Decision Analytics Journal 10 (2024) 100390

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A hybrid correlation-based deep learning model for email spam classification using fuzzy inference system

Femi Emmanuel Ayo [a], Lukman Adebayo Ogundele [a], Solanke Olakunle [a], Joseph Bamidele Awotunde [b,*], Funmilayo A. Kasali [c]

[a] Department of Mathematical Sciences, Olabisi Onabanjo University, Ago-Iwoye, 120107, Ogun State, Nigeria
[b] Department of Computer Science, Faculty of Information and Communication Sciences, University of Ilorin, Ilorin 240003, Nigeria
[c] Department of Computer Science and Mathematics, Mountain Top University, Ogun State, Nigeria

ARTICLE INFO

Keywords:
Deep learning
Neural networks
Fuzzy logic
Spam classification
Feature selection
Email filtering

ABSTRACT

Spam emails are unwanted and unsolicited messages. The major problems in email spam detection methods are low detection rates and a high likelihood of false alarms. This study proposes a hybrid correlation-based deep learning model for email spam classification using a fuzzy inference system. Using a rule-based hybrid feature selection technique, we choose the most crucial features from a preprocessed spam-based dataset. The selected features are then loaded into a deep learning model for spam classification, and fuzzy logic is employed to categorize each spam class into its severity levels to decrease misclassification. Compared with other machine learning methods, the proposed method shows better F1-score results for both test set spambase and test set non-spambase of 96.5% and 96.4%, respectively. Similarly, the developed method shows better accuracy, error rate, and processing time of 94.0017%, 5.9983%, and 0.5 s, respectively. The proposed approach also shows a reduced misclassification based on the fuzzy inference system.

1. Introduction

Email is a means for fast and cheap communication [1,2]. Emails continue to grow in popularity in electronic transactions and corporate communications despite the rise of alternative types of instant messaging and social networking. Emails are used worldwide as a means for electronic communication. A study that has been monitoring email user data since 1993 predicts that 4.37 billion people will be using email globally in 2023. This constitutes over half of the world's population and is up 2.7% from the previous year [3]. Due to the popularity of email for electronic transactions, more people are using it, which has increased the number of spam emails—unwanted emails sent to recipients without their consent. Unwanted messages that are not requested are referred to as spam or garbage emails. The person who sends spam emails has no prior relationship with the recipients and instead collects addresses from various places including phone books and completed forms. One of the primary ways that risks like viruses, worms, and phishing assaults are delivered to users of email is through spam messages, which is quickly becoming a serious problem [4–7].

Email spam is a growing issue that affects not just regular internet users but also poses a serious challenge for businesses and organizations [8,9]. The average monthly amount of spam emails sent increased by an astonishing 7700% of global email traffic between 2012 and 2023 [10–12]. Many approaches are being put forth to identify email spam, but machine learning techniques top the list of currently used automatic email spam detections [13].

Email spam detection techniques are becoming more and more important since it is now harder for a user to tell the difference between legitimate and spam emails only by reading the subject line or email content. Lack of feature selection approaches that could increase classification accuracy, high false alarm rate, low detection rate, and lengthy processing time are the shortcomings of the present email spam detection methods.

In order to address the shortcomings of current email spam detection methods, a hybrid correlation-based deep learning model for classifying email spam utilizing a fuzzy inference method was built in this study. To choose the most crucial characteristics, the created technique combined the Correlation-based Feature Selection Subset Evaluator (CfsSubsetEval) and Rule-Based Genetic Search (RBGS) methodologies. A deep learning model was then trained using the chosen features to determine whether or not an email message is spam. Fuzzy logic was used to classify spam emails as normal, harassing, suspicious, or fraudulent in order to gauge the severity of a spam instance that had been identified.

* Corresponding author.
E-mail addresses: ayo.femi@oouagoiwoye.edu.ng (F.E. Ayo), ogundele.lukman@oouagoiwoye.edu.ng (L.A. Ogundele), solanke.olakunle@oouagoiwoye.edu.ng (S. Olakunle), awotunde.jb@unilorin.edu.ng (J.B. Awotunde), fkasali@mtu.edu.ng (F.A. Kasali).

https://doi.org/10.1016/j.dajour.2023.100390
Received 21 August 2023; Received in revised form 18 December 2023; Accepted 22 December 2023
Available online 23 December 2023

The primary contributions of this research include:

(1) The design of a rule-based hybrid feature selection method to improve detection accuracy.

(2) The application of an automated detection technique based on deep learning.

(3) The development of a fuzzy logic spam classification model to reduce the possibility of misclassifying or removing legitimate emails.

The remaining work are arranged as follows: related work is explained in Section 2, materials and methods were discussed in Section 3, results were discussed in Section 4, and conclusion section conclude the work.

2. Related work

Authors in [14] evaluated the use of case-based reasoning for categorizing brief text messages. They chose the best feature types and representations for short texts before evaluating the performance of the k-Nearest Neighbor (kNN), Support Vector Machine (SVM), and Naive Bayes (NB) classifiers [15,16]. The findings revealed that different features and classifiers are required for short text messages than long text messages. The results of the experiments also showed that SVM and NB demonstrated better classification performances than kNN. The conclusion indicate that SVM outperformed other related classifiers in term of classification accuracy. The study lacks a robust framework for spam detection.

In [17], the authors presented an improved online SVM for spam filtering. They argued that the performance of the traditional SVM always produce overfitted classification value and increased processing cost. Therefore, they presented an improved online SVM to resolve overfitting problem and reduced computational cost. They conducted test with related algorithms on the same benchmark datasets. It was demonstrated that the suggested approach can deliver excellent results at a significantly lower cost. A sizable benchmark set of email data was used for the test. Although, the results still showed some level of high computational cost.

Authors in [18] provided a unique probabilistic feature selection strategy for text categorization. They introduced a new probabilistic feature selection method for text classification called Distinguishing Feature Selector (DFS), which is based on filters. The provided approach was contrasted with various well-known filtering techniques, such as chi-square, information gain, Gini index, and deviation from Poisson distribution. Different datasets, classification methods, and success criteria were all compared. In terms of classification accuracy, dimension reduction rate, and processing time, the findings clearly demonstrated that DFS performed better than other comparable approaches. Other pattern categorization issues cannot be solved using the method.

Authors in [19] created an online active multi-field learning system for effective email spam filtering. The created technique offers an online active multi-field learning approach based on various concepts and addresses a number of issues with email spam filtering. The created multi-field learner combined the prediction results from various classifiers using a novel compound weight schema, and each classifier computes the mean arithmetic of numerous conditional probabilities derived from feature strings in accordance with a string-frequency index data structure. The evaluation revealed that in terms of classification accuracy and minimal space–time complexity, the created multi-field learner for email spam classification performed better than the alternatives. The study is not adaptable with other pattern classification problems.

In [20], the authors described a customized spam filtering with natural language features. In order to do attribute selection, they first employed bag-of-words approaches. As a second step, they used cutting-edge baseline classifiers like Radom Forest (RF), NB, and SVM. They also applied two meta-learning algorithms: Adaptive Boosting (AdaBoost) and Bootstrap Aggregating (Bagging). The evaluation results showed the choice of RF as the weak learner for each of the applied meta-learning algorithms, called Boosted RF and Bagged RF, respectively. The authors justified the choice of RF based on its high classification accuracy for spam filtering. Although, no method for data imbalance.

Authors of [21] also created an improved genetic programming method for identifying unwanted emails. The method that has been created for detecting spams works by creating an ensemble of classifiers. The created method utilized an ensemble of classifiers for email spam filtering and the greedy stepwise search strategy to extract the most significant features from two benchmark datasets (Enron and spamassassin). Genetic Programming (GP), Bayesian, NB, J48 decision tree, RF, and SVM were some of the relevant machine learning methods that were compared to the constructed ensemble of classifiers. The study's findings demonstrated that, in terms of performance accuracy and false positive rate, the constructed ensemble of classifiers outperformed related methods. The study did not include methods for data imbalance problem and feature selection method.

Based on Bayesian theory, authors in [22] presented a three-way decision method for email spam filtering. In order to decrease the likelihoods of misclassification, the created approach offers three decision alternatives for handling incoming emails. In order to prevent emails from being incorrectly categorized, the created approach offers the option of rejecting the result of a categorization. The developed method collects additional information for rejected cases for further examination of the classification decision. The authors established a loss function to analyze the cost of a classification decision, two automatically calculated threshold values on the posterior odds ratio based on the loss function, and the final choice is based on the action with the overall lowest cost. The test results demonstrated that the developed approach outperformed the baseline in terms of processing cost and classification error rate reduction. The study did not include methods for data imbalance problem and feature selection method.

Authors of [23] presented a hybrid classifiers strategy for email spam detection in a different study. For the best classification of email spam, the study created an ensemble of classifiers using Boosted Bayesian, Boosted Naive Bayes, and SVM. The authors selected features using a greedy step-by-step feature search strategy. Boosted Bayesian and boosted naive Bayes were selected as committee members and SVM as the president in the combination of classifiers. Members of the committee were chosen based on the authors' prior research, which showed that boosting with adaboost improves the performance of probabilistic classifiers. In terms of accuracy and false positive rates, the hybrid classifiers technique performed better than other comparable individual classifiers. The study did not include methods for data imbalance problem and feature selection method.

The authors of [24] presented a classification-based approach to email filtering. In order to choose the most crucial aspects for email spam categorization, the method used the Term Frequency and Inverse Term Frequency (TF-IDF) to examine the body of Email messages. In order to reduce the dimensionality of the extracted features, the authors additionally applied an adaption approach in which only significant terms are taken into consideration after consulting a dictionary. Five classification algorithms – Naive Bayes, SVM, Bayesian logistic regression, J48 decision tree, and Random Forest – were comparison tested. The approach was evaluated using Enron dataset. Based on precision, recall, F1-score, and accuracy criteria, the evaluation's findings demonstrated that Bayesian logistic regression beat other comparable classification algorithms. The outcomes also demonstrated that the dictionary-based filtering proposed had acceptable performance and quick filtering execution. The disadvantage of the method is that no strong framework for e-mail spam misclassification.

Authors in [25] presented machine learning methods for Short Message Service (SMS) spam filtering. Using benchmark datasets from diverse backgrounds (Singaporean, American, Indian English, etc.), they created an SVM model for SMS spam filtering. Prior research was done by the authors using benchmark datasets for English in Singapore and India. The developed method showed best results on Indian English and promising results on other related SMS datasets in term of high precision for SMS spam filtering. The developed method has no robust framework for spam detection.

In another study by authors in [26] a spam detection using ensemble learning method was presented. The created ensemble method offers a variety of possible combinations of the four classifiers Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, and Decision Tree. The developed ensemble methods is based on a voting mechanism to determine the best accuracy of different combination of classifiers. The results showed that the developed ensemble classifiers based on a voting mechanism produces better classification accuracy than individual classifier. Additionally, they have developed a client–server mobile application for spam filtering. In essence, the mobile application employed the client side to transmit user click data from the mobile to the server. The received data is then classified by a machine learning model on the server before being sent back to the client with the forecast. In comparison to all other combinations, the authors found that the multinomial NB, Bernoulli NB, and decision tree combination produced the best accuracy results on the email dataset. The study did not include feature selection method.

In [27], the authors presented a natural language processing based on composite features for email spam filtering. The modified composite features include character-based features, word-based features, tag-based features, structural features, and bag-of-words features. To minimize feature dimension based on the chosen composite features, the authors employed Term Frequency-Inverse Document Frequency-Class Frequency (TF-IDF-CF). The base classifier for spam detection was then the Naive Bayes classifier. They tested both the composite feature model and the individual feature model. According to the test results, the devised approach generated excellent classification accuracy and a low false positive rate. The study did not include method for data imbalance problem.

When attempting to identify spam SMS messages, authors in [28] presented a word embedding method. To choose the most crucial features, they employed the Continuous Bag Of Words (CBOW) method. As the primary classifier for SMS message spam filtering, they used a deep learning model based on feedforward neural network. The accuracy of the developed deep learning method was found to outperform the traditional SVM method. The method is not suitable for long text spam detection.

Similarly, authors in [29] presented a machine learning method for email spam filtering. They used data transformation method for dataset normalization. In addition, they selected features using distributed bag-of-words with cosine similarity and autoencoder feature representation. Their ability to handle small feature sizes regardless of the data source is their customized feature representation's key advantage over the accuracy of baseline classifiers. The accepted cosine similarity metric may result in a space complexity problem, which is one of the method's noted shortcomings.

In [30], authors suggested a new machine learning-based solution for email spam identification. They adapted Naive Bayes, decision tree and RF as the baseline classifiers. The authors used three benchmark datasets for their evaluation. The most pertinent feature set was chosen using the TF-IDF feature selection method. After that, the email spam classification is done using the modified classifiers. According to the experimental findings, RF is more accurate than other approaches. High precision and less overfitting are two benefits of their technology. The study did not include method for data imbalance problem.

For the purpose of identifying SMS spam, authors in [31] created a weighted feature augmented Hidden Markov Model (HMM). The HMM approach was used in the model's formulation to weight and label SMS words for spam detection. The experimental findings demonstrated that, in terms of classification accuracy, the devised method outperformed Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). Additionally, a dataset of Chinese SMS messages was used to confirm the created method's categorization accuracy and speed. After careful consideration, the evaluation revealed that the created method outperformed comparable methods in terms of accuracy and classification speed. The developed method is not language sensitive.

In [32], authors presented sentiment and spam classification model using the Yet Another Clustering Algorithm (YACA) with transferability. In order to analyze sentiment in Twitter comments and categorize spam in YouTube comments, the authors provided a YACA model with a domain transferable feature technique. Three Twitter datasets and four YouTube spam datasets were used to evaluate the proposed approach. Modern clustering techniques were compared to the outcomes. The findings demonstrated that the proposed technique outperformed comparable cutting-edge sentiment analysis systems. The authors also acknowledged that on all datasets, YACA and kNN performed within 1% of one other and significantly better than other cutting-edge algorithms for spam categorization. The presented method showed better accuracy and provide transferability across the datasets used. The study did not include method for data imbalance problem.

Authors in [33] developed a collaborative method with fingerprints for spam detection. The developed collaborative method created spam database from the recommendations of other users. The created spam database was matched with any incoming email for spam classification using near duplicate similarity matching method. In order to reduce false alarm rate in spam classification, they calculated cumulative weights from both email and fingerprint signatures of spam emails. The fingerprint signatures of newly classified spam was periodically used to update the spam database for new attacks spam detection. The system was tested with Spam Assassin dataset and the results showed a comparatively better classification accuracy. The study did not include feature selection method.

Authors in [34] presented a model that categorizes emails into two groups: spam and non-spam. To determine which extreme values fall outside of the designated range, Density-based spatial clustering of applications with noise (DBSCAN) and Isolation Forest are employed. The effective features are chosen using the Chi-Square, Recursive Feature Elimination, and Heatmap feature selection approaches. To produce a comparison study, the proposed model was implemented in both deep learning and machine learning. Various machine learning algorithms such as Multinomial Naïve Bayes, Random Forest, K-Nearest Neighbor, and Gradient Boosting are utilized to introduce collaborative learning. Artificial neural networks, gradient descent neural networks, and recurrent neural networks are used in deep learning. To aggregate the output of several classifiers, an ensemble approach was built. The authors ensemble approaches was compared to a single classifier, and the results showed higher prediction accuracy. For both machine learning and deep learning applications of email spam detection, the authors' suggested model produced higher accuracy and lower error rates. The proposed model showed overfitting problem.

In order to detect spam effectively, authors in [35] suggested the MOGA–CNN–DLAS, a Multi-Objective Genetic Algorithm and Convolutional Neural Network-Based Deep Learning Architectural Scheme. This suggested MOGA–CNN–DLAS strategy effectively replaces the Softmax regression approach at the CNN's output layer with SVM. There are two reasons for the suggested MOGA–CNN–DLAS strategy. The initial reason centers on the incorporation of the possibility to facilitate the acquisition of implicit semantics that may be obtained from the extensive collection of unlabeled data. The second reason stems from the restricted range of resources that can be investigated to enhance the effectiveness of the training process. The suggested MOGA–CNN–DLAS technique uses CNN for the feature extraction process, word embedding for the word representation of Twitter content, and a Multi-Objective Optimization Algorithm for better classification. Using the stochastic gradient descent back-propagation technique, the initial stage involves creating process weights that help train the CNN. It has been found that CNN has the ability to extract related syntactic and lexical features from the Twitter dataset separately, which may allow it to extract dominating traits straight from the training set. Next, in order to find the most correlated set of features that could be obtained from the SVM framework in the second step, a feature selection procedure influenced by Multi-Objective Optimization is incorporated. Because SVM can define a hyperplane by extracting a manageable number of features from CNN and because it has the optimal properties of the Multi-Objective Optimization Algorithm, the authors used it in their proposed approach. Following CNN training, the network's optimal factors are ascertained by applying a feature selection procedure influenced by multi-objective optimization. In order to feed the top hidden layer output into the SVM for classification, the optimal feature set obtained from the Multi-Objective Optimization Algorithm is concatenated with it in the third phase. The suggested MOGA–CNN–DLAS scheme's experimental outcomes contributed to improved optimality metrics, recall value, accuracy, precision, and F1-score. The suggested approach might produce extremely complex results.

Authors in [36] proposed a better Bayesian method by concentrating on the area where Bayesian may misidentify labels and managing those errors to enhance classification performance. The authors determine the likelihood that an instance belongs to a class using the Bayes theorem, assigning the instance the class label with the highest probability. According to the authors, when the probability calculated for spam and non-spam classes are near to each other, the Bayesian classifier's prediction is weak. To distinguish between weak and strong predictions, they consequently use a threshold. Their two-layer Bayesian approach—Basic Bayesian, or BBayes, and Corrected Weak Region Bayesian, or CWRBayes—concerns strong and weak predictions, respectively, and leads to a hybrid technique. Although they employ different feature selection processes, both BBayes and CWRBayes have the same classification mechanism. Following their implementation and evaluation on two datasets of spam emails, the proposed approach outperform the naïve Bayesian baseline and several additional Bayesian variants. Features selection techniques that might improve classification accuracy are absent from the proposed approach.

Authors in [37] proposed a semantic graph neural network for the classification of spam emails. The paper presented a Semantic Graph Neural Network (SGNN) approach to tackle the difficult email classification problem. By projecting email onto a graph and using the SGNN model for classification, this method transforms the email classification problem into a graph classification challenge. The four main stages of the proposed solution are data preprocessing, graph construction, training graph neural networks, and graph classification. The dataset was manually cleansed during the data preprocessing stage to get rid of noisy and imbalanced data. Email document and word nodes make up the bulk of the graph that is constructed during the graph development phase. Embedding vectors are included in each node according to the characteristics of the nodes nearby. The authors fed the graph to the created graph during the graph neural network training phase in order for it to learn high-dimensional features. In the end, the authors used a convolutional neural network to train a graph classification model on the email content and word graph in order to solve the email classification challenge. Since the email features come from the semantic network, the authors did not use word embeddings for numerical vector representation. Several publicly available datasets are used to test the proposed method performance. The proposed method provides good accuracy in the email classification test when tested against a few available datasets, according to experiments conducted using the public dataset. In terms of spam classification, the performance of the proposed method outperforms the most advanced deep learning-based approach. The proposed method is limited to text-based email and spam identification. When there are few training datasets available, the proposed method may cause overfitting.

Authors in [38] suggested a hybrid method for email spam detection called GDTPNLP, which combines Genetic Decision Tree Processing and Natural Language Processing. The hybrid algorithm of GDTPNLP that has been suggested combines the advantages of the decision tree and the genetic process. Emails sent from one end to the other using voice assistance and text can be sent using the authors' suggested bidirectional technique. The genetic process takes the content out of a test input or a produced email and uses genetic cross-checking to compare it to a trained set that is readily available. Those results are intelligently categorized using the decision tree approach, which also yields an estimated result on whether the current message is spam or regular mail. Every mail message is given a confidence-threshold level by the GDTPNLP proposed algorithm, which then cross-checks the message with a trained set. If the trained set value is greater than the current message threshold level, the proposed algorithm flags the mail as spam. A similar procedure is applied to voice mail messages. Nonetheless, the authors noted that the idea of data extraction for voice mail message extraction differs slightly from that of other concepts. Google Assistant is used to gather voice data from users, and a speech synthesizer is used to extract the voice data into a text message. Following the conversion of the speech variables into text, the collected input is categorized as preprocessing and is subsequently analyzed one at a time in a manner similar to the text message analysis method. Lastly, the input is subjected to the Genetic Decision Tree Processing with Natural Language Processing process, which evaluates the audio message in question to identify whether or not it contains spam. The mail is routed to the user's inbox if the procedure detects spam content in the input; if not, it is sent to the spam folder of the mailbox. This suggested method offers a means of recognizing spam emails in both text-based and speech-activated formats. A greater rate of spam detection is provided by the GDTPNLP technique. The suggested method is not suitable for real-time applications and is not time-efficient.

Authors in [39] suggested using conventional classifiers along with bidirectional LSTM (BiLSTM) for Email spam detection. The authors used deep learning and machine learning algorithms, including Long Short-Term Memory, Bidirectional-Long Short-Term Memory, Artificial Neural Network, Naive Bayes Classifier, and Random Forest, to determine which model is more accurate at classifying the emails as spam or non-spam. The study is divided into data pre-processing and classification. The data pre-processing technique is used to removes noisy data. During the classification phase, email spam filtering was accomplished by comparing the performances of Bi-LSTM versus traditional classifiers. When compared to other machine learning models, the simulations and findings showed that Bi-LSTM is the most accurate and highly preferred model. Different performance grades were also displayed by the other machine learning models. According to these findings, the Bi-LSTM outperformed the other classifiers in terms of accuracy in the following order: BI-LSTM > Naïve Bayes > SVM > Random Forest > LSTM > Decision Tree > ANN. The suggested method lacks feature selection strategies that could increase classification accuracy.

Authors in [40] suggested a Random Forest with TF-IDF and Synthetic Minority Over-sampling TEchnique (SMOTE) for the classification of Email spam and non-spam. The study is divided into the preprocessing, feature extraction and classification phases. The preprocessing phase was used to remove noise from the dataset. In the feature extraction stage, features were extracted from the dataset using feature extraction methods including bag-of-words and term frequency-inverse document frequency. The authors employed over-sampling and under-sampling approaches to address the imbalance in the used SMS dataset. In the classification phase, several machine learning techniques, including support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression, were applied on the extracted keywords with TF-IDF and BOW independently. The experimental findings demonstrated that, in comparison to other machine learning algorithms, the random forest classifies SMS as either spam or non-spam more reliably. With the use of TF-IDF features and the oversampling technique, the suggested model is effectively trained to distinguish between spam and non-spam SMS categories. With notable accuracy, the performance of the suggested method was also evaluated using the spam email dataset. Using TF-IDF features on balanced data, Random Forest outperformed other machine learning methods in terms of accuracy. The ensemble architecture of random forests contributes significantly to their performance, and oversampling helps further by providing a large and well-balanced feature set for training. In comparison to BOW, TF-IDF provides weighted features, which also contributes to the high accuracy of the random forest. Because more trees are needed for a more accurate forecast, which makes the model slower, the suggested method has a high level of complexity.

Authors in [41] proposed an enhanced Fruitfly Optimization method for email classification using a stacked residual recurrent neural network. The proposed method include the preprocessing, parameter tunning and classification phases. The preprocessing phase was used to make the Email dataset be in the compatible format necessary for machine learning computation. The parameter tunning phase used the improved fruitfly optimization algorithm to optimally tune the parameters of the stacked residual recurrent neural network. The classification phase was then used for the Email classification using the stacked residual recurrent neural network based on the tuned parameters. A series of simulations were conducted to test the proposed method on public datasets. The comparison results demonstrate the improvements of the proposed improved Fruitfly Optimization with Stacked Residual Recurrent Neural Network method over other similar approaches that have achieved high classification accuracy. Large-scale real-time email datasets were not used to evaluate the proposed approach. High processing requirements may also arise from the proposed approach.

Authors in [42] suggested an SMS spam filtering using support vector machines. The study includes dataset collection, preprocessing and classification phases. The dataset collection include the collection of spam and non-spam messages. The preprocessing phase was used to remove unwanted data from the comma separated values (CSV) format of the Kaggle dataset. The preprocessing phase was also used to transform the dataset into a machine-readable form by converting to vector or by doing discretization. The classification phase used support vector machines for the training, testing and design of the suggested SMS spam filtering model using support vector machines. The suggested model was validated on spam messages and confusion matrix was used to measure the performances of the suggested SMS spam filtering model. In comparison to the Naïve Bayes algorithm, the results indicated that the SMS spam filtering model that uses support vector machines has the best accuracy. The proposed support vector machine-based SMS spam filtering method was not evaluated on large datasets. For big datasets, the proposed support vector machine method of SMS spam filtering may also result in a lengthy training period.

Authors in [43] proposed an email spam detection using gated recurrent neural network. The study includes dataset preparation, preprocessing, feature extraction and classification phases. The dataset preparation adapted the raw data that was gathered from the Kaggle website. The preprocessing was done to remove stop words and other noises. The feature extraction was used to select the best feature set for the detection of email spam. The classification phase used the gated recurrent neural network for the classification of email spam. The simulation of the proposed email spam detection using gated recurrent neural network on the test dataset showed the effectiveness of the method. The result showed that the gated recurrent neural network produced better accuracy when compared to similar models. The proposed method could result in a lengthy training period.

Table 1 displays an overview of the related work that has been reviewed. It can be seen that most of the related work did not consider feature selection method as a technique that can increase classification accuracy of classifiers. Also, some of the methods lacks merit in dealing with data imbalance problems therefore resulting in poor classification accuracy. Some of the related work also showed processing complexity due to the lack of preprocessing and feature selection techniques. Therefore, the identified issues in literature include the need for feature selection method and a robust classifier to correctly classify spam class into their severity levels. If these problems are resolved properly, email spam classification will be more accurate and have a lower error rate.

2.1. *Artificial neural networks*

The Artificial Neural Network (ANN) is a system of intricately linked computations that was created to mimic the functioning of the brain [44]. A kind of ANN employed in real-world applications is the MultiLayer Perceptron (MLP). A MLP network is made up of weighted input nodes, one or more computational hidden nodes, and an output node for making wise choices. The nodes are highly linked and the weights attached to the input nodes are used for incremental learning through weight adjustments. The backpropagation method is the conventional learning algorithm for MLP networks. The network is initially trained using random weight selection and weight adjustment to determine new weights with the least amount of error. The ANN was adopted because of its fault tolerance ability which could be an advantage for data imbalance problem. The ANN can also perform parallel processing which can reduce processing time complexity and increase accuracy.

2.2. *Genetic search*

Based on the idea of a Genetic Algorithm (GA), genetic search looks for the optimum solutions. GA is the employment of a computer software to mimic the actions of natural creatures [45]. Authors first proposed GA in [46], and it has found use in a number of machine learning applications [47,48]. An initial population of randomly generated individual programs is used in GA. The GA creates new individuals by automated genetic recombination and mixing. On a variety of fitness measures, the best new member of the population is assessed. Better individual programs replace the existing population of better individual programs at each iteration. To choose the best feature subset for email spam categorization, the genetic search was modified. The best optimal feature subset selection can improve the ANN classifier's accuracy.

2.3. *Correlation-based feature selection subset evaluator (CfsSubsetEval)*

The CfsSubsetEval determines the degree of inter-correlation between a chosen feature subset of predictors and the classification accuracy of the classifier. It is decided which feature subsets to use based on their high correlation with the baseline classifier and low inter-correlation with the other predictors. The CfsSubsetEval was modified as the feature selection method because it assesses the correlation between the performance of the baseline classifier and the performance of the selected features. The CfsSubsetEval will choose a feature subset to improve the baseline classifier's accuracy. Therefore, CfsSubsetEval will lead to better classification accuracy and low false alarm rate.

2.4. *Fuzzy logic*

Fuzzy logic is described as a multi-variable logic that can capture real values between 0 and 1 inclusively [49]. The goal of fuzzy logic is to eliminate uncertainty in classification choices. Fuzzy logic is a concept used to control instability in an unstable system in the context of information systems. The benefit of employing fuzzy logic for spam classification is that it can correct the mistake of classifying a legitimate email as spam. The fuzzy logic was adapted to further classify a spam class into their severity levels to reduce cases of misclassification and therefore increase accuracy of the baseline classifier.

3. Materials and methods

In this study, a Hybrid Correlation-based Deep Learning model with Fuzzy Inference System (HCDL-FIS) for email spam categorization

Table 1
Overview of reviewed work.

Authors & year	Method	Strength	Limitation
Healy et al. (2005) [14] Sculley & Wachman (2007) [17]	SVM Online SVM	- High classification accuracy. - Increase accuracy.	- No robust framework for spam detection. - High computational time.
Uysal & Gunal (2012) [18]	Probabilistic feature selection	- High classification accuracy. - Good dimension reduction rate. - low processing time.	- Not adaptable with other pattern classification problems.
Liu & Wang (2012) [19]	SVM	- Lower standards for labels. - Limited space and time expenses.	- No method for data imbalance problem. - No feature selection method.
Shams & Mercer (2013) [20]	Bagged random forest	- Better accuracy.	- No method for data imbalance problem.
Trivedi & Dey (2013) [21]	Enhanced genetic programming	- Better accuracy. - Reduce false positive rate.	- No method for data imbalance problem. - No feature selection method.
Zhou et al. (2014) [22]	Naïve Bayes	- Reduce error rate. - Reduce computational cost.	- No method for data imbalance problem. - No feature selection method.
Trivedi & Dey (2016) [23]	Boosted NB + SVM	- Better accuracy. - Low false positives.	- No method for data imbalance problem. - No feature selection method.
Bahgat et al. (2016) [24] Kaliyar et al. (2018) [25] Gupta et al. (2019) [26]	Bayesian logistic regression SVM Ensemble NB + Decision tree	- Faster filtering execution. - High precision. - Better accuracy.	- No design framework for spam detection. - No robust framework for spam detection. - No feature selection method.
George & Vinod (2018) [27]	Naïve Bayes	- High classification accuracy. - Low false positive rate.	- No method for data imbalance problem.
Lee & Kang (2019) [28]	Word embedding technique + feedforward neural network.	- High accuracy.	- Not suitable for long text spam detection.
Diale et al. (2019) [29]	SVM	- Good accuracy. - Good generalization.	- Space complexity issue.
Gaurav et al. (2020) [30]	Random forest	- High accuracy. - Reduced over fitting.	- No method for data imbalance problem.
Xia & Chen (2021) [31]	Enhanced Hidden Markov model	- Higher accuracy. - Faster training. - High filtering speed.	- It is not language sensitive.
Ghiassi et al. (2022) [32]	YACA with transferability	- Better accuracy. - Reduction in model dimensionality. - Provide transferability across datasets.	- No method for data imbalance problem.
Rajendran et al. (2022) [33]	Hybrid filtering with fingerprints	- Increase accuracy. - Reduction in false alarm rate.	- No method for data imbalance problem. - No feature selection method.
Hossain et al. (2021) [34]	Ensemble of machine learning and deep learning	- Higher accuracy. - Lower error rate.	- Overfitting problem.
Jacob (2022) [35]	Genetic algorithm and Convolutional neural network	- High optimality, recall value, accuracy, precision, and F1-score.	- High processing time.
Nosrati et al. (2023) [36]	Bayesian method	- High accuracy.	- No features selection techniques that might improve. classification accuracy.
Pan et al. (2022) [37]	Semantic graph neural network	- High accuracy.	- Limited to text-based email and spam identification. - Overfitting problem.
Ismail et al. (2022) [38]	Genetic decision tree and natural language processing	- It can classify spam emails in both text-based and speech-activated formats. - High accuracy.	- It is not suitable for real-time applications. - It is not time-efficient.
Shaik et al. (2023) [39]	Bidirectional LSTM	- Accurate and highly preferred model.	- It lacks feature selection strategies that could increase classification accuracy.
Abid et al. (2022) [40]	Random forest with TF-IDF and SMOTE	- High accuracy. - Ensemble architecture.	- High processing time.
Alshahrani et al. (2023) [41]	Fruitfly optimization with stacked residual recurrent neural network method	- High classification accuracy.	- Lack of use of large-scale real-time email datasets. - High processing requirements.
Prasanna et al. (2021) [42]	Support vector machines	- High classification accuracy.	- It is not evaluated on large datasets. - Lengthy training period.
Mani et al. (2023) [43]	Gated recurrent neural network	- High classification accuracy.	- Lengthy training period.

was created. Data collection, data preprocessing, feature extraction, spam detection, and spam classification are the five main stages of the developed HCDL-FIS. The Spambase dataset is first collected during the data collecting phase. The Spambase dataset was transformed into the Attribute Relation File Format (ARFF) during the data preprocessing stage. Following preprocessing, training and testing sets were created from the Spambase dataset. The feature extraction stage then receives the preprocessed train and test sets. The most crucial features for email spam detection are chosen during the feature extraction phase using a rule-based hybrid feature selection method. A rule-based engine, the Genetic search method, and CfsSubsetEval are all components of the hybrid feature selection. The subset evaluator determines how each attribute and class are related. The attribute-class association with the highest correlation is then preferred for selection. This procedure is referred to as feature evaluation. The genetic search method examines each attribute's merits in light of this feature evaluation, and it delivers the features with the highest fitness values. The feature subset with the fewest number of subset features is returned by the rule-based engine

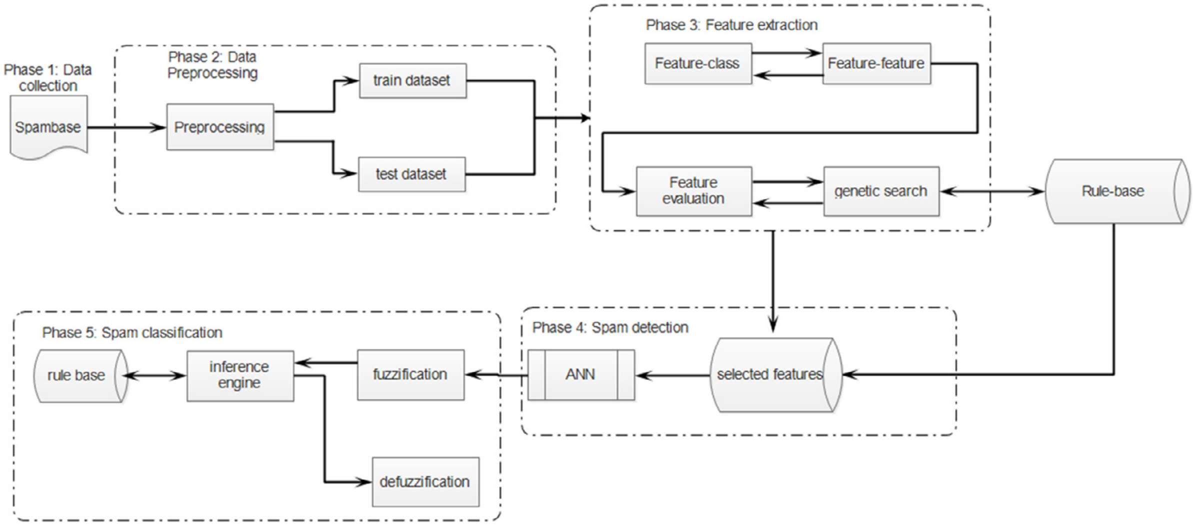


Fig. 1. The HCDL-FIS architecture.

Table 2
Definitions of the variables in the adapted dataset.

Number	Type of variable	Description
1–48	word_freq_WORD	Proportion of words that match WORD in the e-mail
49–54	char_freq_CHAR	Proportion of characters that match CHAR in the e-mail
55	capital_run_length_average	Typical continuous orders of upper case letters
56	capital_run_length_longest	Maximum continuous orders of upper case letters
57	capital_run_length_total	Total amount of upper case letters in the e-mail
58	class	Class of spam

if two feature subsets have the same fitness value. The spam detection phase then receives the chosen features. A deep learning model is used in the phase of spam detection to categorize messages as spam or not. In order to avoid misclassification, the spam classification phase uses fuzzy logic to divide a given spam category into various grade levels. The architecture of HCDL-FIS is shown in Fig. 1.

3.1. Data collection

The UCI Machine Learning Spambase dataset was used. There are 4601 instances and 58 attributes in the collection. In total, there are 58 examples, 57 of which are continuous, and 1 has a nominal class designation. 1813 spam instances and 2788 non-spam instances make up the total, with respective percentages of 39.4% and 60.6%. The majority of the training instances let you know how frequently a word or character appeared in the email. The attributes in the dataset are defined as shown in Table 2. In a similar vein, Table 3 lists the attributes in the Spambase dataset along with their respective data types.

3.2. Data preprocessing

The Spambase dataset was initially transformed into an ARFF format and the Java WEKA API for machine learning was used to process the dataset for the implementation of the HCDL-FIS architecture. The dataset was transformed into an ARFF format to make it suitable for machine learning training.

3.3. Feature extraction

The most crucial features are chosen from the preprocessed dataset during the feature extraction phase using a hybrid feature selection method. A rule-based engine, the Genetic search method, and CfsSubsetEval are all components of the hybrid feature selection. The link between each attribute and the class is calculated by the subset evaluator. Next, preference is given to the attribute-class association with the highest correlation. It is known as feature evaluation. The genetic search algorithm examines the merits of each attribute based on this feature evaluation and returns the features with the highest fitness value. The rule-based engine delivers the feature subset with the fewest subset features when two feature subsets with the same fitness value are present. The justification for the hybrid feature selection approach is to balance the strengths of the filter (CfsSubsetEval) and wrapper (genetic search algorithm) methods for optimal feature set selection.

Definition 1 (*Subset*)**.** The feature V_i that forms a strong relationship with the class label c is selected as v_i if $p\left(V_i = v_i\right) > 0$ as shown in (1):

$$p\left(C = c|V_i = v_i\right) \neq p(C = c) \tag{1}$$

Definition 2 (*CfsSubsetEval*)**.** The correlation between a composite test consisting of the components added together and the outside variable may be predicted from (2) if the correlation between each of the components of a test and the outside variable is known as well as the inter-correlation between each pair of components.

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{n + n(n-1)\overline{r_{ii}}}} \tag{2}$$

where r_{zc} is the association among all the components and the outside variable, n is the total number of components, r_{zi} is the average of the associations among the components and the outside variable, and r_{ii} is the typical inter-correlation among components.

Definition 3. A genetic search is one that is motivated by the process of natural evolution. A linear combination of an accuracy term and a simplicity term serves as the fitness function in this genetic search.

$$Fitness\,(X) = \frac{3}{4}A + \frac{1}{4} = \left(1 - \frac{S+F}{2}\right) \tag{3}$$

where S is the number of training instances, X is a feature subset, A is the accuracy of the classifier, and F is the number of feature subgroups.

Table 3
List of the variables in the adapted dataset.

S/N	Variable	Type	S/N	Variable	Type	S/N	Variable	Type
1	word_freq_make	Double	27	word_freq_george	Double	53	char_freq_$	Double
2	word_freq_address	Double	28	word_freq_650	Double	54	char_freq_#	Double
3	word_freq_all	Double	29	word_freq_lab	Double	55	capital_run_length_average	Double
4	word_freq_3d	Double	30	word_freq_labs	Double	56	capital_run_length_longest	Double
5	word_freq_our	Double	31	word_freq_telnet	Double	57	capital_run_length_total	Double
6	word_freq_over	Double	32	word_freq_857	Double	58	label{1=spam,0=no spam}	Double
7	word_freq_remove	Double	33	word_freq_data	Double			
8	word_freq_internet	Double	34	word_freq_415	Double			
9	word_freq_order	Double	35	word_freq_85	Double			
10	word_freq_mail	Double	36	word_freq_technology	Double			
11	word_freq_receive	Double	37	word_freq_1999	Double			
12	word_freq_will	Double	38	word_freq_parts	Double			
13	word_freq_people	Double	39	word_freq_pm	Double			
14	word_freq_report	Double	40	word_freq_direct	Double			
15	word_freq_addresses	Double	41	word_freq_cs	Double			
16	word_freq_free	Double	42	word_freq_meeting	Double			
17	word_freq_business	Double	43	word_freq_original	Double			
18	word_freq_email	Double	44	word_freq_project	Double			
19	word_freq_you	Double	45	word_freq_re	Double			
20	word_freq_credit	Double	46	word_freq_edu	Double			
21	word_freq_your	Double	47	word_freq_table	Double			
22	word_freq_font	Double	48	word_freq_conference	Double			
23	word_freq_000	Double	49	char_freq_;	Double			
24	word_freq_money	Double	50	char_freq_(	Double			
25	word_freq_hp	Double	51	char_freq_[	Double			
26	word_freq_hpl	Double	52	char_freq_!	Double			

Definition 4 (*Rule Evaluation*)**.** If more than one feature subset ($F_>$) with a similar fitness value exists, the rule-based engine returns a feature subset (V_i) with less features (X_F); otherwise, it returns the feature subset with the greatest fitness value (F_hi) to the base classifier as in (4).

$$R = \begin{cases} V_i, \ if \ V_i \in F_> \bigcap X_F \\ V_i, \ if \ F_{hi} \bigcap \emptyset \end{cases} \tag{4}$$

3.4. Spam detection

The developed ANN-based detection model for email spam detection can be defined in the following equations:

Step 1: The ANN-based model combine the input x in the training dataset and some randomly generated weights w to get the output y as in (5).

$$\text{y} = \sum_{i=1}^{n} x.w \tag{5}$$

Step 2: A bias b can be added to the output in (5) to enhance the adjustment of the entire function to produce the necessary output as in (6).

$$y = \sum_{i=1}^{n} x.w + b \tag{6}$$

Step 3: The generated output y from (6) can be transformed through an activation function to make the network a non-linear activation function. The adapted activation function is the log sigmoid as in (7).

$$\hat{y} = \frac{1}{(1 + e^{-y})} \tag{7}$$

Step 4: The difference between the target output d and the predicted output $\hat{y}$ computes the error term as in (8).

$$e_n = d_n - \hat{y}_n \tag{8}$$

Step 5: The new weights of the network can be computed for better predictions by finding the product of the error term e_n, learning rate ŋ and the input x_n at that instance. This is shown in (9).

$$w_n = ŋ.e_n.x_n \tag{9}$$

Step 6: The weights of the network can be updated by summing the old weight w and the new weight w_n to arrive at the adjusted weight as in (10).

$$w_n + 1 = w + w_n \tag{10}$$

The justification for the use of hybrid correlation-based deep learning for email spam detection is to benefit from deep learning's nonlinear relationships and capacity to scale effectively with huge datasets.

3.5. Spam classification

The following concepts and definitions of fuzzy logic are applied in the fuzzy extension to the classification of email spam:

3.5.1. Fuzzy input

The input defined in (11) and reported in Table 2 together make up the fuzzy input for the classification of email spam. The defined membership of the fuzzy input show the degree of their presence in the set between 0 and 1 inclusive.

$$A = \{\text{WORD, CHAR, AVERAGE, LONGEST, TOTAL}\} \tag{11}$$

3.5.2. Membership variables

According to Eq. (12), the membership variables represent the level of membership for the specified membership set A. It is employed to demonstrate the level of categorization for a specific class attribute value. The grades specified in (12) can be assumed for the input and output variables.

$$m_A(x) = \{\text{normal, harassing, suspicious, fraudulent}\} \tag{12}$$

3.5.3. Fuzzification

Since the membership variables consist of four variables, the triangle membership function as given in (13) was adapted. The extreme values were calibrated using the triangle membership function. Table 4 show the fuzzy range of values for the fuzzification procedure.

$$\mu_A(s; [u, v, w]) = \begin{cases} 0, & if \ s = u \\ \dfrac{s-u}{w-u}, & if \ s \in [u, w] \\ \dfrac{v-s}{w-v}, & if \ s \in [v, w] \\ 0, & if \ s \geq w \end{cases} \tag{13}$$

Table 4
Fuzzy value interval.

Membership variable	Value interval
Normal	$0.1 \leq x < 0.3$
Harassing	$0.3 \leq x < 0.6$
Suspicious	$0.6 \leq x < 0.8$
Fraudulent	$0.8 \leq x \leq 1.0$

where s represent the x- coordinate of real values and u, v, w represent the y- coordinate between 0 and 1.

The justification for the use of intervals in Table 4 is because the linguistic variables are four and the adapted membership function is triangular. Therefore, the value interval can be assumed using $\mathrm{xi}/\sum \mathrm{n}, xi = 1$ to 4 and $\mathrm{n} = 4$. In other words, x_i is the individual linguistic variable {1 = Normal, 2 = Harassing, 3 = Suspicious, 4 = Fraudulent} and $\sum n$ is the total number of linguistic variables $n = 4$. For example, Normal is $1/4 = 0.25$; Harassing is $2/4 = 0.5$; Suspicious is $3/4 = 0.75$ and Fraudulent is $4/4 = 1$. Hence, the range of intervals in Table 4.

3.5.4. Knowledge base

The knowledge base of the developed fuzzy inference system for spam classification consist of $2^4 = 16$ rules, where the power represent the number of membership variables. The rule definition were based on the knowledge of experts in the domain as shown in Table 5. The fuzzy inference system evaluate its rules by taking the minimum based on the AND function.

3.5.5. Reasoning engine

The fuzzy reasoning engine make decisions based on the facts defined in the knowledge base for email spam classification. The purpose of the reasoning engine is to make prediction concerning the class of an email from the knowledge base of the system. The Root Mean Square (RMS) was used as the reasoning method as in (14).

$$\sqrt{\sum R^2} = \sqrt{R_1^2 + R_2^2 + R_3^2 + \cdots + R_n^2} \tag{14}$$

$R_1^2 + R_2^2 + R_3^2 + \cdots + R_n^2$ represent facts in the knowledge base with similar output. The method compute the centre of gravity from the grand sum of the square of facts with the same output in the knowledge base.

3.5.6. Defuzzification

Defuzzification is the process of changing from fuzzy to Boolean values for clear decision making. The value from this stage is intended to be used to determine the class of a particular email spam. The Centre of Gravity (CoG) method was used due to its clarity and precision as in (15).

$$\mathrm{CoG}\left(Y^*\right) = \frac{\sum \mu y\left(X_i\right) x_i}{\sum \mu y\left(X_i\right)} \tag{15}$$

where $\mu y\left(X_i\right)$ stands for the RMS for facts with the same output and x_i denotes the mid-points of their respective value interval.

3.6. Algorithms

The algorithms that summarizes the developed HCDL-FIS for email spam detection and classification are depicted in algorithms 1 and 2 respectively.

Algorithm 1: Detection algorithm

The most crucial features are chosen from the preprocessed dataset during the feature extraction phase using a hybrid feature selection method. A rule-based engine, the Genetic search method, and CfsSubsetEval are all components of the hybrid feature selection. The link between each attribute and the class is calculated by the subset evaluator. Next, preference is given to the attribute-class association with the highest correlation. It is known as feature evaluation. The genetic search algorithm examines the merits of each attribute based on this feature evaluation and returns the features with the highest fitness value. The rule-based engine delivers the feature subset with the fewest subset features when two feature subsets with the same fitness value are present. After the feature selection (line 15), the selected feature then serve as input into the neural network by randomizing the weight based on the input vectors x. The net Z is then computed (line 18) and the output computed based on the net using the sigmoid activation function (line 19). The error of the network is then computed and the iteration continues until the error is insignificantly small (line 23). The trained network is then used for the assignment of email into a spambase or non-spambase class (line 26).

Algorithm 2: Fuzzy-based classification

In order to remove email spambase misclassification, the fuzzy logic was used to resolve misclassification uncertainty. The input to the algorithm is the spambase variables defined in Table 2. The spambase class using the triangular membership function is classified as normal, harassing, suspicious and fraudulent based on the fuzzy value interval in Table 4. The algorithm will then return the fuzzy classification showing the severity of the spambase class.

4. Results and discussion

The program was implemented on a Windows 10 computer with an Intel Pentium CPU clocked at 2.40 GHz and 4.00 GB of RAM. Due to Java's support for pure object-oriented design, flexibility, portability, and the rich graphical interface, it was used to code the created HCDL-FIS. The implementation in Java made use of the genetic algorithm library, CfsSubsetEval library, and ANN Weka API. The Integrated Development Environment (IDE) utilized was NetBean. The dataset was edited using Notepad++. Modeling and creating the fuzzy rules for spam classification were done using MATLAB R2012b.

Fig. 2 depicts the dataset preparation for testing and evaluating the created HCDL-FIS in the arff format. The interface for creating and refining the neural network model for email spam detection is shown in Fig. 3. The dataset for the training, creation, and detection of email spam was loaded using the java neural network API. Control buttons like the Start button and the Accept button are included on the interface. The system is trained using the Start button. The training procedure starts with the pre-defined settings when the button is pressed. The Accept button is also intended to show the results of the anticipated classes. Either spambase or non-spambase classes can be anticipated.

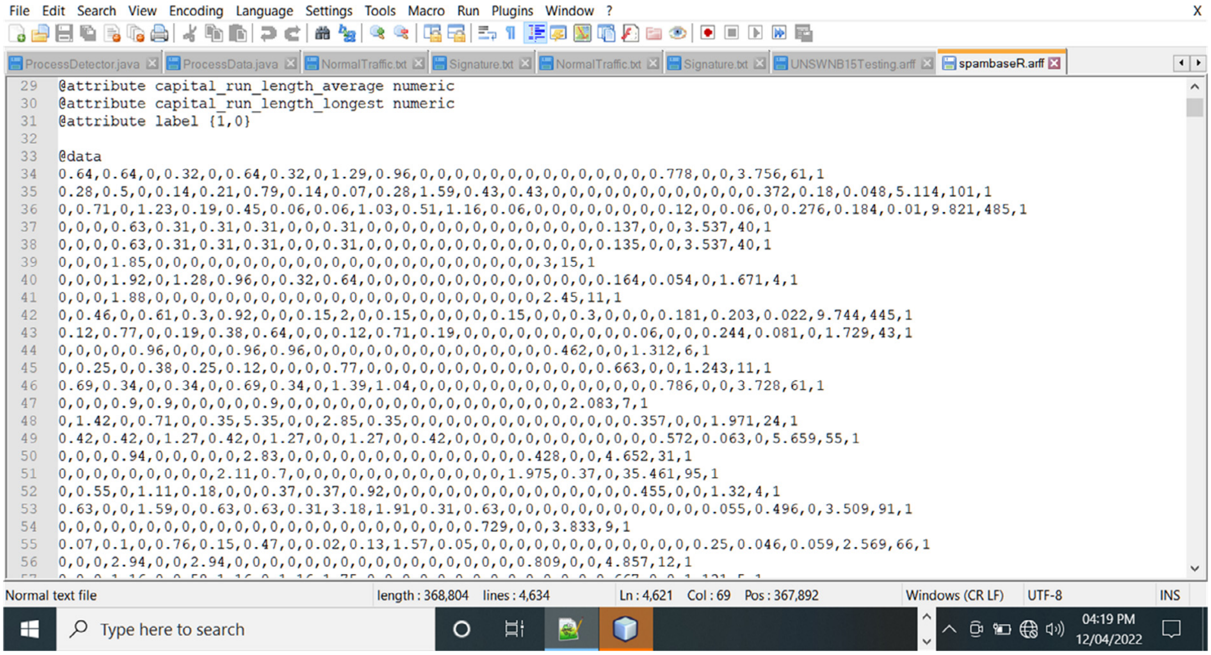
The method uses a genetic algorithm to search through the training set and choose the most crucial attributes. The parameter settings for the genetic search include population size of 20, the number of generations used is 20, the probability of crossover is 0.6, which indicates the rate at which beneficial materials are exchanged between individuals in the population to produce better individuals, the probability of mutation is 0.033, which indicates the rate at which genes are mixed in the population of individuals for better solutions, and the random number seed is 1, which indicates the starting point in the genetic search.

The number of inputs for the chosen neural network was set to 28 representing the number of attributes selected by the developed HCDL-FIS, the number of hidden layers was set to 8, the number of epochs was set to 500, the learning rate was set to 0.3, the momentum was set to 0.2, the number of hidden layers was set to 8, and the number of output nodes is 2 representing the spambase and non-spambase class.

Table 6 show the combination of several search methods and attribute evaluators for features selection. The least selected features were determined to be 10, 15, and 20 features for FilteredSubsetEval + GreedyStepwise, GreedyStepwise + CfsSubsetEval, and RankSearch + CfsSubsetEval, respectively. BestFirst + ConsistencySubsetEval and ConsistencySubsetEval + GreedyStepwise had 25 features each. Ranker + ChiSquaredAttributeEval, and InfoGainAttributeEval + Ranker had

Table 5
Knowledge base for spam classification.

#No	Word	char	ave	long	tot	Spam classification (Conclude)	Non Zero Min no
1	Normal	Normal	Normal	Normal	Normal	Normal	0.25
2	Normal	Harassing	Harassing	Harassing	Harassing	Harassing	0.25
3	Normal	Suspicious	Suspicious	Suspicious	Suspicious	Suspicious	0.25
4	Normal	Fraudulent	Fraudulent	Fraudulent	Fraudulent	Suspicious	0.25
5	Harassing	Normal	Normal	Normal	Normal	Normal	0.25
6	Harassing	Harassing	Harassing	Harassing	Harassing	Harassing	0.5
7	Harassing	Suspicious	Suspicious	Suspicious	Suspicious	Suspicious	0.5
8	Harassing	Fraudulent	Fraudulent	Fraudulent	Fraudulent	Fraudulent	0.5
9	Suspicious	Normal	Normal	Normal	Normal	Normal	0.25
10	Suspicious	Harassing	Harassing	Harassing	Harassing	Harassing	0.5
11	Suspicious	Suspicious	Suspicious	Suspicious	Suspicious	Suspicious	0.75
12	Suspicious	Fraudulent	Fraudulent	Fraudulent	Fraudulent	Fraudulent	0.75
13	Fraudulent	Normal	Normal	Normal	Normal	Normal	0.25
14	Fraudulent	Harassing	Harassing	Harassing	Harassing	Harassing	0.5
15	Fraudulent	Suspicious	Suspicious	Suspicious	Suspicious	Suspicious	0.75
16	Fraudulent	Fraudulent	Fraudulent	Fraudulent	Fraudulent	Fraudulent	0.9



```
@attribute capital_run_length_average numeric
@attribute capital_run_length_longest numeric
@attribute label {1,0}

@data
0.64,0.64,0,0.32,0,0.64,0.32,0,1.29,0.96,0,0,0,0,0,0,0,0,0,0,0,0.778,0,0,3.756,61,1
0.28,0.5,0,0.14,0.21,0.79,0.14,0.07,0.28,1.59,0.43,0.43,0,0,0,0,0,0,0,0,0,0,0.372,0.18,0.048,5.114,101,1
0,0.71,0,1.23,0.19,0.45,0.06,0.06,1.03,0.51,1.16,0.06,0,0,0,0,0,0,0.12,0,0.06,0,0.276,0.184,0.01,9.821,485,1
0,0,0,0.63,0.31,0.31,0.31,0,0,0.31,0,0,0,0,0,0,0,0,0,0,0,0.137,0,0,3.537,40,1
0,0,0,0.63,0.31,0.31,0.31,0,0,0.31,0,0,0,0,0,0,0,0,0,0,0,0.135,0,0,3.537,40,1
0,0,0,1.85,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,15,1
0,0,0,1.92,0,1.28,0.96,0,0.32,0.64,0,0,0,0,0,0,0,0,0,0,0,0.164,0.054,0,1.671,4,1
0,0,0,1.88,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.45,11,1
0,0.46,0,0.61,0.3,0.92,0,0,0.15,2,0,0.15,0,0,0,0,0.15,0,0,0.3,0,0,0,0.181,0.203,0.022,9.744,445,1
0.12,0.77,0,0.19,0.38,0.64,0,0,0.12,0.71,0.19,0,0,0,0,0,0,0,0,0.06,0,0,0.244,0.081,0,1.729,43,1
0,0,0,0,0.96,0,0,0,0.96,0.96,0,0,0,0,0,0,0,0,0,0,0,0.462,0,0,1.312,6,1
0,0.25,0,0.38,0.25,0.12,0,0,0,0.77,0,0,0,0,0,0,0,0,0,0,0,0.663,0,0,1.243,11,1
0.69,0.34,0,0.34,0,0.69,0.34,0,1.39,1.04,0,0,0,0,0,0,0,0,0,0,0,0.786,0,0,3.728,61,1
0,0,0,0.9,0.9,0,0,0,0,0.9,0,0,0,0,0,0,0,0,0,0,0,0,0,2.083,7,1
0,1.42,0,0.71,0,0.35,5.35,0,0,2.85,0.35,0,0,0,0,0,0,0,0,0,0,0.357,0,0,1.971,24,1
0.42,0.42,0,1.27,0.42,0,1.27,0,0,1.27,0,0.42,0,0,0,0,0,0,0,0,0,0.572,0.063,0,5.659,55,1
0,0,0,0.94,0,0,0,0,0,2.83,0,0,0,0,0,0,0,0,0,0,0,0.428,0,0,4.652,31,1
0,0,0,0,0,0,0,0,0,2.11,0.7,0,0,0,0,0,0,0,0,0,0,1.975,0.37,0,35.461,95,1
0,0.55,0,1.11,0.18,0,0,0.37,0.37,0.92,0,0,0,0,0,0,0,0,0,0,0,0.455,0,0,1.32,4,1
0.63,0,0,1.59,0,0.63,0.63,0.31,3.18,1.91,0.31,0.63,0,0,0,0,0,0,0,0,0,0,0.055,0.496,0,3.509,91,1
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.729,0,0,3.833,9,1
0.07,0.1,0,0.76,0.15,0.47,0,0.02,0.13,1.57,0.05,0,0,0,0,0,0,0,0,0,0,0.25,0.046,0.059,2.569,66,1
0,0,0,2.94,0,0,2.94,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.809,0,0,4.857,12,1
```

Fig. 2. Spambase dataset editor.

57 features each. GeneticSearch + ConsistencySubsetEval, RankSearch + ConsistencySubsetEval, PrincipalComponents + Ranker, and GeneticSearch + CfsSubsetEval + RuleEval had 37, 54, 48 and 28 features respectively. In the end, the 28 features of the produced GeneticSearch + CfsSubsetEval + RuleEval helped the neural network to be more accurate. According to these findings, the base classifier's overall performance is closely correlated with the number of features used and its accuracy.

4.1. Evaluation

The created fuzzy-based forensic analysis and classification of email data using deep learning was evaluated using the following performance indicators.

- **True positive rate (TP):** The number of instances accurately classified in the normal class. This is indicated in (16).

$$TP = \frac{TP}{TP + FN} \tag{16}$$

where FN is number of instances incorrectly classified in the normal class.

- **False positive rate (FP):** The number of occurrences that were wrongly assigned to the normal class. It is indicated in (17).

$$FP = \frac{FP}{FP + TN} \tag{17}$$

where TN is the number of occurrences that were correctly assigned to the normal class.

- **Precision:** This is a measurement of the precision assuming that a certain class that was expected to be positive is actually positive. It is indicated in (18).

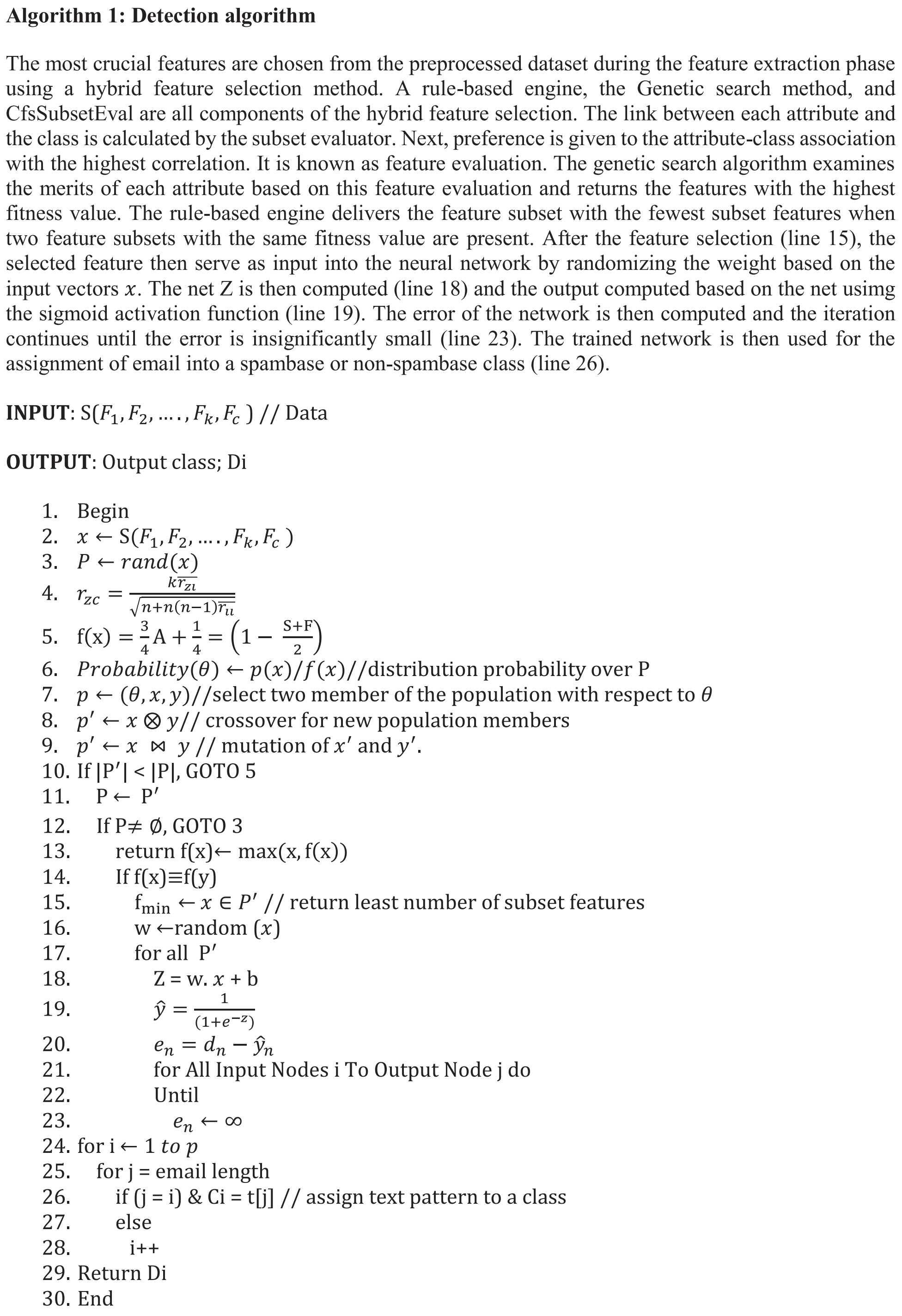
$$Precision = \frac{TP}{TP + FP} \tag{18}$$

- **Recall:** This is a measurement of the quantity of tagged instances that a prediction model properly detects as depicted in (19).

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

- **F1-score:** The precision and recall calculated using a specific threshold are represented by the harmonic mean. It serves to assess the classification's quality as shown in (20).

$$\mathrm{F1-score} = \frac{2\,(precision * recall)}{precision + recall} \tag{20}$$



Algorithm 1: Detection algorithm

The most crucial features are chosen from the preprocessed dataset during the feature extraction phase using a hybrid feature selection method. A rule-based engine, the Genetic search method, and CfsSubsetEval are all components of the hybrid feature selection. The link between each attribute and the class is calculated by the subset evaluator. Next, preference is given to the attribute-class association with the highest correlation. It is known as feature evaluation. The genetic search algorithm examines the merits of each attribute based on this feature evaluation and returns the features with the highest fitness value. The rule-based engine delivers the feature subset with the fewest subset features when two feature subsets with the same fitness value are present. After the feature selection (line 15), the selected feature then serve as input into the neural network by randomizing the weight based on the input vectors x. The net Z is then computed (line 18) and the output computed based on the net usimg the sigmoid activation function (line 19). The error of the network is then computed and the iteration continues until the error is insignificantly small (line 23). The trained network is then used for the assignment of email into a spambase or non-spambase class (line 26).

INPUT: $S(F_1, F_2, \ldots., F_k, F_c)$ // Data

OUTPUT: Output class; Di

```
1.  Begin
2.  $x \leftarrow S(F_1, F_2, \ldots., F_k, F_c)$
3.  $P \leftarrow rand(x)$
4.  $r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{n+n(n-1)\overline{r_{ii}}}}$
5.  $f(x) = \frac{3}{4}A + \frac{1}{4} = \left(1 - \frac{S+F}{2}\right)$
6.  $Probability(\theta) \leftarrow p(x)/f(x)$//distribution probability over P
7.  $p \leftarrow (\theta, x, y)$//select two member of the population with respect to $\theta$
8.  $p' \leftarrow x \otimes y$// crossover for new population members
9.  $p' \leftarrow x \bowtie y$ // mutation of $x'$ and $y'$.
10. If |P′| < |P|, GOTO 5
11.    $P \leftarrow P'$
12.    If P≠ ∅, GOTO 3
13.       return f(x)← max(x, f(x))
14.       If f(x)≡f(y)
15.          $f_{min} \leftarrow x \in P'$ // return least number of subset features
16.          w ←random ($x$)
17.          for all  P′
18.             Z = w. $x$ + b
19.             $\hat{y} = \frac{1}{(1+e^{-z})}$
20.             $e_n = d_n - \hat{y}_n$
21.             for All Input Nodes i To Output Node j do
22.             Until
23.                $e_n \leftarrow \infty$
24. for i ← 1 $to\ p$
25.    for j = email length
26.       if (j = i) & Ci = t[j] // assign text pattern to a class
27.       else
28.          i++
29. Return Di
30. End
```

- **Accuracy:** This is the proportion of occurrences that were correctly categorized relative to all of the instances. It is indicated in (21).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{21}$$

4.2. Discussion

In the light of the literature, the results of the study has been able address data imbalance problems using hybrid of feature selection subset evaluator and rule-based genetic search. The feature selection subset evaluator was used as a filter method to extract important features and the rule-based genetic search was used as a wrapper method with cross validation with the base classifier. The hybrid feature selection help to improve poor classification accuracy of most previous methods. The reduced dataset from the hybrid feature selection also help to reduce processing time of the baseline classifier. The use of fuzzy logic has help to reduce spam misclassification common with most of the previous methods. The hybrid correlation-based deep learning model resolved the problems of low accuracy and high error rate. Similarly, the fuzzy logic resolve any uncertainty in deep learning classification by categorizing spam class into their severity levels.

The method in this study can be useful to companies and application developers to identify unwanted email contents sent by attackers and

Algorithm 2: Fuzzy-based classification

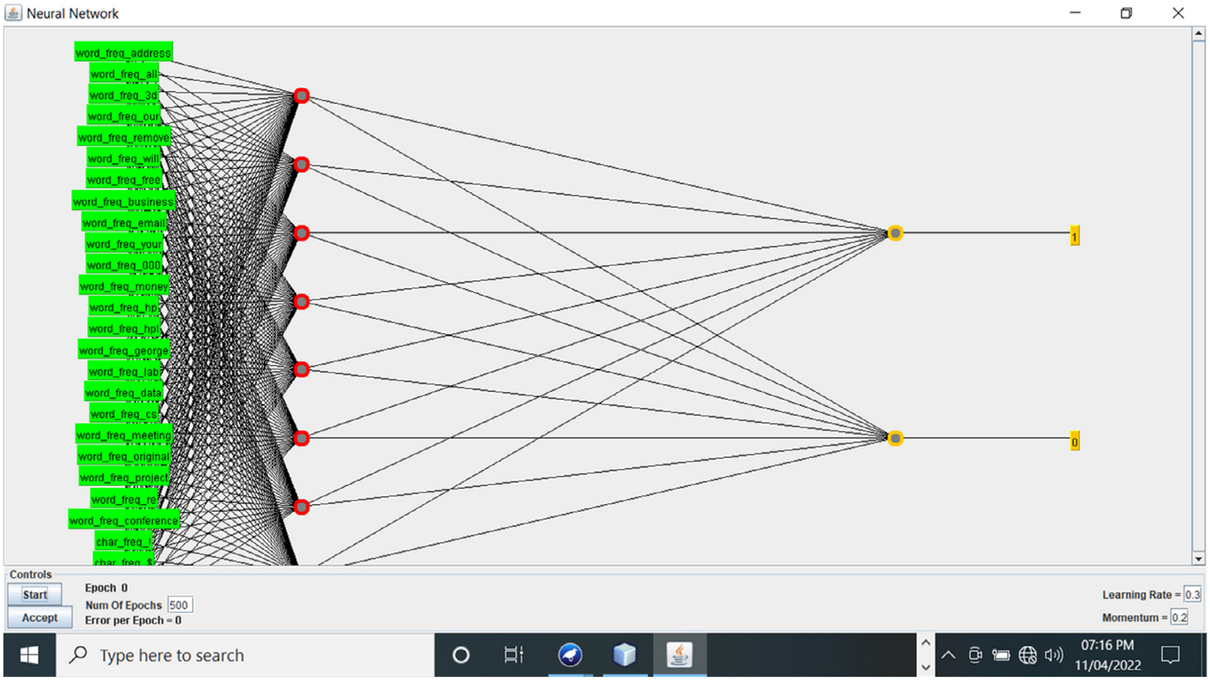
In order to remove email spambase misclassification, the fuzzy logic was used to resolve misclassification uncertainty. The input to the algorithm is the spambase variables defined in Table 2. The spambase class using the triangular membership function is classified as normal, harassing, suspicious and fraudulent based on the fuzzy value interval in Table 4. The algorithm will then return the fuzzy classification showing the severity of the spambase class.

INPUT: Di: spam class, wc: word, character, average, long, total

OUTPUT: Ci: classification level

PROCESS:

```
Begin
accept Di as input
for i = 1 to j of all wc in Di
   if $(0.1 \le Ci < 0.3)$ then
        Ci ← normal
    end if
    else  if $(0.3 \le Ci < 0.6)$ then
         Ci ← harassing
     end if
     else if $(0.6 \le Ci < 0.8)$ then
          Ci ← suspicious
     end if
     else if $(0.8 \le Ci \ge 1)$ then
          Ci ← fraudulent
     end if
     end for
return Ci
End
```



Epoch 0
Num Of Epochs 500
Error per Epoch = 0
Learning Rate = 0.3
Momentum = 0.2

Fig. 3. Neural network interface.

for the security of devices in end-to-end communication. The detail discussion of the results of the developed hybrid correlation-based deep learning model for email spam classification using fuzzy inference system is as follows:

Table 7 displays the categorization outcomes for test set spambase using several machine learning methods on reduced features. On the test set's spam class, the majority of machine learning algorithms attained F1-scores of at least 78.5% in terms of categorization rates.

Table 6
Feature selected by various attribute selectors.

SN	Attribute selector	#	Features selected
1	GreedyStepwise + CfsSubsetEval	15	4,5,7,16,21,23,24,25,27,42,44,46,52,53,55
2	BestFirst + ConsistencySubsetEval	25	1,3,5,7,10,12,16,17,19,20,21,23,24,26,37,38,42,45,46,50,52,53,55,56,57
3	GeneticSearch + ConsistencySubsetEval	37	1,2,3,4,5,7,8,10,11,12,15,16,19,20,21,22,23,24,25,27,29,32,33,34,37,40,41,45,46,48,49,50,52,53, 55,56,57
4	Ranker + ChiSquaredAttributeEval	57	52,53,56,21,7,55,16,24,57,23,5,25,19,3,11,27,17,2,26,8,6,20,10,9,18,12,54,50,15,1,37,45,46,30, 13,35,29,28,31,42,43,14,32,39,36,33,44,34,41,22,48,40,51,49,4,38,47
5	RankSearch + CfsSubsetEval	20	5,7,11,15,16,20,21,22,23,24,25,26,27,29,42,44,46,52,53,56
6	RankSearch + ConsistencySubsetEval	54	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35, 36,37,38,39,41,42,43,44,45,46,47,48,50,52,53,54,55,56,57
7	InfoGainAttributeEval + Ranker	57	52,53,56,7,21,55,16,24,57,25,23,27,5,19,26,3,11,17,2,8,6,20,10,9,18,12,54,30,37,45,50,15,29,46, 35,1,28,31,42,13,43,32,39,44,34,33,36,14,41,48,22,40,51,49,4,38,47
8	PrincipalComponents + Ranker	48	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35, 36,37,38,39,40,41,42,43,44,45,46,47,48
9	ConsistencySubsetEval + GreedyStepwise	25	1,3,5,7,10,12,16,17,19,20,21,23,24,26,37,38,42,45,46,50,52,53,55,56,57
10	FilteredSubsetEval + GreedyStepwise	10	7,16,21,23,24,25,27,52,53,55
11	GeneticSearch + CfsSubsetEval + RuleEval (approach)	28	2,3,4,5,7,12,16,17,18,21,23,24,25,26,27,29,33,41,42,43,44,45,48,52,53,54,55,56

Table 7
Comparison of various models on reduced features for test-set spambase.

Algorithm	TP	FP	Precision	Recall	F1-score	AUC	Time (s)
J48 tree	0.896	0.058	0.909	0.896	0.903	0.948	1.48
Decision Table	0.857	0.088	0.863	0.857	0.860	0.952	4.05
Bagging	0.896	0.058	0.909	0.896	0.903	0.948	2.52
kNN	0.880	0.075	0.884	0.880	0.882	0.906	0.6
Logistic	0.867	0.047	0.924	0.867	0.894	0.968	1.74
RBF Network	0.962	0.288	0.769	0.769	0.800	0.892	2.44
Naïve Bayes	0.971	0.326	0.659	0.971	0.785	0.953	0.15
Bayesian Logistic Regression	0.899	0.165	0.845	0.846	0.845	0.867	0.9
Naïve Bayes Multinomial	0.982	0.331	0.659	0.982	0.788	0.964	0.13
Multilayer Perceptron	0.903	0.098	0.857	0.903	0.879	0.958	62.73
GeneticSearch + CfsSubsetEval+RuleEval + NN	0.967	0.036	0.964	0.964	0.965	0.977	0.5

NN is Neural Network.

GeneticSearch + CfsSubsetEval + RuleEval + NN has a higher F1-score (96.5%) than bagging and J48 tree, which are the scores that comes the closest to it (90.3%). Additionally, in all the methods, a sizable portion of the occurrences were categorized as spam accurately, whereas a far smaller portion of the instances were incorrectly classified as spam. The stark contrast in scores between true positive and false positive results can be used to explain this. The true positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is lower than the closest result of the Naïve Bayes Multinomial with 96.7% < 98.2%, and the false positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is lower than the closest value of the Logistic with 3.6% < 4.7%. The outcomes showed that the best technique for classifying spam is GeneticSearch + CfsSubsetEval + RuleEval + NN. Overall, the test set spambase performance of the various machine learning algorithms was rather well distributed. In comparison to most of the other algorithms, the time it took to generate the model for GeneticSearch + CfsSubsetEval + RuleEval + NN exhibited faster processing time of 0.5 s. These results showed that the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) is relatively consistent and can effectively categorize emails as spambase. Hence, the justification for the precision value of 0.964 and F1-score of 0.965. The result also justifies the need for fuzzy logic to resolve misclassification uncertainty.

Table 8 displays the effectiveness of various machine learning models for the test set no-spam with reduced features. On the test set no-spam class, the majority of machine learning algorithms attained F1-scores of at least 79.6% for classification rates. GeneticSearch + CfsSubsetEval + RuleEval + NN has a higher F1-score (96.4%) than RBFNetwork, which is nearest at 94.9%. Additionally, all the approaches accurately classified a sizable portion of the cases as no-spam, while just a small portion of the occurrences were incorrectly labeled as no-spam. The significant score discrepancies between the true positive and the false positive scores can be used to explain this. The true positive of GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly higher than the closest result of the Logistic with 96.8% > 95.3%, and the false positive of GeneticSearch + CfsSubsetEval + RuleEval + NN is greater than the Naïve Bayes Multinomial with 3.7% > 1.8%. The application of fuzzy logic to resolve misclassification is the justification for the higher false positive in the GeneticSearch + CfsSubsetEval + RuleEval + NN. The findings indicated that GeneticSearch + CfsSubsetEval + RuleEval + NN is the most effective method for spam identification. Overall, the test set no-spam performance of the various machine learning methods appeared to be rather evenly distributed. In comparison to most of the other algorithms, the time it took to generate the model for GeneticSearch + CfsSubsetEval + RuleEval + NN exhibited faster processing time of 0.5 s. These results showed that the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) is consistent and can effectively categorize emails as non-spambase. Hence, the justification for the precision value of 0.963 and F1-score of 0.964.

Table 9 show the performance summary statistics of various machine learning models on reduced features. The mean absolute error

Table 8
Comparison of various models on reduced features for test-set non-spambase.

Algorithm	TP	FP	Precision	Recall	F1-score	AUC	Time (s)
J48 tree	0.942	0.104	0.933	0.942	0.937	0.948	1.48
DecisionTable	0.912	0.143	0.907	0.912	0.909	0.952	4.05
Bagging	0.942	0.104	0.933	0.942	0.937	0.948	2.52
kNN	0.925	0.120	0.922	0.925	0.924	0.906	0.6
Logistic	0.953	0.133	0.917	0.953	0.935	0.968	1.74
RBFNetwork	0.712	0.038	0.949	0.949	0.949	0.949	2.44
Naïve Bayes	0.674	0.029	0.973	0.674	0.796	0.953	0.15
Bayesian Logistic Regression	0.835	0.101	0.892	0.892	0.892	0.892	0.9
Naïve Bayes Multinomial	0.669	0.018	0.983	0.669	0.796	0.964	0.13
Multilayer Perceptron	0.902	0.097	0.935	0.902	0.918	0.958	62.73
GeneticSearch + CfsSubsetEval + RuleEval + NN	0.968	0.037	0.963	0.964	0.964	0.971	0.5

KEY:
Mean absolute error (MAE).
Root mean squared error (RMSE).
Relative absolute error (RAE).
Root relative squared error (RRSE).

Table 9
Performance summary statistics of different models on reduced features.

Algorithm	Accuracy (%)	Error (%)	MAE	RMSE	RAE (%)	RRSE (%)
J48 tree	92.3712	7.6288	0.0893	0.2494	18.7017	51.0312
DecisionTable	89.0024	10.9976	0.1979	0.2957	41.4307	60.5071
Bagging	93.1971	68 029	0.1102	0.2246	23.0704	45.9727
kNN	90.7411	9.2589	0.0956	0.3074	20.0117	62.9112
Logistic	91.9148	8.0852	0.1344	0.2541	28.1431	51.9953
RBFNetwork	81.0476	18.9524	0.2497	0.3526	52.2794	72.1503
Naïve Bayes	79.0698	20.9302	0.2164	0.4624	45.3241	94.6274
Bayesian Logistic Regression	85.9813	14.0187	0.1402	0.3744	29.355	76.6232
Naïve Bayes Multinomial	79.2219	20.7781	0.1781	0.3605	37.2862	73.7698
Multilayer Perceptron	90.2195	9.7805	0.1158	0.2626	24.2485	53.7461
GeneticSearch + CfsSubsetEval + RuleEval + NN	94.0017	5.9983	0.1118	0.251	23.4029	51.3699

of the GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly higher than that of the Bagging with 11.18 > 11.02, plus the accuracy of the GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly better than the closest result of Bagging with 94.0017% > 93.1971%. From looking at the accuracy alone, it can be seen that the GeneticSearch + CfsSubsetEval + RuleEval + NN outperformed the Bagging method. These results immediately reveals that the GeneticSearch + CfsSubsetEval + RuleEval + NN has also outperformed other related methods. The possible reason for the better result of the GeneticSearch + CfsSubsetEval + RuleEval + NN is because the processing power of the neural network was enhanced with the rule-based selected features that produced the reduced features. According to Table 9, the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) has 5.9983% error which justify its high accuracy for email spambase classification.

Table 10 displays the classification outcomes for different machine learning models using the complete feature set for the test set spambase. On the test set spambase class with all available features, the majority of machine learning methods achieved F1-scores of at least 73.2% classification rates. GeneticSearch + CfsSubsetEval + RuleEval + NN has an F1-score of 98.4%, which is somewhat higher than Bayesian Logistic Regression's score of 94.8%. A significant portion of the training instances in each technique were accurately categorized as spam using the entire feature set, while just a tiny portion of the instances were incorrectly labeled as spam. The significant score discrepancies between the true positive and the false positive scores can be used to explain this. Additionally, the false positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is lower than the closest value of the Bagging with 1.8% < 4.1%, and the true positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly higher than that of the Bayesian Logistic Regression with 97% > 95.5%. The results showed that GeneticSearch + CfsSubsetEval + RuleEval + NN is best for spam class classification on the full feature set. Overall, the results demonstrated that the various machine learning methods performed fairly evenly on the test set spambase using the entire feature set. The processing time for the GeneticSearch + CfsSubsetEval + RuleEval + NN model was 0.6 s as opposed to 0.03 s for the Naive Bayes Multinomial technique. As a result, the Naive Bayes Multinomial had the quickest processing time for the entire feature set. These results showed that the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) is relatively consistent and can effectively categorize emails as spambase. Hence, the justification for the precision value of 0.982 and F1-score of 0.984. The result also justifies the need for fuzzy logic to resolve misclassification uncertainty.

Table 11 show the classification results of various machine learning models on full feature set for test set no-spam. Most of the machine learning algorithms on test set no-spam class with full feature set obtained F1-score of higher than 80% classification rates. Compared to Bayesian Logistic Regression, which has the closest score of 95.8%, GeneticSearch + CfsSubsetEval + RuleEval + NN has a slightly higher F1-score of 98.3%. A significant portion of the instances were accurately categorized as no-spam using the full feature set in all of the techniques, whereas just a tiny portion of the instances were incorrectly labeled as no-spam. The stark contrast in scores between the true positive and false positive results can be used to explain this. Additionally, the false positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is lower than the closest value of the Bayesian Logistic Regression with 1.9% < 3%, and the true positive of the GeneticSearch + CfsSubsetEval + RuleEval + NN is higher than the closest result of the Bagging with 98% > 95.9%. This result can be attributed to the resultant small feature set of the GeneticSearch + CfsSubsetEval + RuleEval + NN compared to the other methods. This is due to the developed approach still doing feature reduction on the entire feature set automatically. Overall, the results demonstrated fairly equal performance across the various machine learning algorithms on the test set no-spam on the whole feature set. The time taken to build the model for GeneticSearch + CfsSubsetEval + RuleEval + NN showed a processing time of 0.6 s > 0.03 s of the Naïve Bayes Multinomial method. This means that the Naïve Bayes Multinomial obtained the fastest processing time on the full feature set. These results showed that

Table 10
Performance summary statistics of different models on full feature set for test set spambase.

Algorithm	TP	FP	Precision	Recall	F1-score	AUC	Time (s)
J48 tree	0.903	0.059	0.908	0.903	0.906	0.937	3.03
Decision Table	0.855	0.100	0.847	0.855	0.851	0.945	12.46
Bagging	0.907	0.041	0.935	0.907	0.921	0.978	4.44
kNN	0.867	0.075	0.882	0.867	0.874	0.899	0.7
Logistic	0.888	0.050	0.921	0.888	0.904	0.971	1.53
RBF Network	0.865	0.231	0.789	0.789	0.789	0.789	2.29
Naïve Bayes	0.952	0.305	0.670	0.952	0.786	0.940	0.23
Bayesian Logistic Regression	0.955	0.052	0.948	0.947	0.948	0.947	0.58
Naïve Bayes Multinomial	0.721	0.163	0.742	0.721	0.732	0.848	0.03
Multilayer Perceptron	0.890	0.061	0.905	0.890	0.897	0.962	27.44
GeneticSearch + CfsSubsetEval + RuleEval + NN	0.97	0.018	0.982	0.982	0.984	0.979	0.6

Table 11
Performance summary statistics of different models on full feature set for test set no- spambase.

Algorithm	TP	FP	Precision	Recall	F1-score	AUC	Time (s)
J48 tree	0.941	0.097	0.937	0.941	0.939	0.937	3.03
Decision Table	0.900	0.145	0.905	0.900	0.902	0.945	12.46
Bagging	0.959	0.093	0.941	0.959	0.950	0.978	4.44
kNN	0.925	0.133	0.914	0.925	0.919	0.899	0.7
Logistic	0.950	0.112	0.929	0.950	0.939	0.971	1.53
RBF Network	0.769	0.135	0.851	0.854	0.851	0.854	2.29
Naïve Bayes	0.695	0.048	0.957	0.695	0.805	0.936	0.23
Bayesian Logistic Regression	0.712	0.03	0.959	0.959	0.958	0.959	0.58
Naïve Bayes Multinomial	0.837	0.279	0.822	0.837	0.830	0.848	0.03
Multilayer Perceptron	0.939	0.110	0.929	0.939	0.934	0.962	27.44
GeneticSearch + CfsSubsetEval + RuleEval + NN	0.98	0.019	0.981	0.981	0.983	0.979	0.6

Table 12
Performance summary statistics of different models on full features set.

Algorithm	Accuracy (%)	Error (%)	MAE	RMSE	RAE (%)	RRSE (%)
J48 tree	92.6103	7.3897	0.0912	0.2562	18.6861	52.4351
DecisionTable	88.1982	11.8018	0.198	0.2922	41.4655	59.7881
Bagging	93.8709	6.1291	0.107	0.2173	22.4027	44.4628
kNN	90.176	9.824	0.0924	0.3036	19.3577	62.1283
Logistic	92.5658	7.4332	0.12	0.2434	25.1307	49.816
RBF Network	80.6564	19.3436	0.2616	0.3633	54.789	74.355
Naïve Bayes	79.6131	20.3869	0.2066	0.4527	43.2668	92.6423
Bayesian Logistic Regression	81.3736	18.6264	0.1863	0.4316	39.0034	88.3224
Naïve Bayes Multinomial	79.1567	20.8433	0.2068	0.4246	43.3133	86.8832
Multilayer Perceptron	91.98	8.02	0.1151	0.2712	24.0931	55.4976
GeneticSearch + CfsSubsetEval + RuleEval + NN	94.5449	6.4551	0.0908	0.2285	19.0053	46.7573

KEY:
Mean absolute error (MAE).
Root mean squared error (RMSE).
Relative absolute error (RAE).
Root relative squared error (RRSE).

the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) is consistent and can effectively categorize emails as spambase. Hence, the justification for the precision value of 0.981 and F1-score of 0.983.

Table 12 show the performance summary statistics of various machine learning models on full features set. The mean absolute error of the GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly lower than that of the J48 tree with 9.08 > 9.12, plus the accuracy of the GeneticSearch + CfsSubsetEval + RuleEval + NN is slightly better than Bagging with 94.5449% > 93.8709%. From looking at the accuracy alone, it can be seen that the GeneticSearch + CfsSubsetEval + RuleEval + NN outperformed the Bagging method. These results immediately reveals that the GeneticSearch + CfsSubsetEval + RuleEval + NN has also outperformed other related methods. The hybrid feature selection and rule evaluation strategy, which resulted in the reduced features, may have improved the neural network's processing capability, which could account for the superior performance of the GeneticSearch + CfsSubsetEval + RuleEval + NN.

The performance comparison for fewer features for test set spambase is shown in Fig. 4. Besides Naïve Bayes Multinomial and Naïve Bayes methods, the developed method showed high true positive value of 96.7%. Similarity, the developed approach is slightly better in term of false positive than that of the closest result of Logistic with 3.6% < 4.7%. The F1-score of the developed method is superior with 96.5% than the closest results of J48 tree and Bagging with a joint F1-score value of 90.3%. The time taken to build the developed approach is very negligible compared to the closest time of kNN with 0.5 s < 0.6 s. These results immediately reveals that the developed method outperformed other related methods. According to Table 12, the proposed approach (GeneticSearch + CfsSubsetEval + RuleEval + NN) has 6.4551% error which justify its high accuracy for email spambase classification.

Fig. 5 show the performance comparison on reduced features for test set no-spam. The closest result of the Logistic showed a value of 95.3%, while the true positive of the created technique showed the maximum value of 96.8%. Similarity the developed approach is higher in term of false positive than that of the best result of Naïve Bayes Multinomial with 3.7% > 1.8%. The application of fuzzy logic to resolve misclassification is the justification for the higher false positive of the developed approach in comparison with the result of Naïve Bayes Multinomial. The F1-score of the developed method is superior to the closest result of RBFNetwork with values of 96.4% and 94.9% respectively. The time taken to build the developed approach is very

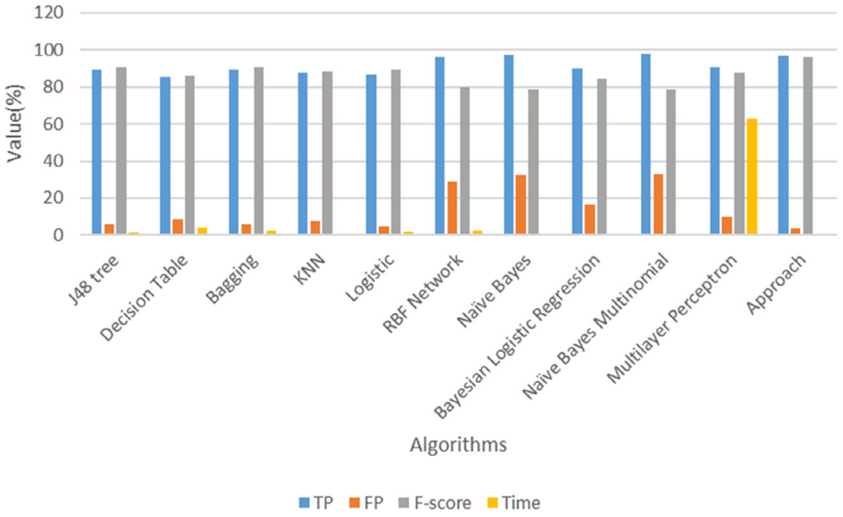


Fig. 4. Performance comparison on reduced features for test set spambase.

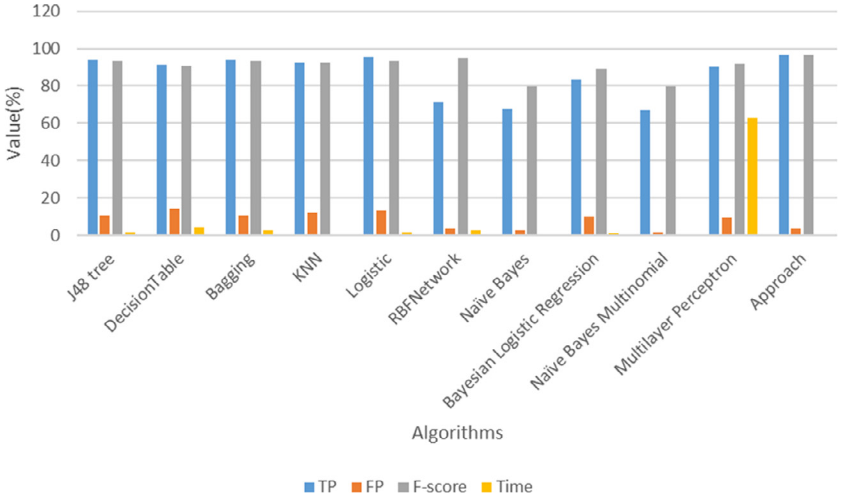


Fig. 5. Performance comparison on reduced features for test set no-spam.

negligible compared to the closest time of kNN with 0.5 s < 0.6 s. The results in term of F1-score immediately reveals that the developed method outperformed other related methods.

The summary statistics based on accuracy and MAE on the reduced feature set are displayed in Fig. 6. The accuracy of the developed approach is somewhat greater than that of the closest result of Bagging with 94.0017% > 93.1971%, while the mean absolute error is higher than that of the J48 tree with 11.18 > 8.93. It is clear that the created methodology outperformed the other similar methods just by looking at accuracy.

The performance comparison on full features for test set spambase is shown in Fig. 7. When compared to the nearest outcome of Bayesian Logistic Regression, which showed a value of 95.5%, the true positive of the created approach showed the highest value of 97%. Similarity the developed approach is slightly better in term of false positive than that of the closest result of Bagging with 1.8% < 4.1%. With values of 98.4% and 94.8%, respectively, the F1-score of the developed approach is marginally higher than the nearest result of Bayesian Logistic Regression. The time taken to build the developed approach is very negligible compared to the closest time of kNN with 0.6 s < 0.7 s. These results immediately reveals that the developed method outperformed other related methods on full features for test set spambase.

Fig. 8 show the performance comparison on full features for test set no-spam. In contrast to the nearest Bagging result, which had a value of 95.9%, the true positive of the developed approach had the highest value of 98%. Similarity the developed approach is lower in term of false positive than that of the closest result of Bayesian Logistic Regression with 1.9% < 3%. With values of 98.3% and 95.8%, respectively, the F1-score of the developed approach is just marginally superior to the closest outcome of Bayesian Logistic Regression. The time taken to build the developed approach is very negligible compared to the closest time of kNN with 0.6 s < 0.7 s. The results in term of F1-score immediately reveals that the developed method outperformed other related methods.

The summary statistics based on accuracy and MAE for the entire feature set are displayed in Fig. 9. The accuracy of the developed approach is marginally greater than that of the closest result of Bagging with 94.5449% > 93.8709%, and the mean absolute error is marginally lower than that of the J48 tree with 9.08 < 9.12. The developed method outperformed the other similar methods when the mean absolute error and accuracy were examined.

In Fig. 10, the fuzzy input variables for spam classification are added between the 0 and 1 range in the fuzzy inference system editor. The fuzzy input parameters consist of:

i. WORD: proportion of WORD-matched characters in the email
ii. CHAR: proportion of CHAR-matched characters in the email
iii. AVE: average number of uppercase letters in a row without a break

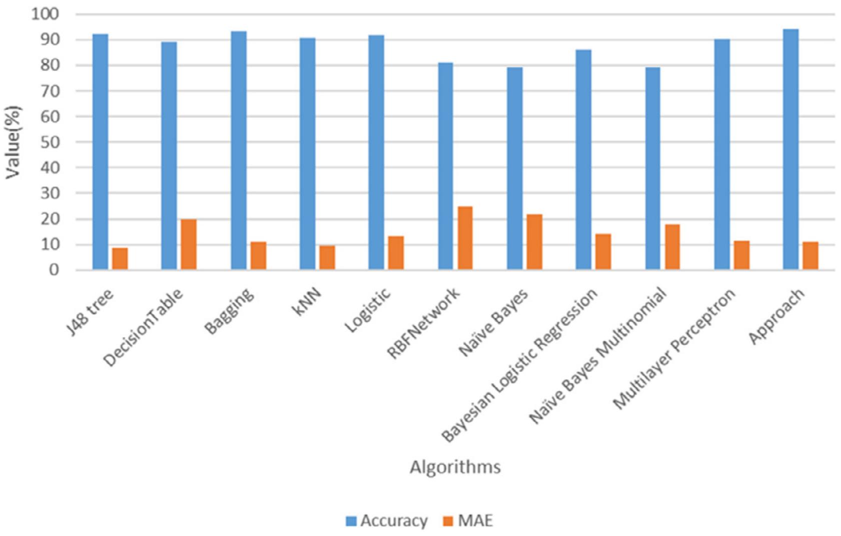


Fig. 6. Summary statistics based on accuracy and MAE on reduced feature set.

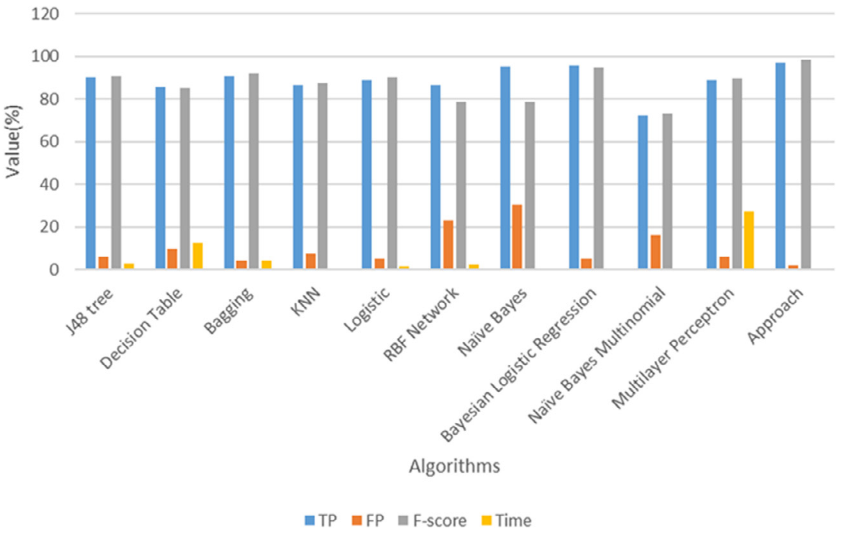


Fig. 7. Performance comparison on full features for test set spambase.

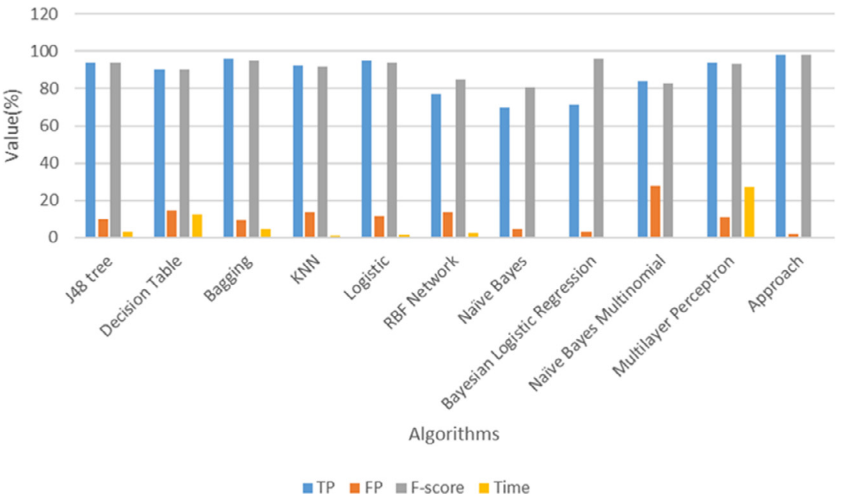


Fig. 8. Performance comparison on full features for test set no-spam.

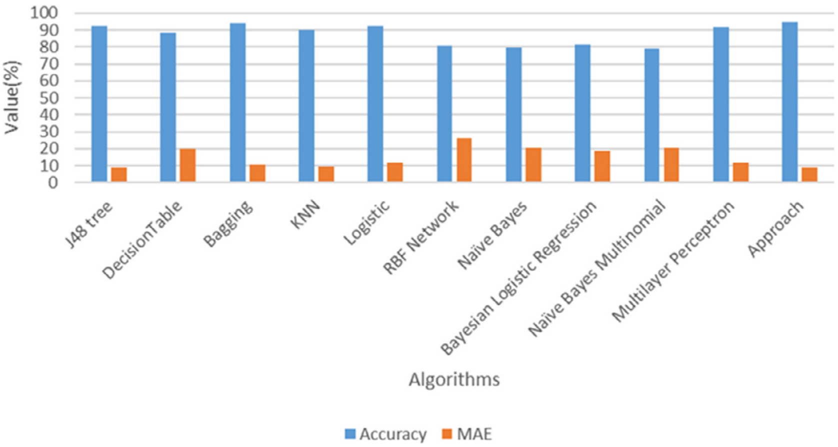
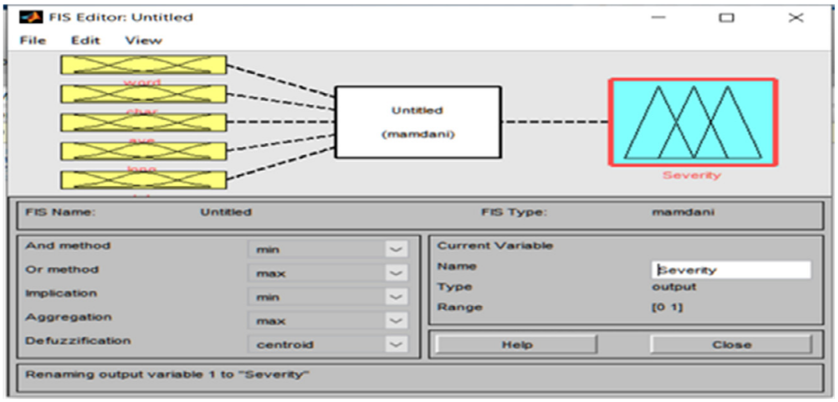


Fig. 9. Summary statistics based on accuracy and MAE on full feature set.



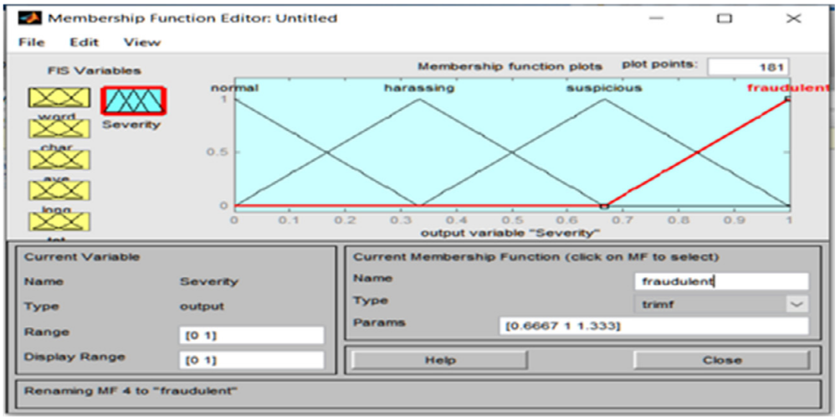
FIS Name: Untitled
FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: Severity
Type: output
Range: [0 1]

Renaming output variable 1 to "Severity"

Fig. 10. Editor for fuzzy input variables.



plot points: 181

Current Variable
Name: Severity
Type: output
Range: [0 1]
Display Range: [0 1]

Current Membership Function (click on MF to select)
Name: fraudulent
Type: trimf
Params: [0.6667 1 1.333]

Renaming MF 4 to "fraudulent"

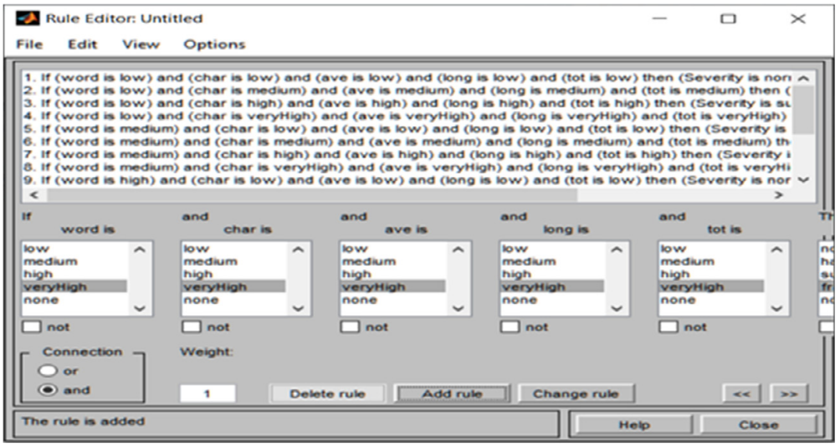
Fig. 11. Membership function editor.

iv. LONG: length of the longest continuous capital letter sequence
v. TOT: total number of capital letters in the email

The membership function editor for each of the fuzzy variables is shown in Fig. 11. The editor that makes it possible to define linguistic variables for various fuzzy variables within a defined fuzzy range of values denoted by low, medium, high, and extremely high. Within the predetermined range of values, these linguistic variables enable fuzzification of the fuzzy variables.

Fig. 12 depicts the rule editor for the linguistic and fuzzy variables. Rules are added and specified in the rule editor depending on subject-matter expertise. The rule of thumb defined a total of 16 rules.

The rule viewer modifications for email spam classification are shown in Table 13. The outcome of the rule modification revealed the amount of email spam severity. The findings demonstrated that when all the input factors are medium (rule #1), email spam is classified as harassing. When the five variables are in the following order: medium, high, very high, very high, and very high (rule #2), an email spam is categorized as suspicious. Another outcome is that email spam is labeled as fraudulent when all five factors are extremely high (rule #3). The interpretation of the other outcomes is similar. These findings demonstrated that an email spam will typically produce a level of severity that is harassing. Other times, an email spam will produce a fraudulent severity level.



1. If (word is low) and (char is low) and (ave is low) and (long is low) and (tot is low) then (Severity is norı
2. If (word is low) and (char is medium) and (ave is medium) and (long is medium) and (tot is medium) then (
3. If (word is low) and (char is high) and (ave is high) and (long is high) and (tot is high) then (Severity is su
4. If (word is low) and (char is veryHigh) and (ave is veryHigh) and (long is veryHigh) and (tot is veryHigh)
5. If (word is medium) and (char is low) and (ave is low) and (long is low) and (tot is low) then (Severity is
6. If (word is medium) and (char is medium) and (ave is medium) and (long is medium) and (tot is medium) th
7. If (word is medium) and (char is high) and (ave is high) and (long is high) and (tot is high) then (Severity i
8. If (word is medium) and (char is veryHigh) and (ave is veryHigh) and (long is veryHigh) and (tot is veryHi
9. If (word is high) and (char is low) and (ave is low) and (long is low) and (tot is low) then (Severity is nor

Connection: ○ or ● and

Weight: 1

The rule is added

Fig. 12. Editor the fuzzy rules.

Table 13
Results of fuzzy email spam classification.

No	Word	Char	Ave	Long	Tot	Spam severity level
1	0.5	0.5	0.5	0.5	0.5	Harassing
2	0.5	0.657	0.882	0.827	0.824	Suspicious
3	0.918	0.898	0.973	0.827	0.824	Fraudulent
4	0.136	0.0833	0.209	0.118	0.0648	Normal
5	0.591	0.0833	0.627	0.882	0.0648	Harassing
6	0.336	0.343	0.3	0.864	0.231	Harassing
7	0.882	0.935	0.882	0.936	0.972	Fraudulent
8	0.282	0.38	0.355	0.773	657	Harassing
9	0.609	0.88	0.845	0.918	0.898	Harassing
10	0.3	0.806	0.645	0.445	0.324	Harassing

Table 14 presents Friedman descriptive statistics for the email spam classification machine learning techniques. The Friedman test was carried out to demonstrate the significant value of the outcome of this research in comparison to other outcomes for classifying email spam. To compare the effectiveness of the various email spam classification systems, the Friedman test was utilized. Therefore, Table 14 showed an upward performances from GeneticSearch + CfsSubsetEval + RuleEval + NN (median = 0.96400), to J48 tree (median = 0.89600), to Bagging (median = 0.89600), to kNN (median = 0.8800), to Multilayer Perceptron (median = 0.87900), to Logistic (median = 0.86700), to Decision Table (median = 0.85700), to Bayesian Logistic Regression (median = 0.84500), Naïve Bayes Multinomial (median = 0.78800), Naïve Bayes (median = 0.78500), and RBF Network (median = 0.76900). The Friedman test mean rank differences between the various techniques of classifying email spam are shown in Table 15. The result of the Friedman test is shown in Table 16, along with a determination of whether there was a statistically significant overall difference between the mean ranks of the various email spam classification techniques. So, the values of the developed method is significant for the email spam classification, $x^2\,(10) = 21.927$, p = 0.015, since $p < 0.05$.

5. Conclusion

A hybrid correlation-based deep learning model of email spam classification using fuzzy inference system was developed in this study. In order to choose the most crucial characteristics from a preprocessed spambase dataset, a hybrid feature selection approach that combines the Genetic search method, CfsSubsetEval, and a rule evaluation was created. The CfsSubsetEval determines how each attribute and class are related. Next, preference is given to the attribute-class association with the highest correlation. The genetic search algorithm examines the merits of each attribute based on this feature evaluation and returns the features with the highest fitness value. The rule-based engine returns the feature subset with the fewest amount of subset features when two feature subsets with the same fitness value are compared. The chosen features are subsequently fed into a deep learning model for spam classification, and during the classification stage, fuzzy logic is used to categorize a specific spam category into its severity levels. The developed method showed better F1-score results for both test set spambase and test set non-spambase of 96.5% and 96.4% respectively, compared with other machine learning methods. Similarly, the developed method showed better accuracy, error rate, and processing time of 94.0017%, 5.9983%, and 0.5 s respectively when compared with other machine learning methods. The developed method also showed reduce misclassification based on fuzzy inference system. The Friedman test was conducted to state the significant value of the results obtained from comparing the developed technique and other machine learning techniques. As a result, the results of the devised approach are significant for the categorization of email spam at p = 0.015. Future research can focus on developing a more reliable deep learning architecture with cutting-edge optimization methods. For more accurate

Table 14
Friedman descriptive statistics.

Technique	n	Percentiles		
		25th	50th (Average)	75th
J48 tree	5	0.47700	0.89600-2	0.90600
Decision Table	5	0.47250	0.85700-6	0.86150
Bagging	5	0.47700	0.89600-2	0.90600
kNN	5	0.4775	0.8800-3	0.8830
Logistic	5	0.45700	0.86700-5	0.94050
RBF Network	5	0.52850	0.76900-10	0.88100
Naïve Bayes	5	0.49250	0.78500-9	0.97100
Bayesian Logistic Regression	5	0.50500	0.84500-7	0.87250
Naïve Bayes Multinomial	5	0.49500	0.78800-8	0.98200
Multilayer Perceptron	5	0.47750	0.87900-4	0.90300
GeneticSearch + CfsSubsetEval+ RuleEval + NN	5	0.50000	0.96400-1	0.96600

Table 15
Mean ranks for Friedman test.

Technique	Mean rank
J48 tree	6.50
Decision Table	4.20
Bagging	6.50
kNN	5.40
Logistic	5.20
RBF Network	4.80
Naïve Bayes	6.50
Bayesian Logistic Regression	4.80
Naïve Bayes Multinomial	7.30
Multilayer Perceptron	6.60
GeneticSearch + CfsSubsetEval + RuleEval + NN	8.20

Table 16
Test statistics for Friedman.

	Test statistics[a]
n	5
x^2 (Chi-Square)	21.927
df	10
$Asymp.Sig.$	0.015

email spam classification, the fuzzy logic linguistic variables can be expanded to include more severity classes. With the Nigeria dataset, machine learning methods can also be used for multilabel email spam detection.

Declaration of competing interest

The authors declared they have no conflict of interest.
No funder or grant was received for the research work.

Data availability

Data will be made available on request.

References

[1] R.K. Kumar, G. Poonkuzhali, P. Sudhakar, Comparative study on email spam classifier using data mining techniques, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 1, 2012, pp. 14–16.

[2] M. Abdullahi, A.D. Mohammed, S.A. Bashir, O.O. Abisoye, A review on machine learning techniques for image based spam emails detection, in: 2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA), IEEE, 2021, pp. 59–65.

[3] Y. Lin, (n.d.). How Many People Use Email in 2023? [2023 Update]. https://www.oberlo.com/statistics/how-many-people-useemail#:~:text=According%20to%20a%20study%20that,half%20of%20the%20global%20population.

[4] A. Sharaff, N.K. Nagwani, A. Dhadse, Comparative study of classification algorithms for spam email detection, in: Emerging Research in Computing, Information, Communication and Applications, Springer, New Delhi, 2016, pp. 237–244.

[5] A.F. Yasin, Spam reduction by using E-mail history and authentication (SREHA), Int. J. Comput. Netw. Inf. Secur. 8 (7) (2016).

[6] J.B. Awotunde, Y.J. Oguns, K.A. Amuda, N. Nigar, T.A. Adeleke, K.M. Olagunju, S.A. Ajagbe, Cyber-physical systems security: Analysis, opportunities, challenges, and future prospects, Blockchain Cybersecur. Cyber-Phys. Syst. (2023) 21–46.

[7] K. Raghavendar, I. Batra, A. Malik, A robust resource allocation model for optimizing data skew and consumption rate in cloud-based IoT environments, Decis. Anal. J. 7 (2023) 100200.

[8] A. Bilgram, P.G. Jensen, K.Y. Jørgensen, K.G. Larsen, M. Mikučionis, M. Muñiz, et al., An investigation of safe and near-optimal strategies for prevention of Covid-19 exposure using stochastic hybrid models and machine learning, Decis Anal. J. 5 (2022) 100141.

[9] S. Magdy, Y. Abouelseoud, M. Mikhail, Efficient spam and phishing emails filtering based on deep learning, Comput. Netw. 206 (2022) 108826.

[10] T.A. Almeida, A. Yamakami, Facing the spammers: A very effective approach to avoid junk e-mails, Expert Syst. Appl. 39 (7) (2012) 6557–6561.

[11] U. Maqsood, S. Ur Rehman, T. Ali, K. Mahmood, T. Alsaedi, M. Kundi, An intelligent framework based on deep learning for SMS and e-mail spam detection, Appl. Comput. Intell. Soft Comput. 2023 (2023) 1–16.

[12] S. Douzi, F.A. AlShahwan, M. Lemoudden, B. Ouahidi, Hybrid email spam detection model using artificial intelligence, Int. J. Mach. Learn. Comput. 10 (2) (2020) 316–322.

[13] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Syst. Appl. 36 (7) (2009) 10206–10222.

[14] M. Healy, S. Delany, A. Zamolotskikh, An assessment of case-based reasoning for short text message classification, in: Procs. of 16th Irish Conference on Artificial Intelligence and Cognitive Science, (AICS-05), 2005, pp. 257–266.

[15] J.B. Awotunde, F.E. Ayo, R. Panigrahi, A. Garg, A.K. Bhoi, P. Barsocchi, A multi-level random forest model-based intrusion detection using fuzzy inference system for internet of things networks, Int. J. Comput. Intell. Syst. 16 (1) (2023) 31.

[16] F.E. Ayo, J.B. Awotunde, O.A. Olalekan, A.L. Imoize, C.T. Li, C.C. Lee, CB-FISKD: A combinatorial-based fuzzy inference system for keylogger detection, Mathematics 11 (8) (2023) 1899.

[17] D. Sculley, G.M. Wachman, Relaxed online SVMs for spam filtering, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 415–422.

[18] A.K. Uysal, S. Gunal, A novel probabilistic feature selection method for text classification, Knowl.-Based Syst. 36 (2012) 226–235.

[19] W. Liu, T. Wang, Online active multi-field learning for efficient email spam filtering, Knowl. Inf. Syst. 33 (1) (2012) 117–136.

[20] R. Shams, R.E. Mercer, Personalized spam filtering with natural language attributes, in: 2013 12th International Conference on Machine Learning and Applications, Vol. 2, IEEE, 2013, pp. 127–132.

[21] S.K. Trivedi, S. Dey, An enhanced genetic programming approach for detecting unsolicited emails, in: 2013 IEEE 16th International Conference on Computational Science and Engineering, IEEE, 2013, pp. 1153–1160.

[22] B. Zhou, Y. Yao, J. Luo, Cost-sensitive three-way email spam filtering, J. Intell. Inf. Syst. 42 (1) (2014) 19–45.

[23] S.K. Trivedi, S. Dey, A combining classifiers approach for detecting email spams, in: 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, 2016, pp. 355–360.

[24] E.M. Bahgat, S. Rady, W. Gad, An e-mail filtering approach using classification techniques, in: The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November (2015) 28-30, Beni Suef, Egypt, Springer, Cham, 2016, pp. 321–331.

[25] R.K. Kaliyar, P. Narang, A. Goswami, SMS spam filtering on multiple background datasets using machine learning techniques: A novel approach, in: 2018 IEEE 8th International Advance Computing Conference (IACC), IEEE, 2018, pp. 59–65.

[26] V. Gupta, A. Mehta, A. Goel, U. Dixit, A.C. Pandey, Spam detection using ensemble learning, in: Harmony Search and Nature Inspired Optimization Algorithms, Vol. 74, Springer, Singapore, 2019, pp. 661–668.

[27] P. George, P. Vinod, Composite email features for spam identification, in: Cyber Security, Springer, Singapore, 2018, pp. 281–289.

[28] H.Y. Lee, S.S. Kang, Word embedding method of sms messages for spam message filtering, in: 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2019, pp. 1–4.

[29] M. Diale, T. Celik, C. Van Der Walt, Unsupervised feature learning for spam email filtering, Comput. Electr. Eng. 74 (2019) 89–104.

[30] D. Gaurav, S.M. Tiwari, A. Goyal, N. Gandhi, A. Abraham, Machine intelligence-based algorithms for spam filtering on document labeling, Soft Comput. 24 (13) (2020) 9625–9638.

[31] T. Xia, X. Chen, A weighted feature enhanced hidden Markov model for spam SMS filtering, Neurocomputing 444 (2021) 48–58.

[32] M. Ghiassi, S. Lee, S.R. Gaikwad, Sentiment analysis and spam filtering using the YAC2 clustering algorithm with transferability, Comput. Ind. Eng. (2022) 107959.

[33] P. Rajendran, A. Tamilarasi, R. Mynavathi, A collaborative abstraction based email spam filtering with fingerprints, Wirel. Pers. Commun. 123 (2) (2022) 1913–1923.

[34] F. Hossain, M.N. Uddin, R.K. Halder, Analysis of optimized machine learning and deep learning techniques for spam detection, in: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE, 2021, pp. 1–7.

[35] W.S. Jacob, Multi-objective genetic algorithm and CNN-based deep learning architectural scheme for effective spam detection, Int. J. Intell. Netw. 3 (2022) 9–15.

[36] V. Nosrati, M. Rahmani, A. Jolfaei, S. Seifollahi, A weak-region enhanced Bayesian classification for spam content-based filtering, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 22 (3) (2023) 1–18.

[37] W. Pan, J. Li, L. Gao, L. Yue, Y. Yang, L. Deng, C. Deng, Semantic graph neural network: A conversion from spam email classification to graph classification, Sci. Program. 2022 (2022) 1–8.

[38] S.S. Ismail, R.F. Mansour, A. El-Aziz, M. Rasha, A.I. Taloba, Efficient E-mail spam detection strategy using genetic decision tree processing with NLP features, Comput. Intell. Neurosci. 2022 (2022) 1–16.

[39] C.M. Shaik, N.M. Penumaka, S.K. Abbireddy, V. Kumar, S.S. Aravinth, Bi-LSTM and conventional classifiers for email spam filtering, in: 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), IEEE, 2023, pp. 1350–1355.

[40] M.A. Abid, S. Ullah, M.A. Siddique, M.F. Mushtaq, W. Aljedaani, F. Rustam, Spam SMS filtering based on text features and supervised machine learning techniques, Multimedia Tools Appl. 81 (28) (2022) 39853–39871.

[41] H.J. Alshahrani, K. Tarmissi, A. Yafoz, A. Mohamed, A. Motwakel, I. Yaseen, M. Mahzari, Improved fruitfly optimization with stacked residual deep learning based email classification, Intell. Autom. Soft Comput. 36 (3) (2023).

[42] P.Prasanna. Bharathi, G. Pavani, K. Krishna Varshitha, V. Radhesyam, Spam SMS filtering using support vector machines, in: Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020, Springer Singapore, 2021, pp. 653–661.

[43] S. Mani, G. Gunasekaran, S. Geetha, Email spam detection using gated recurrent neural network, Int. J. Prograssive Res. Eng. Manage. Sci. (IJPREMS) 3 (2023) 90–99.

[44] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, New Jersey, 1999.

[45] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT, Massachusetts, 1992.

[46] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99, Springer, USA.

[47] R. Alcalá, M.J. Gacto, F. Herrera, J. Alcalá-Fdez, A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 15 (05) (2007) 539–557, World Scientific: Singapore.

[48] A. Fernandez, V. Lopez, M.J. del Jesus, F. Herrera, Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges, Knowl.-Based Syst. 80 (2015) 109–121, New York: USA, Elsevier.

[49] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.