pay-as-you-go model. Commercial cloud providers like Google, Amazon, Yahoo, and Microsoft provide such services all over the world to their users. Virtualization techniques are used effectively to share physical machine (host) among multiple performance-isolated platform. Cloud Service Providers (CSP) are responsible to enable the cloud services to its users. Basically, the service providers are responsible to provide processor, memory, and software as a service to their end users on proper time.

Basically, cloud uses a meter-enabled services where not only the end users manage to reduce the capital cost of hardware or software while maintain QoS. This is also called as the "pay- as-you-go" model on subscription basis. Thus, cloud computing fulfills the following need of the end users by:

- a. *Dynamism*: One of the most important needs of computing resource that may be scaled up or down due to heterogeneous user's demand.
- b. *Abstraction*: The end users are completely abstracted from the core functionality of cloud computing they do not need to take care of OS, plug-ins, web-security, software, or platform all these things are totally abstracted from end users.
- c. *Resource Sharing*: Cloud shares its resources by not only ensuring usage optimality of the resources but making the resource adaptive to share application and multifaceted network resources.

The above needs should be provided by CSP within stipulated to ensure efficient cloud service to its users with QoS. Also, SLA are signed among CSP and cloud users. However, this requires quite a bit of involvement in research fraternity and

business houses. There are many parallel fields which have been evolved and tested to provide the SLA. Domain like security, resource provisioning, load balancing, etc. are being studied and reported in the literature. One of the main issues identified in load balancing in Cloud.

1.1 Load Balancing in Cloud Computing

Load balancing is a process of assigning load to individual node from collective system to make resource utilization more effective and improve the response time of the user's job. Simultaneously load balancing algorithms ensure equal distribution of workloads by removing the overloaded or underloaded condition across all the node that may arise in cloud environment. Due to the dynamic nature we cannot predict number of requests that are issued at each second. This unpredictable nature is the ever-changing behavior of the cloud. A good load-balancing algorithm is very needful to find out the way to maximize the resource utilization and enhance the throughput, performance, scalability, fault-tolerance, and response time by avoiding the bottleneck. By effective utilization of resources, it not only saves the energy but also increases the green computing. These parameters are also used to check the proposed load balancer's is good enough or not to balance the load. So, the efficient load balancing must ensure the enhancement of user service, which directly affects the service, provided by the CSP to make profit. Many load balancing algorithms have been proposed so far [2–4]. But none of them ensures the best result so far. Allocating the task evenly in the entire system is a NP-complete problem [5]. Cloud uses virtualization technology [6] to utilize its resources optimally by sharing physical computational resources as virtual machine. Each VM can run separate users' job so multiple users' job simultaneously can run on single host. As, VM is the most powerful feature of load balancing, management of online infrastructure and advanced fault tolerances can be seen in the literature [6–8]. So, a good load balancing technique must be dynamic and adaptive in contrast to any situation that may arise in distributed environment [9]. Depending upon the initiation whether it is initiated by sender or receiver it can be divided into two parts, sender initiated-where a sender initiates the load balancing algorithms or receiver initiated-the receiver is initiator of the load balancing algorithms. Symmetric-It combines both the above-stated policies [10].

Further, load balancing algorithm is broadly categorized as static and dynamic based on its working strategy. Static load balancing algorithm finds out the best possible solution if the environment is stable and compatible. Static algorithm cannot adapt in dynamic environment due to its inflexibility that leads to use of adapt dynamic load balancing strategies. Dynamic algorithm is mostly preferred in the situation where the parameters are changing due to its flexible nature and adaptability, in contrast to the heterogeneous environments. Dynamic algorithm can further be classified as centralized, decentralized, and hierarchical.

According to Escalante et al. [11], any good load balancing strategy must have the following:

- I. Improve the overall performance of the entire system at very lower cost.
- II. To improve the performance always set a backup plan to recover from failure of the system. Failure may be partial or total failure or it will be single or multipoint failure.
- III. Increase the scalability of the system to make the system more stable.
- IV. Setting the priority of the resources and the jobs so that high priority jobs will get better chance to be executed.
- V. Minimizes the average waiting time of the jobs and also the migration time if required.

Based on the above strategies, there are two types of migration policy found in virtual machines, Cold Migration and Live migration. The former migration is a preemptive strategy, where the virtual machines are shut down before migrate from one host to another. After migration, it will restart on another host. Whereas in Live migration, preemption is a very important feature of virtualization, where we migrate a running VM seamlessly from one physical host to another without disconnecting the services provided by the VMs like storage memory and CPU. VMs continue to run after recopy the dirty pages on the destination host. In this proposed work, we have adapted a live migration procedure which dynamically migrate the VMs from overloaded host to underloaded one. The literature survey on Live migration policy we have found in [12] to meet the Service Level Agreement (SLA) and managing power consumption effectively, though there was a trade-off noticed in between consumption of the power and QoS. In this approach, we have noticed VMs have migrated from overloaded to underloaded host if the number of requested CPU becomes greater than the current capacity of the host. This work has some issues such as

- How to spot out the overloaded host?
- Figure out the proper VM that needs to be migrated.
- The way to know the selected host for the particular VMs that are migrated?

In this paper, we try to focus on first two problems by predicting future requirements of expected CPU usage of a host. If the host has adequate resources to meet its requirements or not. From this, the host is determined as overloaded or underloaded. To make provision for incorporating the prediction the existing load balancing strategy using simulated annealing [13] has been further expanded by adding migration policy as an attribute. The prediction of CPU requirements is proposed by implementing linear regression-based resource utilization prediction. It is going to predict the resource utilization of each host using the concept of linear regression method. The next section details the use of linear regression as a prerequisite.

2 Prerequisite to the Proposed Work-Linear Regression

The proposed work in this paper uses regression as a machine learning approach for prediction of resources. Regression used as a popular statistical technique to calculate the relationship between one or more inputs and the output variables. To establish a linear regression model in between input x and output y variable we can take help of the equation of straight line. The line is chosen so that it will best fit the data. The best fit line tries to eliminate total prediction error as much as possible. Prediction error for all data point may retain as small as possible between the point and regression line.

$$y = \beta_0 + \beta_1 x \quad (1)$$

where β_0, β_1 are regression coefficient is used to measure how they are best to fit based on the prediction from the k previous utilization. The output variable y (dependent variable) is predicted or expected value of utilization also it is the magnitude of the residual ξ_i which is basically the contrast between predicted outcome and actual outcome at the data point i . We find x is the predictor or current resource utilization value. The parameter k is assigned to 12 empirically in our model thus the utilization time slot is set on 5(five) minutes in our proposed model.

$$\xi_i = y_i - \bar{y}_i \quad (2)$$

Initially, if it is assumed that for all the data point $\xi_i = 0$ then from the equation, we can say the all points lie on the model. To find the regression coefficient, minimization residual is required. Least square method [14, 15] may be used for this purpose. Here the coefficient parameters are selected in a way such that sum of the square for all residuals are minimized [16].

$$S(\beta_0, \beta_1) = \sum_{i=1}^n \xi_i = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \quad (3)$$

The least squares estimators of β_0 and β_1 are $\hat{\beta}_0$ and $\hat{\beta}_1$ will be satisfied.

$$\frac{\partial s}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \quad (4)$$

$$\frac{\partial s}{\partial \beta_1} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0 \quad (5)$$

Simplifying Eqs. 4 and 5 yield

$$\hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^n y_i - \hat{\beta}_1 \sum_{i=1}^n x_i = \bar{Y} - \hat{\beta}_1 \bar{X} \quad (6)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i x_i - \frac{1}{n} (\sum_{i=1}^n y_i) (\sum_{i=1}^n x_i)}{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} \quad (7)$$

where \bar{X} and \bar{Y} are the means of the X_i and Y_i observations, respectively.

3 Problem Formulation Using Simulated Annealing (SA) and Linear Regression (LR)

We have already proposed SA- [1] based load-balancing strategy. But we found some limitations in our previous work.

- i. We assumed the network is homogeneous so the host has the same resource capacity. (i.e., same CPU, memory, and bandwidth).
- ii. In case of heterogeneous host-based network, the resource capacity will not be the same as homogeneous network.
- iii. Whenever a VM is provisioned to any host the amount of available or residual resource of individual machine will be reduced for next allocation. So, there will be always an error in calculation of residual capacity if the hosts are heterogeneous in nature.
- iv. Simulated Annealing is a local search algorithm and it has been reported in the literature survey that the such localized search algorithm is mostly best fitted for finding localized optimum; achievement of global optimum is difficult.
- v. Migration policy for VM's from overloaded to underloaded host is also not considered here.

Basic objective of the work is to propose more effective load balancing technique by eliminating the above limitation in [1]. So, we further extended our existing work with the machine learning-based prediction technique to migrate VM from one host to another. We consider N number of Datacentres with M number of heterogeneous hosts. Initially, the VMs are allocated based on Simulated Annealing-based load balancing strategy. To meet the variable need of the end users resource usage of the VMs may experience a dynamic workload which means usage of resources also may vary over the time period.

Let, $J = 1, 2, 3, \dots, m$ are the set of different hosts where $I = 1, 2, 3, \dots, n$ are set of VMs needed to be deployed over the time on the host. As per our assumption, the environment is homogeneous in nature so from [1] we can say collective load balancing factor of all VMs deployed on a particular host must be lesser than the residual capacity ($rcap$), thus we can say that the load is balanced for a particular host, otherwise we must initiate our load balancing algorithm.

$$r_{cap} = c - pj \quad (8)$$

$$r_{cap} \leq \sum_{i=1}^n CPU - LBi_1 \quad (9)$$

We also proposed a cost function to allocate these N number of jobs into M number of processors such that the objective or cost function is minimized.

$$f(C) = W_1 * ci(NIC \div MIPS) + W_2 * delay_cost \quad (10)$$

$$\Delta E = C(j) - C(i) \quad (11)$$

where the control parameters W_1 and W_2 have been set here $W_1 = 0.6$ and $W_2 = 0.4$ based on their output found after rigorous test run on the testbed. In Eq. 10 $C(i)$ is the execution cost and $delay_cost$ defines calculated delay cost that has to be given by the CSP to any cloud user as penalty when it is found that job ending time is more than the estimated completion time of CSP. The acceptance probability pr_{ij} as per simulated annealing may be defined by,

$$pr_{ij} = \min \left[1, \exp\left(-\left(C(j) - \frac{C(i)}{c}\right)\right) \right] \quad (12)$$

3.1 Linear Regression-Based Resource Utilization Procedure for VM Migration

As we proposed, resource usage prediction method to optimize the overloaded and underloaded host by the help of live migration process. On the first step, it forecasts the overloaded host and all VMs can be migrated from the overloaded hosts before the violation of service level agreement. In second step, it forecasts the underloaded hosts and in final step all the VMs migrated from overloaded to underloaded hosts. Predicting the future requirements of the resources is very essential for efficient resource management in Datacenters. Our regression model uses a linear function to guesstimate the prediction function. With the help of the prediction function, it is possible to show the linear relationship between the hereafter and current resource utilization. To assess the prediction function in contrast to the past resource utilization history we take 1-h interval initially. In this section, we try to describe the resource utilization methodology is as follows.

1. Detection of overloaded host using resource utilization procedure.
2. Detection of underloaded host using resource utilization procedure.
3. Host selection to place the VMs.
4. Live migration process.

3.1.1 Methodology to Resource Utilization Prediction of the Host-Based on Linear Regression

To calculate the current load, we have examined current resource demand, i.e., CPU, memory, and bandwidth. VMs are provisioned on a host by acquiring the resources from the host. Load of an individual VM can be calculated as prediction of future resource demands is very much important for cloud Datacenter. This future resource prediction can be done with the help of LR model. History of the past resource requirements of the host is required to predict the resource utilization. Past 1-h resource requirement is considered here. Further we will discuss how LR model helps to predict the resource requirements then it will be applied to detect overloading and underloading condition detection. To calculate total load on a host we can accumulate all VM's currently acquired resources running on a host. If m number of VMs deployed over n th host then the average load on the n th host we examine these three main parameters and load on VM can be calculated as given in [17].

$$VM(cpu)_{used} = \frac{\sum VM_j^{mips}}{\sum PM_i^{mips}} \quad (13)$$

$$VM(bw)_{used} = \frac{\sum VM_j^{bps}}{\sum PM_i^{bps}} \quad (14)$$

$$VM(ram)_{used} = \frac{\sum VM_j^{ram}}{\sum PM_i^{ram}} \quad (15)$$

Hosts resource utilization has a linear relationship with the deployed VMs entire load utilization along with CPU, memory bandwidth based on abovementioned multiple parameters (as given in 16).

$$VM(util) = VM(cpu)_{used} + VM(bw)_{used} + VM(ram)_{used} \quad (16)$$

To provisioning a VM, CPU plays the most important role. So, we can say any load on VM is directly proportionate to CPU need.

$$VM(load) = VL = \frac{\sum VM_j^{mips}}{\sum PM_i^{mips}} \quad (17)$$

So, the total load on host is equal to total VMs currently running on host. So, the average load on a physical machine's n^{th} host is

$$PM(load) = \frac{\sum_{j=1}^m VL_j}{m} \quad (18)$$

Though summing up the resources directly may not guarantee accurate host utilization. To calculate host utilization, multiple factors like CPU, memory, and bandwidth are considered by taking the product and combined them for virtual and as well as physical machines also.

$$h(u) = \frac{VM_j(cpu)_{used}}{1 - pm(cpu)} * \frac{VM_j(ram)_{used}}{1 - pm(ram)} * \frac{VM_j(bw)_{used}}{1 - pm(bw)}, \text{ where } j = 1, 2, \dots, m \quad (19)$$

Our proposed algorithm 1 predicts the CPU utilization of the host far past 1 h with 5-min interval. The history of 1-h CPU utilization is significant to increase short-term future utilization. Then it analyzes the host whether it is now overloaded or underloaded and perform migration based on the different resource parameters of the physical machine. The LR-based algorithm is initiated to constructs the function in between past CPU utilization and upcoming CPU need of the each and every host by the Eq. 1 as discussed earlier.

Algorithm 1: UPLRegA-Utility Prediction Linear Regression Analysis Algorithm

Input: *HostList h*

Output: *Predict_utilization*

- 1 *Perquisite: k past utilization interval to approximate the prediction function;*
 - 2 *Initialize the β_0 and β_1 by availing minimal randomized values;*
 - 3 *For $i = 1$ to k interval do;*
 - 4 $x_i \leftarrow Utiliz_History(i);$
 - 5 $y_i \leftarrow Utiliz_History(i + 1);$
 - 6 $\hat{y}_i = \beta_0 + \beta_1 x;$
 - 7 */* Go for loss calculation */;*
 - 8 $\xi_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2;$
 - 9 *Revise the β_0 value using Eq. 6 with the help of i samples of k ;*
 - 10 *Revise the β_1 value using Eq. 7 with the help of i samples of k ;*
 - 11 *End For;*
 - 12 *// Using regression function;*
 - 13 $Predict_utilization = \beta_0 + \beta_1 * Current_total_Utilization(h);$
 - 14 *return Predict_utilization;*
-

In Algorithm 1, initially, we predict small random for the coefficient parameters and β_1 in next phase the prediction function is determined from the k previously utilized history in the host h . From line 3 to 11 of 1, the LR model calculates the dissimilarities in between the actual and anticipated exertion at each data point i here it may be the succeeding exertion value which is actual for the previous data point (line 4 and 5 of Algorithm 1). Anticipated utilization is estimated based on the current utilization and the β_0 and β_1 's value (in line 6 of 1). To calculate the loss sum of square of the residuals are used overall i past data point (line 7 and 8 of 1). The β_0 and β_1 's values are updated on line 9 and 10 to minimize the difference between actual and predicted utilization. Finally, the *Predict_utilization* function is used to

forecast the utilization based on recent total utilization. In 2 *MigrateLive_VMs* are used to initiate two different queues first one is the underloaded host pool consisted host *id* which are underloaded and second one the list of VMs that need migration as their host become overloaded.

Algorithm 2: *MigrateLive_VMs*

Input: *hostList h*

Output: *queue₁ : VMlistthatareneedtomigrate, queue₂ : underloadedhostwhereVMscanbemigrated*

```

1 For each hostlist do;
2 If(Detect_overload(host) is true) ;
3 Select VM with host id(VM, h) to migrate from the corresponding host and
  put it into the queue1;
4 ELSE If(Detect_underload(host) is true));
5 Select host id h and put back into the queue2;
6 queue2 := h;
7 End For;
```

Algorithm 3 is used here to detect the overloaded host *h* its return value is true if the host is seemed to be overloaded and return value is false till further allocation of VMs is possible on the host *h*. Here, a host is considered overloaded when the present use of resources in a host is 85% (the value has been found empirical measures and experimental analysis) of the total usage in initial 1 h, then the host becomes overloaded. Also, from Algorithm 1, if we get the *predicted_util* value greater than current total utilization value then a particular host becomes overloaded.

Algorithm 3: *Detection of overloaded host*

Input: *host h*

Output: 1 if host is overloaded else 0

```

1 If (utiliz_history_length < 12);
2 If (current_total_utiliz(host) > 0.85 * (total_utiliz(host)));
3 return TRUE;
4 else;
5 return FALSE;
6 end if;
7 end if;
8 else;
9 predicted_util := UPLRegA(host);
10 If (predicted_util > curent_total_utiliz(host));
11 return TRUE;
12 Else;
13 return FALSE;
14 end if;
15 end else
```

Algorithm 4: Detection of underloaded host

Input: *host h*
Output: Boolean 1: if host is underloaded else 0

- 1 If (*utiliz_history_length* < 12);
- 2 If (*Current_Total_Utilz(host)* = 0;
- 3 return **TRUE**;
- 4 else;
- 5 return **FALSE**;
- 6 end if;
- 7 end if;
- 8 else;
- 9 *prediction_Util* := *UPLRegA(host)*;
- 10 If (*prediction_Util* > *Curent_Total_Util(host)*);
- 11 return **TRUE**;
- 12 Else;
- 13 return **FALSE**;
- 14 end if;
- 15 end else;
- 16 *prediction_Util* := *UPLRegA(host)*;
- 17 If(*(prediction_Utilization)* < 0.15 * (*Total_Utilz(host)*);
- 18 return **TRUE**;
- 19 else;
- 20 return **FALSE**;
- 21 end If;
- 22 end else;

Algorithm 4 is used to detect the underloaded host, to detect this underloaded condition we consider the current total utilization is equal to zero. Now, if in line 11 and 12 (of Algorithm 4) after 1 h past 12 utilization by using Algorithm 1(*UPLRegA*) we predict the host utilization. That will be less than equal to 15% as we consider 85% total utilization in overloaded condition so it is complement, i.e., 15% is considered for underloaded condition. Algorithm 3 is used to select the virtual machines to migrate and to maintain the QoS without violation of SLA.

Algorithm 5: Algorithm for Load balancing in VMs using Simulated Annealing

Input: *Index_map_table* consisting of *VMid* and *associated job's request id*
Output: Load balanced VMs

- 1 Allocate *VM's* in the available host *h* by checking the hosts residual capacity using Eq. 8;
- 2 If ($r_{cap} \leq \sum_{i=1}^n CPU - LBi_1$) then go to next step;
- 3 Initialize a random set of control parameters and process unit vectors with large positive value;
- 4 Do While value of control parameters value touches the minimality;
- 5 induce and invoke another cost function $f(J')$ from $f(J)$;
- 6 calculate ΔE using Eq. 11;
- 7 if $\Delta E = 0$ and $\exp(\frac{-(C(J)-C(i))}{C}) > \text{allocatorandomly}(0, 1)$ do $f(J')$ from $f(J)$;
- 8 reduce the temperature with certain rate;
- 9 end while;
- 10 end if

To increase the efficiency of the simulated annealing-based load balancing we incorporated live migration policy in the proposed work, where initially VMs are allocated using Algorithm 5 until the first 1 h is passed. Later, the live migration of VMs is performed among the hosts which are detected overloaded using Algorithm 3 to underloaded host using Algorithm 4 using the live migration using Algorithm 6.

Algorithm 6: Linear regression-based live migration

Input: (*utiliz_history_length*)
Output: live migration of VMs based on prediction

- 1 while (*utiliz_history_length* < 12) ;
- 2 Call Algorithm 5;
- 3 Increase *utiliz_history_length*;
- 4 current optimal solution stored and index table updated globally;
- 5 End while;
- 6 If (*utiliz_history_length* > 12) then;
- 7 Send the current *hostlist* to *UPLRegA(hostlist)*;
- 8 Call Algorithm 5 to predict host condition;
- 9 End if;

4 Overview of Simulation Tool CloudAnalyst

Cloud computing infrastructure demands the deployment of the environment in such a way that a remarkable number of users can get cloud-enabled services simultaneously. This is the prime objective of cloud computing. Cloud Computing infrastructure is allotted and expanded in different geographic locations. There are several parameters like resources, memory, multiple numbers of network components, etc. which must be considered. Most of the parameters have firsthand influence in the performance of any load balancing approach. It is the most inconvenient for the developer to predict these parameters before. So, examining the result of the execution of the proposed algorithm based on real-world condition on physical infrastructure is not possible. The obvious solution is to use a simulator where virtually setup of a real-world environment is possible. Among different available simulators CloudAnalyst [13] is used here for simulation. This simulator was first proposed by University of Melbourne. The researchers can test their algorithm visually and can arrange different experiments by setting different control parameters. The flexibility and the different configuring capabilities are the best feature of the CloudAnalyst. The output of the simulator is pictorial representation in form of tables and charts which facilitate the developer to understand and analyze their result (Fig. 1).

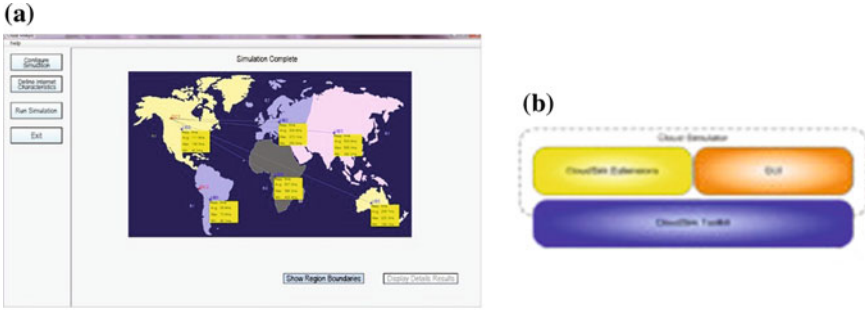


Fig. 1 A view showing **a** GUI **b** Framework of CloudAnalyst

5 Simulation with Results and Analysis

To check the performance of our proposed algorithm we implement our proposed algorithm on CloudAnalyst which is designed and deployed over the CloudSim toolkit. CloudAnalyst considers the scenario of social networking site like FaceBook by partitioning the world into six different “Regions” as given in Table 1.

All Userbase (UB) has been formulated at a single timezone for the geographic distributed UBs . There will be timeslot when there are many users (defiend as *peak hours*) and timeslot when there is normal or less users (defiend as *off-peak hours*). The value of *off-peak hours* has been set as one-tenth of online users by empirical study. Each Datacenter consists of 8 GB RAM, 200GB storage space and 10000 MB available bandwidth. Each Datacenter has 4 CPU which consist 10000 $MIPS$ capacity. The simulation study uses the X86 architecture in Linux platform. It is also assessed that each user’s job needs atleast 100 instructions to be executed. Our proposed approach is compared with five already proposed algorithms among them two are traditional scheduling policy-based algorithm. (I) First Come First Serve (FCFS), (II) Round Robin (RR), and the three others are soft computing-based optimization technique. (III) Genetic Algorihm(GA) [18], (IV) Stothastic Hill Climbing (SHC) [19] and V) Simulated Annealing (SA) [1]. We found all these

Table 1 Overall Setup of the simulation environment

S. no	User base	Region	Concurrent online users during peak hrs.	Online concurrent users during off-peak hrs.
1.	UB1	0—N. America	4,75,000	82,000
2.	UB2	1—S. America	6,25,000	2,00,000
3.	UB3	2—Europe	3,75,000	74,000
4.	UB4	3—Asia	8,45,000	1,14,000
5.	UB4	4—Africa	1,24,000	14,000
6.	UB4	5—Oceania	1,84,000	44,200

Table 2 Testing environment settings and measured average Response Time (*RT*) in (*ms*) using one DC

S. no	Cloud configuration	Configuration of Datacenter (One DC with)	Measured <i>RT</i> in ms for SA with migration policy	Measured <i>RT</i> in ms for SA-based load balancing	Measured <i>RT</i> in ms for GA-based load balancing	measured <i>RT</i> in ms for SHC-based load balancing	Measured <i>RT</i> in ms for RR-based load balancing	<i>RT</i> in ms for FCFS-based load balancing
1.	CC1	25 VMs	325.32	326.86	329.01	329.02	330	330.11
2.	CC2	50 VMs	325.1	326.24	328.97	329.01	329.42	329.65
3.	CC3	75 VMs	231.02	235.46	244	329.34	329.67	329.44

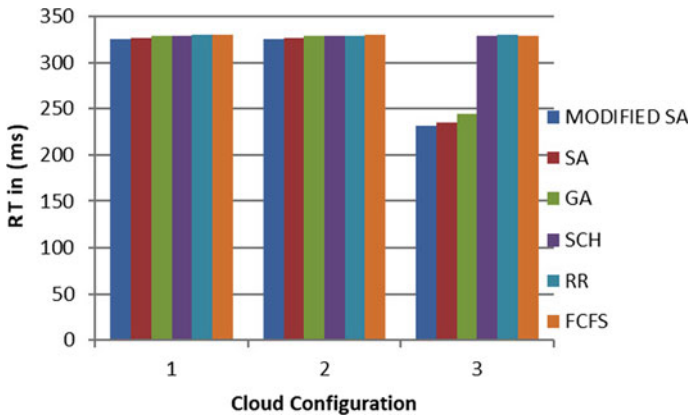


Fig. 2 Graph depicting of effectiveness of proposed algorithm with SA, ACO, GA, SHC, and FCFS using one Datacenters

Table 3 Testing environment settings and measured average Response Time (*RT*) in (*ms*) using two DCs

S. no	Cloud configuration	Configuration of Datacenter (One DC with)	Measured <i>RT</i> in ms for SA with migration policy	Measured <i>RT</i> in ms for SA-based load balancing	<i>RT</i> in ms for GA-based load balancing	<i>RT</i> in ms for SHC-based load balancing	<i>RT</i> in ms for RR-based load balancing	<i>RT</i> in ms for FCFS-based load balancing
1.	CC1	25 VMs each	349.81	352.31	360.77	365.44	376.27	376.34
2.	CC2	50 VMs each	343.61	347.51	355.72	360.15	372.49	372.52
3.	CC3	75 VMs each	341.08	346.05	355.32	359.73	369.48	370.56
4.	CC4	25, 50 VMs each	338.32	345.21	350.58	356.72	367.91	368.87
5.	CC5	25, 75 VMs each	338.11	345.52	351.56	357.23	369.45	36.23
6.	CC6	75, 50 VMs each	337.87	34.86	352.01	357.04	356.01	361.01

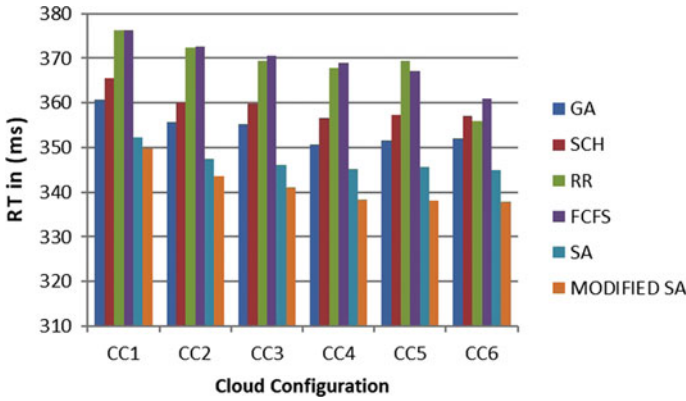


Fig. 3 Graph depicting of effectiveness of proposed algorithm with SA, ACO, GA, SHC, and FCFS using two Datacenters

load-balancing algorithm is working effectively but no one considered migration of VM if required from overloaded to underloaded host. So, our proposed modified simulated annealing strategy maybe become very useful when we introduced LR-based migration policy which predicts the host’s future resource requirement and allocates VM accordingly to the proper host. We found the proposed strategy is not only good but also outperforms those five existing strategies when compared. We consider several scenarios initially starting with one DC with 25, 50, 75 VMs as given in Table 2. Further the proposed algorithm is tested for two DCs, three DCs, four DCs, five DCs, and six DCs. The results are tabulated in Tables 3, 4, 5, 6, and 7, respectively. The comparison of the proposed work with existing methods (as named above) are depicted in corresponding Figs. 6, 2, 3, 4, 5 and 7.

Table 4 Testing environment settings and measured average Response Time (RT) in (ms) using three Datacenters

S. no	Cloud configuration	Datacenter configuration (One DC with)	Measured RT in ms for SA with migration policy	measured RT in ms for SA	Measured RT in ms for GA-based load balancing	Measured RT in ms for SHC-based load balancing	RT in ms for RR-based load balancing	Measured RT in ms for FCFS-based load balancing
1.	CC1	25 VMs each	336.38	342.13	350.32	356.82	366.17	363.34
2.	CC2	50 VMs each	332.5	340.86	350.19	355.25	363.52	363.52
3.	CC3	75 VMs each	331.03	338.34	346.01	350.73	360.18	361.56
4.	CC4	25, 50, 75 VMs each	328.72	336.89	345.98	350.01	361.21	360.87

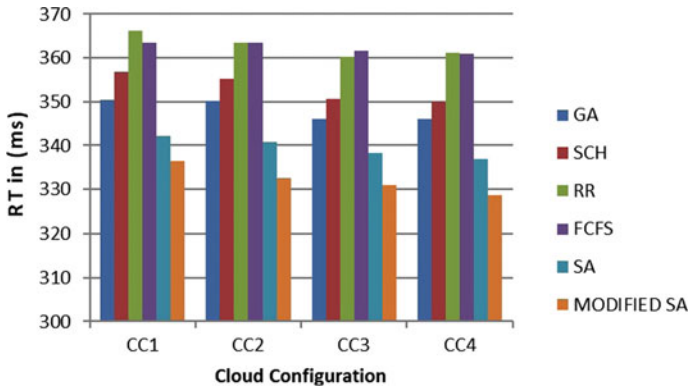


Fig. 4 Graph depicting of effectiveness of proposed Algorithm with SA, ACO, GA, SHC, and FCFS using three Datacenters

Table 5 Testing environment settings and measured average Response Time (*RT*) in (*ms*) using four Datacenters

S. no	Cloud configuration	Configuration of Datacenter (One DC with)	Measured <i>RT</i> in ms for SA with migration policy	Measured <i>RT</i> in ms for SA-based load balancing	<i>RT</i> in ms for GA-based load balancing	Measured <i>RT</i> in ms for SCH-based load balancing	<i>RT</i> in ms for RR-based load balancing	Measured <i>RT</i> in ms for FCFS-based load balancing
1.	CC1	25 VMs each	329.1	336.89	348.85	354.35	359.35	360.95
2.	CC2	50 VMs each	330.33	337.25	345.54	350.71	356.93	359.97
3.	CC3	75 VMs each	326.67	332.32	340.65	346.46	352.09	358.44
4.	CC4	25, 50, 75 VMs each	324.71	331.23	337.88	344.31	351	355.94

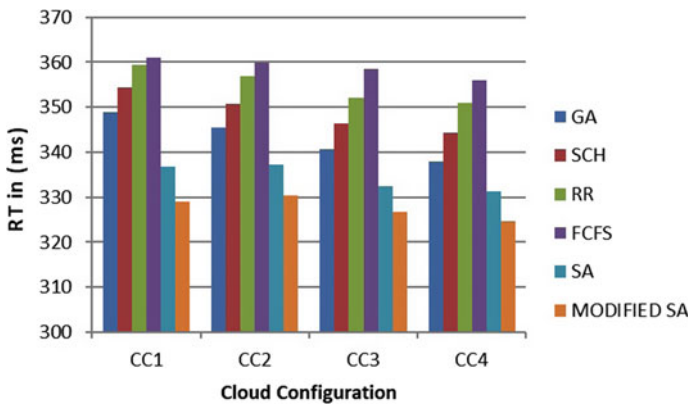


Fig. 5 Graph depicting of effectiveness of proposed Algorithm with SA, ACO, GA, SHC, and FCFS using four Datacenters

Table 6 Testing environment settings and measured average Response Time (*RT*) in (*ms*) using five Datacenters

S. no	Cloud configuration	Configuration for Datacenter (One DC with)	Measured <i>RT</i> in ms for SA with migration policy	Measured <i>RT</i> in ms for SA-based load balancing	Measured <i>RT</i> in ms for GA-based load balancing	measured <i>RT</i> in ms for SHC-based load balancing	Measured <i>RT</i> in ms for RR-based load balancing	<i>RT</i> in ms for FCFS-based load balancing
1.	CC1	25 VMs each	323.13	329.02	335.64	342.86	348.57	352.05
2.	CC2	50 VMs each	314.24	318.45	326.02	332.84	339.76	345.44
3.	CC3	75 VMs each	312.61	317.2	322.93	329.46	335.88	342.79
4.	CC4	25, 50, 75 VMs each	311.2	314.24	319.98	326.64	334.01	338.01

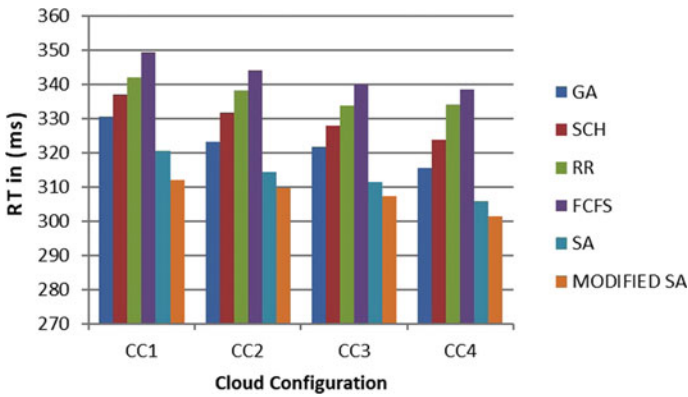


Fig. 6 Graph depicting effectiveness of proposed algorithm with SA, ACO, GA, SHC, and FCFS using five Datacenter

Table 7 Testing environment settings with values of measured average Response Time (*RT*) in (*ms*) using six Datacenters

S. no	Cloud configuration	Configuration of Datacenter (One DC with)	Measured <i>RT</i> in ms for SA with migration policy	Measured <i>RT</i> in ms for SA-based load balancing	Measured <i>RT</i> in ms for GA-based load balancing	<i>RT</i> in ms for SHC-based load balancing	Measured <i>RT</i> in ms for RR-based load balancing	measured <i>RT</i> in ms for FCFS based load balancing
1.	CC1	25 VMs each	311.82	320.45	330.54	336.96	341.87	349.26
2.	CC2	50 VMs each	309.56	314.23	321.01	331.56	338.14	344.04
3.	CC4	25, 50, 75 VMs each	307.23	311.21	321.54	327.78	333.67	339.87
4.	CC3	75 VMs each	301.23	305.87	315.33	323.56	334.01	338.29

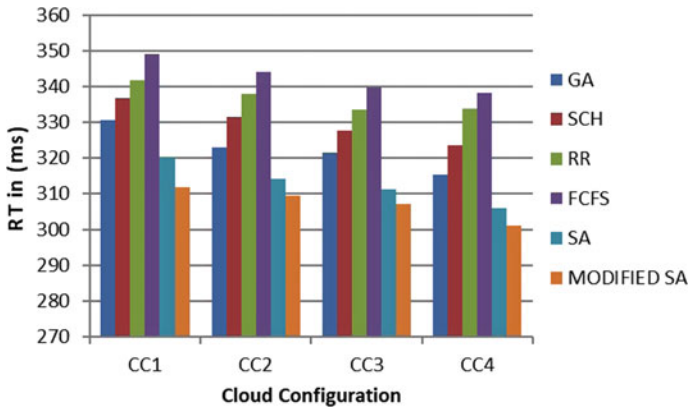


Fig. 7 Graph depicting of effectiveness of the proposed Algorithm with SA, ACO, GA, SHC, and FCFS using six Datacenters

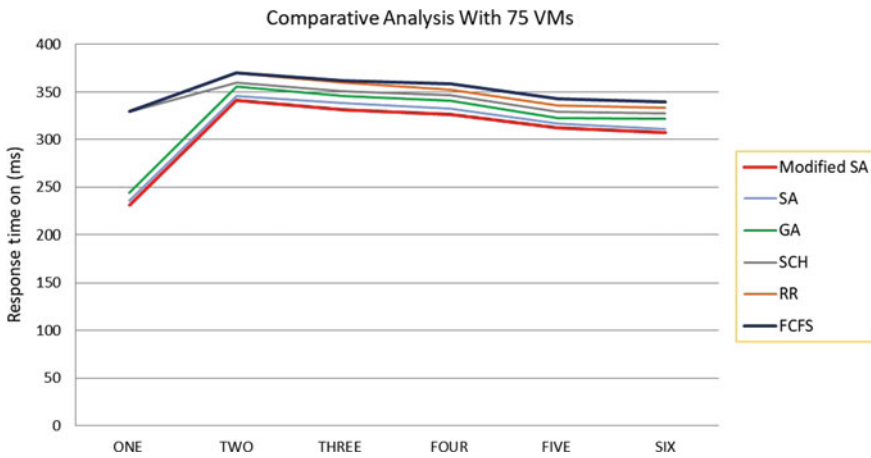


Fig. 8 Comparative analysis of the proposed LR prediction for load balancing using SA with existing Ant Colony Optimization (ACO), Genetic Algorithm (GA), SHC, and FCFS using 75 VMs and varying values of Datacenter

The proposed algorithm **UPLRegA** was compared with SA [1], GA [18], SHC [19], RR, and FCFS in case of one DC with 75 VM. The result showed that the average response got reduced time by 1.71, 5.31, 29.85, 29.92, and 29.87% as depicted in Fig. 8.

6 Conclusion and Future Work

The algorithm **UPLRegA** presented in this paper highlights a load prediction method in every physical host, based on LR technique. The proposed technique analyzes the future resource requirements from the past 1-h resource utilization and approximates the effective function which is used to predict short-term future requirement on the basis of the present requested utilization of each host. The policy is compared with some existing strategy and the result is quite encouraging. Incorporation of live migration policy has become very effective and also outperforms existing strategy those are not considered the migration time. Choosing of simulated annealing ensures the algorithm may be failed to find the best global solution but at least it can reach the most optimal or effective one. In this work, it has been considered that the all users jobs have equal priority but the social world milieu may differ, such cases need to be tested in future study. Variation of the simulated annealing or hybridization of other approaches may produce more encouraging results. Linear regression is sensitive to outliers for this reason we use multiple LR taking CPU usage, memory usage, bandwidth usage as independent variables, which can perform better than simple linear regression, combining other predictive models may produce more effective results also considering the ranking of different users requests, based upon time, throughput and other parameters may provide a better prediction which can be included as future scope of our present work.

References

1. Mandal G, Dam S, Dasgupta K, Dutta P (2018) Load balancing strategy in cloud computing using simulated annealing. In: Proceedings of the international conference on computational intelligence, communications, and business analytics. Springer, Singapore, pp. 67–81
2. Moharana SS, Ramesh RD, Powar D (2013) Analysis of load balancers in cloud computing. *Int J Comput Sci Eng* 2(2):101–108
3. Nuaimi KA, Mohamed N, Nuaimi MA, Al-Jaroodi J (2012) A survey of load balancing in cloud computing: challenges and algorithms. In: Proceedings of second symposium on network cloud computing and applications (NCCA), pp 137–142
4. Mesbahi M, Rahmani AM (2016) Load balancing in cloud computing: a state of the art survey. *Int J Mod Educ Comput Sci* 8(3):64
5. Baca DF (1989) Allocating modules to processors in a distributed system. *IEEE Trans Soft Eng* 15(11):1427–1436
6. Nathuji R, Schwan K (2007) Virtualpower: coordinated power management in virtualized enterprise systems. In: ACM SIGOPS operating systems review, vol 41, No 6. ACM, pp. 265–278
7. Liu H, Jin H, Liao X, Hu L, Yu C (2009) Live migration of virtual machine based on full system trace and replay. In Proceedings of the 18th ACM international symposium on High performance distributed computing. ACM, pp. 101–110
8. Nagarajan AB, Mueller F, Engelmann C, Scott SL (2007) Proactive fault tolerance for HPC with Xen virtualization. In: Proceedings of the 21st annual international conference on Supercomputing. ACM, pp 23–32

9. Li K, Xu G, Zhao G, Dong Y, Wang D (2011) Cloud task scheduling based on load balancing ant colony optimization. In: Proceedings of the 2011 sixth annual ChinaGrid conference. IEEE, pp 3–9
10. Alakeel AM (2010) A guide to dynamic load balancing in distributed computer systems. *Int J Comput Sci Inf Secur* 10(6):153–160
11. Escalante D, Korty AJ (2011) Cloud services: policy and assessment. *Educ Rev* 46(4)
12. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr Comput Pract Exp* 24(13):1397–1420
13. Wickremasinghe B, Calheiros RN, Buyya R (2010) Cloudanalyst: a cloudsim-based visual modeller for analysing cloud computing environments and applications. In: Proceedings of 2010 24th IEEE international conference on advanced information networking and applications (pp 446–452). IEEE
14. Das R, Kephart JO, Lefurgy C, Tesauro G, Levine DW, Chan H (2008) Autonomic multi-agent management of power and performance in data centers. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track. International foundation for autonomous agents and multiagent systems, pp 107–114
15. Seber GA, Wild CJ (2003) Nonlinear regression. Wiley series in probability and statistics. Wiley-Interscience, Hoboken, NJ
16. Farahnakian F, Liljeberg P, Plosila J (2013) LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: Proceedings of the 2013 39th euromicro conference on software engineering and advanced applications. IEEE, pp. 357–364
17. Sajitha AV, Subhajini AC (2018) Dynamic VM consolidation enhancement for designing and evaluation of energy efficiency in green data centers using regression analysis. *Int J Eng Technol* 7(3.6):179–186
18. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S (2013) A genetic algorithm (GA) based load balancing strategy for cloud computing. *Procedia Technol* 10:340–347
19. Mondal B, Dasgupta K, Dutta P (2012) Load balancing in cloud computing using stochastic hill climbing—a soft computing approach. In: Proceedings of C3IT-2012, vol 4. Elsevier, Procedia Technology, pp 783–789

Tracking Changing Human Emotions from Facial Image Sequence by Landmark Triangulation: An Incircle-Circumcircle Duo Approach



Md Nasir, Paramartha Dutta and Avishek Nandi

Abstract Intelligent recognition of human emotions from face images is a challenging proposition in the field of affective computing which becomes even more difficult when one has to deal with characterizing the nature of transition of human emotion from a relevant sequence of face images. In the present scope, we considered a triangulation mechanism derived from the landmark points of the face images. Resulting in a number of triangle formulations which are found to be sensitive to different emotions like anger, disgust, fear, happiness, sadness, and surprise. Accordingly a pair of circles, viz, incircle and circumcircle corresponding to these triangles are taken into account and geometric features arising out of such pair are utilized for classification of different emotional transitions from various face image sequences. Results of the proposed method obtained by application on various benchmark image databases are found to be quite impressive and encouraging compared to existing state-of-the-art technique.

Keywords Facial feature points (FFP) · Geometric area signature matrix (GASM) · Active appearance model (AAM) · Multilayer perceptron (MLP)

1 Introduction

Automatic facial expression recognition is a viable way to describe individual characteristics of human emotions. In [7], authors have introduced six different basic categories of human emotion: anger, disgust, fear, happiness, sadness, and surprise. In [13], authors have argued against the universal relationship between core facial expression and emotion by supplying examples of variations between different cultures. Ekman et al. have provided a framework that categorizes facial expressions such that some expressions are universal and some of those are culture specific. In the study [6], authors have found universality in the relationship between facial

Md Nasir (✉) · P. Dutta · A. Nandi
Department of Computer and System Sciences, Visva-Bharati University, Santiniketan, India
e-mail: nasir.vb@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,
Studies in Computational Intelligence 870,
https://doi.org/10.1007/978-981-15-1041-0_8

muscles and basic emotions (anger, disgust, fear, happiness, sadness, and surprise). They also mentioned in their study that there are constants natures in facial behavior across the culture, religion, sex, and for both adults and children. Nonuniversal emotional expressions are not constant in nature, they are modified by cultural factor [3]. Our research indicates the study on universal facial behavior associated with basic emotions. For a person, it is possible to sense the face and interprets actual information of facial expression but it is very challenging task for a machine to recognize facial expression accurately. In [14] it is observed that in a message communication between humans, 7% of effect of communication is contributed by verbal part of the message, 38% is contributed by vocal part and facial expression contributes 55% of effect of the message. In [8], they used geometric feature-based approach for human emotion recognition. Two different techniques they applied for facial expression recognition: one has used Support Vector Machine (SVM) as classifier and another technique has used multiclass AdaBoost with dynamic time warping on boosted feature vectors. They tested their techniques on popular image sequence database CK+ by giving a well performance. In [4], active appearance-based model is utilized to identify the features, viz, shape and texture used to track expression label. In the study [10], it is found that feature extraction play the vital role in facial expression recognition. Authors in [5, 9] they used Artificial Neural Network(ANN) as classifier to recognize human emotion with the advantage that if any element of ANN is unable to do the computation, network can continue the task without any interruption due to their parallel nature. According to [21] several approaches have been applied to static images to recognize human emotion and provided very well performances. Static-based approaches have some limitations. They are unable to retrieve real facial information properly because an emotion might be changed with respect to several time stages. Dynamic-based approaches are very useful to detect such missing information. In recent years, many several approaches [5, 8, 11, 21] have used image sequences to recognize dynamic nature of an emotion. In [5], they only focused on neutral and peak expression of image sequence to analyze the dynamic behavior of emotion. For peak expression detection from the video, they applied double local binary pattern (DLBP) and to compute effective facial feature for the expression they employed Taylor expansion theorem in their recognition system. DLBP is a variant of Local Binary Pattern (LBP) that describes the feature of local region of an image. According to DLBP algorithm it basically divide the image into different blocks first, next it computes pixel-level information in each block, after that generates histogram of each block and finally concatenates all histograms into single vector to represent an image. Unlike LBP, DLBP reduces the dimension of the feature as well as improve the recognition performance. LU et al. [11] proposed facial expression recognition system which used facial feature points to extract facial feature from image sequences and computed canonical correlation to classify a sequence into one of the basic expressions.

Motivation: Above mentioned studies motivated us to give focus on finding temporal changes of an emotion so that it can be recognized perfectly. Our proposed technique is different from [5] as authors used only neutral and peak expression on the image sequence as input to discriminate human emotions. As a result, it would

be difficult to analyze the time-variant behavior of an emotion due to missing of information related to intermediate frames. To overcome from this difficulty, we considered image sequences as input with intermediate frames including neutral face is represented by the first frame and the last frame in a sequence represents peak expression which is nothing but a basic emotion. Authors in [11, 21], they experimented their facial expression recognition method only on Extended Cohn-Kanade (CK+) image sequence database. But our proposed method implemented on different image sequence databases CK+, MUG, and MMI.

Contributions: (1) Location detection of Facial Feature points: Active Appearance Model (AAM) [18] is applied on each frame in every sequence to get landmark points on face images. Among them we only considered the important landmarks on major portions: eyebrows, eyes, nose, and mouth regions according to authors in [16]. (2) Geometric shape formation: Those crucial landmarks are taken into the account to form geometric shapes on face like triangle, circumcircle, and incircle. A triangulation mechanism used to generate the area of both circles, viz, circumcircle and incircle. (3) Generation of geometric feature: For every frame in each sequence, geometric area signatures are computed by taking the ratio between areas of both circles formed by triangulation mechanism. These area signatures are used as prominent feature vectors representing a sequence to explore the time-variant nature of an expression and those are fed into MultiLayer Perceptron (MLP) to categorize a sequence into basic expression.

This paper is organized as follows: Sect. 2 describes the proposed methodology including facial points detection, facial feature extraction, and recognition of facial expression. Experiment and results are presented in Sect. 3. Comparison of our results with other state-of-the-art method is described in Sect. 4. Finally, we draw the conclusion of our work in Sect. 5.

2 Proposed Methodology

In this study, we have proposed an effective dynamic-based approach for facial expression recognition. It has three major components to build automatic facial expression recognition system [5]. To fix the limitation of static-based approaches we used popular image sequence databases CK+, MUG, and MMI. We have taken 10 frames into the account from each sequence to analyze temporal changes of facial expression. Figure 1 shows a sequence of images taken from CK+ database. First frame contains facial information of neutral face, last frame contains information of fear expression. Among three components, [22] Feature extraction is a very powerful policy to make our Facial Expression Recognition System more robust. We have chosen geometric area-based information to retrieve features on face images. That is why we need Facial Feature Points on every frame in each sequence with geometric positions.



Fig. 1 Image sequence data start with neutral image and end with fear expression image

2.1 Facial Feature Points Detection

Due to changes in human emotions over the time periods, the movement of facial muscle points plays the vital role in detection of dynamic behavior of an emotion. We have applied Active Appearance Model for each frame of the sequence to identify the geometric location of those Facial Feature Points on face. AAM model is a combination of shape, motion, and appearance models. The main aim of this model is to compute a parametric description of a deformable object through optimization. A deformable shape instance of an object is indicated by a vector $S = [x_1, y_1, \dots, x_L, y_L]$ of size $2L$. This vector contains L landmark points coordinates (x_i, y_i) where $i = 1, 2, \dots, L$. It uses N number of training images I_1, I_2, \dots, I_N for annotating all L landmark points. The training shapes $[S_1, S_2, \dots, S_N]$ are obtained by using Procrustes Analysis to get N -similarity free shapes through discarding similarity transformation from original image. After that by using Principal Component Analysis (PCA) it computes mean shape and n shape eigenvectors \bar{s} and U_s , respectively. Finally, it computes new shape instance with parameter P by using formula $S_p = \bar{s} + U_s P$, here $P = [p_1, p_2, \dots, p_n]$. It generates common shape from newly generated shape instance by warping the texture associated with the shape. Next, holistic feature function, i.e., $F(I_i)$, where $i = 1, 2, \dots, N$ is used for extracting features from training images and reference shape is created by warping those features. A vector of warped images is calculated as $a_i = F(I_i)(W(p_i)), \forall i = 1, \dots, N$. By using PCA

on resultant vector, it calculates mean appearance vector \bar{a} and n appearance eigenvectors U_a . At last a new appearance instance a_c is computed by augmenting mean appearance vector \bar{a} with eigenvectors U_a , here $a_c = \bar{a} + U_a c$ mathematical formula is used, where $c = [c_1, c_2, \dots, c_m]$ are appearance vector parameters. We got 68 feature points that cover entire face for each frame of the sequence. Through analyzing every sequence we have noticed that only 23 points among them are very sensitive with respect to facial expressions [17]. Over the frames, movement of other feature points is very negligible. They have no significant role in changing the expression from neutral to one of the basic emotions: anger, disgust, fear, happiness, sadness, and surprise. Those 23 points are selected from major portions of the face. Three points are taken from left eyebrow, three points are from right eyebrow, four points are taken from left eye, four points are from right eye, three points chosen from nose, and six points are selected from mouth region.

2.2 Facial Feature Extraction

Our proposed method extracts facial features from face image sequence by using geometric area signature. We have considered 10 frames from the sequence of various basic expressions as input to detect the changes in emotional transition from neutral to other basic expressions. We have used 23 facial informative landmark points for every frame to construct triangle shape on face. By taking every combination of three landmark points from them, we have formed total $n = 1771$ triangles for the neutral image which is start frame of the sequence. For each triangle we have extracted area of circumcircle and incircle as geometric feature vector of length n . In this way, all possible triangles and feature vectors are generated for remaining frames in the sequence up to the basic expressions. Finally we got the complete feature vector of size $m \times n = 10 \times 1771 = 17710$ by concatenating each frame feature vector. Here $m = 10$ is number of frames per sequence and the complete feature vector represents the sequence. Figure 2 shows formation of incircle and circumcircle from landmarks.

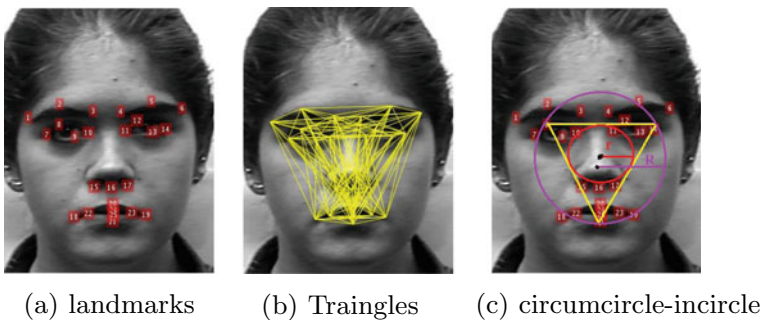


Fig. 2 Circumcircle and incircle formation from landmarks

A. Formation of Geometric Area Signature Matrix

We have computed three side lengths a , b , and c of the triangle by using Euclidean distance between possible pairs of every three landmark coordinates. Then we have calculated perimeter of the triangle $p = a + b + c$, $s = p/2$ and area of triangle

$$\Delta_i = \sqrt{s(s-a)(s-b)(s-c)} \quad (1)$$

Next, we have determined the radiuses of incircle (r) and circumcircle (R) of the triangle by the following formulas:

$$r_i = \frac{\Delta_i}{s} \quad (2)$$

and

$$R_i = \frac{a \times b \times c}{4 \times \Delta_i} \quad (3)$$

We have chosen area of circumcircle (κ) and incircle (τ) as the geometric feature and areas of both circles are calculated as follows:

$$\kappa_i = \pi R_i^2 \quad (4)$$

and

$$\tau_i = \pi r_i^2 \quad (5)$$

Each frame is represented by the normalized feature vector computed as below

$$\vartheta_i = \frac{\kappa_i - \tau_i}{\kappa_i} \quad (6)$$

Finally, geometric shape signature matrix is generated by concatenating all feature vectors of the sequence like as

$$GASM_k = \sum_{m=1}^{10} \sum_{n=1}^{1771} \vartheta_n^m \quad (7)$$

Here, k is used to denote a specific sequence and a particular triangle is denoted by i .

2.3 Recognition of Facial Expression

We have used MultiLayer Perceptron as a classifier to recognize facial expression from image sequence data. It is a three-layer neural network [2] consisting one input

layer, one hidden layer, and one output layer. Size of input neuron depends on size of feature vector and number of output neurons is related to the number of facial expressions to be recognized.

MLPs are utilized for supervised learning problems with the capability to approximate nonlinear functions, they train on a set of input–output pairs and learn to model the correlation between those inputs and outputs. It is a feedforward network that works in two passes, wherein the forward pass, input signal comes from the input layer and it is passed through the hidden layers to the output layer, and the decision of the output layer is measured by comparing its outcomes with known outcomes. In the backward pass, backpropagation is used to adjust weights and biases repeatedly to get minimum error. Here, activation function is used for passing the outputs to next layer and it is formulated by $y = \phi(\sum_{i=1}^l w_i x_i + b)$ where w is the weights vector, x is the inputs vector, b is the bias and ϕ is the nonlinear activation function.

An arbitrary number of hidden layers that are the actual computational engine of the MLP and it is one of the various dependencies of network performance to get better recognition results. In the implementation of our proposed method on different databases, it is observed that usage of 10 hidden neurons in the hidden layers gives better results than usage of other number of hidden neurons.

We have used 10 hidden neurons to take input signal from input layer and passes its output signal to the output layer to classify emotions. Scaled Conjugate Gradient backpropagation is used as a network training function that updates weight and bias values according to the scaled conjugate gradient method. Network gives the best performance on expression recognition by minimizing Mean Squared Error (MSE). MSE is a statistical estimator that computes the difference between predicted output and actual output. The formula of MSE computation is

$$e_k = \frac{1}{2} \sum (t_k - y_k)^2 \quad (8)$$

where k is number of output neurons, t denotes actual output and y denotes predicted output. Algorithm 1 shows the network training process.

3 Experiment and Results

For the verification of our proposed method, it is experimented on popular benchmark image sequence databases: Extended Cohn-Kanade (CK+), MUG, and MMI. We have utilized Dlib Shape Predictor Model implementation library for 68 landmark points of FAST-AAM for Python code available on GitHub link “<https://github.com/davisking/dlib-models>” [15]. We have considered those databases for classification of image sequences into different basic classes in following order: 70% of total sequences are used for training phase, 15% are taken for validation phase, and 15% are reserved for testing phase. Here 15% of total size of dataset is held out as validation dataset for detecting the effect of overfitting of MLP classifier. During the training

Algorithm 1: Network Training Algorithm

Input: Image Sequence data D.

Output: Emotion recognized into basic expressions.

```

1 begin
2 for each sequence s in D
3 for each frame f in s
4 Extract landmark points from f;
5 Load those points into single array A;
6 for every three points in A
7 Generate triangle;
8 Compute area of incircle & circumcircle;
9 Normalize area values;
10 Store normalized area values into feature vector v;
11 end
12 Store vector v into GASM;
13 end
14 Concatenate each column in GASM into single vector;
15 Store this single vector into SCORE Matrix;
16 end
17 SCORE Matrix is fed into MLP;
18 end

```

periods of the classifier model, those held out dataset is used to check the improvement of results by comparing the accuracy derived at validation phase with the accuracy obtained from the dataset used in training phase. If no improvement is occurred, then it stop the training process immediately and following which the model can be generalized. In this way overfitting of the MLP model is overruled. We have earmarked 10 frames for every image sequence start with neutral image and end with basic expression. Geometric features extracted from both databases are fed into MLP to categorize emotions and it provides recognition results after network is trained 50 times. Figure 3 shows the recognition rates of our method under training, validation, and testing phases implemented on CK+, MUG, and MMI databases and Fig. 4 shows the graphical representation of the emotions classified for all three datasets CK+, MMI, and MUG.

Details analysis of results on CK+, MUG, and MMI databases are given in Sects. 3.1, 3.2 and 3.3 respectively.

3.1 Results on CK+ Database

In [12], it is found that 327 of 593 image sequences from CK+ database were labeled with seven emotions. They are anger (AN), contempt (CON), disgust (DI), fear (FE), happiness (HA), sadness (SA), and surprise (SU). This dataset contains both posed and non-posed facial expressions of 210 adults captured by Panasonic Ag-7500 video recorder with Horita synchronized time-code generator. They were 18–50 years of

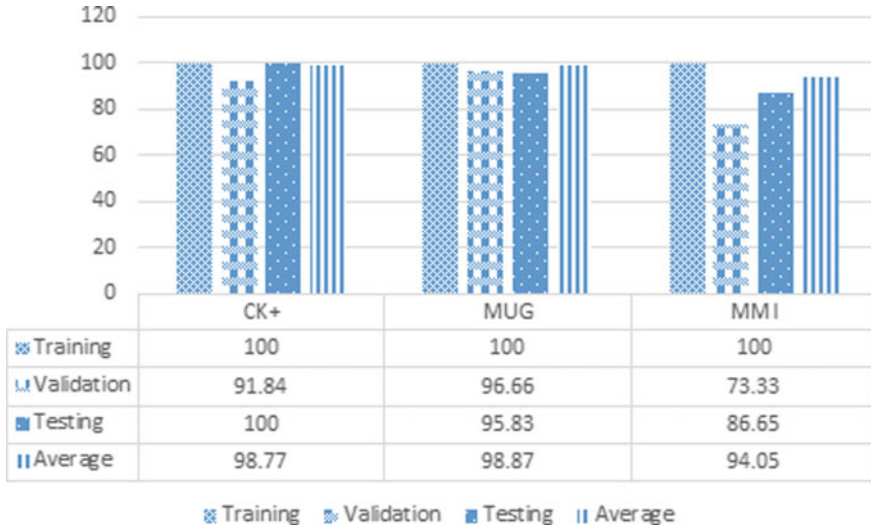


Fig. 3 Comparison in different recognition results

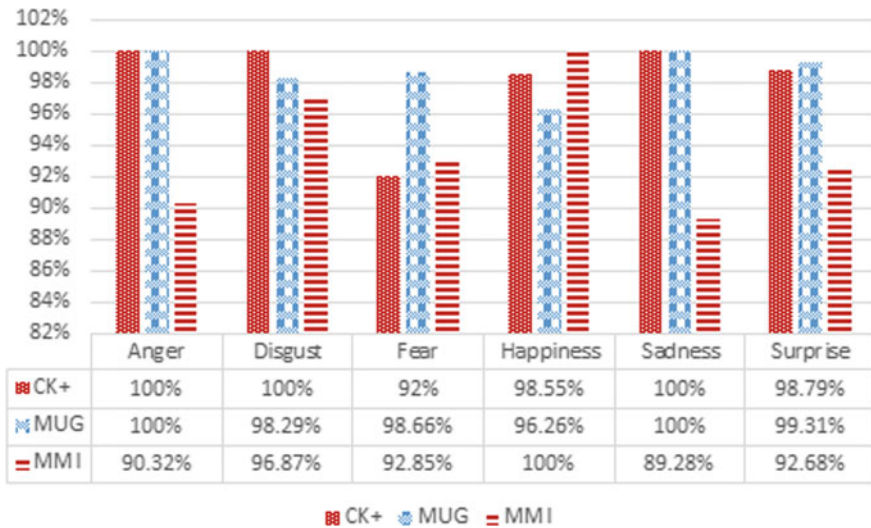


Fig. 4 Graphical representation of the emotions classified for all three datasets CK+, MMI, and MUG

age, among them 69% female, 81%, Euro-American, 13% Afro-American, and 6% other groups. It contains the following number of sequences of individual expression anger (45), contempt (18), disgust (59), fear (25), happiness (69), sadness(28), and surprise (83). Our method performed on CK+ database and it gives 98.77% as average recognition rate. Table 1 shows confusion matrix generated by our approach

Table 1 Confusion matrix on CK+ database

	AN	CON	DI	FE	HA	SA	SU
AN	45	0	0	0	0	0	0
CON	0	18	0	0	0	0	0
DI	0	0	59	0	0	0	0
FE	0	0	0	23	1	1	0
HA	0	1	0	0	68	0	0
SA	0	0	0	0	0	28	0
SU	0	1	0	0	0	0	82

Table 2 The recall, precision, and F-score of area signature on CK+ database using MLP

	Recall	Precision	F-score
AN	1	1	1
CON	1	0.9	0.95
DI	1	1	1
FE	0.92	1	0.96
HA	0.99	1	0.99
SA	1	0.97	0.98
SU	0.99	1	0.99

on this database where the rows correspond to the predicted class (Output Class), and the columns show the actual class (Target Class). We have computed the Table 1 by considering diagonal elements of the matrix as number of image sequences are properly classified and elements of other cell as number of misclassification of image sequences. Anger, contempt, disgust, and sadness these image sequences are individually recognized with 100% recognition rate. In case of fear image sequences, they are classified with 92% classification rate. Happiness and surprise, these sequences are recognized with 98.55% and 98.79% recognition rate, respectively. Table 2 shows the sensitivity of our method in classification by measuring Recall, Precision, and F-score computed from confusion matrix on CK+ database. Recall is formulated by $Recall = \frac{TP}{TP+FN}$, Precision is computed by $Precision = \frac{TP}{TP+FP}$, and F-score is calculated by $F - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$. Here TP(True Positive) = expressions are correctly identified, FP(False Positive) = expressions are incorrectly identified as actual class and FN(False Negative) = expressions are incorrectly identified into other classes except actual class. Table 3 shows the sample pictorial representation of emotions classified for CK+ dataset.

Table 3 Pictorial representation of emotions classified on CK+







Image	Actual class	Predicted class	Outcome
	Anger	Anger	Properly classified
	Disgust	Disgust	Properly classified
	Fear	Happiness	Misclassified
	Happiness	Happiness	Properly classified
	Sadness	Sadness	Properly classified
	Surprise	Surprise	Properly classified

Table 4 Confusion matrix on MUG database

	AN	DI	FE	HA	SA	SU
AN	149	0	0	0	0	0
DI	0	115	0	0	0	2
FE	1	0	148	0	0	1
HA	4	0	0	103	0	0
SA	0	0	0	0	133	0
SU	0	1	0	0	0	144

Table 5 The recall, precision, and F-score of area signature on MUG database using MLP

	Recall	Precision	F-score
An	1	0.97	0.98
DI	0.98	0.99	0.98
FE	0.99	1	0.99
HA	0.96	1	0.98
SA	1	1	1
SU	0.99	0.98	0.98

3.2 Results on MUG Database

MUG database [1] consists of 801 image sequences with six basic expressions: anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA), and surprise (SU). All such sequences are taken from 86 different subjects, among them 35 women and 51 men participated. They are of 20–35 years age. Videos are captured by the camera with rate of 19 frames per second. Subjects are not wearing glasses and no occlusions are there except for a few hair falling on the face.

Table 4 shows confusion matrix on MUG database where the rows correspond to the predicted class (Output Class), and the columns show the actual class (Target Class). Here anger and sadness, these are perfectly recognized with 100% recognition rate. Disgust sequences are identified with 98.29% recognition rate. Fear is classified with 98.66% classification rate. Happiness and surprise are recognized with 96.26% and 99.31% recognition rate, respectively. Overall performance of our technique on this database is achieved with average recognition rate 98.87%. Further, performance justification of our proposed method on MUG database is shown in Table 5. Table 6 shows the sample pictorial representation of emotions classified for MUG dataset.

Table 6 Pictorial representation of emotions classified on MUG

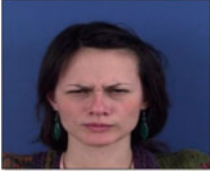
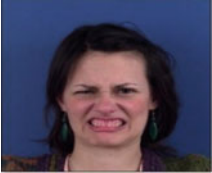
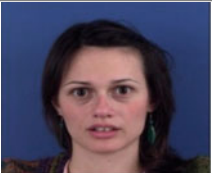

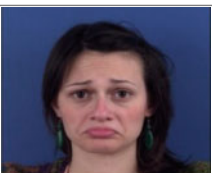
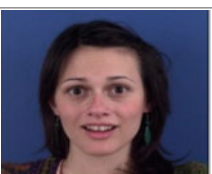
Image	Actual class	Predicted class	Outcome
	Anger	Anger	Properly classified
	Disgust	Surprise	Misclassified
	Fear	Anger	Misclassified
	Happiness	Anger	Misclassified
	Sadness	Sadness	Properly classified
	Surprise	Surprise	Properly classified

Table 7 Confusion matrix on MMI database

	AN	DI	FE	HA	SA	SU
AN	28	1	0	0	2	0
DI	1	31	0	0	0	0
FE	0	0	26	1	0	1
HA	0	0	0	42	0	0
SA	2	1	0	0	25	0
SU	0	0	3	0	0	38

Table 8 The recall, precision, and F-score of area signature on MMI database using MLP

	Recall	Precision	F-score
AN	0.9	0.9	0.9
DI	0.97	0.94	0.95
FE	0.93	0.89	0.91
HA	1	0.98	0.99
SA	0.89	0.93	0.91
SU	0.93	0.97	0.95

3.3 Results on MMI Database

Experimentation on MMI database [20] is very challenging task for facial expression analysis compared to CK+ and MUG databases. Our proposed method is implemented on this database by considering 202 image sequences with six basic emotions: anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA), and surprise (SU). This dataset consists of expressions of both frontal images and side view images and data are organized in session units of audiovisual recording. Total 238 recording clips are taken from 28 different subjects and all expressions are captured by two attempts, one is wearing glasses and another is without wearing glasses. Experimental results on MMI database are obtained by considering frontal images for every sequence. Table 7 shows the confusion matrix generated by implementing our approach on MMI database where the rows correspond to the predicted class (Output Class), and the columns show the actual class (Target Class).

Our proposed system achieved average recognition rate 94.05% on MMI database with maximum recognition rate 100% in case of happiness and minimum recognition rate 89.28% in case of sadness. Table 8 shows the Recall, Precision, and F-score of area signature of MLP classifier. Table 9 shows the sample pictorial representation of emotions classified for MMI dataset.

Table 9 Pictorial representation of emotions classified on MUG

Image	Actual class	Predicted class	Outcome
	Anger	Sadness	Misclassified
	Disgust	Anger	Misclassified
	Fear	Surprise	Misclassified
	Happiness	Happiness	Properly classified
	Sadness	Anger	Misclassified
	Surprise	Surprise	Properly classified

Table 10 Comparison with existing method [21] using CK+ Database

Input	Method	Feature type	Recognizer	Avg recognition rate
Sequence database	Yaddaden et al. [21]	Variance-based feature	KNN	88.34%
		Distance-based feature	SVM	92.54%
Sequence Database	Our approach	Circumcircle-Incircle area-based feature	MLP	98.77%

4 Comparison of Our Results with Other State-of-the-Art Method

We have compared the discrimination power of our method with [21] by investigating on CK+ database. In [21], 309 sequences are used for classification with six basic expressions excluding contempt expression and 10–60 frames of each sequence are utilized by them. But in our approach, 327 sequences are considered with seven expressions including contempt expression and we have utilized exactly 10 frames for every sequence. Authors in [21], they introduced two geometric based approaches. One is classification through SVM classifier with distance-based features achieved average classification rate 92.54% and another approach is classification through KNN recognizer with variance-based features reached 88.34% average accuracy. Our proposed method obtained better performance than these two procedures which has reached 98.77% as average classification rate. Table 10 shows that our method performs better than existing method Yaddaden et al. [21]

5 Conclusion

In this article, we observed that the best representation of image sequences by extracting powerful discriminating geometric features [19] arising out of the areas of incircle-circumcircle pair applied on triangles derived from landmark triangulation. The efficiency of our geometric shape-based method is vindicated by comparing the outcomes with other existing method [21]. Analysis of results on different image sequence databases provides potential insights on emotional transition of facial expression. Recognition rate computed by our proposed procedure encourages us to find out optimum features for every sequence in order to ensure improved accuracy.

Acknowledgement The authors like to avail this opportunity to express their gratitude to Dr. A. Delopoulos for providing MUG database and Prof. Maja Pantic for providing MMI database free

for carrying out this work. The authors would also like to thank the Department of Computer and System Sciences, Visva-Bharati, Santiniketan for the infrastructure support. The authors gratefully acknowledge the support of DST-INSPIRE Fellowship (INSPIRE Reg. no. IF160285, Ref. No.: DST/INSPIRE Fellowship/IF160285) for pursuing Doctoral Research in Department of Science and Technology, Ministry of Science and Technology, Government of India.

References

1. Aifanti N, Papachristou C, Delopoulos A (2010) The mug facial expression database. In: Proceedings of the 11th international workshop on image analysis for multimedia interactive services WIAMIS 10. IEEE, pp 1–4
2. Boughrara H, Chtourou M, Amar CB, Chen L (2016) Facial expression recognition based on a mlp neural network using constructive training algorithm. *Multimed Tools Appl* 75(2):709–731
3. Collier G (1985) *Emotional expression*. Laurence Erlbaum Associates Inc, Publishers, Hillsdale, NJ
4. Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. *IEEE Trans Pattern Anal Mach Intell* 6:681–685
5. Ding Y, Zhao Q, Li B, Yuan X (2017) Facial expression recognition from image sequence based on lbp and taylor expansion. *IEEE Access* 5:19409–19419
6. Ekman P (2004) Emotions revealed. *BMJ* 328(Suppl S5):0405184
7. Ekman P, Friesen WV (1971) Constants across cultures in the face and emotion. *J Personal Soc Psychol* 17(2):124
8. Ghimire D, Lee J (2013) Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines. *Sensors* 13(6):7714–7734
9. Goyal R, Mittal T (2014) Facial expression recognition using artificial neural network. *HCTL Open Int J Technol Innov Res* 10:1–10
10. Lajevardi SM, Lech M (2008) Facial expression recognition from image sequences using optimized feature selection. In: Proceedings of the 2008 23rd international conference image and vision computing New Zealand. IEEE, pp 1–6
11. Lu K, Zhang X (2010) Facial expression recognition from image sequences based on feature points and canonical correlations. In: Proceedings of the 2010 international conference on artificial intelligence and computational intelligence, vol 1. IEEE, pp 219–223
12. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: Proceedings of the 2010 IEEE computer society conference on computer vision and pattern recognition-workshops. IEEE, pp 94–101
13. Manstead AS, Fischer AH (2002) Beyond the universality-specificity dichotomy. *Cogn Emot* 16(1):1–9
14. Mehrabian A (2007) *Nonverbal communication*. New Brunswick
15. Sagonas C, Antonakos E, Tzimiropoulos G, Zafeiriou S, Pantic M (2016) 300 faces in-the-wild challenge: database and results. *Image Vis Comput* 47:3–18
16. Samal A, Iyengar PA (1992) Automatic recognition and analysis of human faces and facial expressions: a survey. *Pattern Recognit* 25(1):65–77
17. Tie Y, Guan L (2013) Automatic landmark point detection and tracking for human facial expressions. *EURASIP J Image Video Process* 2013(1):8
18. Tzimiropoulos G, Pantic M (2013) Optimization problems for fast AAM fitting in-the-wild. In: Proceedings of the IEEE international conference on computer vision, pp 593–600
19. Uddin MZ, Hassan MM, Almogren A, Alamri A, Alrubaian M, Fortino G (2017) Facial expression recognition utilizing local direction-based robust features and deep belief network. *IEEE Access* 5:4525–4536

20. Valstar M, Pantic M (2010) Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In: Proceedings of the 3rd international workshop on EMOTION (satellite of LREC): corpora for research on emotion and affect, Paris, France, p 65
21. Yaddaden Y, Adda M, Bouzouane A, Gaboury S, Bouchard B (2017) Facial expression recognition from video using geometric features
22. Zhang L, Tjondronegoro D (2011) Facial expression recognition using facial movement features. *IEEE Trans Affect Comput* 2(4):219–229

Recognizing Human Emotions from Facial Images by Landmark Triangulation: A Combined Circumcenter-Incenter-Centroid Trio Feature-Based Method



Avishek Nandi, Paramartha Dutta and Md Nasir

Abstract Human emotion reflected in facial expression is generated by coordinated operation of muscular movement of facial tissue which associates with the emotional state of the human subject. Facial expression is one of the most significant non-articulated forms of social communication and it is highly adopted by scientific community for the purpose of automated emotion analysis. In the present scope, a triangular structure is induced with three points, viz., circumcenter, incenter, and centroid are considered as the geometric primitive for extraction of relevant features. Information extracted from such features is utilized for the purpose of discrimination of one expression from another using MultiLayer Perceptron (MLP) classifier in images containing facial expressions available in various benchmark databases. Results obtained by applying this method found to be extremely encouraging.

Keywords Active appearance model (AAM) · Triangulation · Facial expressions · Circumcenter-incenter-centroid trio feature · Multilayer perceptron (MLP)

1 Introduction

Expression recognition in human face has much significance in the field of affective computing from intelligent emotional assistance, interest estimation for e-learning system to criminal tendency estimation [25]. The human face is capable of generating 7,000 different kinds of facial expression, only Six of these expressions are recognized by behavioral scientist Ekman at el, namely, Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), and Surprise (SU) as “atomic expressions” [14, 15]. They also proved that these six expressions are unique among different

A. Nandi (✉) · P. Dutta · Md Nasir
Visva-Bharati University, Santiniketan 731235, India
e-mail: avisheknandi10@gmail.com

Md Nasir
e-mail: nasir.vb@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,
Studies in Computational Intelligence 870,
https://doi.org/10.1007/978-981-15-1041-0_9



Fig. 1 Reference image for each of the basic emotions of faces

racess, religions, cultures, and groups [13]. These six basic expressions are portrayed in Fig. 1.

The universality of the basic facial expressions and emotion is argued in [6, 11, 22]. Ekman et al. tried to settle the issue of universality dichotomy in their research facial expression across different cultures in [14] but they also mentioned that the universality phenomena of basic emotions are only valid when subjects are having strong emotional behavior. In the strong emotional condition, the facial expressions are not masked by the cultural influences and the expressions are static across religion, sex, race, culture, and educational background. But according to Russell, the facial expressions are sensitive to the context of the situation in which expressions are displayed [26]. But our research follows the universality of facial expressions which is argued by the Ekman and well accepted in the research community. This paper is organized as follows, apart from the first introductory section, Sect. 2 contains a survey of relevant literature. Section 3 emphasizes the motivation and contribution of our work. Section 4 discusses about the methodologies along with the essential flow diagram and algorithm. Landmark points generated by Active Appearance Model (AAM), the landmark selection and triangulation formation are emphasized in Sects. 4.1, 4.2 and 4.3, respectively. The circumcenter-incenter-centroid trio feature descriptor, MultiLayer Perceptron (MLP), and Classification Learning are discussed in Sects. 4.4, 4.5 and 5, respectively. Results and comprehensive analysis with respect to CK+, JAFFE, MMI, and MUG databases are reported in Sects. 6 and 7, respectively. Section 8 figures overall conclusions.

2 Literature Survey

Some recent studies of automatic facial expression recognition include the following researches. The work of Happy and Routray introduced salient facial patches to differentiate one expression to another; they also have used their own learning free landmark localization method for detection of facial landmarks in a robust and autonomous way [17]. In contrast to that Almaev and Valstar used local Gabor patterns to extract local dynamic features for detecting facial action units in a real-time manner in [2]. Another significant work of Yuan et al. introduces a hybrid combination of Local binary Patterns (LBP) with Principal Component Analysis (PCA) features which describes local and holistic facial features in a fused manner [30]. The Histogram oriented Gradients (HoG) features of facial components also explored by Chen et al. which are used to detect deformation features respective to facial expressions [9]. Martin et al. used Active Appearance Model (AAM) features of gray scale and edge images to achieve a greater robustness in a varying light condition [23]. The work of Cheon and kim introduce a differential AAM feature which is the computation of the Directed Hausdorff Distance (DHD) in between neutral face image and Excited face image with K-Nearest Neighbor (KNN) classifier [10]. Another interesting work by Barman and Dutta also uses AAM features to compute shape and distance signature along with statistical features to boost up expression recognition performance [3]. They also have extended their work of distance signature generated out of AAM landmark points with newly introduced texture the signature generated out of salient facial patched localized by AAM landmarks along with stability index [4, 5]. Most of the facial expression recognizer uses MultiLayer Perceptron (MLP), Radial Basis Function Network(RBF), Support Vector Machine (SVM) for the purpose of classification of facial expressions [8, 16, 19]. In comparison with MLP the fuzzy MLP classifier performs better because it can identify decision surfaces in case of nonlinear overlapping classes, whereas an MLP is restricted to crisp boundaries only [7].

3 Motivation and Contribution

The methods mentioned previously have a major drawback that it does not effectively classify facial expressions in a robust manner. To overcome this issue, we proposed a circumcenter-incenter-centroid trio feature as a more accurate shape descriptor in this context which is able to recognize facial expressions in a person independent manner. The circumcenter-incenter-centroid trio feature shows that it inherently captures the person independent information and also ensures good accuracy in different groups and ages of people.

The contribution of the present article has the following points of merits.

- An effective triangulation formation on the face is proposed.
- A novel feature descriptor based on circumcenter, incenter, and centroid of a face triangulation is introduced.
- A good person independent expression recognition with distance slope measure of circumcenter-incenter-centroid trio feature is achieved.

4 Methodology

The flow of the computation involves image space and feature space operations as depicted in Fig. 2. The algorithm with steps of computations is given below.

Algorithm 1: Proposed Training Algorithm for Facial Expression Recognition

- 1 INPUT: A set of N expression images of size $n \times m$.
 - 2 STEP1: Extract the Landmark points using AAM fitting algorithm.
 - 3 STEP2: Select the salient landmark points $L = [(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)]$
 - 4 STEP3: Generate the set of Triangulation T using Landmark points, where
 $T = [(x_i, y_i), (x_j, y_j), (x_k, y_k)]$ and $i, j, k \in [1, L]$.
 - 5 STEP4: for t_i in T Compute three triangle center
 $p_{ijk} = [(p_1, p_2), (p_3, p_4), (p_5, p_6)]$
 - 6 Compute distance triplet $D_{ijk} = [d_1, d_2, d_3]$ and slope triplet $S_{ijk} = [s_1, s_2, s_3]$
as
 - 7 $d_1 = \text{dist}((p_1, p_2), (p_3, p_4))$
 - 8 $d_2 = \text{dist}((p_3, p_4), (p_5, p_6))$
 - 9 $d_3 = \text{dist}((p_1, p_2), (p_5, p_6))$
 - 10 $s_1 = \text{slope}((p_1, p_2), (p_3, p_4))$
 - 11 $s_2 = \text{slope}((p_3, p_4), (p_5, p_6))$
 - 12 $s_3 = \text{slope}((p_1, p_2), (p_5, p_6))$
 - 13 STEP5: Combine distance D and slope S to get enhanced feature set.
 - 14 STEP6: Divide the dataset in 70%, 15% and 15% ratio for the purpose of training, validation and testing.
 - 15 STEP7: Train MLP classifier with training data and use validation data to overcome overfitting.
 - 16 STEP8: Test the Classifier on the basis of test database.
 - 17 OUTPUT: Classified expression level for input image database.
-

4.1 Active Appearance Model (AAM)

AAMs are shape predictor model works by optimizing the geometric parameters of deformable bodies of a class of shape models [12]. We have used AAM by

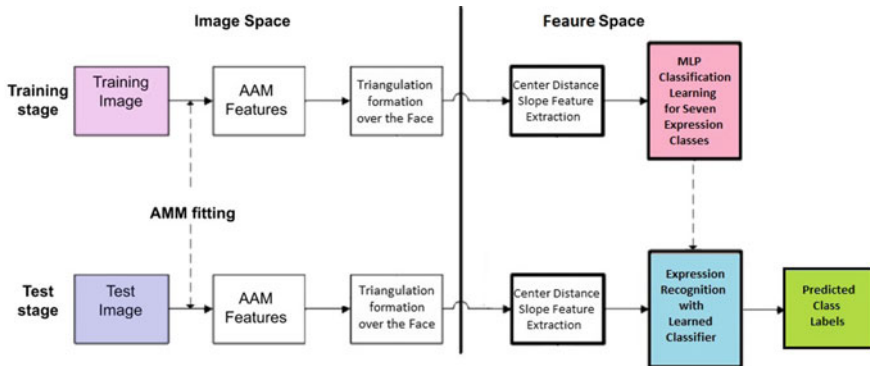


Fig. 2 Image space and feature space flow of computation

Tzimiropoulo et al for generating face description by inducing sixty-eight landmark points on the face [27, 28]. Python and Dlib implementation of [27] can be found on “<https://github.com/davisking/dlib-models>”. We have used Python implementation of [27] for our research.

A deformable shape object can be expressed as $S = [(x_1, y_1), \dots, (x_L, y_L)]^T$, a L element vector comprising L landmark coordinate points $(x_i, y_i), \forall i = 1, \dots, L$. The AAM model is trained with a manually annotated set of N training images I_1, I_2, \dots, I_N where each image consisting of L landmark points. The training of AAM has four steps of computation.

1. First holistic features are extracted using the $F()$ operator, i.e. $F(I_i), \forall i = 1, \dots, T$.
2. Warping of extracted features from candidate image I_i according to the reference shape by $W()$ operator, i.e. $F(I_i)(W(s_i)), \forall i = 1, \dots, N$. Where vector shape parameters are defined as $a = [a_1, a_2, \dots, a_n]^T$.
3. The Warped images are vectorized as $a_i = F(I_i)(W(s_i)), \forall i = 1, \dots, N$ where $a_i \in \mathbb{R}^{M \times 1}$.
4. Finally, PCA is computed on the extracted vectors generating

$$\{\bar{a}, U_a\} \quad (1)$$

where \bar{a} is mean appearance vector and U_a is orthonormal basis eigenvectors. $a_c = \bar{a} + U_a c$ is the new appearance model instance where $c = [c_1, c_2, \dots, c_m]$ are appearance vector parameters.

4.2 AAM Landmark Selection

The landmarks generated by AMM describe geometrical positions and the shape of facial components. We have selected 21 principal landmark points as Barman and Dutta identified these points as salient landmark points in their research [3]. These points are mainly corner points and mid-points of facial components. The selected principal landmark points are as follows—two corner points and one midpoint on both eyebrows, two corner points of eyes and two middle points of eyelids, two corner points on nostril and one on middle of nose, four points on outer lips region and four on inner lips region as depicted on Fig. 3a. We left out outer points intentionally because those points are very less sensitive toward expressional changes.

4.3 Triangulation

The triangulation structure is formed by fixing three pivot points from the set of 21 points $\gamma = [(x_1, y_1), (x_1, y_1), \dots, (x_{21}, y_{21})]$. Triangulation is formed using the formula (2).

$$\delta = \langle \sigma_i, \sigma_j, \sigma_k \rangle \quad (2)$$

where σ_i, σ_j and σ_k are $\in \gamma$ and $\sigma_i \neq \sigma_j \neq \sigma_k$. The possible number of triangulation using 21 points is $\binom{21}{3} = 1330$. Figure 3b depicts the formed triangulation constituting principal landmark points.

It is needless to mention that the shape information of the triangulations is sensitive toward the expressional variation on the face. The shape of a triangle is highly correlated to geometrical positions of different types of centers. We have considered three types of classical triangle center as centroid, incenter, and circumcenter for this work. Triangle centers have this property that they are immune to similarity transformation (rotation, reflection, and translation) so only the change of shape is reflected irrespective of size and position of the triangle.

4.4 Circumcenter-Incenter-Centroid Trio

We have considered three types of triangle centers as Centroid, Incenter, and Circumcenter of to form the triangulation.

- **Centroid** of a triangle is the “Center of Gravity” point of the triangle and also the single intersection point of three lines bisecting in middle point of each side.
- **Incenter** of a triangle is the meeting point of three angel bisector and also the center of incircle of the triangle.
- **Circumcenter** of a triangle is the point of congruence of three side normal and also the center of circumcircle encompassing the triangle.

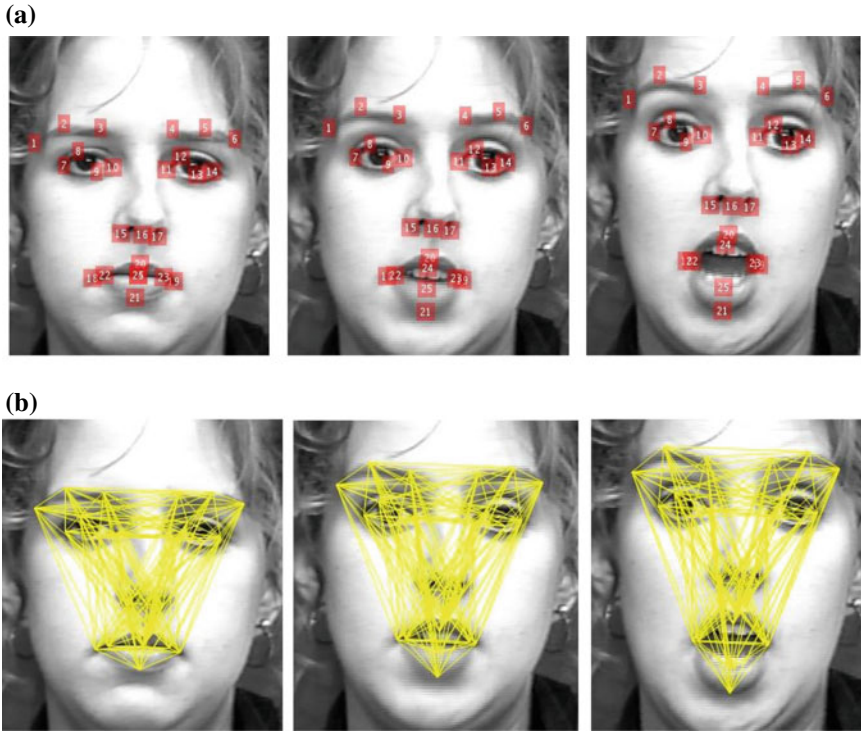


Fig. 3 Principal Landmark selection and Triangle formation Example. **a** Principal landmark points plotted on Face. **b** Triangulation formation

Incenter and centroid of a triangle always remain inside the triangle whereas circumcenter may get outside for obtuse triangles. The features we have considered are three types of distance and three types of slope vectors originating from centroid-incenter, centroid-circumcenter, and incenter-circumcenter pair of the centers of a triangle. These three distances and three slope features are combined to make a six-feature set and which are computed for each triangle in the triangulation set δ of Eq. 2.

4.5 MultiLayer Perceptron

MultiLayer perceptron is a feedforward neural network having at least three layers one input layer, one or more hidden layer, and one output layer. MLP is learned with a backpropagation algorithm which uses supervised methods of learning. MLP uses nonlinear synaptic activation function to learn the properties of high dimensional data. The sigmoid $y(v_i) = \frac{1}{(1+e^{-v_i})}$ and hyperbolic tangent $y(v_i) = \tanh(v_i)$ activation function is most popular activation function in the literature. Layers of MLP are

interconnected with a weight matrix W where w_{ij} is the weight connecting i 'th node of current layer to the j 'th node of the following layer.

Backpropagation learning of MLP is done by changing the connection weight after processing each single input vector on basis of error computed in the output layer. The error is formulated as $e_j(n) = d_i(n) - y_j(n)$ where d is target value and y the value computed by the MLP and this error is backpropagated to output layer to input layer. Connection weights w_{ij} is adjusted to minimize the error $\eta(n) = \frac{1}{2} \sum e_j^2(n)$. The change of weight is computed using Eq. 3

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \quad (3)$$

η is the learning rate and y_i output of the previous neuron. The output layer weight are updated using the formulae (4) from Chap. 4 of the book [18].

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \phi'(v_j(n)) \quad (4)$$

The derivative of activation function is ϕ' . The hidden layer weights are updated using the formulae (5) from Chap. 4 of the book [18].

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n) \quad (5)$$

5 Classification Learning

These three-distance feature and three-slope feature are combined and learned with MultiLayer Perceptron (MLP) with scale conjugate backpropagation learning for classification of expressions to six different atomic expression classes [24]. We have used MATLAB implementation of [24] is used for the training the model for the purpose of pattern recognition and classification learning. The 70% of dataset is used for training purposes, 15% of the dataset is used for testing purposes and the rest of the 15% of the dataset is used for validation checking. Validation dataset used to overrule the overfitting of MLP classifier at the time of training.

6 Results

We have tested the proposed machine with four well-known expression database, the CK+ [20], JAFFE [21], MMI [29] and MUG [1] database. The computing environment we have used is Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz with 4 GB RAM. Dlib Shape Predictor Model implementation library for 68 landmark points

of AAM for Python can be found on GitHub link “<https://github.com/davisking/dlib-models>” [27] used for landmark extraction. The AAM model achieved an 100% detection accuracy in CK+, JAFFE, MUG database and 99.2% in MMI database.

6.1 Extended Cohn-Kanade (CK+) Database

The CK+ expression dataset is a combination of posed and spontaneous expressions having 327 image sequences from neutral to peak expression. This dataset is consist of six basic expressions Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), Surprise (SU) and also Contempt (CO) expression. Only the peak expressions are selected form this database. The proposed MLP classifier achieved a 99% overall accuracy in this database. We have also computed N-Fold accuracy for training, testing, validation, and overall dataset reflected in Fig. 4. Some example images with their true class level and predicted class levels are added for reference in Table 1.

6.2 Japanese Female Facial Expression (JAFFE) Database

The JAFFE database is a posed expression database containing 213 images of 10 different female adults showing six atomic facial expressions and there is no occlusion like spectacle, hair falling on face in JAFFE dataset. This dataset is consist of six basic expressions Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), Surprise (SU) and also Neutral (NE) expression. In this dataset, the system obtained an overall precision of 97.18%. We have also computed N-Fold accuracy

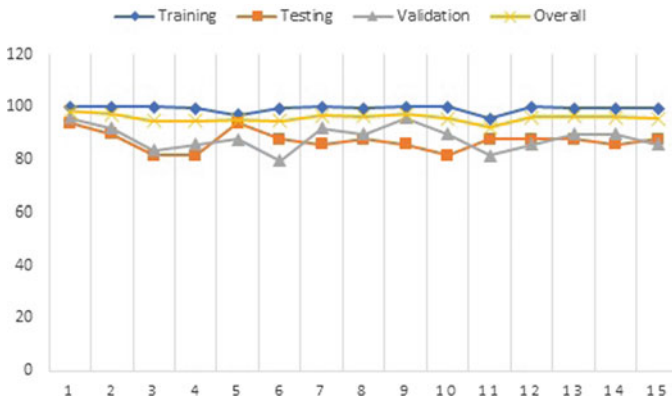


Fig. 4 N-Fold training, testing, validation, and overall accuracy of CK+ database [20]

Table 1 Visual representation of sample images of CK+ database with expression levels predicted by our system

Input image			
True expression level	Anger	Disgust	Fear
Expression level predicted by our system	Anger	Disgust	Fear
Verdict	Truly classified	Truly classified	Truly classified
Input image			
True expression level	Happiness	Sadness	Surprise
Expression level predicted by our system	Happiness	Anger	Surprise
Verdict	Truly classified	Misclassified	Truly classified

for training, testing, validation, and overall dataset reflected in Fig. 5. Some example images with their true class level and predicted class levels are added for reference in Table 2.

6.3 MMI Database

MMI is a posed expression dataset having multiple phases of data collection. The Phase-III of the MMI dataset contains 400 images with single Facial Action Unit (FAU) coded expression levels we have manually annotated 222 of them to six basic expression levels Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA) and Surprise (SU). Effective learning of the system with 222 manually annotated expressions results in an overall 96.87% accuracy. We have also computed N-Fold accuracy for training, testing, validation, and overall dataset reflected in Fig. 6. Some example images with their true class level and predicted class levels are added for reference in Table 3.

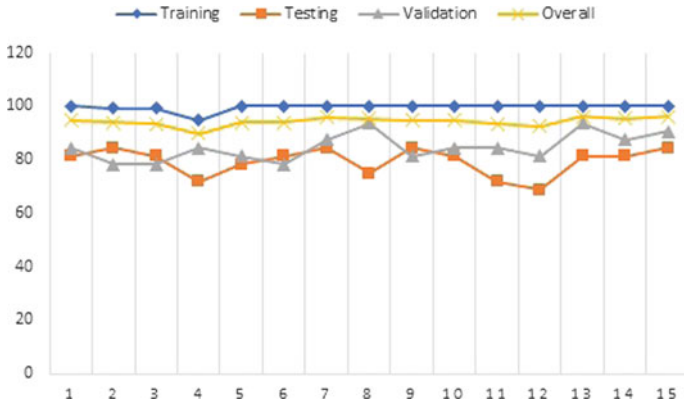


Fig. 5 N-Fold accuracy plot for training, testing, validation and overall images of JAFFE database [21]

Table 2 Visual representation of sample images of JAFFE database with expression levels predicted by our system

Input image			
True expression level	Anger	Disgust	Fear
Expression level predicted by our system	Anger	Disgust	Anger
Verdict	Truly classified	Truly classified	Misclassified
Input image			
True expression level	Happiness	Sadness	Surprise
Expression level predicted by our system	Happiness	Sadness	Surprise
Verdict	Truly classified	Truly classified	Truly classified

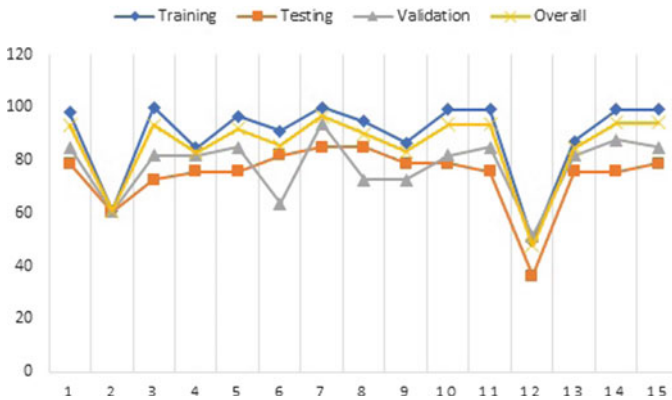
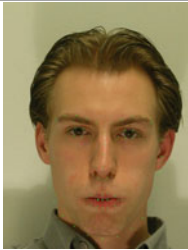
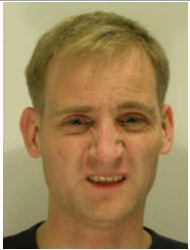
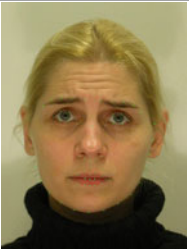

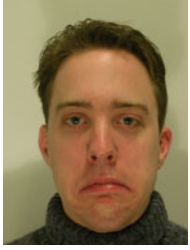
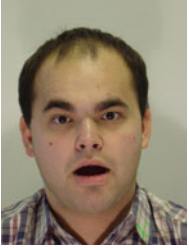


Fig. 6 N-Fold accuracy plot for training, testing, Validation, and overall images of MMI database [29]

Table 3 Visual representation of sample images of MMI database with expression levels predicted by our system

Input image			
True expression level	Anger	Disgust	Fear
Expression level predicted by our system	Anger	Anger	Fear
Verdict	Truly classified	Misclassified	Truly classified
Input image			
True expression level	Happiness	Sadness	Surprise
Expression level predicted by our system	Happiness	Sadness	Surprise
Verdict	Truly classified	Truly classified	Truly classified

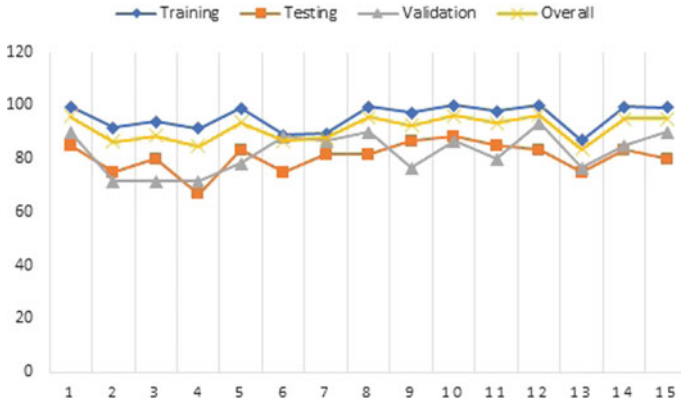


Fig. 7 N-Fold accuracy plot for training, testing, Validation, and overall images of MUG database [1]

6.4 Multimedia Understanding Group (MUG) Database

MUG is a mixed expression dataset with posed and spontaneous expressions containing 401 images from 26 subjects. This dataset is consist of six basic expressions Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), Surprise (SU), and also Neutral (NE) expression. Our system accomplishes an 97.26% of overall accuracy in MUG dataset. We have also computed N-Fold accuracy for training, testing, validation, and overall dataset reflected in Fig. 7. Some example images with their true class level and predicted class levels are added for reference in Table 4.

7 Discussions

The four different types of performance measure of training, validation, testing, and overall accuracy are shown in table no Table 5. The confusion matrix of the CK+, JAFFE, MMI, and MUG is presented in Tables 8, 9, 10 and 11, respectively. The MLP classifier shows very good recognition rate in CK+ database with 99.08% overall accuracy and the Table 6 shows that Anger, Contempt, Disgust, Happiness, and Surprise expressions are recognized with 100% precision. In the JAFFE dataset overall accuracy is 97.18% obtained and the Table 6 shows that the Neutral, Surprise, Anger, and Disgust expressions show 100% recognition rate. The MMI dataset shows 100% precision in Happiness expression with an overall 96.87% accuracy. The MUG database shows 100% Recognition rate for Disgust, Fear, Happiness, Sadness, and Surprise expression. Good accuracy in all four datasets implies that the system

Table 4 Visual representation of sample images of MUG database with expression levels predicted by our system

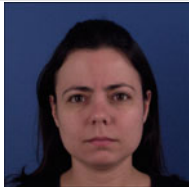
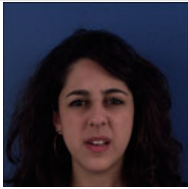
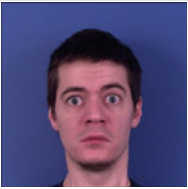

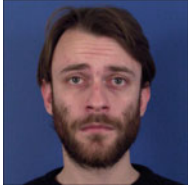
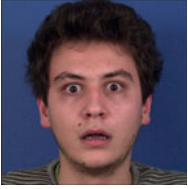
Input image			
True expression level	Anger	Disgust	Fear
Expression level predicted by our system	Fear	Disgust	Fear
Verdict	Misclassified	Truly classified	Truly classified
Input image			
True expression level	Happiness	Sadness	Surprise
Expression level predicted by our system	Happiness	Sadness	Surprise
Verdict	Truly classified	Truly classified	Truly classified

Table 5 Accuracy comparison table with training, testing, validation, and overall precisions

	CK+	JAFFE	MMI	MUG
Training	100	100	100	100
Validation	100	96.8	90.91	93.33
Testing	93.88	84.38	87.88	88.33
Overall	99.08	97.18	96.87	97.26

learns the expressions of a human face image in a person’s independent manner. The circumcenter-incenter-centroid trio feature efficiently grabs the expression related cues showing effective and efficient learning of the system. Comparison with other machine learning technique ate also portrayed in Table 7.

Table 9 Confusion JAFFE [21]

	NE	HA	SA	SU	AN	DI	FE
NE	30	0	0	0	0	0	0
HA	1	29	1	0	0	0	0
SA	1	0	29	0	0	0	1
SU	0	0	0	30	0	0	0
AN	1	0	0	0	29	0	0
DI	0	0	0	0	0	29	0
FE	0	0	1	0	0	0	31

Table 10 MMI confusion [29]

	AN	DI	FE	HA	SA	SU
AN	60	0	0	0	1	1
DI	1	23	0	0	0	0
FE	1	0	22	1	0	0
HA	0	0	0	45	0	0
SA	0	0	1	0	27	0
SU	1	0	0	0	0	38

Table 11 MUG confusion [1]

	NE	AN	DI	FE	HA	SA	SU
NE	21	3	0	0	0	0	1
AN	1	55	1	0	0	0	0
DI	0	0	71	0	0	0	0
FE	0	0	0	47	0	0	0
HA	0	0	0	0	87	0	0
SA	0	0	0	0	0	48	0
SU	0	0	0	0	0	0	66

8 Conclusions

Overwhelming degree of accuracy on different benchmark databases such as CK+, JAFFE, MMI, and MUG vindicates the effectiveness and efficiency of the proposed method with judiciously chosen geometric feature set for the said purpose (Table 12).

Table 12 Confusion matrix of JAFFE, CK+, MMI, and MUG databases

	AN	CO	DI	FE	HA	SA	SU
AN	45	0	0	0	0	0	0
CO	0	18	0	0	0	0	0
DI	0	0	59	0	0	0	0
FE	0	0	0	24	1	0	0
HA	0	0	0	0	69	0	0
SA	1	1	0	0	0	26	0
SU	0	0	0	0	0	0	83

Table 8: CK+ Confusion [20].

	NE	HA	SA	SU	AN	DI	FE
NE	30	0	0	0	0	0	0
HA	1	29	1	0	0	0	0
SA	1	0	29	0	0	0	1
SU	0	0	0	30	0	0	0
AN	1	0	0	0	29	0	0
DI	0	0	0	0	0	29	0
FE	0	0	1	0	0	0	31

Table 9: Confusion JAFFE [21].

	AN	DI	FE	HA	SA	SU
AN	60	0	0	0	1	1
DI	1	23	0	0	0	0
FE	1	0	22	1	0	0
HA	0	0	0	45	0	0
SA	0	0	1	0	27	0
SU	1	0	0	0	0	38

Table 10: MMI Confusion [29].

	NE	AN	DI	FE	HA	SA	SU
NE	21	3	0	0	0	0	1
AN	1	55	1	0	0	0	0
DI	0	0	71	0	0	0	0
FE	0	0	0	47	0	0	0
HA	0	0	0	0	87	0	0
SA	0	0	0	0	0	48	0
SU	0	0	0	0	0	0	66

Table 11: MUG Confusion [1].

Acknowledgements The authors like to avail this opportunity to express their gratitude to Dr. A. Delopoulos for providing MUG database and Prof. Maja Pantic for providing MMI database free for carrying out this work. The authors would also like to thank the Department of Computer and System Sciences, Visva-Bharati, Santiniketan for their infrastructure support. The authors gratefully acknowledge the support of UGC NET-JRF Fellowship (UGC-Ref. No.: 3437/(OBC)(NET-JAN 2017)) for pursuing Doctoral Research in Department of Science and Technology, Ministry of Science and Technology, Government of India.

References

1. Aifanti N, Papachristou C, Delopoulos A (2010) The mug facial expression database. In: Proceedings of the 11th international workshop on image analysis for multimedia interactive services WIAMIS 10. IEEE, pp 1–4
2. Almaev TR, Valstar MF (2013) Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In: Proceedings of the 2013 humane association conference on affective computing and intelligent interaction. IEEE, pp 356–361
3. Barman A, Dutta P (2017) Facial expression recognition using distance and shape signature features. Pattern Recogn Lett
4. Barman A, Dutta P (2019) Facial expression recognition using distance and texture signature relevant features. Appl Soft Comput 77:88–105
5. Barman A, Dutta P (2019) Influence of shape and texture features in facial expression recognition. IET Image Process.
6. Bente G, Krämer NC, Eschenburg F (2008) Is there anybody out there. Mediated interpersonal communication, pp 131–157

7. Bhattacharjee D, Basu DK, Nasipuri M, Kundu M (2010) Human face recognition using fuzzy multilayer perceptron. *Soft Comput* 14(6):559–570
8. Boughrara H, Chtourou M, Amar CB, Chen L (2016) Facial expression recognition based on a mlp neural network using constructive training algorithm. *Multimed Tools Appl* 75(2):709–731
9. Chen J, Chen Z, Chi Z, Fu H (2014) Facial expression recognition based on facial components detection and hog features. In: *Proceedings of the international workshops on electrical and computer engineering subfields*, pp 884–888
10. Cheon Y, Kim D (2009) Natural facial expression recognition using differential-aam and manifold learning. *Pattern Recogn* 42(7):1340–1350
11. Collier G, Collier GJ (2014) *Emotional expression*. Psychology Press
12. Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. *IEEE Trans Pattern Anal Mach Intell* 6:681–685
13. Ekman P (1992) An argument for basic emotions. *Cogn Emot* 6(3–4):169–200
14. Ekman P (2004) Emotions revealed. *BMJ* 328(Suppl S5):0405184
15. Ekman P, Friesen WV (1971) Constants across cultures in the face and emotion. *J Personal Soc Psychol* 17(2):124
16. Er MJ, Wu S, Lu J, Toh HL (2002) Face recognition with radial basis function (RBF) neural networks. *IEEE Trans Neural Netw* 13(3):697–710
17. Happy S, Routray A (2014) Automatic facial expression recognition using features of salient facial patches. *IEEE Trans Affect Comput* 6(1):1–12
18. Haykin SS et al (2009) *Neural networks and learning machines/Simon Haykin*. Prentice Hall, New York
19. Kotsia I, Pitas I (2006) Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Trans Image Process* 16(1):172–187
20. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: *Proceedings of the 2010 IEEE computer society conference on computer vision and pattern recognition-workshops*. IEEE, pp 94–101
21. Lyons M, Akamatsu S, Kamachi M, Gyoba J (1998) Coding facial expressions with gabor wavelets. In: *Proceedings of the third IEEE international conference on automatic face and gesture recognition*. IEEE, pp 200–205
22. Manstead AS, Fischer AH (2002) Beyond the universality-specificity dichotomy. *Cogn Emot* 16(1):1–9. <https://doi.org/10.1080/0269993014000103>
23. Martin C, Werner U, Gross HM (2008) A real-time facial expression recognition system based on active appearance models using gray images and edge images. In: *Proceedings of the 2008 8th IEEE international conference on automatic face & gesture recognition*. IEEE, pp 1–6
24. Møller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw* 6(4):525–533
25. Pham TH, Philippot P (2010) Decoding of facial expression of emotion in criminal psychopaths. *J Pers Disord* 24(4):445–459
26. Russell J (1997) Reading emotions from and into faces: resurrecting a dimensional-contextual perspective. *The psychology of facial expression*, pp 295–320
27. Sagonas C, Antonakos E, Tzimiropoulos G, Zafeiriou S, Pantic M (2016) 300 faces in-the-wild challenge: database and results. *Image Vis Comput* 47:3–18
28. Tzimiropoulos G, Pantic M (2013) Optimization problems for fast AAM fitting in-the-wild. In: *Proceedings of the IEEE international conference on computer vision*, pp 593–600
29. Valstar M, Pantic M (2010) Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In: *Proceedings of the 3rd international workshop on EMOTION (satellite of LREC): corpora for research on emotion and affect*, p 65
30. Yuan L, Wu CM, Zhang Y (2013) Facial expression feature extraction using hybrid PCA and LBP. *J China Univ Posts Telecommun* 20(2), 120–124 (2013)

Stable Neighbor-Node Prediction with Multivariate Analysis in Mobile Ad Hoc Network Using RNN Model



Arindrajit Pal, Paramartha Dutta, Amlan Chakrabarti
and Jyoti Prakash Singh

Abstract In mobile ad hoc networks (MANETs), mobile nodes are communicating with each other without use of any fixed infrastructure. Here, each node works as a receiver as well as transmitter point in the network. This network maintains the wireless connections with the neighbor nodes and establishes a connecting link between the source–destination (s-d) pair. The route in this type of network is highly unstable due to the mobility of the nodes. So, to construct a steady path between s-d pair, it is obvious to build a path through the stable neighbor nodes. In this article, we propose a stability index (SIN) which depends on the various parameters of the nodes such as past SIN values in different time intervals, node velocity, etc. In this paper, we establish a time series prediction model with multivariate analysis for predicting the stability index (SIN) of a node in reference to its neighbor nodes for the future time frame based on their past observation. For this purpose, we use the Elman recurrent neural network (ERNN)-based learning tool to determine the behavior of the mobile nodes of the network in the future time frame.

Keywords Mobile ad hoc network · Multivariate analysis · Stable neighbors · Elman Recurrent Neural Network · Time series prediction

A. Pal (✉) · A. Chakrabarti
A. K. Choudhury School of Information Technology, University of Calcutta,
Kolkata, West Bengal, India
e-mail: arindrajit@gmail.com

A. Chakrabarti
e-mail: acakcs@caluniv.ac.in

P. Dutta
Visva-Bharati University, Santiniketan, West Bengal, India
e-mail: paramartha.dutta@gmail.com

J. P. Singh
National Institute of Technology, Patna, Bihar, India
e-mail: jps@nitp.ac.in

1 Introduction

The backbone infrastructure in Mobile Ad hoc Network (MANET) is not available. So, it is forced to create the infrastructure as and when required. The mobile nodes in MANET search and maintain a path from a source s to a destination d to transfer data packets between them. The nodes establish a link among the neighbor which is in the communication range of each other. This neighborhood is constantly varying due to the movement of the nodes in the network [20, 23]. So, the path from the source to destination breaks frequently [18, 25]. The path from a source to destination can be made stable by selecting all intermediate nodes which are more stable. Some research works have been proposed to select nodes which are more stable compared to other [16]. However, most of the earlier works considered one parameter only to evaluate the stability measure of a node. But node stability depends on many factors such as earlier stability value, speed of the node, energy of the node, packet delivery ratio, etc.

In different effective multivariate parameters of the mobile ad hoc networks, the speed of the mobile nodes is one of the important parameters for measuring the stability of the network. The effect of the speed variation on various ad hoc networks with different traffic patterns was studied by Pal et al. [17]. The average path length between source–destination was predicted by Pal et al. [18] using moving average (MA) model. In another work, Pal et al. [19] made a comparative analysis between the autoregressive integrated moving average (ARIMA) model and artificial neural network for path length prediction in the network using Ad hoc On-demand Distance Vector (AODV) routing protocol. This analysis shows that the multilayer perceptron (MLP) model can perform better results than the ARIMA model. Pal et al. [16] used a metaheuristic technique based on a biogeographic model to estimate the node stability of the mobile network. They proposed a stable neighbor selection model for establishing a stable path formed between the source and destination node. The multivariate analysis [13] can be used for obtaining the relation between a set of variables. The relation between the dependent variables which are correlated with the independent variables based on the parameters of the observation can be obtained by this analysis. The univariate tests and the confidence intervals which are based on the univariate normal distribution may not justify the different network-related problems because a large number of parameters are involved here. So, the multivariate normal distribution method may be exploited in this process. The means, variances, and covariances of a multivariate data set may be used to check the correlated or independent variables from the data set. It also shows the linear trends of the variables and serves useful approximation inferences involving sample mean vectors. Multivariate analysis is involved with the study of p number of variables as

$$Y = [Y_1, Y_2, \dots, Y_p]$$

The expectations are the value of the random $p \times 1$ vector, which can be defined as

$$E(Y) = \begin{bmatrix} E(Y_1) \\ E(Y_2) \\ \vdots \\ E(Y_p) \end{bmatrix}$$

Let, $N_p(\mu, \Sigma)$ be the multivariate normal distribution of y , where, μ is the mean vector and Σ is the covariance matrix then,

$$\mu(E(Y)) = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$$

and the density

$$g(y) = \frac{1}{(\sqrt{2\pi})^p |\Sigma|^{1/2}} e^{-\frac{(y-\mu)\Sigma^{-1}(y-\mu)}{2}}$$

In this article, we have proposed a stable neighbor finding process which can select the intermediate nodes between a source–destination pair based on the stability of the neighbor nodes using the analysis of multivariate parameters. The time series prediction model is used to predict future value based on past observations. We have applied this model to predict the position of the neighbor nodes in the future based on their past observation. So, we considered the stability index (SIN) of each node at time t_k and calculated it for time t_{k+1} . We used a multivariate statistical model to find the stability of a neighbor node with respect to the other nodes.

Let us assume that the $\{x_t, x_{t-1}, x_{t-2}\}$ are the datasets at time $\{t, (t - 1), (t - 2)\}$, respectively. Now, the time series forecast the unknown data value at x_{t+1} based on the past data set. Recently, Artificial neural network (ANN) [4, 9, 11] have become very popular prediction model to be applied in various facets of life [14]. Generally, an ANN model consists of three layers such as input layer, hidden layer, and output layer. The data provided to the input layer pass through the hidden layers and generate the output values through the output layer. The weight function and the activation function control the process of the neurons of ANN. There are various types of ANN models like Multilayer perceptron (MLP), Radial basis function (RBF), Recurrent neural network (RNN) model, etc. The Elman recurrent neural network (ERNN) [6] is an efficient RNN model which was developed by Elman. The ERNN model can be used for time series and nonlinear prediction models [2, 3]. Figure 1 shows the architecture of ERNN network. In this model, the outputs of the hidden layer are connected to another buffer layer called the recurrent layer. The output of the recurrent layer is feedback to the hidden layer. This model helps to generate the temporal patterns in the future time frame. Each hidden neuron is connected to only one recurrent layer neuron with a weight value. So, each layer of ERNN contains

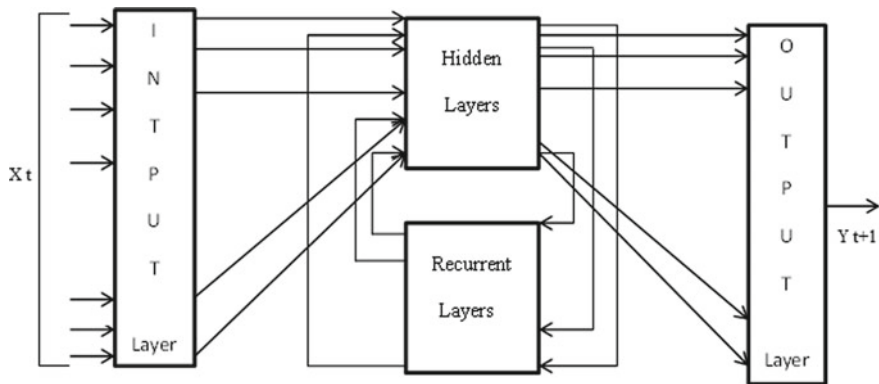


Fig. 1 The block diagram of ERNN model

one or more neurons which produce information from one layer to another layer by computing a nonlinear function of their weighted sum of inputs.

In this paper, we proposed a stable neighbor node prediction model in MANET which can predict the stable neighbor nodes of each mobile node. These stable nodes can be used to establish the path between source and destination node which will be more stable than the other paths. To find the stable nodes, we used predictive RNN model using the past behavior of the nodes. We also analyzed the multivariate parameters of the mobile nodes for the prediction process.

The remainder of this paper is constructed as follows. In Sect. 2, we compile the related works. The methodology is discussed in Sect. 3. In Sect. 4, the simulation model is designed. The results and discussion of performance evaluation are presented in Sect. 5. Finally, conclusions and our future works are proposed in Sect. 6.

2 Related Works

The issue of mobility and neighborhood changes in mobile ad hoc network was studied by Singh and Dutta [23]. They modeled the variation of neighbor nodes due to movement using Autoregressive model. In their next work [24], they considered the link characteristic due to the mobility of nodes and modeled that using Autoregressive models. They further considered the problem of path length in [25] and modeled them using time series. Node stability problem was taken up by Pal et al. [16] to predict the stable nodes in a mobile ad hoc network using Biogeography-based optimization. Guo et al. [8] proposed a system to find the status of a link by probabilistic models. Torkestani and Meybodi [27] proposed a method to find the best stable link between two nodes. Jiang et al. [12] developed a model to select the best path between a source–destination pair using the more reliable links. Biradar et al. [1] proposed a model to find a stable link for nodes moving with a random waypoint mobility pattern.

Singh et al. [26] proposed a regression model to estimate end-to-end delay in ad hoc network considering parameters path length from source to destination and previous delay. Wang et al. [28] proposed a stable route-finding scheme using a weight-based route strategy. The route expiration time, the error count, and the hop count were used to assign the weights to each route from the source node to the destination node. Finally, the largest weight path is selected for routing the packets. Gerharz et al. [7] estimated the stability of paths and introduced statistical metrics to identify stable paths in a mobile ad hoc network using statistical methods. Sarma and Nandi [22] used received signal strengths to compute the link and route stability in mobile ad hoc network. They added relevant fields in route request/reply packets to achieve this. An excellent survey of routing algorithms using link stability is given in [15].

3 Methodology

The mobile wireless network environment is highly unstable and dynamic. So, the link established for route creation based on one metric is always considered as the suboptimal decision. Here, we describe the key ideas to develop a network based on the multivariate analysis. In this section, we have done the following operations:

- Evaluated the effective time lags for inputs of the ERNN model.
- Design a suitable ERNN model for prediction.

3.1 Time Lag Selection Methodology

The choice of lag numbers for input variables is an important part of this network. Here, we consider the input variables like velocity of a nodes (Ve), energy of a nodes (En), packet delivery ratio (PDR) of a nodes, and throughput (Th) of a nodes for multivariate analysis. We also find out the stability index (SIN) of the network based on these parameters. We have selected the autocorrelation function (ACF) and partial autocorrelation function (PACF) for deciding the effective lag values which are correlated with the ERNN model. The PACF is same as the autocorrelation between x_t and x_{t-k} except the linear dependence of x_{t-1} through to x_{t-k+1} have been discarded, i.e., the correlations with all the elements within the lag are partially out. The correlation is measured to consider the acceptable parameters of the network. Here, we considered the Pearson's correlation model to evaluate the correlation.

The linear dependencies between two variables can be measured by the correlation coefficient as follows:

$$Z = cov(X, Y) / \sqrt{var(X)var(Y)}$$

where, the *cov* and *var* are the covariance and variance, respectively.

Now, the autocorrelation(ρ_k) for lag k is

$$\rho_k = \frac{\sum_{t=k+1}^n (x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^n (x_t - \bar{x})^2}$$

where,

n is number of samples,

x_t is the sample value at time t , and

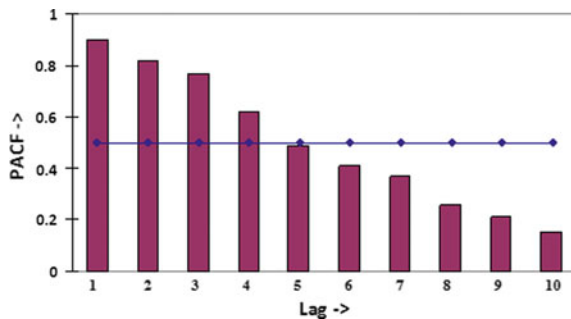
\bar{x} is the sample mean

The correlation value 0 means there is no linear correlation within the variables. If the correlation is +1 or -1, then it is called the perfect positive or negative linear correlation, respectively, between the variables. When the correlation is less than +0.5 or -0.5, then it is considered as the weak correlation between two variables. To modify this correlation factor, we considered the Student T-test for the significance of Z . In this experiment, we observed that the threshold value is very low according to the T-test. But the threshold for the coefficient Z is more important for a certain number of inputs. We have selected the threshold equal to 50% of Z , i.e., only the higher values greater than or equal to 0.5 correlation will be chosen. We have also observed that the increase of this value cannot find a suitable number of entries for evaluating. The lower value of Z may select more data but those extra values have no extra impact for prediction. So, we have considered that the 50% of the threshold value Z is optimal for the prediction process. We have considered the 95% Confidence Interval (CI) of the PACF for the time lags which are significantly correlated.

Figure 2 shows the PACF value of the stability index (SIN). It shows that the first four lag values are greater than the upper confidence interval. So, we consider up to these four lag values to the input of the ERNN model. The upper lag values have no effect on predicting the output of the ERNN network. Though, if we consider the more upper values, then it decreases the performance of the network.

In Fig. 3, we observe the cross-correlation between the SIN and all the multivariate parameters. The straight horizontal lines to both sides of the x-axis represent that the confidence interval value is +0.5 and -0.5, respectively. In this figure, we calculate

Fig. 2 Partial autocorrelation of the stability index (SIN). The straight lines around zero represent the upper and lower bounds of the PACF 95% CI



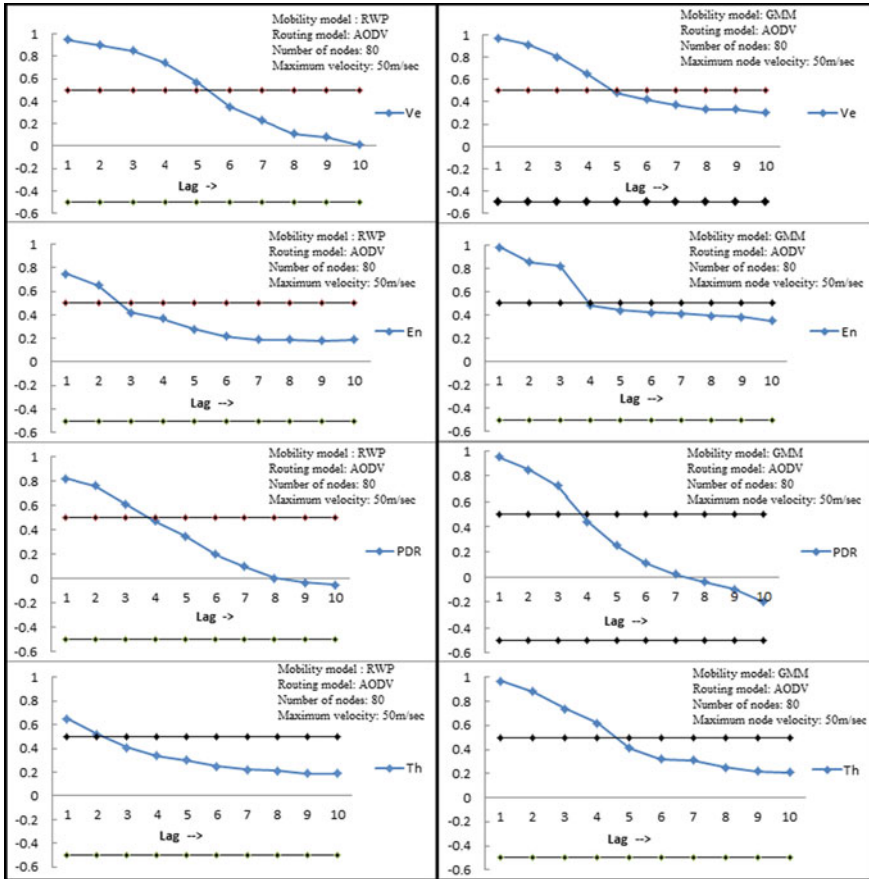


Fig. 3 Pearson cross-correlation between the SIN and the multivariate parameters Ve, En, PDR and Th for RWP and GMM mobility model. The lines define the confidence band selected

the cross-correlation values in different lags and plotted those values into the graph. We only consider those lag values which are greater than the confidence interval. These graphs show the effective lag values for prediction models. If the values are greater than the 95% CI, then there are fewer local minima and the accuracy will be higher. The past values should be considered up to a certain interval after that those past values have no influence in the prediction model and it increases the local minima and also the overhead of the system.

3.2 ERNN Configuration

In this part, we have developed the optimize ERNN model for predicting the stable neighbors of the network. This model consists of four different layers. The input layer (X) has n numbers of input nodes. It is a vector with n values, i.e., X_n . There is a hidden layer (H) with n nodes after the input layer. The recurrent layer (R) which has the inputs from the hidden layer and the outputs of this layer considers the feedback of the hidden layer. So, the outputs from H move to the layer R at time t and from R its moves to H at time $(t + 1)$. The output layer (Y) is connected with the layer H . Through the learning technique, this network generates the temporal output in Y for the future time domain at time $(t + 1)$ from the past data sets of $\{\dots, (t + 1), t\}$. We first optimized the ERNN model by data normalization and selected the optimum number of hidden neurons. The important parameters which influence the network complexity and the number of local minima are the multivariate input variables and their lag numbers, the structure of the hidden layers, activation function, and the learning algorithm. Some other parameters, like the sampling size of learning data, the distribution of data between learning, test, and validation phases, must be considered into account. Now, the input values to the hidden layer (H_t) at time t are

$$H_t = \sum_{i=1}^n w_i \cdot X_t + \sum_{j=1}^m r_j \cdot R_{t-1}$$

where,

w_i = weight function between the input and hidden layers

X_t = input vector of input layer

r_j = weight function between the recurrent and hidden layers

R_t = output of the recurrent layer

Now, the output (Y_{t+1}) of the ERNN network is

$$Y_{t+1} = \phi \left(\sum_{i=1}^n p_i \cdot H_t \right)$$

where, ϕ is the activation function and p_i is the weight function between the hidden layer and output layer.

Here we consider the input vector

$X_t = [\text{SIN}, \text{Ve}, \text{En}, \text{PDR}, \text{Th}]_t$

and the output value is

$Y_{t+1} = [\text{SIN}]_{(t+1)}$

The optimal lag values at time t of the multivariate variables can be obtained from Table 1.

4 Simulation Model

This network model was developed in the Matlab simulator [5]. We have considered two mobility models for the movement pattern of the mobile nodes like random-way point (RWP) [10] and Gauss–Markov mobility (GMM) [21]. The nodes are distributed uniformly in an area of $200 \times 200 \text{ m}^2$ in the simulator. The node velocity varies in between 0 and 50 m/sec. The maximum transmission range of a node is the 20-meter radius. Fixed 80 nodes are moving within the area, and at the time of simulation, no node can leave or enter the simulation area. The s-d pair nodes are disjoint and randomly chosen from the set of mobile nodes. About 3000 data packets can be transmitted by any node within a time lag/interval. The network bandwidth is considered as 2 MBps. The MAC layer protocol uses the IEEE 802.11b distributed coordination function (DCF).

5 Results and Discussion

The results would be discussed in this section to validate our proposed model. Calculating the errors between the predicting output values with the real-simulated values done by the Normalized Root Mean Square Error (NRMSE). The NRMSE can be calculated using the following equation:

$$\text{NRMSE} = \sqrt{(x - y)^2/x^2}$$

where x represents the real value and y is the prediction value.

The selection of variables for the ERNN model with their desire time lag is evaluated in this phase to optimize the network. The significance of cross-correlation is already examined, and here we applied this phenomenon for accuracy. Figure 3 shows the Pearson cross-correlation (PCC) between the stability index (SIN) and all the parameters available in this network. These velocity (Ve), energy (En), packet delivery ratio (PDR) and throughput (Th) of each node are the four parameters which are considered for multivariate analysis. The value of PCC in lag 0 should not be considered for forecasting approach. Here, we already fixed the 50% bound. We have observed the influence of all the multivariate parameters on the output considering one by one. From Fig. 3 we have found appropriate different time lags of these parameters to be considered for future data analysis. In Table 1, we have shown the percentage of error and the variance for each multivariate parameters separately. The lag values are measured as shown in Figs. 3 and 4. We have shown the NRMSE values in different lags for different parameters. The bold numbers represent the optimal lag values of all the parameters. These results are used to check the actual impact of the multivariate variables in the network. These values can be used as the input of the ERNN model for optimizing. The results as shown in the table have a great significance in multivariate analysis. The 95%CI is lower which means there are

Table 1 Error prediction of stability index and different multivariate inputs separately

Effective lag values					% of NRMSE	Value of 95% CI
SIN	Ve	En	PDR	Th		
3	–	–	–	–	6.25	0.58
3	–	–	–	2	6.11	0.54
3	–	–	3	–	5.72	0.48
3	–	3	–	–	6.02	0.59
3	2	–	–	–	5.89	0.26
3	2	3	3	2	5.26	0.11

Table 2 Error prediction of stability index and different multivariate inputs separately

Optimized lag values of multivariate parameters SIN = 3, Ve = 2, En = 3, PDR = 3, Th = 2

Number of hidden layers	% of NRMSE	Value of 95% CI
1	0.6	0.2
2	0.5	0.3
3	0.5	0.4
4	0.4	0.2
5	0.5	0.3
6	0.6	0.4
7	0.7	0.5
8	0.8	0.6

less local minima and high accuracy. In the next step, we were trying to determine the optimal number of hidden nodes in the ERNN model. Table 2 represents the optimal number of hidden nodes of the ERNN model by the bold numbers. The optimal multivariate parameters are chosen from Table 1. These values show that the increasing number of hidden nodes initially decreases the error. However, after a certain limit, it further increases the NRMSE value. Here, we have taken four nodes in the hidden layer.

In Fig. 4, we have shown the percentage of accuracy between the original simulated values and the predicted values with the multivariate analysis. Each graph shows the observation with two mobility models like RWP and GMM. We have changed the parameters for each experiment as mentioned in the graphs. In each case, it has been shown that the experiment gives a good result after certain lag values. This means our model can guess a better observation after a few time lags and consistently increase the accuracy in all respect.

Figure 5 shows the comparison between the univariate and multivariate analysis with respect to the number of link fault. For univariate analysis, we considered the

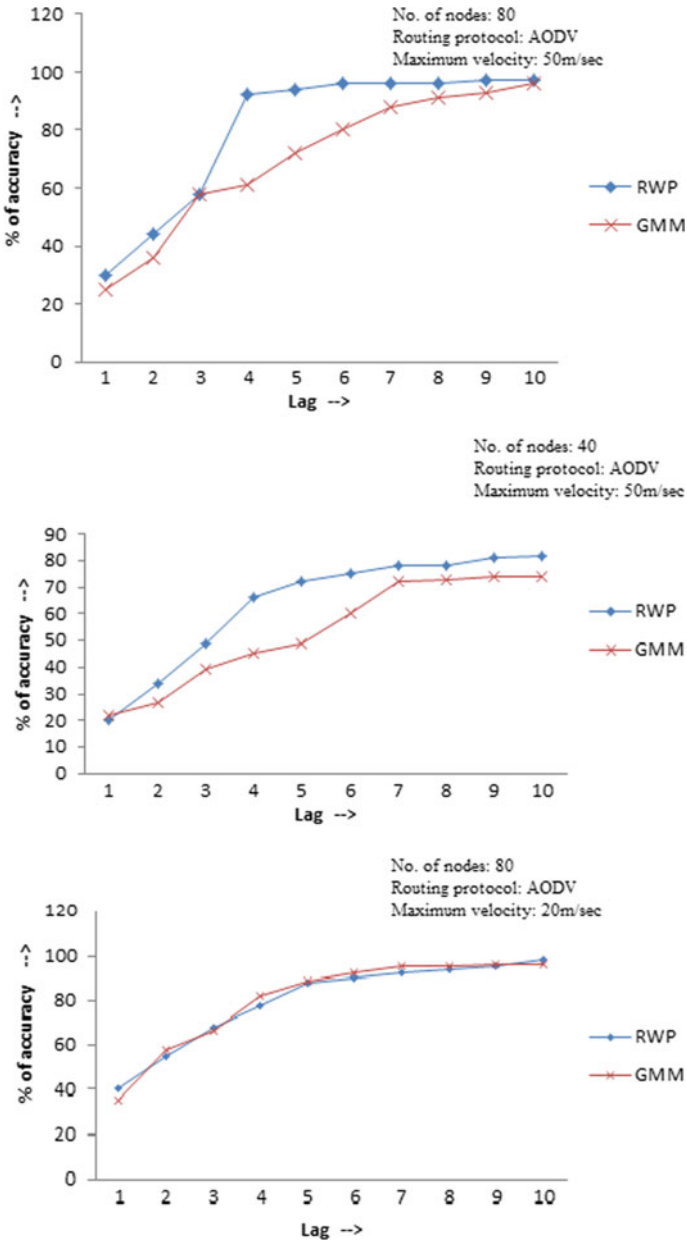


Fig. 4 Percentage of accuracy between the predicted and the original values

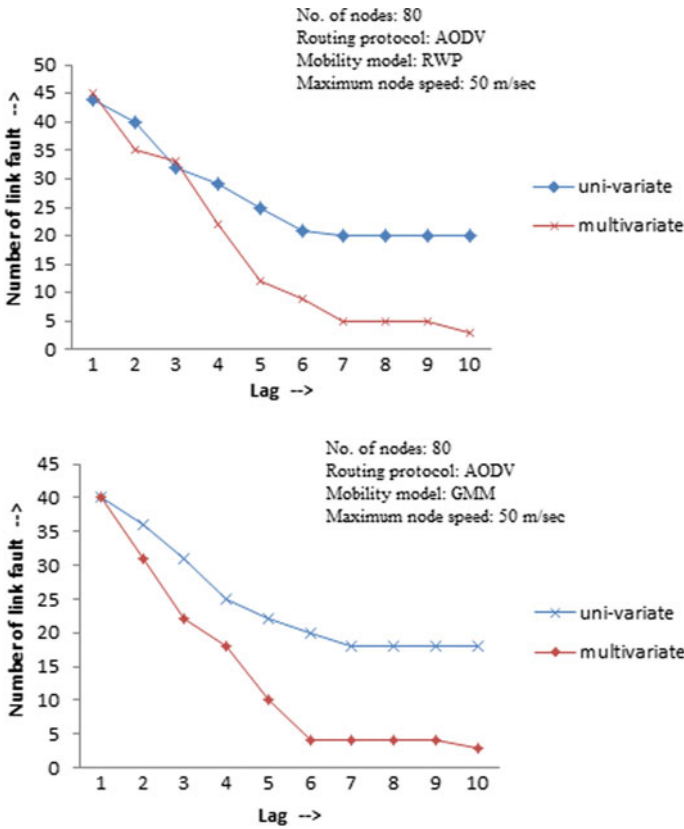


Fig. 5 Compare the number of link faults between univariate and multivariate prediction analysis for different mobility models

prediction model for one variable only and calculate SIN at time $(t + 1)$. The variable is chosen from Table 1 with the optimal lag value. Next, we have considered the multivariate parameters with the same network. It has been shown that the multivariate analysis can predict the SIN more accurately and decrease the link fault of the network.

In Fig. 6, we have done a comparative analysis of the NRMSE between univariate and multivariate parameters. It shows that the prediction using multivariate parameters shows a better result than the prediction by univariate parameters. So, it proves that the different parameters of the mobile network have some impact on predicting future values.

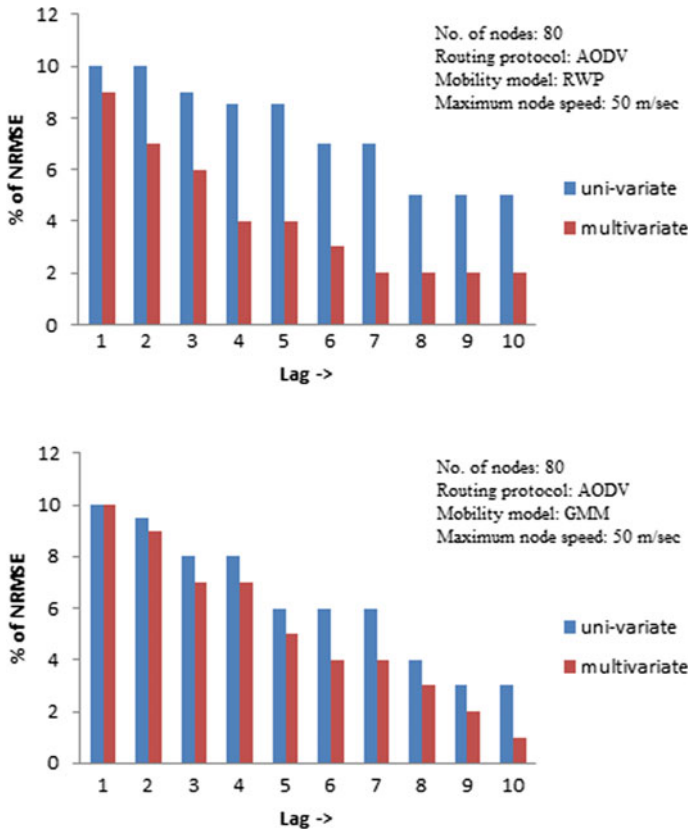


Fig. 6 Compare the NRMSE between univariate and multivariate prediction analysis for different mobility models

6 Conclusion

In this paper, we have proposed a technique to find stable neighbor nodes using a multivariate method. Our proposed mechanism provides a better QoS than the univariate models. Here, we identified four parameters, that is, like velocity (V_e), energy (E_n), packet delivery ratio (PDR) and throughput (Th) which can affect the node stability of the mobile network. We develop an ERNN-based machine learning technique for predicting the stability index (SIN) on the future time frame based on the past data values. We consider the multivariate parameters as the inputs of the ERNN model. We have also found that these multivariate parameters have an impact on the prediction process, and this impact also changes for different lag values. The appropriate selection of the lag values of different parameters increased the stability index (SIN) of the mobile nodes. We compared the error analysis between the univariate and multivariate models in this paper. It shows that the multivariate

method predicts more accurately than the univariate method. All these experimental results show that our method is more justified. This encourages us to study in the future how to develop this methodology to a routing model to minimize the link fault. It seems obvious that the multivariate parameters have a greater impact on prediction accuracy.

References

1. Biradar RC, Manvi SS (2012) Neighbor supported reliable multipath multicast routing in manets. *J Netw Comput Appl* 35(3):1074–1085
2. Cacciola M, Megali G, Pellicano D, Morabito FC (2012) Elman neural networks for characterizing voids in welded strips: a study. *Neural Comput Appl* 21(5):869–875
3. Chandra R, Zhang M (2012) Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. *Neurocomputing* 86:116–123
4. Crone SF (2005) Stepwise selection of artificial neural network models for time series prediction. *J Intell Syst* 14(2–3):99–122
5. Demuth H, Beale M (1998) *Neural network toolbox: for use with matlab*, natick, ma: the math works. Inc. OpenURL
6. Elman JL (1990) Finding structure in time. *Cogn Sci* 14(2):179–211
7. Gerharz M, de Waal C, Martini P, James P (2003) Strategies for finding stable paths in mobile wireless ad hoc networks. In: *Proceedings of the 28th annual IEEE international conference on local computer networks*, 2003. LCN'03. IEEE, pp. 130–139
8. Guo L, Peng Y, Wang X, Jiang D, Yu Y (2011) Performance evaluation for on-demand routing protocols based on opnet modules in wireless mesh networks. *Comput Electr Eng* 37(1):106–114
9. Hu Y, Hwang J (2002) Neural network solutions to signal processing problems. *Handbook of neural network signal processing*
10. Hyytiä E, Virtamo J (2007) Random waypoint mobility model in cellular networks. *Wirel Netw* 13(2):177–188
11. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. *Computer* 29(3):31–44
12. Jiang S, He D, Rao J (2005) A prediction-based link availability estimation for routing metrics in manets. *IEEE/ACM Trans Netw* 13(6):1302–1312
13. Kim K, Timm N (2006) *Univariate and multivariate general linear models: theory and applications with SAS*. Chapman and Hall/CRC
14. Kumar A, Singh JP (2019) Location reference identification from tweets during emergencies: a deep learning approach. *Int J Disaster Risk Reduct* 33:365–375
15. Moussaoui A, Boukream A (2015) A survey of routing protocols based on link-stability in mobile ad hoc networks. *J Netw Comput Appl* 47:1–10
16. Pal A, Dutta P, Chakrabarti A, Singh JP, Sadhu S (2019) Biogeographic-based temporal prediction of link stability in mobile ad hoc networks. *Wirel Pers Commun* 104(1):217–233
17. Pal A, Singh JP, Dutta P (2012) The effect of speed variation on different traffic patterns in mobile ad hoc network. *Procedia Technol* 4:743–748
18. Pal A, Singh JP, Dutta P (2013) The path length prediction of manet using moving average model. *Procedia Technol* 10:882–889
19. Pal A, Singh JP, Dutta P (2015) Path length prediction in manet under aodv routing: comparative analysis of arima and mlp model. *Egypt Inform J* 16(1):103–111
20. Singh JP, Dutta P (2011) Temporal modeling of link characteristic in mobile ad hoc network. *J Comput Inf Technol* 19(3):143–154

21. Roy RR (2010) Handbook of mobile ad hoc networks for mobility models. Springer Science and Business Media
22. Sarma N, Nandi S (2010) Route stability based QoS routing in mobile ad hoc networks. *Wirel Personal Commun* 54(1):203–224
23. Singh JP, Dutta P (2010) Temporal modeling of node mobility in mobile ad hoc network. *J Comput Inf Technol* 18(1):19–29
24. Singh JP, Dutta P (2011) Temporal modeling of link characteristic in mobile ad hoc network. *J Comput Inf Technol* 19(3):143–154 (2011)
25. Singh JP, Dutta P (2012) The temporal effect of mobility on path length in manet. *Int J Wirel Inf Netw* 19(1):38–48
26. Singh JP, Dutta P, Chakrabarti A (2014) Weighted delay prediction in mobile ad hoc network using fuzzy time series. *Egypt Inform J* 15(2):105–114
27. Torkestani JA, Meybodi MR (2011) A link stability-based multicast routing protocol for wireless mobile ad hoc networks. *J Netw Comput Appl* 34(4):1429–1440
28. Wang NC, Huang YF, Chen JC (2007) A stable weight-based on-demand routing protocol for mobile ad hoc networks. *Inf Sci* 177(24):5522–5537

A New Approach for Optimizing Initial Parameters of Lorenz Attractor and Its Application in PRNG



Ramen Pal and Somnath Mukhopadhyay

Abstract Lorenz attractors are sensitive to its initial parameter values. The values of the initial parameters or seeds are very important in terms of simulating the strange chaotic behavior and security of using it in different applications. Using a single initial seed or a set of seed values for an application is a threat to its security. The unique seed for each execution of an application will make the system more secure and robust. In last two decades lot of researches are carried out to estimate the initial parameter of different chaotic maps. They solved this problem by checking the similarity between an observed (benchmarked) and an experimental (attractor with the estimated seed) attractor in either time-domain or in state-space. All the solutions over time-domain are not able to consider the sensitivity on the seed of the attractor. The solutions over state-space considers this. However, these solutions also require an observed attractor to check the similarity with the experimental attractor. This research overcomes the aforementioned constraint and flaw of the past models. In this research, a Real Coded Genetic Algorithm(RCGA) based optimization technique for optimizing the initial parameter values for the Lorenz attractor is proposed. A Pseudorandom Number Generator (PRNG) based on the Lorenz attractor is also proposed. The optimized seed for the Lorenz attractor is used for the PRNG. NIST specified statistical tests for testing the randomness of the output bit sequence of the PRNG is done here.

Keywords Lorenz attractor · Lyapunov dimension · Initial parameter estimation · Real coded GA · PRNG · NIST

R. Pal (✉) · S. Mukhopadhyay
Department of Computer Science and Engineering, Assam University, Silchar, India
e-mail: ramen.pal673@gmail.com

S. Mukhopadhyay
e-mail: som.cse@live.com

© Springer Nature Singapore Pte Ltd. 2020
J. K. Mandal et al. (eds.), *Algorithms in Machine Learning Paradigms*,
Studies in Computational Intelligence 870,
https://doi.org/10.1007/978-981-15-1041-0_11

1 Introduction

Lorenz attractor is a three dimensional system. This system can be represented by using Ordinary Differential Equation [11]. It has deterministic but provides strange (random) behavior with respect to some initial values [10]. In the last three decades, a lot of researches are carried out to use this strange behavior to solve practical problems in different domains like Cryptography, PRNG, Image Processing, etc. Estimating the initial seed for these type of chaotic map is necessary to minimize the dependency on a single or a set of seeds for making these systems more secure and robust. So, estimating the initial parameters of the chaotic map is a thrust area of research. Estimation techniques can be modeled either in time-domain or in state-space [7]. R. Konnur et al. proposed an all parameter estimation technique in the time-domain by considering the least-squares approach. They used Rossler, Lorenz chaos and Rossler hyperchaos system as an observed and experimental system. They applied this experimental attractor with the estimated parameter in decoding communications [2]. G. Fei et al. proposed a parameter estimation technique for the Lorenz attractor based on a quantum-behaved particle swarm optimization problem. They estimated the value for a single parameter(β) by calculating the average of the outcome of twenty independent experiments. They considered the value for other two parameters(γ, ρ) as the value estimated by [18]. They tried to estimate the value for β near to the value of the aforementioned attractor [12]. Xu et al. modeled parameter estimation techniques for their proposed chaotic map based on Whale Optimization Algorithm and Multi Versed Optimizer algorithm. First they used Gaussian Mixture Model(GMM) to represent the trajectory of their observed attractor into state-space. Then they estimated the parameter of the experimental attractor by minimizing a cost function. This cost function is a likelihood function. It is used for calculating the distance between the time series obtained for the experimental attractor with the GMM obtained for the observed attractor [5]. S.T. Kingni modeled a parameter estimation technique for their proposed chaotic map by minimizing a cost function. To formulate the cost function, they considered the similarity measurement of chaotic time domains of the observed and the experimental attractors [15]. Y. Tang proposed a parameter estimation technique for Logistic chaotic and Mackey-Glass system by using Particle Swarm Optimization. They also considered a observed chaotic map to formulate their objective function by calculating similarity between observed and experimental attractors for estimate the parameter [26]. W. D. Chang proposed a parameter estimation technique for the Chen and Lü system based on a differential evaluation-based approach. They also formulated their objective function by considering the parameter of an observed attractor for calculating the similarity with the experimental attractor. They shown that the estimated parameter is close to the benchmarked parameter of the observed attractor [8]. From the above discussion we can see that all the researches to solve the parameter estimation problem always considers a benchmarked or observed chaotic map as a primary constraint. The outcome of their proposed methods is quite similar to the benchmarked parameter. So, these researches failed to estimate the parameters in broader range. This research considers

the importance of optimizing the seeds for Lorenz systems in broader range. This research doesn't consider a benchmarked parameter for optimizing the initial seed. RCGA is a candid tool to efficiently solve these types of optimization problems with fast execution rate [19, 21]. The optimization is done by using a proposed RCGA technique. This proposed optimization method do not required the time series data of the experimental attractor. So, sensitivity to the initial parameter of the resultant Lorenz attractor will also be preserved.

PRNG is a tool to generate a pseudorandom bit sequence of a finite length. From the last two decades, it can be observed that the researchers had shown strong interest to consider the chaotic behavior of chaotic maps to model their PRNG. Chaotic maps, like Logistic, Skewtent, Tinkerball, Henon, Lorenz, Rossler, Chen, Chebyshev, Quantum, etc. were considered for designing a PRNG [3, 13, 17, 20, 25]. These PRNG's are mainly dependent on predefined seed or a set of seed values. Thus, the outcome of these methods are not unique in each execution. This research also considers the need for generating a unique, optimal and pseudorandom bit sequence. So, in this paper a PRNG based on Lorenz system is proposed. This research shows that a unique, optimal and pseudorandom bit sequence can be generated by using a unique optimized seed for Lorenz system in each execution of the proposed method. NIST specified statistical test is done to check the randomness of the output bit sequence.

Section 2 considers the basic discussion of Lorenz system and the Lyapunov dimension. The proposed optimization technique for Lorenz system is discussed in Sect. 3. In Sect. 4 the proposed algorithm for PRNG is discussed. Section 5 considers the result and analysis of the proposed method. In Sect. 6 the conclusion and future scope of the research are given.

2 Three Dimensional Lorenz System

In 1963, E. Lorenz prescribed a primitive Ordinary Differential Equation(ODE) based mathematical model on atmospheric dynamics. In that paper, authors had discussed that this model is sensitive to initial parameters and provides strange long term behavior on small change of these initial values. Initial parameters consists of three variables namely, σ , β and ρ . σ represents the Prandtl number, ρ represents the Rayleigh number and β represents the height of the fluid layer in convection problem [22]. This mathematical model is represented in Eq. 1, where $\sigma, \beta, \rho > 1$. For $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$, the system functions as a strange attractor [18]. Researchers are using this nonlinear behavior from a linear equation to solve many practical problems [10, 18, 23]. This research considers the finding of optimal initial parameters in each run. It also considers to use these findings to propose a PRNG.

$$\begin{aligned}
\frac{dX}{dt} &= -\sigma X + \sigma Y, \\
\frac{dY}{dt} &= \rho X - Y - XZ, \\
\frac{dZ}{dt} &= \beta Z + XY
\end{aligned}
\tag{1}$$

Right hand side of the Eq. 1 represent the velocity vector. It is used to simulate the moving point of a particle from (X, Y, Z) to another point [14].

2.1 Lyapunov Dimension

Harshness and chaos of a dynamical system are calculated mathematically using Hausdorff dimension. Strange attractor has fractional Hausdorff dimension values [9]. It cannot be measured numerically but can be estimated using Lyapunov dimension [24]. Lyapunov dimension for Lorenz attractor can be calculated using the Eq. 2 [4, 16].

$$D = 3 - \frac{2 \times (\sigma + \beta + 1)}{\sigma + 1 + \sqrt{(\sigma - 1)^2 + 4\rho\sigma}}
\tag{2}$$

3 Proposed Method for Initial Seed Optimization for the Lorenz Attractor

The proposed model is a hybrid approach based on a RCGA technique and three dimensional Lorenz chaotic map. It is an one-step process. n number of seed values in real numbers for the Lorenz chaotic map are taken randomly to represent the initial population. The seed for the Lorenz chaotic map consists of three parameters, namely σ , β , and ρ . For each seed in the initial population, the value of σ will be in the range 8 to 13, the value of β will be in the range 3 to 5, and the value of ρ will be in the range 26 to 41. The Crossover Probability($Cross_p$) and the Mutation Probability($Mute_p$) are user given input. This algorithm is executed for 200 generations. It is seen that this algorithm gives satisfactory outcomes in each execution, with the aforementioned prerequisites. The proposed method is presented in Algorithm 1.

By using Eq. 2, the Lyapunov dimension for the seed ($\sigma = 10, \beta = 8/3, \rho = 28$) calculated is 2.4013. This proposed method will give an optimized seed. The Lyapunov dimension of this seed will be close to the Lyapunov dimension 2.4013. In this algorithm, firstly the fitness for each chromosome is evaluated by calculating the Lyapunov dimension of each chromosome. After that elitism operation is performed. In this operation, worst chromosome from the current generation and the fittest chromosome from the previous generation are selected first. Then a comparison on

Algorithm 1 Optimal seed for Lorenz attractor optimization

Input: Number of chromosomes(n), Initial mating pool(m)
Output: Optimized seed value for Lorenz attractor(Opt_{seed})

- 1: **begin**
- 2: Initial chromosomes are encoded by randomly taken real values for σ, β and ρ , i.e. the initial parameters for Lorenz systems
- 3: **for** $g= 1$ to 200 **do** ▷ g =Number of Generations
- 4: **for** each chromosome (j) from the M **do**
- 5: fitness[j]= Calculate Lyapunov dimension by using Equation 2. ▷ Fitness Evaluation
- 6: **end for**
- 7: Perform Elitism operation.
- 8: Using Restricted Binary Tournament Selection, generate a mating pool of same size as of m .
- 9: Perform real coded Crossover operation on the mating pool by using Algorithm 2.
- 10: Perform real coded Mutation operation on the mating pool by using Algorithm 3.
- 11: **end for**
- 12: Select the chromosome with the lowest fitness value from the final mating pool and store it as Opt_{seed} .
- 13: **end**

the basis of their fitness value is done. The chromosome with the lowest fitness value is selected or survived in the current generation. Another one is rejected for survival. After that a restricted Binary Tournament Selection is done. In this operation a criterion for selection is imposed, i.e., the selected chromosome should have the fitness value greater than or equal 2.4013. So, for selection there will be below mentioned three possibilities:

1. If the fitness values for both of the chromosomes are greater than or equal to the aforementioned threshold, then the chromosome with the lower fitness value will get selected.
2. If the fitness value of one chromosome is less than the threshold, and the other one is greater than the threshold, then the chromosome with the fitness value greater than the threshold will get selected.
3. If the fitness value for both of the chromosomes is lower than the threshold, then random selection will be repeated until anyone from the above two conditions satisfied.

After selection, the size of the mating pool will be n . After that crossover operation is performed on that mating pool. The proposed crossover technique on real coded chromosome is presented in Algorithm 2.

In this method $Cross_p$, is the crossover probability is taken as user input. This experiment consider the value for $Cross_p = 0.8$. In this algorithm, two chromosomes (X_1 and X_2) from the population are chosen randomly. After that a probabilistic value C_p is chosen randomly. If this C_p is less than or equal to the $Cross_p$, then the selected chromosome will go for crossover, otherwise they will go directly to the output mating pool(M_{new}). In this crossover operation, first the differences between the seed element of both of the chromosome are calculated, i.e. $D_1 = \sigma_{X_1} - \sigma_{X_2}$, $D_2 = \beta_{X_1} - \beta_{X_2}$ and $D_3 = \rho_{X_1} - \rho_{X_2}$. Then three fractional values, X, Y and Z are

Algorithm 2 Proposed real coded crossover technique

Input: Mating pool(m), Number of chromosome(n), Crossover probability($Cross_p$)

Output: Offspring(M_{new})

```

1: begin
2: for  $k=1$  to  $n/2$  do
3:   Randomly Select two chromosomes  $C_1$  and  $C_2$  from the mating pool.
4:   Randomly Select a probabilistic value  $C_p$ .
5:   if  $C_p \leq Cross_p$  then
6:     Calculate the absolute difference between  $\sigma, \beta$  &  $\rho$  of both of the chromosomes and store
       it in  $D_1, D_2, D_3$ .
7:     Select three random values  $x, y, z$  within the range  $0$  to  $D_1, 0$  to  $D_2$  &  $0$  to  $D_3$  respectively.
8:     Randomly select an integer value for  $opt$ .  $\triangleright 1 \leq opt \leq 2$ 
9:     if  $opt = 1$  then
10:       Perform addition operation of  $x$  and  $\sigma, y$  and  $\beta, z$  and  $\rho$ .  $\triangleright \sigma, \beta, \rho \in C_1$ 
11:       Perform absolute subtraction operation of  $x$  and  $\sigma, y$  and  $\beta, z$  and  $\rho$ .  $\triangleright \sigma, \beta, \rho \in C_2$ 
12:     else
13:       Perform absolute subtraction operation of  $x$  and  $\sigma, y$  and  $\beta, z$  and  $\rho$ .  $\triangleright \sigma, \beta, \rho \in C_1$ 
14:       Perform addition operation of  $x$  and  $\sigma, y$  and  $\beta, z$  and  $\rho$ .  $\triangleright \sigma, \beta, \rho \in C_2$ 
15:     end if
16:     Boundary restriction is done here.
17:   end if
18:   Store  $C_1$  and  $C_2$  in  $M_{new}$ 
19: end for
20: end

```

selected randomly, where, $0 \leq X \leq D_1, 0 \leq Y \leq D_2$ and $0 \leq Z \leq D_3$. After that select a integer value Opt randomly, where $1 \leq Opt \leq 2$. If the value for Opt is 1, then X, Y and Z are added to the σ, β and ρ of the first chromosome respectively and X, Y and Z are subtracted to the σ, β and ρ of the second chromosome. Otherwise the process will be performed in reverse order. The output of the aforementioned process is the offsprings, which will go to the resultant mating pool. This crossover process will run for $\frac{n}{2}$ times. After that mutation operation is performed. The proposed method for mutation on real coded chromosome is presented in Algorithm 3.

In this method $Mute_p$, is the mutation probability is taken as user input. This experiment consider the value for $Mute_p = 0.1$. In this technique, first a chromosome from the input mating pool is selected randomly. Then a probabilistic value P_m is chosen randomly. If P_m is less than or equal to the $Cross_p$ then the chromosome will get mutated, otherwise it will go directly to the output mating pool as an offspring. In a mutation operation a probabilistic value (X) is chosen randomly, where, $0 \leq X \leq P_m$. Then one integer value (Opt) is also chosen randomly, where, $1 \leq Opt \leq 2$. If the value of Opt is equals to 1 then, $(X \times 100)\%$ of values from σ, β, ρ are calculated and then added with them respectively. For example, $\sigma = \sigma + \sigma \times (X \times 100)\%$. Otherwise $(X \times 100)\%$ of values from σ, β, ρ are calculated and then subtracted with them respectively. The output of the aforementioned process is an offspring, which will go to the resultant mating pool. The mutation process will be executed for $\frac{n}{2}$ times.

Algorithm 3 Proposed real coded mutation technique

Input: Mutation probability(P_m), Mating pool(m), Number of chromosomes(n)

Output: Offspring(M_{new})

```

1: begin
2: for k=1 to  $n/2$  do
3:   Randomly Select one chromosome  $C_1$  from  $m$ .
4:   Randomly select a probabilistic value  $Mute_p$ .
5:   if  $Mute_p \leq P_m$  then
6:     Randomly select the value for  $X$ .  $\triangleright 0 \leq X \leq P_m$ 
7:     Randomly select a integer value  $Opt$ .  $\triangleright 1 \leq Opt \leq 2$ 
8:     if  $Opt = 1$  then
9:       Modify the values for  $\sigma, \beta$  and  $\rho$  of  $C_1$  by subtracting  $(X \times 100)\%$  values from them.
10:    else
11:      Modify the values for  $\sigma, \beta$  and  $\rho$  of  $C_1$  by adding  $(X \times 100)\%$  values to them.
12:    end if
13:  end if
14:  Store  $C_1$  in  $M_{new}$ .
15: end for
16: end

```

The aforementioned process will execute for 200 generations. After that, the chromosome from the resultant mating pool, with the lowest fitness value is selected as an optimal seed (Opt_{seed}). Opt_{seed} will be used as an input to the proposed PRNG, which is discussed in Sect. 4. The results of this experiment are discussed in Sect. 5.

4 Proposed Method for Lorenz System Based PRNG

The proposed method for Lorenz system based PRNG is presented in Algorithm 4. This method will produce a q bit pseudorandom sequence($bits_{seq}$) by using Opt_{seed} as a seed for the Lorenz attractor. Lorenz attractor equation can be solved by using Runge–Kutta ODE solvers. Dormand–Prince fourth and fifth order method belong to Runge–Kutta family of ODE solvers [1]. This research considers Dormand–Prince fourth and fifth order method to solve the Lorenz attractor equation (1). In this proposed method, first the beginning and end limit for the integration of the Dormand–Prince fourth and fifth order method is set to 0 and q respectively. Then the initial points for the Lorenz attractor trajectories are initialized to [1, 1, 1]. Then the next point(O_{vec}) of the trajectories is calculated by using Dormand–Prince fourth and fifth order Ordinary Differential Equation(ODE) solver.

Then a probabilistic value(J) and a integer value(K) are chosen randomly, where, $0 \leq J \leq 1$ and $1 \leq K \leq 10$. O_{vec} contains three values for the three dimensions, these three values are mapped to a probabilistic value(f) by using Eqs. 3 and 4. This process will be executed for q number of times, to get q numbers of probabilistic values. After that the mean value of these probabilistic values is calculated. Then each

Algorithm 4 Proposed method for Lorenz attractor based PRNG

Input: Opt_{seed} , The length of the output bit sequence(q)

Output: q bit length of output bit sequence(bit_{sec})

1: **begin**

2: Set the beginning and end limit of the integration by 0 and q respectively.

3: Set the initial point or the output vectors(O_{vec}) of the Lorenz attractor trajectories to $[1, 1, 1]$.

4: **for** $i=1$ to q **do**

5: Calculate the next point in trajectories or $O_{vec}(i)$ by using Dormand-Prince 4th and 5th order Ordinary Differential Equation(ODE) solver.

6: Select a probabilistic value J randomly. $\triangleright 0 \leq J \leq 1$.

7: Select an integer value K randomly. $\triangleright 1 \leq K \leq 10$

8: Calculate the value for $temp$ by using Equation 3.

9: calculate $f(i)$ by using Equation 4.

10: **end for**

11: Calculate the mean value(m) of f .

12: Map each fractional values from f to a binary value by using the Equation 5.

13: **end**

of these fractional value is mapped to a binary value by using Eq. 5, to get a q length of bit sequence (bit_{sec}).

$$temp = \frac{\sum_{n=1}^q O_{vec}(i) \times J}{K} \quad (3)$$

$$f(i) = (\lceil temp \rceil) - temp \quad (4)$$

$$bit_{seq}(i) = \begin{cases} 0 & \text{if, } f_i < mean(f) \\ 1 & \text{Otherwise.} \end{cases} \quad (5)$$

5 Results and Analysis

The outcome of the execution of the proposed Algorithm 1 is an optimized seed. In each execution, this method provides a unique optimized seed. For example, seeds like, $Opt_{seed_1}(\sigma = 11.719533166824359, \beta = 3.247502492906329, \rho = 32.745210572679130)$, $Opt_{seed_2}(\sigma = 10.697153652222013, \beta = 3.695306541101536, \rho = 34.681133008510570)$ are the resultant optimized seeds of this proposed method. The Lyapunov dimension of Opt_{seed_1} is 2.4013 and for the Opt_{seed_2} is 2.40131742527066. The attractor for the Lorenz system with the seed Opt_{seed_1} is presented in Fig. 1. The attractor for the Lorenz system with the seed Opt_{seed_2} is presented in Fig. 2. It can be observed that both of the attractors showcase the well-known butterfly effect. The generation wise improvement in terms of the objective function over 200 generations of Algorithm 1 is shown as a bar graph in Figs. 3 and 4. Here,

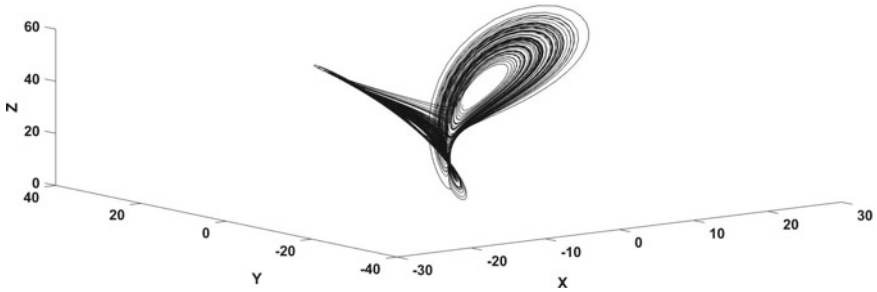


Fig. 1 Attractor of the Lorenz system with: $\sigma = 11.719533166824359$, $\beta = 3.247502492906329$, $\rho = 32.745210572679130$

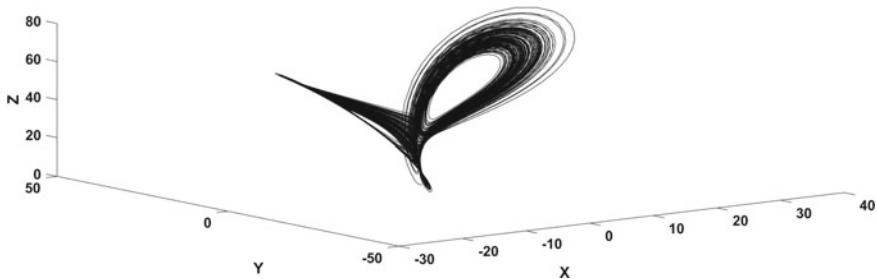


Fig. 2 Attractor of the Lorenz system with: $\sigma = 10.697153652222013$, $\beta = 3.695306541101536$, $\rho = 34.681133008510570$

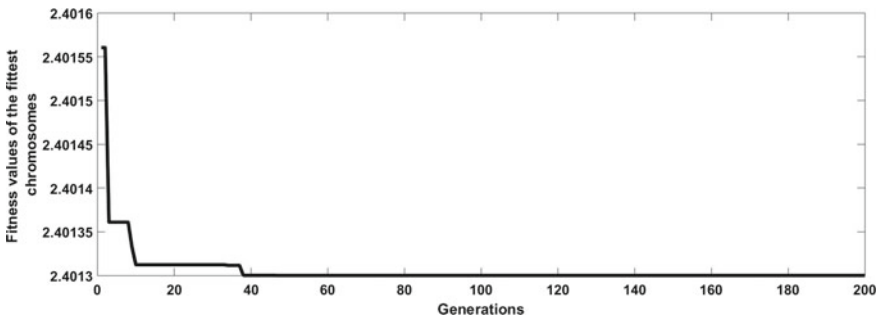


Fig. 3 Generation wise improvement in different execution with: $\sigma = 11.719533166824359$, $\beta = 3.247502492906329$, $\rho = 32.745210572679130$

the X-axis represents the Generations and the Y-axis represents the fitness value of the fittest chromosome in each generation. From this bar graph, it can be observed that the proposed method minimizes the fitness value over the generations.

The randomness of the output bit sequence of the proposed PRNG (Algorithm 4) is tested by using NIST statistical test suit for randomness [6]. NIST specified 15 tests are done on the output bit sequence. An output bit sequence of different

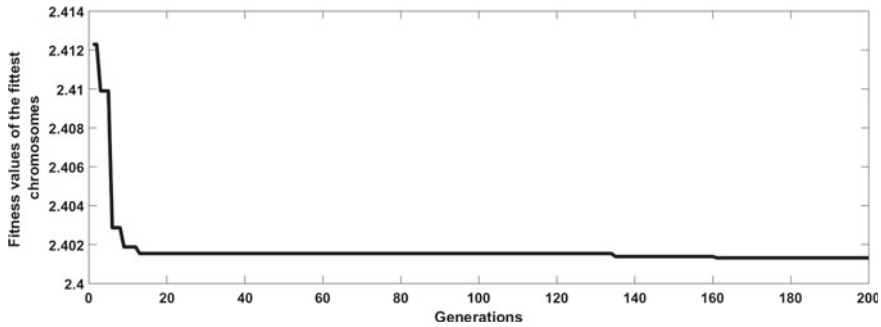


Fig. 4 Generation wise improvement in different execution with: $\sigma = 10.697153652222013$, $\beta = 3.695306541101536$, $\rho = 34.681133008510570$

execution is tested and each of them passes through all the tests. The results of these 15 statistical tests are given in Table 1.

P-value is a probabilistic value. The output p-value for a test should be greater than or equal to 0.01 to pass that test. From Table 1, it can be observed that all the output probabilistic values (*P – values*) of all tests are greater than or equal 0.01. So, with 99% confidence, it can be concluded that the output bit sequence of this proposed PRNG is pseudorandom [6].

Table 1 NIST test result for the proposed method

Index	Test index	P-value	Result
1	Frequency	0.5762	Pass
2	Block frequency	0.1565	Pass
3	Runs	0.5395	Pass
4	Longest-Run-of-Ones	0.2660	Pass
5	Binary matrix rank	0.2701	Pass
6	DFT (Spectral)	0.5164	Pass
7	Non overlapping template matching	0.3663	Pass
8	Overlapping template matching	0.5578	Pass
9	Maurers <i>Universal statistical</i>	0.3282	Pass
10	Linear complexity	0.4815	Pass
11	Serial 1	0.7847	Pass
	Serial 2	0.5273	Pass
12	Approximation entropy	0.0223	Pass
13	Cumulative sums (Cusums)	0.8595	Pass

(continued)

Table 1 (continued)

Index	Test index	P-value	Result
14	Random excursion		
	-4	0.9996	Random
	-3	0.9991	Random
	-2	0.9970	Random
	-1	0.9626	Random
	1	0.9626	Random
	2	0.9970	Random
	3	0.9991	Random
	4	0.9996	Random
15	Random excursions variant		
	-9	0.5211	Random
	-8	0.4945	Random
	-7	0.4631	Random
	-6	0.4588	Random
	-5	0.4497	Random
	-4	0.4751	Random
	-3	0.4990	Random
	-2	0.4450	Random
	-1	0.1859	Random
	1	0.1859	Random
	2	0.5854	Random
	3	0.7353	Random
	4	0.3914	Random
	5	0.3778	Random
	6	0.5688	Random
	7	0.8751	Random
8	0.8453	Random	
9	0.8545	Random	

6 Conclusion and Future Scope

Lorenz system acts as a strange attractor, depending on the initial parameters or seed. This research has shown that seed for an experimental Lorenz system can be optimized by using evolutionary search algorithms without considering an observed attractor. Lorenz systems with these optimized initial parameter behaves like a strange attractor. This research has also shown that without using time-domain or state-space approach the initial parameter of a Lorenz attractor can be estimated using an optimization approach based on the Lyapunov dimension based objective function. A PRNG based on the estimated initial parameter based Lorenz attractor is proposed.

This PRNG generates a pseudorandom bit sequence of finite length. The randomness of the bit sequence is tested by using NIST specified statistical test suit. The output of this proposed PRNG passed the aforementioned tests with a 99% confidence.

This research considers the initial parameter estimation of the three dimensional Lorenz attractor. In future optimization of initial parameter for the four-dimensional and five-dimensional chaotic attractors can also be done by using evolutionary algorithms. A industry-standard system can also be developed, which can optimize initial parameters for any type of (One-dimensional to Five-dimensional) chaotic attractor.

References

1. Dormand JR, Prince PJ (1980) A family of embedded rungekutta formulae. *J Comput Appl Math* 6(1):19–26
2. Konnur R (2003) Synchronization-based approach for estimating all model parameters of chaotic systems. *Phys Rev E, Stat Nonlinear Soft Matter Phys* 67:027,204–1–027,204–4
3. Akhshani A, Akhavan A, Mobaraki A, Lim SC, Hassan Z (2014) Pseudo random number generator based on quantum chaotic map. *Commun Nonlinear Sci Numer Simul* 19(1):101–111
4. Leonov GA, Kuznetsov NV, Korzhemanova NA, Kusakin DV (2016) Lyapunov dimension formula for the global attractor of the lorenz system. *Commun Nonlinear Sci Numer Simul* 41:84–103
5. Xu G et al (2018) A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation. *Entropy* 20(1):86:1–86:23
6. Bassham LE et al (2010) Sp 800-22 rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., Gaithersburg, MD, United States
7. Jafari S et al (2014) A new cost function for parameter estimation of chaotic systems using return maps as fingerprints. *Int J Bifurcation Chaos* 24(10):1450,134:1–1450,134:18
8. Chang WD (2007) Parameter identification of chen and lü systems: a differential evolution approach. *Chaos Solut Fractals* 32:1469–1476
9. David A, Russell EO, James D, Hanson, (1980) Dimension of strange attractors. *Phys Rev Lett* 45(14):1175–1178
10. Farmer JD (1982) Chaotic attractors of an infinite-dimensional dynamical system. *Phys D: Nonlinear Phenom* 4(3):366–393
11. Fowler A, Gibbon J, McGuinness M (1982) The complex lorenz equations. *Phys D: Nonlinear Phenom* 4(2):139–163
12. Gao Fei THQ, Li, Z-Q (2008) Parameters estimation online for lorenz system by a novel quantum-behaved particle swarm optimization. *Chinese Phys B* 17:1196–1201
13. Hu H, Liu L, Ding N (2013) Pseudorandom sequence generator based on the chen chaotic system. *Comput Phys Commun* 184(3):765–768
14. Immler F (2018) A verified ode solver and the lorenz attractor. *J Autom Reason* 61(1):73–111
15. Kingni ST, Jafari HSPWS (2014) Three-dimensional chaotic autonomous system with only one stable equilibrium: analysis, circuit design, parameter estimation, control, synchronization and its fractional-order form. *Eur Phys J Plus* 129:1–16
16. Leonov GA (2001) Lyapunov dimension formulas for hénon and lorentz attractors. *Algebra i Analiz* 3:155–170
17. Liu L, Miao S, Cheng M, Gao X (2016) A pseudorandom bit generator based on new multi-delayed chebyshev map. *Inf Process Lett* 116(11):674–681
18. Lorenz EN (1963) Deterministic nonperiodic flow. *J Atmosph Sci* 20(2):130–141

19. Mukhopadhyay S, Tamal Datta Chaudhuri JKM (2017) A hybrid pso-fuzzy based algorithm for clustering indian stock market data. (eds) Computational Intelligence, Communications, and Business Analytics CICBA 2017 Communications in Computer and Information Science 776:475–487
20. Stoyanov B, Kordov K (2015) Novel secure pseudo-random number generation scheme based on two tinkerbells. *Adva Stud Theor Phys* 9(9):411–421
21. Sanjib Ganguly DS (2015) Distributed generation allocation on radial distribution networks under uncertainties of load and generation using genetic algorithm. *IEEE Trans Sustain Energy* 6(3):688–697
22. Strogatz SH (2014) *Nonlinear dynamics and chaos with applications to physics, biology, chemistry, and engineering*, 2nd edn. CRC Press, Boca Raton, Florida, USA
23. Tucker W (2002) A rigorous ode solver and smale's 14th problem. *Found Comput Math* 2(1):53–117
24. Tucker W (2019) A pseudo random bit generator based on a modified chaotic map. *Int J Netw Secur* 21(3):402–408
25. Ünal Çavuşoğlu AASJSK Shirin Panahi (2019) A new chaotic system with hidden attractor and its engineering applications: analog circuit realization and image encryption. *Analog Integr Circuit Signal Process* 98(1):85–99
26. Yinggan Tang GX (2009) Parameter estimation for time-delay chaotic system by particle swarm optimization. *Chaos Solut Fractals* 40:1391–1398

Author Index

B

Baidya, Arunima, [43](#)
Biswas, Animesh, [1](#)
Biswas, Arindam, [91](#)
Bose, Mahua, [57](#)

C

Chakrabarti, Amlan, [165](#)

D

Dam, Santanu, [109](#)
Das, Abhirup, [25](#)
Das, Amit Kumar, [25](#)
Dasgupta, Kousik, [109](#)
Dutta, Paramartha, [109](#), [129](#), [147](#), [165](#)

J

Jana, Susovan, [71](#)

K

Kundu, Sagarika Saroj, [25](#)

M

Mali, Kalyani, [57](#)
Mandal, Gopa, [109](#)
Md, Nasir, [129](#), [147](#)
Mondal, Kartick Chandra, [43](#)
Mukhopadhyay, Somnath, [181](#)

N

Nandi, Avishek, [129](#), [147](#)
Nandy, Biswadeep Deb, [43](#)

P

Pal, Arindrajit, [165](#)
Pal, Ramen, [181](#)
Parekh, Ranjan, [71](#)

R

Roy, Nilanjana Dutta, [91](#)

S

Sarkar, Arun, [1](#)
Sarkar, Bijan, [71](#)
Singh, Jyoti Prakash, [165](#)