Received: May 12, 2023. Revised: June 10, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Email Net: Efficient Email Classification Approach Based on Graph Similarity Measure

Aruna Kumara B[1]* **Mallikarjun Kodabagi M[1]**

[1]*School of Computing and Information Technology, REVA University, Bangalore, Karnataka, India*
* Corresponding author's Email: aruna777kumara@gmail.com

Abstract: In this Internet Era, Email is a vital form of communication for many academic, personal, and professional users. Despite the availability of alternative means of communication, such as social networks, electronic messages, and mobile apps, email remains an integral part of communication. Email auto-management techniques are necessary for a variety of reasons, such as saving users valuable time, dealing with high-dimensional data, and making email communication easier and more accessible. In this work, a novel email net (improved elephant herd optimization and Graph similarity with the Jaccard index) technique has been proposed for efficient email classification based on graph similarity measure. Initially, the dataset is pre-processed using NLP techniques such as removing email signatures, removing punctuations, removing stop-words, lowercase conversion, tokenization, and stemming for removing irrelevant data. After pre-processing the feature are extracted using bag of words and term frequency – inverse document frequency (TF-IDF). These extracted features are given as input to improved Elephant herding optimization (EHO) for selecting the most relevant features to build a graph-based similarity index for classifying each category of e-mail. The proposed email net was tested on a benchmark dataset and a real-time dataset. Also, the proposed method's performance is compared with other classifiers. According to the experimental results, the proposed approach outperforms all other classifiers with a 98.82% of accuracy.

Keywords: Graph similarity measures, Multiclass classification, Graph classification, Elephant herding optimization.

1. Introduction

Email is used to transmit information on a personal and professional level. The usage of email communications is multiplying even after the availability mobile applications and social media platforms. Especially, after the pandemic, email communication in the education sector increased dramatically as learning shifted to an online mode. According to estimates, 333.2 billion people use email globally as of 2022; by the end of 2025, that number is predicted to reach 376.4 billion [1].

A typical user gets almost 30-50 emails per day, which leads to a flood of emails if a person goes on 15-20 days of vacation. Additionally, users have to allocate a sizable portion of their working hours to dealing with emails. email management is a crucial responsibility shared by both individuals and organizations. For instance, in an academic university, one can classify an incoming email into academics, research, placements, examinations, and others for easy access.

Many machine-learning (ML) techniques exist to classify emails into predefined categories, such as supervised ML, semi-supervised ML, unsupervised ML, content-based learning, and statistical learning, [2]. Some of the algorithms used supervised learning concept are support vector machine (SVM), genetic algorithms (GA) [3], decision trees (DT) [4], random forest (RF) [5], Naïve bayes (NB) [6], k-nearest neighbor (KNN) [7], and artificial neural network (ANN) [8]. The unstructured, noisy, and highly dimensional data in each email makes it difficult to develop an email classifier for a real-time dataset. However, many of these classifiers have produced noticeable outcomes across a range of datasets. Additionally, a few innovative email classifiers were developed, such as semantic-based classifiers [9], tree-based classifiers [10], and graph-based classifiers [11], to address these issues. But improvement is still required when applied to real-

time datasets. A graph-based or tree-based classifier has gained a lot of attention recently for its non-linear properties, which make it adaptable to solving any type of classification problem.

This paper proposes a novel Email Net for efficient email classification-based similarity measures. The following are the key contributions of this work:

- The primary purpose of this work is to present a novel EmailNet for an Efficient Email Classification technique based on graph similarity measure
- Initially, the dataset is pre-processed using NLP languages such as removing email signatures, removing punctuations, removing stop-words, lowercase conversion, tokenization, and stemming for removing irrelevant data.
- The relevant features are extracted using various feature extraction methods like bag of words and term frequency – inverse document frequency (TF-IDF).
- The extracted features are given as input to improved Elephant herding optimization (EHO) for selecting the most relevant features to build a graph-based similarity index for classifying emails into Academic, examination, research, and placement.
- Several factors were assessed to evaluate the proposed method based on precision, accuracy, recall, and F-measure.

The remainder section of this paper is arranged as follows: section 2 explains the related works. Section 3 discusses the proposed EmailNet methodology, which describes building graphs, finding the node similarity between graphs, and grouping emails into predefined categories. Section 4 discusses the results obtained after rigorous experiments on real-time and benchmark datasets. Finally, section 5 concluded the work.

2. Related works

This section describes several studies such as Deep learning (DL) NLP, and ML, and classifies email into various predefined categories This section provides a brief overview of some of the most recent studies.

2.1 Review on email classification techniques

In 2022 Qi Li et al. [12] developed a phishing detection strategy for large email datasets based on long short-term memory (LSTM). The suggested model comprised two important steps: an extension phase for samples and the testing phase. The method combined KNN and K-Means during the extension phase to boost the training sample count to encounter the requirement of learning. Then pre-processing was applied to these data before testing. Later, the pre-processed data was used to train the LSTM model. Finally, the technique used the trained model to categorize phishing emails.

In 2022 Fernández, J.M.et al [13] presented a multi-class E-mail classification based on feature selection and information retrieval. The primary features are chosen for each class from an initial data set of manually labeled emails using three techniques: TF-IDF, logistic regression, and SS3. Documents and search engines are used for indexing the rest of the cases.

In 2022 Hosseinalipour, A. and Ghanbarzadeh, R.,[14] presented a horse herd optimization algorithm for spam detection with increased accuracy and a minimum error rate. First, a discrete algorithm was created from the continuous HOA. The inputs of the resulting algorithm were subsequently transformed from opposition-based to multiobjective. Finally, it was applied to the discrete, multiobjective problem of spam identification.

In 2022 Iqbal, K. and Khan, M.S.,[15] presented an ML-based email classification. In the first stage, data is gathered from the spam base UCI database. The second stage involves normalizing all the dataset's properties with a wider range of values. Point-Biserial correlation, a feature selection technique, is used in the third phase. Finally, eight machine learning classifiers are then applied to selected attributes.

In 2021 Bhatti, P., et al [16] presented an LSTM for email classification into four classes: harassment, suspicious, fraudulent, and normal. The email classes have been determined using a sampling strategy and LSTM. Moreover, the input dataset has been attempted to be balanced using oversampling techniques. The suggested model achieves greater than 90% accuracy.

In 2020 Deshmukh, S et al [17] developed a machine learning for email classification based on text content. The text is pre-processed using NLP techniques such as tokenization and stemming are analyzed. The textual content is used to create feature vectors in the feature construction and representation module. Finally, the email is classified using the NB classifier and logistic regression. The Naive bayes classifier outperforms other models and achieves an accuracy of 77%

Table 1. Comparative analysis with existing techniques

Author with Year	Techniques	Advantages	Disadvantages
Qi Li et al. [14] (2022)	LSTM	The suggested model attains an accuracy of 95% and effectively classifies phishing emails	Yet LSTM cannot perform parallelly.
Fernández, J.M.et al [15] (2022)	Feature Selection and Information Retrieval	The data in this model are retrieved automatically	However, this model only classifies labels that have been trained
Hosseinalipour, A. and Ghanbarzadeh, R.,[16] (2022)	horse herd optimization algorithm	This model attains lower computational complexity	Yet the suggested model has a limited dataset
Iqbal, K. and Khan, M.S.,[17] (2022)	machine learning classifiers	This model effectively classifies email with eight machine classifiers and attains an accuracy of 98.06%	However, the model can also increase more features and size of the dataset
Bhatti, P., et al [18] & (2021)	LSTM	This method efficiently balances the input dataset using over-sampling.	On large datasets, LSTM can be slow to train.
Deshmukh, S et al [19] & (2020)	Naive Bayes classifier and logistic regression	This method is developed as a stand-alone application that can be scaled up or down depending on the organization's needs	Yet NB shouldn't take its probability outputs effectively.
Sharaff, A. et al [20] & (2019)	Extra tree classifier	The suggested method attains an accuracy of 95.5%	Yet this model has time complexity.
Sharaff, A. et al [21] (2019)	latent Dirichlet allocation	Emails should be managed systematically when the topic is vague.	However, this model cannot identify multiple topics in emails
Kusuma, P.D. and Kallista, M. [25] (2023)	Quad Tournament Optimizer	This algorithm successfully found the global optimal solution	Adding more methods to the tournament will improve QTO
Kusuma, P.D. and Novianty, A., [26] (2023)	Multiple Interaction Optimizer	This algorithm effectively solves Order Allocation Problem with minimum cost and lateness	However, this algorithm is not suitable for multiple objective order allocation problems.
Zeidabadi, F.A. and Dehghani, M., [27] & (2022)	Puzzle optimization algorithm	POAs do not require parameter setting since they have no control parameters.	Yet this model provides a slow convergences rate
Kusuma, P.D. and Prasasti, A.L., [28] & (2022)	Guided Pelican Algorithm	This algorithm reduces portfolio problem	However, this algorithm is 1% worse than the Pelican optimization algorithm
Kusuma, P.D. and Kallista, M., [29] & (2022)	Stochastic Komodo Algorithm	The suggested algorithm is very competitive in both unimodal and multimodal functions	Yet this algorithm can give a more comprehensive evaluation

In 2019 Sharaff, A. et al [18] presented an email classification technique using an extra tree classifier with a metaheuristics model for treating spam messages as ham. Initially, pre-processing techniques with normalization are applied to the dataset. After pre-processing the most relevant features are extracted using PSO, BPSO, and GA. The extra-tree classifier is used to extract the chosen features and divide them into ham and spam emails. The suggested method attains 95.5 % of accuracy. yet this model has time complexity.

In 2019 Sharaff, A. et al [19] developed an email categorization based on latent Dirichlet allocation for identifying categorical terms. First, the email data are pre-processed using NLP techniques such as stemming and removing stop words. The textual similarity technique is used to create clusters in the second step. LDA is used to categorize terms after the cluster is formed, and the frequent terms are then calculated. Yet this model cannot identify multiple topics of emails.

2.2 Review on optimization strategies

In 2023 Kusuma, P.D. and Kallista, M.[25] presented a Quad Tournament Optimizer based on four searches that perform in each iteration. The four

searches are neighbourhood searches around the corresponding solution and the global best solution, searches relative to a randomly chosen solution, searches toward the centre between the randomly selected solution and the global best solution, and searches toward the global best solution.

In 2023 Kusuma, P.D. and Novianty, A., [26] presented a Multiple Interaction Optimizer for solving Order Allocation Problem. Initially, agents interact with a few randomly chosen agents from the population. Every contact includes a guided search. Each agent does a local search in the second phase, which linearly shrinks the search space for the iteration.

In 2022 Zeidabadi, F. A. and Dehghani, M., [27] developed a Puzzle optimization algorithm (POA) for solving a puzzle game. The POA advocates statistically simulating the process of solving a challenge as an evolutionary optimizer. The POA does not require parameter configuration because it has no control parameters.

In 2022 Kusuma, P.D. and Prasasti, A.L., [28] presented a Guided Pelican algorithm which is the improvements of the (POA) pelican optimization algorithm, that replicates the pelican birds' hunting behavior. The global best solution is used by GPA as a deterministic target in the beginning, replacing the randomized target. In addition, when calculating local search space size, GPA swaps out the pelican's present location for the size of the search space. Thirdly, GPA uses numerous candidates in both phases, as it did in the original POA.

In 2022 Kusuma, P.D. and Kallista, M., [29] described a stochastic Komodo algorithm which is derived from the behavior of Komodo during foraging and mating calls. Three different Komodo species make up this algorithm: big male, female, and little male. While female Komodos carry out diversification based on the search space radius, males concentrate on intensification. At the start of the iteration, the sorting mechanism is removed and a random distribution of the Komodo is undertaken.

From the above-related works, most of the research is based on the classification of spam email, classification phishing, and ML classifier and some showed noticeable results on various datasets. Additionally, graph-based/tree-based classifiers have recently gained more popularity. Though many graph-based/tree-based classifiers were developed, more research needs to be done on graph-based classifiers on the real-time dataset. Hence, this paper proposes an Email Net for an efficient email classification system based on a graph-based node similarity method.

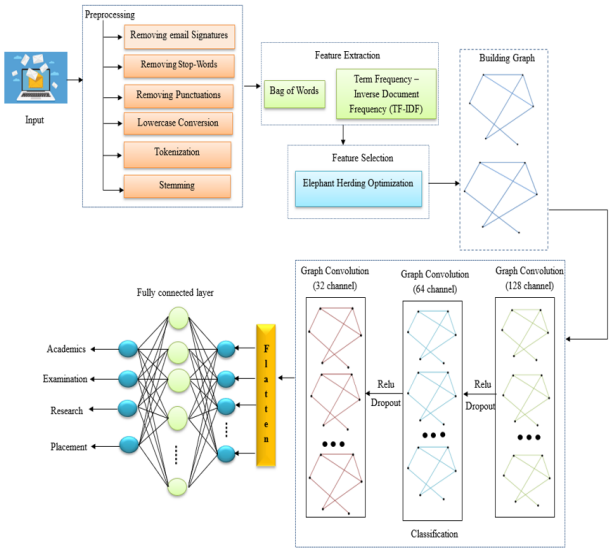


Figure 1. The architecture proposed EmailNet

3. Proposed email net methodology

In this work, a novel email net has been proposed for efficient email classification based on graph similarity measure. The architecture proposed Email Net for the email classification method is illustrated in Fig. 1. Initially, the dataset is pre-processed using NLP techniques such as removing email signatures, removing punctuations, removing stop-words, lowercase conversion, tokenization, and stemming for removing irrelevant data. After pre-processing the feature are extracted using term frequency – inverse document frequency (TF-IDF) and Bag of Words. These extracted features are given as input to improved Elephant herding optimization (EHO) for selecting the most relevant features to build a graph-based similarity index for classifying each category of e-mail.

3.1 Data acquisition

The REVA University email database was used to construct the key datasets for the first experiment. To analyse the dataset, three thousand emails were divided into four categories: placement, examination, Academics, and research. Training and testing samples were split according to an 80:20 ratio for the experiment. Emails from various classes are distributed unevenly in the collection.

3.2 Data pre-processing

Data pre-processing is required to convert the raw email data into the format required. The dataset is pre-processed using NLP techniques such as removing email signatures, removing punctuations, removing stop-words, lowercase conversion,

Table 2. Example of pre-processing

Original Text	After Processing
Hearty congratulations to the authors, who published a research article in UGC indexed journal.	{hearty, congrats, author, publish, research, article, ugc, index, journal}
Dear All, I am happy to inform that, a research article authored by me is published in SCI indexed journal with impact factor 1.8	{happy, inform, research, article, author, publish, sci, index, journal, impact, factor}
Hearty congratulations to you and your research supervisor on having published a research article in SCOPUS indexed journal	{hearty, congrats, research, supervisor, publish, research, article, scopus, index, journal}
Dear All, I am happy to inform that the research article authored by me and my research scholar has been accepted for publication in Web of Science indexed journal.	{happy, inform, research, article, author, research, scholar, accept, publication, web, science, index, journal}

Table 3. Binary bag of words model

Doc/ Word	publish	journal	index	research	author	article	accept	Happy	hearty	congrats
D1	1	1	1	1	1	1	0	0	1	1
D2	1	1	1	1	1	0	0	1	0	0
D3	1	1	1	1	0	0	0	0	1	1
D4	0	1	1	0	1	1	1	1	0	0

tokenization, and stemming for removing irrelevant data [24].

Removing email signatures

The email's body contained the signature section, that is appended to every email message. Email signatures include information such as regards, associate, assistant, email, etc. During email classification, all of this information should be removed.

Removing punctuations

The unwanted punctuation marks such as semicolons, brackets, parentheses, quotation marks, etc. were removed from each email.

Removing Stop Words

A stop word is a word that appears frequently in a document but does not contribute to identifying the main contents of the document, such as "a", "an" or "the".

lowercase conversion

The data in the emails were converted to lowercase.

Tokenization

In tokenization, the text is disassembled into valuable data while its meaning is maintained. This stage involves dividing lengthy paragraphs, commonly known as chunks of text, tokens, which are sentences. Further, these sentences can be broken down into words.

Stemming

Stemming eliminates unnecessary calculations by converting words from multiple tenses to their basic forms.

The volume of email data has significantly decreased after pre-processing. Four email documents belonging to a research category in the dataset were considered to show the working methodology of the proposed method.

Table 2 illustrates examples of pre-processing. Before using pre-processing methods, the length of four email documents was 44 words; after pre-processing, the length was reduced to 23 words, indicating a nearly 48% decrease in the dataset size. In the next stage, a weight was calculated using the Term Frequency – Inverse Document Frequency (TF-IDF) weight factor for the most frequently occurred words in the reduced dataset.

3.3 Feature weight determination and efficient feature extraction

First, the most frequently occurring words in the reduced dataset are used in this phase to create the binary bag of words representation model. Later, a weight was computed for each word to indicate the significance of the words in the document using the TF-IDF weight factor. The binary bag of words representation for the ‘n’ frequently occurring words in the four email documents is shown in Table 3. This study considered the top 10 frequently occurring words.

However, the bag of words denotes the presence of words and ignores the significance of particular keywords in an email text. For instance, the word 'scholar' in the fourth document is more important than other words. But, in this model, words appearing in the document get the value '1'; otherwise, they get a value '0'. Hence, a bag of words model with TF-IDF score was applied instead of '0's and '1's, as in the bag of words model.

TF-IDF obtains the scores for each word in the document by multiplying TF and IDF for specific

Table 4. TF-IDF Model

Doc/ Word	research	publish	author	journal	index	article	Scopus	accept	hearty	congrats
E_1	0	0.032	0.032	0.032	0	0.032	0	0	0.077	0.077
E_2	0	0.026	0.026	0	0	0.026	0	0	0	0
E_3	0	0.028	0	0.028	0	0.028	0.136	0	0.069	0.069
E_4	0.106	0	0.022	0.022	0	0	0	0.107	0	0

words. Therefore, the score for each word is calculated using the formula given below.

$$TFIDF\ (word, doc) = TF\ (word, doc) * IDF\ (word) \quad (1)$$

So, this method needs to calculate two matrices, one (TF) counts the number of times a term/ word appears in every document, and the other one (IDF) measures the importance of every word in all documents. The following formulas were used to calculate both of them:

$$TF(word, doc) = \frac{number\ of\ times\ word\ occur\ in\ document}{number\ of\ words\ in\ document} \quad (2)$$

$$IDF(word) = log(\frac{number\ of\ documents}{1+number\ of\ documents\ with\ word}) \quad (3)$$

First, the TF dictionary was created for the 'n' most frequently occurring words with their TF values to find the significance of each term. Later, the IDF dictionary was created for the same set of words with the respective IDF values. Finally, the TF-IDF score is generated, as shown in Table 4.

3.4 Feature selection using Improved Elephant Herd Algorithm (IEHA)

The improved elephant herd algorithm (IEHA) algorithm is modelled by the behaviour and way of life of elephants. IEHA is a heuristic intelligence system based on elephants' nomadic lifestyles. Elephants exhibit social behaviour and have a complicated structure of females and calves. The IEHA algorithm selects the most pertinent features from the extracted features. The number of elephants in this algorithm represents the features that were extracted from the input layer; the most pertinent features are identified as the best female elephant of the clan after the death of the matriarch; and the irrelevant features are identified as the male elephants with the lowest fitness value. Fig 2 shows the flow chart for the proposed Improved Elephant Herd Algorithm. Clan update and separation operators make up the two stages of the Elephant Herding Optimization algorithm.

The entire population of elephants is initially split up into 'y' clans. Each elephant a_n the new position is influenced by the matriarch a_n . The clan a_n elephant 'y' can be determined using

$$E_{x,p_{i,j}} = E_{p_{i,j}} + \alpha \times \left(E_{best,p_i} - \lambda_{p_{i,m}}\right) \times k \quad (4)$$

where [0,1] is a scaling factor, E_{best,p_i} is the location with the best fitness value inside clan "i," and $E_{x,p_{i,j}}$ represent the old and new positions of elephant "y" in clan i, respectively. With a normal distribution and a value between [0, 1], K is a random number. For each clan, the best elephant is determined using

$$E_{x,p_{i,j}} = \beta \times E_{bt,p_i} \quad (5)$$

where $\beta \in [0,1]$ is a scaling factor that defines how the position of the clan leader $E_{x,p_{i,j}}$ will change for the following iteration depending on the effect of the clan centre E_{bt,p_i} . Eq. (6) is evaluated to determine a clan center's value:

$$E_{bt,p_i,f} = \frac{1}{z_{p_i}} \times \sum_{y=1}^{z_{p_i}} E_{pi,}y,f \quad where\ 1 \le f \le F \quad (6)$$

Where the number of elephants in the clan is signified as z_{p_i} , the f[th] dimension of an individual elephant. In Eq. (5), the update of the matriarch position is related to the information of all members of the clan.

The worst solution individuals are replaced by randomly initialized individuals during the separation procedure. It expands the population of elephants and enhances their capacity for exploration. The least valuable elephants in each tribe are relocated to the position indicated by

$$E_{q,p_i} = E_{Min} + (E_{Max} - \lambda_{Min} + 1) \times W \quad (7)$$

where P_{w,m_x} is the position with the worst fitness value in clan 'i'; E_{Min} and E_{Max} are the upper and lower bound of the elephant's position, respectively;

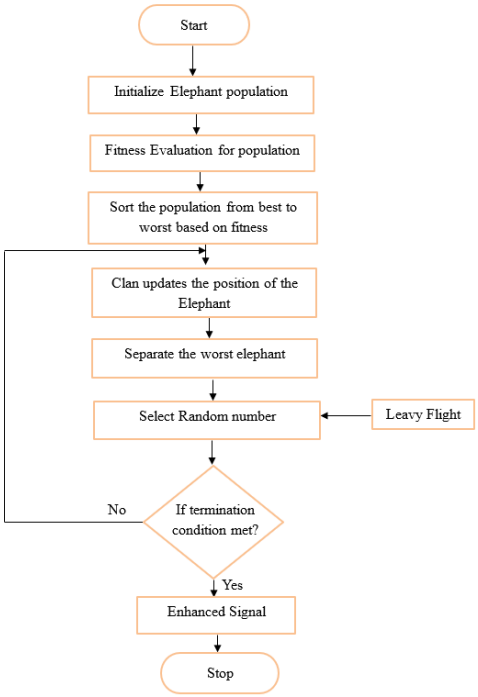


Figure. 2 Flow diagram of the proposed improved EHO

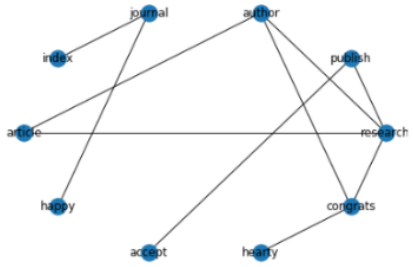


Figure. 3 Graph with weighted edges

W is a random number with a normal distribution in the range [0, 1]. A slower rate of convergence results from difficulties like lack of exploitation and random replacement of the worst person when random numbers are used. To address this issue, the LF mode is combined with the IEHO. The LF is modelled as,

$$LF(W) = \begin{cases} 1 & W < 1 \\ (W)^{-r} & W \geq 1 \end{cases} \quad (8)$$

By joining Eqs. (13) and (14), the improvement of IEHA with LF is attained as,

$$E_{c,p_i} = E_{Min} + (E_{Max} - \lambda_{Min} + 1) \times Wer(W) \quad (9)$$

IEHA renders the most relevant features and specifies them as follows,

$$MT_i = \{mt_1, mt_2, mt_3, \ldots\ldots, mt_n \quad (10)$$

The inputs are then multiplied by the vectors of features; eventually, the randomly selected features are summed. Mathematically, the input layer is represented as follows:

$$L_i = \sum_{i=1}^{n} MT_i c_i + X_i \quad (11)$$

Where the IL is exhibited as L_i, the input features are depicted as MT_i, the weight values are denoted as c_i and the bias value is notated as X_i.

3.5 Building graph

An email-document graph was built in this phase for email classification using the selected features from IEHO. The graph is defined as follows:

$$G = (V, E) \quad (12)$$

$$V = \{f_1, f_2, f_3 \ldots\ldots\ldots f_n \quad (13)$$

$$E = \{e_1, e_2, e_3, \ldots\ldots\ldots e_n\} \quad (14)$$

V indicates the set of nodes, and E denotes the edges. The top 10 words with the highest IEHO were considered as features to build a graph for the research category. These features are represented as nodes and the relationships between them are represented as edges in the graph, as shown in Fig. 3.

Each edge value is calculated using pointwise mutual information (PMI) [27]. The PMI is used to calculate the likelihood that two words will appear together.

A strong semantic association between words is indicated by a high PMI score, whereas a weak semantic correlation is indicated by a low PMI score.

The PMI formula is:

$$MI(W_1, W_2) = \log \frac{P(W_1, W_2)}{P(W_1)P(W_2)} \quad (15)$$

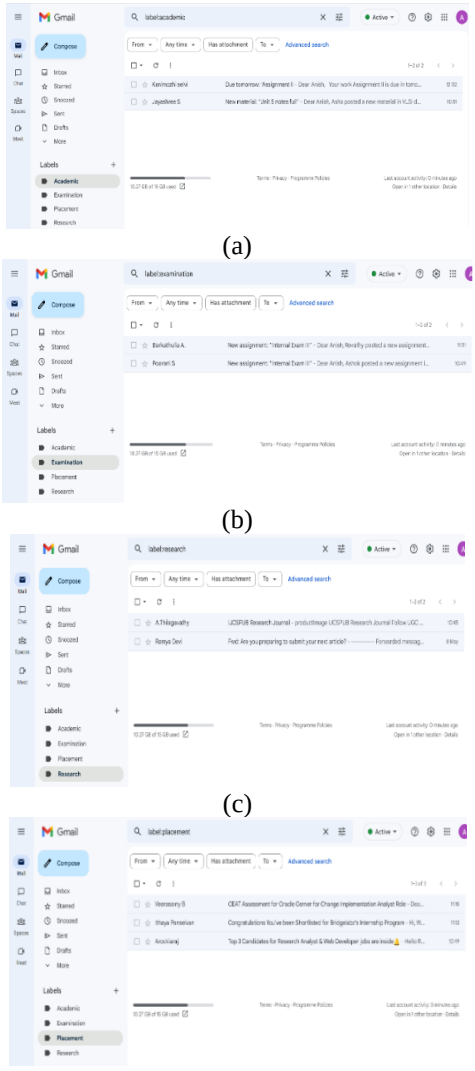
$$P(W) = \frac{Freq(w)}{total\ word\ count} \quad (16)$$

where W1 and W2 are pairs of words. To build a graph, only the edges with positive PMI values were considered while the negative PMI value edges were excluded. Once a graph for each category has been created, it is saved as a template for classifying unseen incoming mail in the future.

3.6 Classification

In this phase, emails were classified into one of the predefined categories based on node similarity (NS) value. The description of node similarity is as follows:

Algorithm 1: Graph similarity (G1, G2)
Input: Two different graphs G1 and G2 **Output:** returns similarity score SS [] between two graphs 1. SS [] ← NULL 2. for each node **i** in **G1** 3. for each node **j** in **G2** 4. SS [i, j] ← SS[i, j] + JS (i, j) 5. return SS



(a) (b) (c) (d)

Figure. 4: (a) Academic email, (b) Examination email, (c) Research email, and (d) Placement email

3.6.1. Find graph similarity

The graph similarity is determined by comparing nodes based on how they are connected. Similarities between two nodes can be detected when their neighbors are similar, i.e., when both source and destination nodes are similar. The NS between two nodes was calculated using the jaccard similarity (JS) metric.

Given two email documents X and Y, the JS is computed as:

$$JS(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|} \quad (17)$$

The algorithm graph similarity computes the node similarity between two graphs. Initially, the similarity score (SS []) between two graphs was initialized to NULL to represent no nodes as compared yet. Later, for each pair of nodes (i, j), the algorithm computes the similarity score between the nodes 'i' and 'j' and adds the result into SS []. Finally, the algorithm returns the final similarity score between two graphs G1 and G2. Finally, the input graph was classified into the category with the maximum similarity score.

4. Result and discussion

In the proposed work, EmailNet architecture is trained using Python. The proposed EmailNet classification technique was trained and tested with the Real-time dataset. Four types of email are distinguished, including Academics, examination, research, and placement. The proposed approach was applied to the real-time REVA dataset in the first experiment, and its performance was measured. The classification results are depicted in Fig. 4.

4.1 Performance analysis

The performance analysis was based on specificity, accuracy, precision, F1-score, and recall.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

$$precision = \frac{TP}{TP+FP} \quad (20)$$

$$recall = \frac{TP}{TP+FN} \quad (21)$$

$$f1 = 2\left(\frac{precision*recall}{precision+recall}\right) \quad (22)$$

Where TP, FP TF, and FN specify false-positives, true-positives, true-negatives, and false- negatives respectively.

Table 5 provides an illustration of different types of Email Classification with specific parameters. The average accuracy, precision, recall, and F1score of the proposed EmailNet are 0.988, 0.94, 0.96, and 0.96 respectively.

The ROC was generated for four classes that include Academic, examination, research, and placement illustrated in Fig. 5. The proposed EmailNet attains higher AUC of 0.99, 0.989, 0.985,

Table 5. The efficiency of the proposed EmailNet framework

Class	Accuracy	Precision	F1 score	Recall
Academic	0.973	0.94	0.97	0.97
Examination	0.985	0.95	0.95	0.96
Research	0.989	0.92	0.94	0.95
Placement	0.99	0.96	0.98	0.98

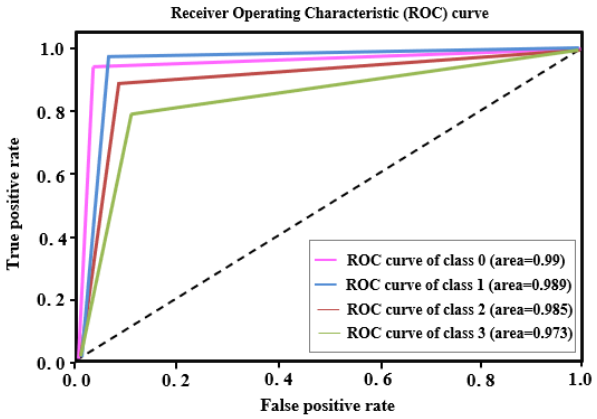


Figure. 5 ROC curve of the proposed email classification model

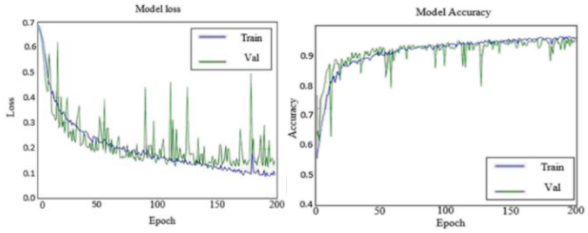


Figure. 6 Performance curve and loss curve of the proposed email classification model

and 0.973, for Placement, Research, Examination, and Academic respectively can be measured via FPR and TPR parameters.

Fig. 6 shows the accuracy curve with accuracy and epochs on both axes; as the epochs are raised, the accuracy of the method increases. Fig. 6 illustrates the relationship between epochs and losses, showing that the model's loss decreases as the epochs are improved. So, the predicted accuracy of 0.95 for the proposed EmailNet is highly reliable for email classification.

4.2 Comparative analysis

For evaluating the efficacy of the proposed EmailNet model, the existing state-of-the-art email methods were compared to the findings of the proposed model.

The performance is analyzed based on the precision, accuracy, Recall, and F1 score metrics.

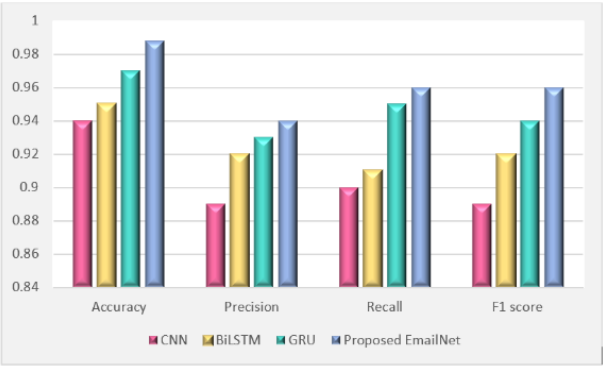


Figure. 7 Comparison of traditional models

Table. 6 Comparison between the existing and proposed technique

Author & Year	Methods	Accuracy (%)
Bhatti, P., et al [16] (2021)	LSTM	90%
Yaseen, Q., [22] 2021)	Bert-base-cased transformer	97.67%
Borg, A.et al [23] (2021)	Word embedding and LSTM	91.2
Hina, M.et al [24] (2021)	SeFACED	95%
Proposed	EmailNet	98.82

The proposed EmailNet performance is compared with conventional approaches such as CNN, BiLSTM, and GRU corresponding findings are illustrated in Fig. 7. As shown in the plot, the accuracy obtained by CNN, BiLSTM, and GRU is 0.94, 0.95, and 0.97% respectively.

Compared to the traditional network, the proposed method achieved 4.85%, 3.84%, and 1.82% higher performance than CNN, BiLSTM, and GRU respectively. Thus, it is seen that EmailNet achieves a better result than other states of art models.

Table 6 compares the proposed model with other existing methods. The comparison of existing models (i.e., LSTM, Bert-base-cased transformer, Word embedding, and LSTM, SeFACED) with the proposed EmailNet using the REVA University email database. The overall accuracy of the proposed method is 8.92%, 1.235, 7.710, and 3.86% is better than existing techniques. The proposed EmailNet achieves 98.82 % of accuracy, which is better than the existing model. From this analysis, we conclude that the proposed model achieves a better range of accuracy in the classification of emails.

5. Conclusion

In this paper, a novel EmailNet has been proposed for efficient email classification based on graph similarity measure. Initially, the dataset is preprocessed using NLP techniques such as

removing email signatures, removing punctuations, removing stop-words, lowercase conversion, tokenization, and stemming for removing irrelevant data. After preprocessing the feature are extracted using bag of words and term frequency – inverse document frequency (TF-IDF). These extracted features are given as input to improved Elephant herding optimization (EHO) for selecting the most relevant features to build a graph-based similarity index for classifying each category of e-mail. The proposed EmailNet achieves a high accuracy of 98.82% for classifying the email. In the future, the proposed EmailNet is tested with numerical and image data. The proposed EmailNet is also planned to implement other optimization techniques to optimize the network to improve its accuracy.

Conflicts of interest

The authors declare that they have no conflict of interest.

Author contributions

The authors confirm contribution to the paper as follows: Study conception and design: Aruna Kumara B and Mallikarjun Kodabagi M; Data collection: Aruna Kumara B; Analysis and interpretation of results: Mallikarjun Kodabagi M; Draft manuscript preparation: Aruna Kumara B and Mallikarjun Kodabagi M. All authors reviewed the results and approved the final version of the manuscript.

Acknowledgments

The authors would like to thank the reviewers for all of their careful, constructive and insightful comments in relation to this work.

References

[1] R. Team, "Email statistics report, 2021-2025", *The Radicati Group*, Inc. Palo Alto, CA, USA, 2021.

[2] T. Suma, and Y. S. Kumara Swamy, "Email classification using adaptive ontologies Learning", In: *Proc. of 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, pp. 2102-2106, 2016.

[3] M. Habib, H. Faris, M. A. Hassonah, J. Alqatawna, A. F. Sheta, and A. M. A. Zoubi, "Automatic Email Spam Detection using Genetic Programming with SMOTE", 2018 *Fifth HCT Information Technology Trends (ITT)*, Dubai, United Arab Emirates, pp. 185-190, 2018.

[4] J. Cui and X. Li, "Content Based Spam Email Classification using Supervised SVM, Decision Trees and Naive Bayes", In: *Proc. of 2nd International Conference on Machine Learning and Computer Application*, Shenyang, China, pp. 1-4, 2021.

[5] P. Saraswat and M. S. Solanki, "Phishing Detection in E-mails using Machine Learning", In: *Proc. of 2*022 2nd *International Conference on Technological Advancements in Computational Sciences (ICTACS)*, Tashkent, Uzbekistan, pp. 420-424, 2022.

[6] P. Trikanjananun, A. Numsomran and V. Tipsuwannaporn, "Improving Naive Bayes by Reducing the Importance of Low-Frequency Words Based on Entropy of Words for Spam Email Classification", In: *Proc. of* 2022 22nd *International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Korea, Republic of, pp. 10-14, 2022.

[7] A. R. Yeruva, D. Kamboj, P. Shankar, U. S. Aswal, A. K. Rao, and C. S. Somu, "E-mail Spam Detection Using Machine Learning – KNN", In: *Proc. of 2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Uttar Pradesh, India, pp. 1024-1028, 2022.

[8] Iqbal, K. and M. S. Khan, "Email classification analysis using machine learning techniques", *Applied Computing and Informatics*, 2022.

[9] E. M. Bahgat, S. Rady, W. Gad, and I. F. Moawad, "Efficient email classification approach based on semantic methods", *Ain Shams Engineering Journal*, Vol. 9, No. 4, pp. 3259-3269, 2018.

[10] A. Sharaff and H. Gupta, "Extra-Tree Classifier with Metaheuristics Approach for Email Classification", *Bhatia, S., Tiwari, S., Mishra, K., Trivedi, M. (eds) Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing*, Vol. 924, 2019.

[11] W. Pan, J. Li, L. Gao, L. Yue, Y. Yang, L. Deng, and C. Deng, "Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification", *Scientific Programming*, Vol. 2022, Article ID 6737080, p. 8, 2022.

[12] Q. Li, M. Cheng, J. Wang, and B. Sun, "LSTM Based Phishing Detection for Big Email Data", In *IEEE Transactions on Big Data*, Vol. 8, No. 1, pp. 278-288, 2022.

[13] J. M. Fernández, and M. Errecalde, "Multi-class E-mail Classification with a Semi-Supervised Approach Based on Automatic Feature Selection and Information Retrieval", In: *Proc. of Cloud Computing, Big Data & Emerging Topics: 10th Conference, JCC-BD&ET 2022, La Plata, Argentina, June 28–30, 2022, Proceedings*, pp. 75-90, 2022.

[14] A. Hosseinalipour, and R. Ghanbarzadeh, "A novel approach for spam detection using horse herd optimization algorithm", *Neural Computing and Applications*, Vol. 34, No. 15, pp. 13091-13105, 2022.

[15] K. Iqbal, and M. S. Khan, "Email classification analysis using machine learning techniques", *Applied Computing and Informatics*, (ahead-of-print), 2022.

[16] P. Bhatti, Z. Jalil, and A. Majeed, "Email Classification using LSTM: A Deep Learning Technique", In: *Proc. of 2021 International Conference on Cyber Warfare and Security (ICCWS) IEEE*, pp. 100-105, 2021.

[17] S. Deshmukh and S. Dhavale, "Automated real-time email classification system based on machine learning", In: *Proc. of International Conference on Computational Science and Applications: ICCSA 2019*, pp. 369-379, 2020.

[18] A. Sharaff and H. Gupta, "Extra-tree classifier with metaheuristics approach for email classification", *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018*, pp. 189-197, 2019.

[19] A. Sharaff and N. K. Nagwani, "Identifying categorical terms based on latent Dirichlet allocation for email categorization", *Emerging Technologies in Data Mining and Information Security: Proc. of IEMIS 2018*, Vol. 2, pp. 431-437, 2019.

[20] B. A. Kumara, M. M. Kodabagi, and T. Choudhury, et al. "Improved email classification through enhanced data preprocessing approach", *Spatial Information Research*, No. 29, pp. 247–255, 2021.

[21] A. K. B, and M. M. Kodabagi, "Efficient Data Preprocessing approach for Imbalanced Data in Email Classification System", In: *Proc. of* 2020 *International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 338-341, 2020.

[22] Q. Yaseen, "Spam email detection using deep learning techniques", *Procedia Computer Science*, Vol. 184, pp. 853-858, 2021.

[23] A. Borg, M. Boldt, O. Rosander, and J. Ahlstrand, "E-mail classification with machine learning and word embeddings for improved customer support", *Neural Computing and Applications*, Vol. 33, No. 6, pp. 1881-1902, 2021.

[24] M. Hina, M. Ali, A. R. Javed, F. Ghabban, L. A. Khan, and Z. Jalil, "Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning", *IEEE Access*, Vol. 9, pp. 98398-98411, 2021.

[25] P. D. Kusuma, and M. Kallista, "Quad Tournament Optimizer: A Novel Metaheuristic Based on Tournament Among Four Strategies", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 2, pp. 268-278, 2023, doi: 10.22266/ijies2023.0430.22.

[26] P. D. Kusuma and A. Novianty, "Multiple Interaction Optimizer: A Novel Metaheuristic and Its Application to Solve Order Allocation Problem", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 2, pp. 440-453, 2023, doi: 10.22266/ijies2023.0430.35.

[27] F. A. Zeidabadi, and M. Dehghani, "POA: Puzzle optimization algorithm", *Int. J. Intell. Eng. Syst*, Vol. 15, No. 1, pp. 273-281, 2022, doi: 10.22266/ijies2022.0228.25.

[28] P. D. Kusuma and A. L. Prasasti, "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179-190, 2022, doi: 10.22266/ijies2022.1231.18.

[29] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp.156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[30] P. D. Kusuma and M. Kallista, "Extended Stochastic Coati Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 156-166, 2023, doi: 10.22266/ijies2023.0630.38.

[31] P. D. Kusuma and F. C. Hasibuan, "Attack-Leave Optimizer: A New Metaheuristic that Focuses on The Guided Search and Performs Random Search as Alternative", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 3. 2023, doi: 10.22266/ijies2023.0630.19.