



SMS sentiment classification using an evolutionary optimization based fuzzy recurrent neural network

Ulligaddala Srinivasarao¹ · Aakanksha Sharaff¹

Received: 17 April 2022 / Revised: 13 September 2022 / Accepted: 30 March 2023 /
Published online: 11 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Sentiment analysis using the inbox message polarity is a challenging task in text mining, this analysis is used to differentiate spam and ham messages in mail. Polarity estimation is mandatory for spam and ham identification, whereas developing a perfect architecture for such classification is the hot demanding topic. To fulfill that, fuzzy based Recurrent Neural network-based Harris Hawk optimization (FRNN-HHO) is introduced, which performs post-classification over the classified messages (spam and ham). Previously the authors tried to classify the spam and ham messages from the collection of SMSs. But sometimes, the spam messages may incorrectly be classified within the ham classes. This misclassification may reduce the accuracy. The sentiment analysis process is performed over the classified messages to improve such classification accuracy. The spam and ham messages from the available data are classified using a Kernel Extreme Learning Machine (KELM) classifier. The sentiment analysis and classification based experimental evaluation is carried out using accuracy, recall, f-measure, precision, RMSE, and MAE. The performance of the proposed architecture is evaluated using three different datasets: SMS, Email, and spam-assassin. The Area under the curve (AUC) of the proposed approach is found to be 0.9699 (SMS dataset), 0.958 (Email dataset), and 0.95 (spam assassin).

Keywords SMS · Spam and ham · Sentiment analysis · Fuzzy recurrent neural network · Harris hawk optimization · Kernel extreme learning machine

1 Introduction

The Internet has provided different platforms to share their ideas and views. SMS, email, and Twitter are the most attractive platforms. User interactions easily contribute to creating subgroups by having high-density connections between participants. The

✉ Ulligaddala Srinivasarao
usrinivasarao.phd2018.cs@nitrr.ac.in

Aakanksha Sharaff
asharaff.cs@nitrr.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology Raipur, Chhattisgarh 492010, India

profuse texts normally provide useful knowledge, which is found valuable for various purposes [26]. With the exponential growth of social networking sites (forums, journals, and social networks), the use of these emerging sources of knowledge has become important [19]. The vocabularies in normally written sentences and SMS messages are different. The sentiment words from SMS big data and general-purpose text corpus are extracted for SMS sentiment analysis by exploring an automated sentiment lexicon constructor [14]. But the presence of spam may reduce the attractiveness of SMS communication. Spam is a random unsolicited message forwarded in large amounts without the receiver's authority. People also struggle with the abuse of SMS by spammers, who try to make fake statements to access users' private details. Emails, social media pages, analysis, and even Facebook had spammers seeking to break through the emergence of the Web everywhere. Spam occurs in several ways, such as through blogs, search results, tweets, and personal communications, through which spammers aim to obtain money [17].

Spamming campaigns have been utilized for both online fraud and the sale of products. Several malicious activities are minimized by the many approaches introduced and investigated by the researchers in an underground economy that reports billions of benefits dollars. With traditional methods, automatic spam detection has been started with the spam detection design [30]. For spam filtering, such simple techniques named blacklisting and content-based machine learning (ML) are utilized in traditional spam detection. Different types of ML, deep learning (DL), and traditional algorithms are used for spam classification and sentiment analysis. Only a few works have concentrated on identifying the sentiment from the tweets, SMS, email, etc., on better identifying spam messages from inbox.

The sentiment analysis aims to examine the informally written text by average citizens and collect people's views about goods and incidents and certain things. Numerous organizations and policymakers are interested in collecting emotions from micro-blogs in various domains like business, disaster management, and health. The increasing usage of social media has provided countless ways for communities to share their opinions and thoughts. The study of emotions attempts to collect and interpret the views of individuals. In these last years, the Sentiment Analysis (SA) field has seen a massive burst in study attention, but still, there remains a growing curiosity. Recently, there are several demanding research topics have been developed on opinion mining and other important domains. Among that, possible sentiment classification from filtered spam is considered the most demanding. Previous research in emotion analysis has often implemented ML techniques to categorize the sentiment polarity from spam SMS into positive, neutral, or negative.

The sentiment analysis from classified inbox messages is a demanding topic, as it enhances the overall efficiency of spam identification. Therefore, to motivate this, a hybrid fuzzy based network is introduced in the proposed architecture, which uses fuzzy concepts and obtains a crisp output during sentiment analysis. In this, the hybrid fuzzy and RNN is considered the major advantage because the fuzzy greatly impacts processing text during detection and classification. Moreover, the RNN can analyze huge amounts of text data with less error rate. These two merits are combined in this proposed architecture and have introduced this architecture for sentiment analysis from the classified messages. Combining the fuzzy RNN and HHO is considered the major novelty in this approach. This approach has hybrid merit of HHO, RNN, and fuzzy, which enhances the overall efficiency of sentiment analysis. This is the first approach which has combined these three processing techniques for analyzing the sentiments from the classified inbox messages. Recently few works like [29, 34, 44] have been developed for sentiment analysis and spam classification, but these techniques haven't

achieved satisfactory results in terms of accuracy and error. Therefore, a hybrid architecture is introduced in the proposed architecture, which analyses the classified emails efficiently and achieves better efficiency.

The major contribution of the proposed algorithm is:

- The proposed KELM classifier achieves the spam and ham classification. This classification is carried out to analyze the sentiments from the ham and spam messages.
- The presence of sentiments can easily allow us to classify spam and ham messages. Therefore, we have included sentiment analysis and spam/ham classification in this proposed work.
- Finally, based on the analyzed sentiments, the spam and ham messages are accurately classified by the proposed framework. Normally, the dataset available for sentimental analysis is found unlabeled. Therefore, Textblog based labelling is used before the sentiment analysis process to label such datasets.

The entire manuscript is organized as follows: The recent related works carried out for SMS classification, and sentiment analysis from SMS messages are discussed in Section 2. A clear and brief description of the proposed classification and sentiment analysis process is carried out in Section 3. Then, the classification, sentiment analysis, and optimization-based results are evaluated and discussed in Section 4. Finally, the overall work is concluded in Section 5.

2 Related Works

A few recent works that perform sentiment analysis and spam classification are reviewed below. Further, the works which have performed sentiment analysis and spam classification separately are also discussed in the below sections:

Analyzing the sentiments from the classified tweets using bidirectional long short-term memory (BiLSTM) was analyzed in [29], which obtains an efficient performance than other existing algorithms. The major goal of this process was to determine whether the given tweet as spam or ham. After that, the emotions related to that particular tweet were also evaluated using the BiLSTM model. This approach has used different methods for sentiment analysis they are RNN (Recurrent neural network), LSTM, 1D CNN, NB, RF, SVM (Support vector machine), and LR (Logistic regression). Extracted tweet features have improved the overall classification performance.

Removal of spam and fake news from the Internet based on sentiment analysis is demanding, and for that, an efficient architecture was developed [44]. A hybrid architecture was used for sentiment analysis, and then an ML like RCNN (Recurrent convolutional neural network), RF (Random forest), and NB (naïve Bayes) was used for fake news detection. The final estimated results indicate that RCNN has achieved better than other algorithms. GUI (Graphical User Interface) based spam detection was performed in [34], which concentrates on classifying the sentiments from the spam. It achieves 97% accuracy on sentiment analysis. For sentiment analysis, the Vader sentiment analyzer was used. Finally, the spam classification was performed using the binary and multi-label classifier (OnevsRest classifier).

2.1 Sentiment Analysis from SMS

The recent flourishing and emergence of SMS have dramatically altered how each individual interacts. Throughout these years, utilizing sentiment analysis to assess public opinion from online media networks has become a popular study topic throughout the view of this trend. LMAEB-CNN integrates CNN and Bi-LSTM, and a multi-head attention mechanism was proposed in [40] for accuracy improvement. It solves over-fitting issues and improves classification accuracy during polarity identification. N-gram and SVM models have attained higher precision results for large datasets, which were found to be better than NB Classifier models. The author in [6] gathered movie comments and achieved classification using maximum entropy, SVM, and NB classifier. Finally, 87.5% precision by this method during classification. Further, the AEB-CNN model (Attention of Emoticon-Based Convolutional Neural Network), which integrates a single direction attention mechanism and CNN model, was proposed in [40] for identifying the sentiment polarities without using any external knowledge.

The author in [41] introduced an emotion analysis to reveal hidden knowledge in Chinese short messages, which can be an important part of tracking and insights on social media. This method introduced an AEB-CNN approach, which integrates the attention-based and emoticons mechanisms with CNN for accuracy enhancement. [25] Have suggested a method to examine the problems of context-conscious SA. Initially, a hybrid lexicon was established based on context-related vague words found in current emotion lexicons. The context-aware polarity analysis was performed secondly by the hybrid lexicon. For handling missing data, the outcomes provided by the hybrid lexicon analysis were more effective, which was activated in the third.

Text standardization with embedding a deep convolutionary character level (Conv-char-Emb) neural network model of unstructured data for SA was proposed in [5]. Lemmatization, tokenization, stemming, detection, and replacing the OOV (out of vocabulary) were performed as pre-processing steps. A character-based embedding with a convolution neural network (CNN) is an easy and powerful strategy for SA, which utilizes attribute representation parameters that are less learnable. Therefore, the suggested approach conducts the normalization and sentiment classification for unstructured words for each SMS. The paraphrase from the noisy and clean texts was detected using DNN (deep neural network) [2]. Initially, the joint RNN-CNN architecture was used for sentence modelling, where the word embedding was taken as the source for CNN. CNN learned the local features, and RNN determined the long-term text dependencies. The text similarity was determined using a pair-wise similarity technique to estimate the major portions of the text.

The polarity from the texts was detected in [12], and an opinion mining technique was developed. Various users, like managers, buyers, and customers, were supported by this system. These users have analyzed the information in a multi-facet format. [27], introduced a modern form of sentiment approach, named sentiment phrase pattern matching (SPPM). It is a technique which determines the sentiments from the response text provided by students. Therefore, it is widely used for Educational purposes. Suggested the possible usage of sentiment mining for estimating the teaching via SMS texts of trainees. They also introduced the tree model to categorize students' feelings through a training course.

2.2 Spam and Ham detection from SMS

While the number of mobile users is growing rapidly, email id access and SMS spam messages are also rising tremendously. Various spam filtering approaches were tested to reduce

the annoyance of spam communications. [35], developed the genetic algorithm (GA) based Bayesian network, normalizing the data's noise. The text data was expanded using semantic and lexicographic dictionaries to achieve efficient spam filtering. This hybrid architecture performs feature selection and spam classification. Finally, it attains efficient performance using the word sense disambiguation approach. [24, 32] developing an SMS thread identification and email thread identification. The spam or ham from the SMS texts was classified using feed-forward neural network architecture. This research is an intermediate product of implementing the word2vec with deep learning and the SVM model [20]. In [33], spam message identification uses subject and content based. The cost-sensitive technique named ensemble learning base regularized deep neural network (EL-RDNN) was introduced in [8]. It comprises two stages they are feature selection and spam classification. With the selected attributes, spam filtering was performed. A multi-objective optimization technique was used for feature selection. Next, [18] developed LSTM (Long Short Term Memory) and RNN (Recurrent Neural Network) using Tensorflow and Keras model for spam and ham detection. The 'Spam SMS Collection' dataset was used in this method. TF-IDF, stop-word removal, tokenization, and vectorization were pre-processing techniques. With these techniques, this approach attained better detection accuracy.

To improve the performance of the Hidden Markov Model (HMM) over SMS, weight and label word based SMS formatting was introduced in [46]. Weight enhanced HMM has achieved better accuracy with rapid training and anti-spam detection. Finally, a performance analysis was carried out to show its effectiveness. Spam classification from SMS using BiLSTM was developed in [1], which utilizes two datasets for processing. Fine-tuning of ML parameters has improved the overall performance. A real-time collected dataset named expanded (ExAIS_SMS) was used to evaluate the BiLSTM performance on spam classification. To avoid spreading incorrect news related to COVID-19, a spam classification from the text data using an automatic approach was developed in [31]. It introduces a dynamic deep ensemble model for effective detection. The convolution and pooling of ensemble architecture were used for feature extraction. Finally, an extremely randomized tree and RF were used for spam classification.

Spam detection using ANN (artificial neural network), referred to as MLP (Multi-Layer Perceptron), was developed in [13]. It leverages the benefits of the Grasshopper Optimization Algorithm (GOA) to improve email classification performance. Performance evaluation was done over the standard dataset. Development on the Internet increases the spam count, which affects the user experience. Therefore, the Deep belief network (DBN) was introduced to enhance spam classification [21]. This DBN was combined with SMOTE (Synthetic Minority Over-Sampling Technique) and DAE (De-Noising Auto-Encoder) to enhance the classification performance. Email based sentiment analysis developed using lexicon methods [36–38].

2.3 Motivation

Recently, some techniques have been developed for spam classification and sentiment analysis. However, only a few works have concentrated on determining the sentiments from the classified inbox messages. This sentiment identification is valuable for determining is classified spam messages from the inbox. Previously developed techniques concentrate on identifying the misclassified messages in the inbox; however, these techniques do not concentrate on improving the overall efficiency of the sentiment analysis process. Considering this as a major contribution, the proposed approach has developed a hybrid architecture.

This hybrid architecture has introduced fuzzy along with RNN (FRNN) to obtain the crisp output. Moreover, the HHO is also hybrid with FRNN to obtain optimal weight that shows better accuracy and less error.

3 Proposed Methodology for Sentiment Analysis from Spam SMS

Short Message Service (SMS) is recognized as one of the most common forms of communication, enabling users of cell phones to send a fast text message at a low cost. Its increasing prevalence and cell phone dependency have increased the assaults triggered by forwarding unsolicited messages such as SMS spam. In this work, we analyze the KELM for Spam detection. Before spam detection, the input SMS needs to get pre-processed using stemming, stop word removal, Tokenization, PoS (Part of speech) tagging, and micro-blogging. Then, the features from the pre-processed data are extracted using Latent semantic analysis (LSA), Independent component analysis (ICA), and lexicon-based features. Next, the dimension of extracted features is reduced using a chi-square, Point-wise mutual information (PMI), and Distinguishing feature selector (DFS) approaches. Then, these extracted feature vectors are used as input for training and testing the spam detection algorithm. The process flow of the proposed sentiment analysis process is shown in Fig. 1.

Moreover, detecting the sentiments from the classified SMS is necessary, which evaluates the emotions from the classified spam and ham SMS. The detected spam and ham SMS sentiments are identified using an optimization-based deep learning approach. The recurrent fuzzy neural network (FRNN) is used for this sentiment analysis. To maximize the classification accuracy, a metaheuristic optimization algorithm (i.e. Harris Hawk Optimization) is hybrid along with the proposed neural network. The weight parameter is optimized to the optimal or near-optimal solution using the HHO algorithm. This merit makes it attain the fastest convergence rate. The training procedure carried out in FRNN intends to decide the connection weights among several neurons for error reduction. The dataset does not contain any sentiment labels, so to label that spam/ham dataset, a Textblog based sentiment analysis approach is used.

3.1 Pre-processing

Pre-processing is a key component of text classification in the framework. Applying pre-processing to several datasets improves the general quality of text classification. It is an important step in ML that removes the datasets' noise. The pre-processing involves some steps to classify the input Short Message Service (SMS) using several methods such as stemming, stop word removal, Tokenization, Latent semantic analysis (LSA), PoS (Part of speech) tagging, Independent component analysis (ICA), micro-blogging features and lexicon-based features. The methods of pre-processing were detailed as follows,

3.1.1 Stemming

The stemming process is carried out to establish the variant form of the input SMS words into the common representation method. The aim of stemming is to reduce an unwanted character from the text, remove the longest suffix from the words and turn the stem into a valid form of words Albalawi et al. [3]. For example, the words: "determination",

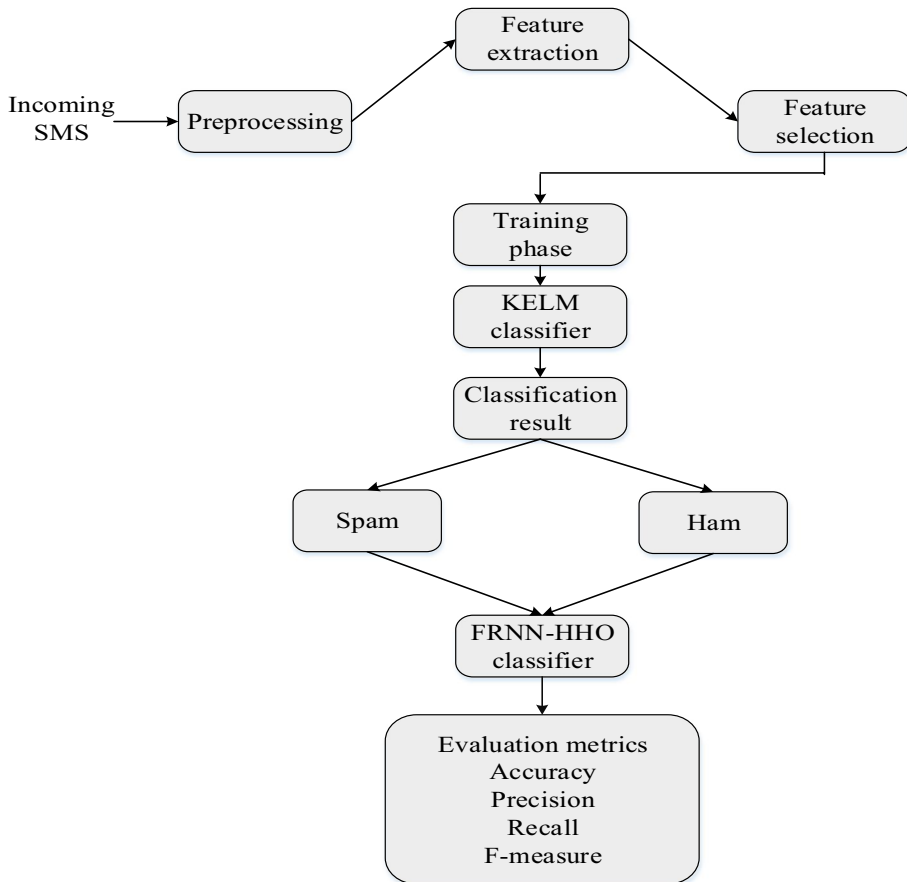


Fig. 1 Flowchart for proposed sentiment analysis

“determined”, and “determining” are reduced in the common representation method of “determine”. Stemming is mainly subject to two errors, namely;

- Over stemming—the formation of two different stem words was stemmed in the same root is known as false positive.
- Under stemming—the different stem, words should form in the same stem roots; if it doesn't form, it is called a false negative.

3.1.2 Stop word removal

Stop words are used to remove the common and frequent words in SMS that were essentially meaningless, and they connect those words to form sentences. Due to the highly frequent, a group of words without adding additional information like determine, preposition, and articles are called stop words. The most frequently used stop words, such as ‘and’, ‘are’, ‘this’, etc., are not much used in documents, so words were removed from the

particular sentence Guo et al. [15]. Further words are not necessary for text mining applications. The stop word can easily scarify without changing the meaning of the sentence.

For example: ['There' 'is' 'a' 'tree' 'near' 'the' 'river'] without stop word, the sentence form as ['There tree near river'].

3.1.3 Tokenization

Tokenization was defined as splitting the paragraph into a sentence or sentence into words in meaningful parts. The tokenization was mainly considered by alphabetic or alphanumeric characters, which delimit the non-alphanumeric characters such as punctuation, white space, etc. The challenges of tokenization are classified into three methods: isolating- the words not separated into small units, agglutinative- the word must separate into small units, and inflectional- consisting of Latin words [16]. Tokenization also provided input cleaning steps: punctuation removal, character removal from numbers, etc.

For example, ["God is great! I won a lottery."] – [God is great I won a lottery] Therefore, the sentences were redefined into tokenization format.

3.1.4 Part of speech (PoS) tagging

PoS tagging determines the grammatical structure such as noun, verb, adjective, etc. It is a fundamental step of pre-processing to edit the text. The PoS is used to translate and normalize the input documents. The different modulation of several parts of speech (PoS) is tagged in the same root. For example: ["The big green fire truck"] it is difficult to mention whether the "fire" is a noun or adjective [16]. Some special features were laid from the pre-processed data to analyze the input documents.

3.1.5 Microblogging features

It is a technique that extracts various microblogging features they are emoticons, URLs, question marks, punctuation marks, all caps, hashtags, elongated words, user names, etc. Wan et al. [45].

- Hashtags: the number of hashtags.
- All-caps: Total words having all characters in uppercase.
- Punctuation:
 - Total contiguous sequences of Question marks, exclamation marks, and both question and exclamation marks;
 - Check whether a question or exclamation mark is included in the last token;
- Emoticons: Emoticon polarities are identified from the regular expression available in the tokenizing script prepared by Christopher Pott:
 - Identify the presence and absence of negative and positive emoticons from any position from the available tweet.
 - Then, it analyzes whether the emoticon of a final token is positive or negative;
- Elongated words: One of the characters from the particular word may get repeated sometimes, for example, 'Noooo'.

3.2 Feature Extraction

Feature extraction is a process that converts a set of input data into its corresponding features. It is considered an essential step in text processing as it directly affects distinguishing different clusters or classes. Moreover, identifying effective features from unstructured data is considered a difficult task. Two categories of feature extraction techniques are introduced in this framework, one for text classification and the next for sentiment classification. The details about the two categories of feature extraction are discussed below:

3.2.1 Latent semantic analysis (LSA)

LSA is defined as the characteristic of algebraic-statistical methods, removing the hidden structure from the words or sentences. The LSA was also known as singular value decomposition (SVD), introduced in Ullah et al. [42]; it identifies the unstructured data hidden in the input document and the relationship between the words or sentences. SVD has been able to reduce noise and increase efficiency. LSA undertakes the words were nearer to the meaning and occur in similar pieces of text. SVD is used to perform the mathematical technique in LSA to optimize the length of the text.

The LSA consists of four main steps, which are described below;

- Term-document matrix – a collection of large text detached words into smaller units of passage or sentence for each application.
- Transformed term-document matrix – as a replacement of operational with the rare term frequency, the accesses of the term-document matrix were often changed. Hence it obtains the frequency in a sub-linear fashion $\log(\text{freq}_{ij} + 1)$.
- Dimension reduction – it reduces the rank of the matrix by using SVD. We assume that ‘L’ is taken as the largest singular value, and the remainder is set to zero. The SVD technique is also closely interrelated with features such as Eigen analysis, factor analysis, principal components analysis, and linear neural networks.
- Retrieval in reduced space is used to reduce the space in the term-document matrix. For example: “document-document, term-term, document-term” these words are represented in the same space, so they need to compute.

The mathematical view of LSA is detailed as follows;

The matrix of SVD was represented as ‘X’.

$$X = T^* S^* D^T \quad (1)$$

where T^* and S^* are represented as an orthonormal matrix, D^T was represented as a diagonal matrix. The problematic representation of ‘X’ was using an orthogonal dimension. The SVD uses the largest ‘L’ singular value to optimize the dimensions used in LSA.

$$X = T_L^* S_L^* D_L^* \quad (2)$$

where T_L - term vectors in LSA, D_L - Document vectors in LSA.

3.2.2 Independent component analysis (ICA)

ICA aims to identify the missing text and useful information from the input document. The ICA is used to separate the neural method words were varied in an unidentified way

[39]. The ICA also includes linear transformation methods such as principal component analysis, factor analysis, and projection pursuit. The problematic description of ICA was detected in two forms which are defined in Eq. (3),

$$\begin{aligned}x_1(t) &= a_{11}s_1 + a_{12}s_2 \\x_2(t) &= a_{21}s_1 + a_{22}s_2\end{aligned}\quad (3)$$

where a_{11}, a_{12}, a_{21} , and a_{22} are some of the parameters to find the distance of the signal. $x_1(t)$ and $x_2(t)$ are used to estimate the duplicate text, $s_1(t)$ and $s_2(t)$ estimate original text. The matrix of ICA is denoted as ‘A’. Here the lower case of bold letters indicates vector, and upper case bold letters indicate matrix.

$$\mathbf{x} = \mathbf{A}s \quad (4)$$

‘A’ is the column of a matrix. Sometimes we have to denote the matrix as a_j , and the model is rewritten as;

$$x = \sum_{i=1}^n a_j s_j \quad (5)$$

The ICA is a generative model in which the data were generated with the mixing components s_j .

3.2.3 Lexicon-based feature extraction

In this section, the information about the domain-specific emotion lexicon (DSEL) that is used for extracting the range of features suitable for emotion classification is discussed by Bandhakavi et al. [7]. The feature vectors obtained using the lexicon knowledge are mostly of length $|E|$, where $|E|$ indicates the total number of emotion classes present in the available dataset. The following features are considered to represent the emotions in the document.

3.2.4 Total Emotion Count (TEC)

It is a feature extraction that captures the emotion-based words from the document. The feature vector corresponding to the word in the given document d is represented as d_{TEC} . For emotion j , the feature value is computed using Eq. (6),

$$d_{TEC}[e_j] = \sum_{w \in d} I((e_j = \arg \max_k Lex(w, k)) \times count(w, d)) \quad (6)$$

$I(\cdot)$, represents the indicator function set as either 0 or 1 (for true or false arguments). The total number of times the word w occurred in a document d is represented as $count(w, d)$. TEC captures the features suggested by the lexicon alone (i.e. the features showing the highest value in the lexicon). However, the emotions shared by each word in the document are not similar. For example, the word “beautiful” is subjected to love and joy, whereas the TEC provides a count of 1 for one class and 0 for another.

3.2.5 Total Emotion Intensity (TEI)

The sum of the scores from the emotional intensity of each word in the document is obtained by TEI. Generally, TEC uses coarse integer counts, but TEI utilizes emotion intensity scores obtained from DSEL to extract the emotion-based features from the document having several emotion classes. The feature vector extracted by TEI from the document d is represented as d_{TEI} . For emotion $j(e_j)$, the feature value is computed using Eq. (7),

$$d_{TEI}[e_j] = \sum_{w \in d} Lex(w, e_j) \times count(w, d) \quad (7)$$

The total number of times the word w occurred in a document d is represented as $count(w, d)$.

3.2.6 Max Emotion Intensity (MEI)

Instead of identifying the average score, identifying the term that shows the highest sentiment bearing to the sentiment class from the entire document is considered the major MEI concept. Therefore, the intensity score of the word having the highest emotion-bearing from the given document is estimated with MEI. The feature vector extracted by MEI from the document d is represented as d_{MEI} . For emotion j , the feature value is computed using Eq. (8).

$$d_{MEI}[e_j] = \arg \max_{w \in d} Lex(w, j) \quad (8)$$

3.2.7 Graded Emotion Count (GEC)

The concept of high-intensity emotion words is extended to extract the document representation. For such extraction, the TEI and TEC variants are developed. Both these variants considered all the words in the document without concern about their emotional intensity. However, understanding the effect of high-intensity words on emotion classification is valuable to achieving better performance. The processing principle of both GEC and TEC are similar, except that it extracts the total number of words and their respective emotions within a threshold value δ from the document. It quantifies the relation between the emotion and its respective classes in the probability distribution form. The resultant intensity scores lie between intervals 0 and 1. In this work, the available interval is divided into 4 different quartiles they are [0, 0.25), [0.25, 0.5), [0.5, 0.75) and [0.75, 1]. From these four intervals, three values are selected as threshold values they are 0.25, 0.5, and 0.75. The features that are extracted using GEC are mostly for these three intervals. The feature vector extracted by GEC from the document d is represented as d_{GEC} . For emotion j , the feature value is computed using Eq. (9).

$$d_{GEC}[e_j] = \sum_{\substack{w \in d \\ Lex(w, j) \geq \delta}} I(e_j = \arg \max_k Lex(w, k)) \times count(w, d) \quad (9)$$

The total number of times the word w occurred in a document d is represented as $\text{count}(w, d)$. The lexicon obtained for the word w from the input k is represented as $\text{Lex}(w, k)$.

3.2.8 Graded Emotion Intensity (GEI)

Similar to GEC, we develop a variant of TEI, named GEI, which is estimated by integrating the word intensity scores found within the fixed threshold δ from the given document d . The thresholds mentioned earlier are used for extracting GEI features using DSELS. Given a document d , and its corresponding feature vector, d_{GEI} , the feature value for the j^{th} emotion is computed as follows:

$$d_{\text{GEI}}[e_j] = \sum_{\substack{w \in d \\ \text{Lex}(w, j) \geq \delta}} \text{Lex}(w, e_j) \times \text{count}(w, d) \quad (10)$$

3.3 Feature Selection

Feature selection is a process that intends to reduce the number of input variables by selecting only the useful variables for the classification process. The main merit of this feature selection is that it reduces the complexity of the learning algorithm, further it increases processing speed and accuracy. In this work, we have used three different and efficient feature selection approaches PMI, chi-square, and DFS.

3.3.1 Chi-square

Chi-square is a filter-based feature selection approach that determines whether the feature is class-dependent or independent [11]. Large values obtained by chi-square represent that both the class and feature are found independent.

$$\chi^2 = \sum \frac{(F_o - F_e)^2}{F_e} \quad (11)$$

where, the expected and observed frequency for each class and feature is represented as F_e and F_o respectively.

3.3.2 Distinguishing feature selection

DFS is an ideal filter-based feature section approach which assigns a high score for distinctive features and a low score for irrelevant features Rehman et al. [28]. It ranks the terms based on four different conditions they are:

- If a term is present in one class and not found in any other classes, then consider the such term as distinctive and assign a high score.
- If a term is rarely present in one class and not found in other classes, consider that term irrelevant and assign a low score.
- If a term is frequently found in all classes, then consider such a term irrelevant and assign a low score.

- If a term occurs in a few classes, then consider such term as relatively distinctive and assign a relatively high score.

The formula applied by DFS for feature selection is given in Eq. (12),

$$DFS(t) = \sum_{a=1}^N \frac{P(C_a|t)}{P(\bar{t}|C_a) + P(t|\bar{C}_a) + 1} \quad (12)$$

where, the conditional probability obtained for a term t that is found in classes other than C_a is represented as $P(t|\bar{C}_a)$.

3.3.3 Point-wise mutual information (PMI)

The relationship between the two features is determined by the PMI approach. A high PMI value indicates the frequent co-occurrence of two features [4]. A feature selection process is extensively applied to identify the mutual information shared among the terms and particular classes. PMI determines the ratio between the estimated co-occurrence for term t_j and class C_a which is defined in Eq. (13),

$$PMI(C_a, t_j) = \log \frac{F(t_j)p(C_a, t_j)}{F(t_j)p(C_a)} = \log \left(\frac{p(C_a, t_j)}{p(C_a)} \right) \quad (13)$$

The features selected by three different filter-based feature selection techniques are then provided to KELM for spam and ham classification. The selected features may contain both sentiment and spam/ham-based features. Sometimes both the spam and ham SMS may carry sentiment words therefore, usage of sentiment-based features along with spam/ham features is essential for the KELM classifier to accurately classify the spam and ham messages.

3.4 Kernel Extreme Learning machine (KELM) based spam classification

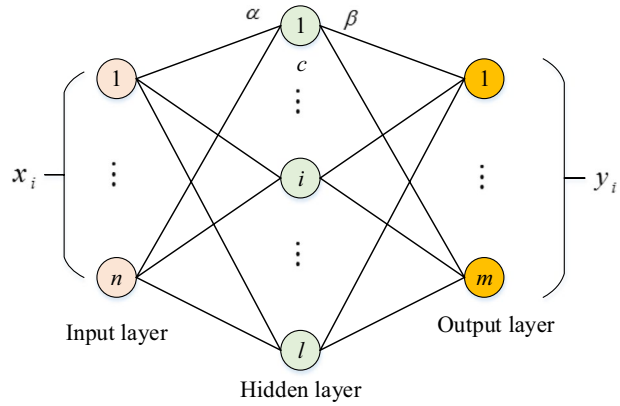
ELM is a type of SLFNN (Single hidden layer feed-forward neural network) whose architecture is depicted in Fig. 2. Three different layers are present in ELM: input, hidden, and output. The usage of the non-linear activation function makes the hidden layer non-linear, whereas the output is linear as it does not comprise any activation function. In a few cases, the ELM fails to attain better results due to the random selection of bias and weight parameters between the input and hidden layers. To overcome such limitation, the ELM approach introduces the kernel function, which eliminates the weight initialization procedure in the input and hidden layers by including the kernel matrix Li et al. [22]. KELM shows some merits like, while using KELM, the determination of hidden layer size is unnecessary. Moreover, it effectively explores the non-linear features. But, the selection of the best kernel function is considered the major contribution of KELM during classification.

Let x represents the total training samples, the output from the neural network is represented as $f(x)$. The SLFNN having l hidden nodes is represented using the Eq. (14),

$$f_{ELM}(x) = W^T \cdot K(w, b, x) \quad (14)$$

where, the activation function of the hidden layer is indicated as $K(w, b, x)$, b indicates the bias weight of the hidden layer, weight obtained between the hidden and output layer is

Fig. 2 Model architecture for ELM



denoted as $W = [\beta_1, \beta_2, \dots, \beta_m]$ and the input weight connecting the input and hidden layer is denoted as w . The output from the hidden layer is defined in Eq. (15),

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_n) \end{bmatrix} \tag{15}$$

The ELM model is represented in Eq. (16),

$$f_{ELM}(x) = h(x)H^T \left(\frac{I}{\rho} + HH^T \right)^{-1} T \tag{16}$$

ELM is extended as KELM using the kernel function, Let

$$\Omega = HH^T \tag{17}$$

The output of the KELM model is represented as,

$$f_{KELM}(x) = h(x)H^T \left(\frac{I}{\rho} + \Omega \right)^{-1} T \tag{18}$$

where,

$$h(x)H^T = \begin{bmatrix} k(x, x_1) \\ \vdots \\ (x, x_n) \end{bmatrix} \tag{19}$$

where, $\frac{1}{\rho}$ represents the regularization parameter and T represents the target output. KELM is better than ELM as it produces less computational time and does not contain any random feature mappings. Further, using the kernel in ELM eliminates the need to set the desired number of hidden nodes.

Next, the sentiments in spam and ham need to be evaluated to identify each SMS’s sentimental intention. Therefore, the optimization hybrid FRNN is used in this work to identify such sentiments from classified SMS. It takes the selected features as input and identifies the sentiments (positive, negative, and neutral) from both the spam and ham SMS.

3.5 Harris hawk optimization (HHO) based Recurrent Fuzzy Neural Network (FRNN) for sentiment analysis

3.5.1 Fuzzy Recurrent neural network (FRNN):

FNN is extensively used in various fields, among which FRNN [10] is identified as the notable one. FRNN contains four layers:the input layer, the membership layer, the fuzzy layer, and the output layer. The input and output provided to the node i of the layer k is represented as $o_i^{(k)}$ and $u_i^{(k)}$. The procedure for FRNN is discussed below:

3.5.2 Layer 1

Initial layer is the input layer containing N nodes,with a parameter resembling each layer. The features selected using the PMI, DFS, and chi-square are given as input in this first layer.

$$O_i^{(1)} = u_i^{(1)} = x_i(t), \text{ where } i = 1 \div N \tag{20}$$

3.5.3 Layer 2

The second layer is the membership layer. In this layer, the gauss function is used as a membership function which changes the data in the nodes of the second layer in crisp form. The neural node in this second layer is denoted as $N \times M$, where M represents the fuzzy rules. Three parameters, θ_{ij} , σ_{ij} and m_{ij} are included in every node.

$$O_{ij}^{(2)} = \exp \left[-\frac{(u_{ij}^{(2)} - m_{ij})^2}{(\sigma_{ij})} \right], \text{ where } i = 1 \div N \text{ and } j = 1 \div M \tag{21}$$

where, the variance and center corresponding to the Gauss distribution function is represented as σ_{ij} and m_{ij} .

$$u_{ij}^{(2)}(t) = O_i^{(1)} + \theta_{ij} O_{ij}^{(2)}(t - 1), \text{ where } i = 1 \div N, j = 1 \div M \tag{22}$$

where, the weight corresponding to the recurrent nodes is represented as O_{ij} . In this second layer, the factor $O_{ij}^{(2)}(t - 1)$ is included at each input node. The previous learning procedure estimates the residual data illustrated by the introduced factor at this second layer. Then, based on Eq. (21), replace $u_{ij}^{(2)}$ from Eq. (22) to obtain Eq. (23).

$$O_{ij}^{(2)} = \exp \left[-\frac{[x_i(t) + \theta_{ij} O_{ij}^{(2)}(t - 1) - m_{ij}]^2}{(\sigma_{ij})^2} \right] \tag{23}$$

3.5.4 Layer 3

In the third layer, fuzzy rules are used. Layers 3 and 4 are concatenated to conclude the available fuzzy rules. Each node in this third layer is resembled with AND operation. The expression for each AND operation is given in Eqs. (24 & 25).

$$O_j^{(2)} = \prod_{i=1}^N O_{ij}^{(2)} \quad (24)$$

$$O_j^{(2)} = \prod_{i=1}^N \exp \left[-\frac{[x_i(t) + \theta_{ij} O_{ij}^2(t-1) - m_{ij}]^2}{(\sigma_{ij})^2} \right] \text{ where, } j = 1 \div M \quad (25)$$

3.5.5 Layer 4

The output layer is the fourth layer which contains P nodes. Initially, the P is set as 1, which is considered the river run-off value. This layer is also liable to convert the fuzzy data into a crisp form.

$$y_k = O_k^{(4)} = \sum_{j=1}^M u_{jk}^{(4)} w_{jk} = \sum_{j=1}^M O_{jk}^{(3)} w_{jk} \quad (26)$$

$$y_k = \sum_{j=1}^M w_{jk} \prod_{i=1}^N \exp \left[-\frac{[x_i(t) + \theta_{ij} O_{ij}^2(t-1) - m_{ij}]^2}{(\sigma_{ij})^2} \right] \text{ where, } k = 1 \div P \quad (27)$$

After initializing the FRNN architecture, the HHO algorithm is integrated with FRNN to accomplish the training process. While compared with other algorithms, the behavior shown by HHO is found inspiring in this classification process.

3.5.6 Harris Hawk optimization (HHO)

In this approach, we hybrid the HHO algorithm Moayedi et al. [23] with FRNN to identify the optimal weight parameter. The hunting process of HHO comprises four activities: tracking, encircling, approaching, and attacking. Normally, the entire hunting process is bagged up in three main phases: exploration, conversion from exploration to exploitation, and exploitation. The flowchart for the HHO algorithm is shown in Fig. 3.

After completing the searching phase, initiate the first stage by discovering the prey (rabbit) position. Then, the hawks define their position based on the randomly generated solution, X_{rand} .

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(L_b + r_4(U_b - L_b)) & q < 0.5 \end{cases} \quad (28)$$

where, X_m indicates the average position and the random number that varies from 0 to 1 is represented as, r . The formula used to estimate X_m is shown in Eq. (29).

$$X_m(t) = \frac{1}{n} \sum_{i=1}^N X_i(t) \quad (29)$$

where, the position and size of the hawk are represented as X_i and N respectively. Next, the escaping energy (E) during hunting is evaluated using the Eq. (30),

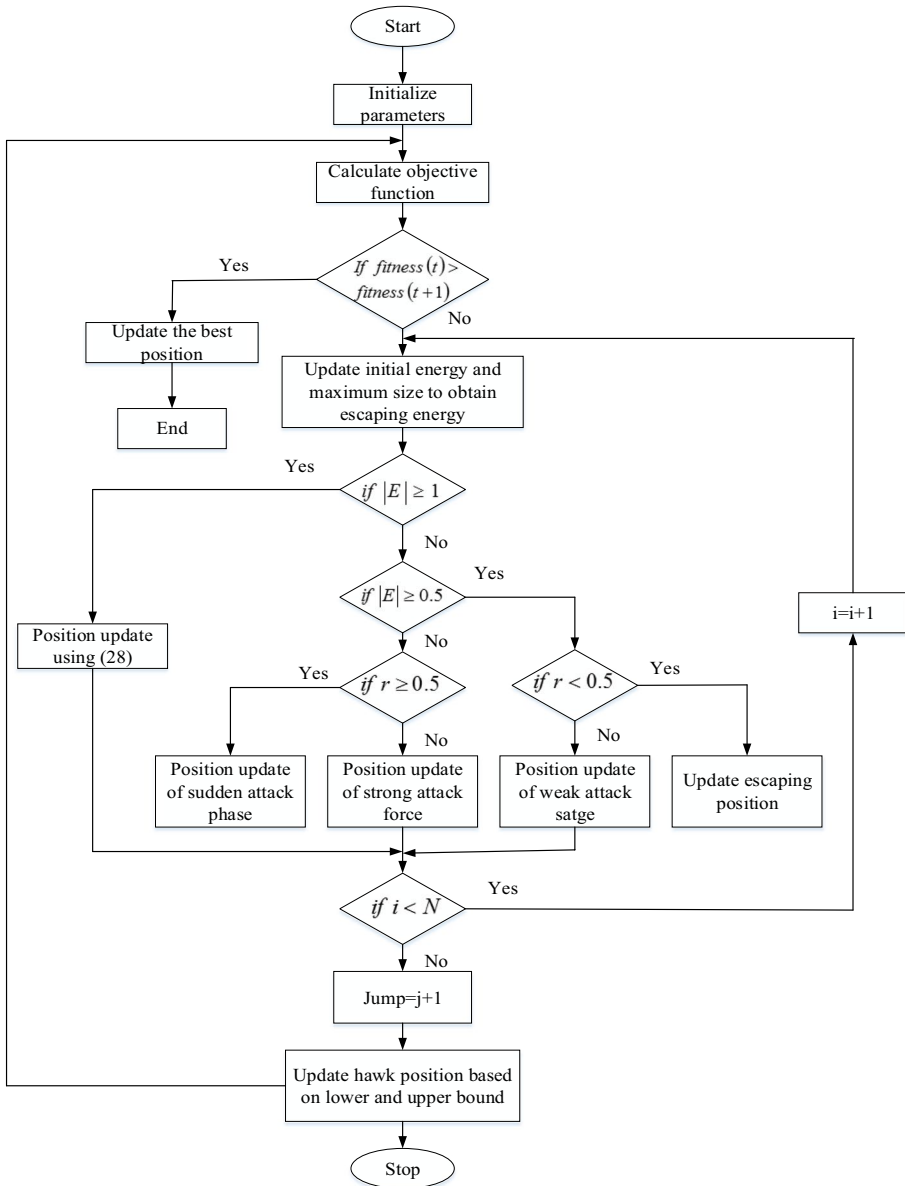


Fig. 3 Flowchart for HHO algorithm

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \tag{30}$$

The flowchart for HHO is depicted in Fig. 3. The HHO process begins with parameter initialization, and then the objective function (i.e. parameter weighting with minimum error) for each parameter is evaluated. If the identified parameter satisfies the defined fitness (minimum error) means, activate the updating process, otherwise search

for the best weight parameter. The process continues till reaching the stopping criteria. Where the maximum size interms of repetition is indicated as, T , and E_0 represents the initial energy which ranges from -1 to 1. The decision to determine whether to initiate the exploration or exploitation phase is identified by evaluating the parameter, $|E|$. During the exploitation phase, the selection of besiege using $|E|$ is also essential, i.e. if $|E|$ is < 0.5 , then take the hard besiege, and if $|E|$ is greater than or equal to 0.5, then we can go for soft besiege. The weight parameter that provides less error during classification is considered the best solution, which is optimally determined by the HHO algorithm. Further, it reduces the computational complexity of FRNN during sentiment classification. The algorithm of the proposed architecture is shown in below algorithm 1.

Input: SMS data
Output: Sentiments from classified SMS
Initialize, input SMS data
Step 1: Pre-process the input data using
 1a) text-based pre-processing techniques such as stemming, stop word removal, tokenization, and POS tagging
 1b) sentiment-based pre-processing using Micro-blogging features.
Step 2: Extract text-based and sentiment-based features using
 2a) LSA and ICA – for text feature extraction
 2b) Lexicon based features – for sentiment based feature extraction
Step 3: Most prominent features that provide high accuracy during text and sentiment classification are selected using,
 3a) Chi-square, DFS, and PMI
Step 4: Selected features are subjected to spam and ham classification using KELM
Step 5: Sentiment analysis for spam and ham SMS is performed using FRNN.
Step 6: The weight parameter selection of FRNN classifier using HHO.
 6a) Initialize solutions randomly.
 6b) Estimate X_m using equation (29)
 6c) Next, evaluate the escaping energy (E) using the equation (30)
Step 7: Performance evaluation to test the effectiveness of the proposed architecture.

4 Result and Discussion

The proposed sentiment analysis from classified SMS messages is implemented in the python platform. In this section, the results that are obtained for the proposed sentiment analysis process are discussed. The experimental analysis is carried out in two different phases they are classification-based results and sentiment-based results. The different performance metrics like accuracy (A), precision (P), recall (R), f-measure (F), RMSE, and MAE are determined to estimate the effectiveness of the proposed approach. The parameter used for tuning the process is shown in Table 1.

4.1 Dataset Description

To test its effectiveness, three different datasets are used in this approach: SMS dataset [9], Email dataset, and spam-assassin [43] dataset. The details about total messages, spam messages, and ham messages are described in Table 2. The SMS dataset is obtained from the following link: <https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>. From the following link, the Email dataset is obtained, <https://www.kaggle.com/datasets/venky73/spam-mails-dataset>. The spam assassin dataset is obtained from <https://www.kaggle.com/datasets/beatoa/spamassassin-public-corpus>.

Table 1 Parameters and their corresponding value

KELM		
Kernel parameter		$\sigma = 2\sigma_0$
Penalty parameter		C = 1000
Activation function		Sigmoid
Hidden neurons		500
Epochs		10
FRNN		
Hidden dimension		100
Learning rate		0.0001
Epoch		25
Activation function		Sigmoid
Output dimension		1
Length of sequence		50
Minimum clip value		-10
Maximum clip value		10
HHO		
Maximum iteration		100
Lower bound		100
Upper bound		100

Table 2 Dataset description

Dataset		Total messages
SMS dataset	Total messages	5574
	Spam	747
	Ham	4827
Email dataset	Total messages	5172
	Spam	1500
	Ham	3672
Spam assassin	Total messages	3252
	Spam	501
	Ham	2751

The word cloud for both the spam and ham messages is shown in Fig. 4. An openly available SMS collection is collected to develop the KELM classifier for spam and ham classification. The SMS messages in the dataset are labeled as ham or spam. The real messages are labeled as ham, whereas the junk messages are labeled as spam. A sample word cloud for both ham and spam messages is shown in Fig. 4.

4.2 Evaluation metrics

4.2.1 Accuracy (A)

The ratio between the total number of correctly predicted SMS and the total number of predicted messages is determined by accuracy.

$$A = \frac{(TP + TN)}{(TP + FPTN + FN)} \tag{31}$$

4.2.2 Precision (P)

It defines the proportion of positive predictions that are truly positive. The formula used to evaluate the precision is shown in Eq. (32).

$$P = \frac{TP}{TP + FP} \tag{32}$$

4.2.3 Recall (R)

It defines the proportion of positives that are truly classified as positive. The formula used to evaluate the recall is shown in Eq. (33).

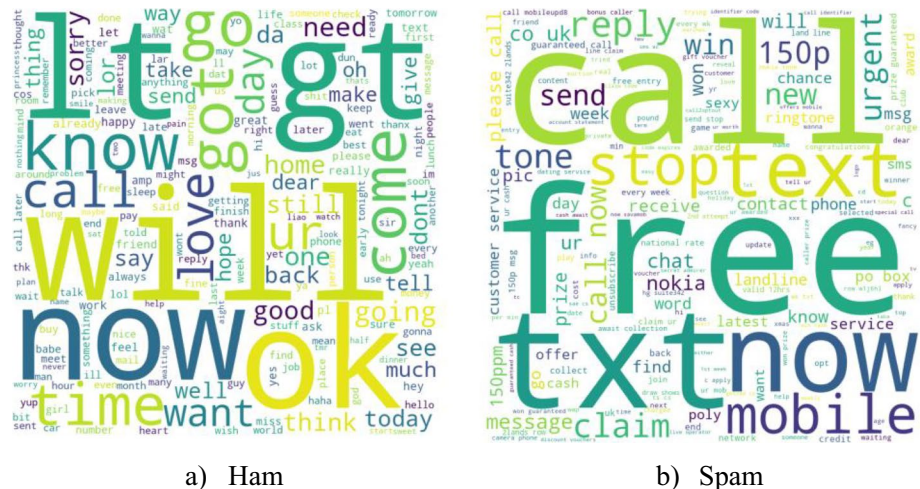


Fig. 4 Word cloud [(a) spam, (b) ham messages]

$$R = \frac{TP}{TP + FN} \quad (33)$$

4.2.4 F1-score (F)

The harmonic from the precision and recall metrics are determined by F1-score. The formula used to evaluate the F1-score is shown in Eq. (34),

$$F = 2 * \frac{P * R}{P + R} \quad (34)$$

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{x}_i - x_i)^2}{n}} \quad (35)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i| \quad (36)$$

where, \hat{x}_i represents the predicted value, n represents the total observations, and x_i represents the actual value.

4.2.5 ROC curve

The plot between TPR (true positive rate) and FPR (False positive rate) is illustrated by the ROC curve [23].

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned} \quad (37)$$

4.3 Performance analysis for spam and ham classification

The classification performance of the proposed KELM is discussed in subsequent paragraphs. The performance metrics like accuracy, precision, F-measure, RMSE, recall, and MAE are evaluated for three different datasets. This evaluation is carried out to show the efficiency of the proposed approach. The three different datasetstaken for evaluation are the SMS, Email, and Spam assassin.

The accuracy attained for a different file and feature sizesare shown in Fig. 5. The accuracy of the proposed KELM classification increases with an increase in file size/feature size. This is mainly due to the efficient performance of the proposed feature selection and KELM classifier techniques. The features selected by the proposed feature selection techniques show promising results in classification. The total size of the file used in our work is 3343, and the total size of the features is 3000.

The cross-validation result attained for the different numbers of hidden neurons is shown in Fig. 6. In this work, used 500 neurons are used, and the cross-validation accuracy for 100, 200, 300, 400, and 500 neurons is evaluated. The fivefold and tenfold cross-validation

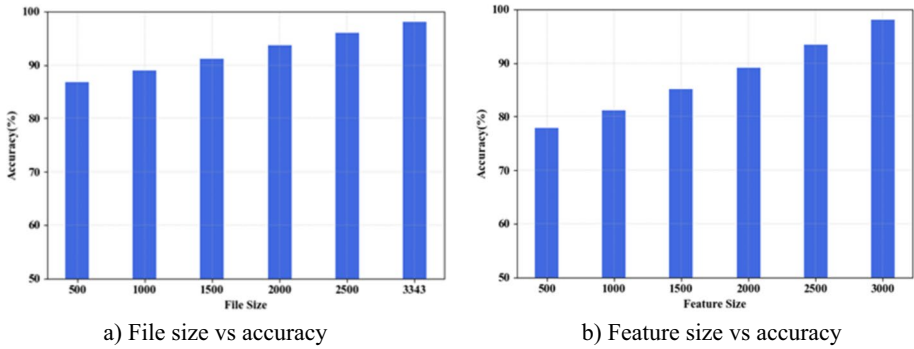


Fig. 5 Accuracy comparison for KELM classifier with (a) different file size and (b) different feature size

Fig. 6 Hidden neuron vs accuracy for 5-fold and 10-fold cross-validation

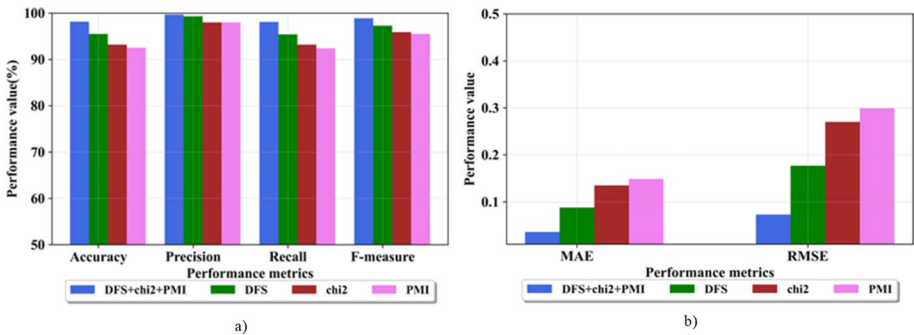
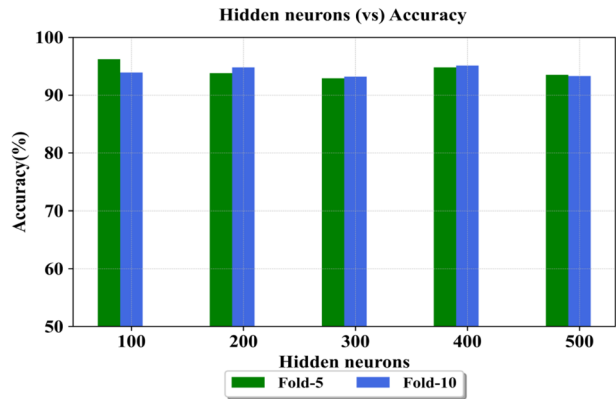


Fig. 7 Performance results of different feature selection techniques [DFS, chi-square, and PMI]

is performed in this method. The accuracy attained at fold-5 for 100 neurons is higher than other neurons. For tenfold, the 400 hidden neurons have attained higher accuracy results.

Table 3 Performance metrics attained with using and without using feature selection

Metrics	Without feature selection	With feature selection			
		DFS	Chi-square	PMI	DFS + Chi + PMI
Accuracy (A) (%)	89.99	95.5	93.22	92.5	98.16
Precision (P) (%)	98.35	99.35	98.84	98.83	99.73
Recall (R) (%)	89.93	95.48	93.257	92.427	98.132
F-measure (F) (%)	93.9	97.381	95.97	95.52	98.92
RMSE	0.4001	0.1776	0.2709	0.2996	0.0735
MAE	0.2	0.088	0.1354	0.1498	0.03678

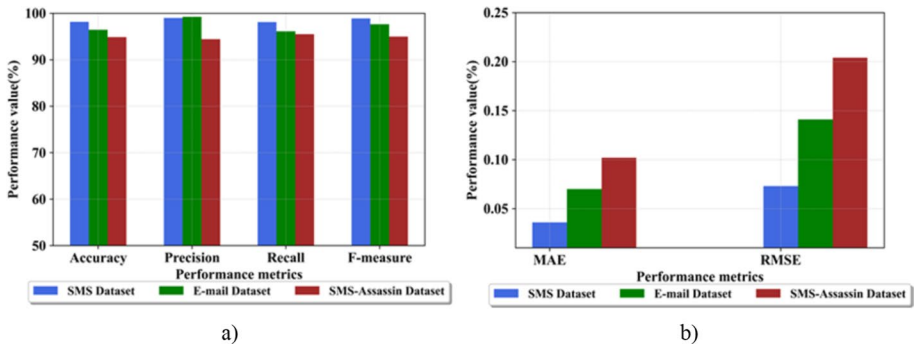


Fig. 8 Overall performance of KELM classifier for different datasets

The accuracy, recall, precision, F-measure, RMSE, and MAE attained by different feature selection techniques are shown in Fig. 7. The results attained by combined features are also illustrated. The features selected by each feature selection technique are found very much valuable for classification purposes. Among these, the accuracy of PMI is found to be less than the other two feature selection techniques.

The effect of the KELM classifier with and without feature selection for spam-based classification is given in Table 3. Different metrics like accuracy, recall, precision, f-measure, RMSE, and MAE are determined for evaluating the performance of proposed feature selection techniques. The presence of a feature selection process highly influences the classification result. With these feature selection techniques, the proposed KELM classifier has high classification accuracy and less error rate. This is because the separate performance shown by the three feature selection techniques is found to be better. Therefore, the combined result provided by the feature selection techniques further maximizes the overall performance of the classification algorithm. Among these three feature selection techniques, the accuracy shown by DFS is higher than the other two techniques, which shows that this algorithm has made a huge difference in classification accuracy.

The performance attained for KELM-based spam/ham classification using three different datasets is shown in Fig. 8. The RMSE for the SMS dataset is much less than the other two datasets. The precision attained by the proposed KELM for the Email dataset is higher than SMS and SMS spam assassin datasets. The proposed approach shows higher

classification accuracy for three different datasets. This is because the error attained by the proposed KELM classifier during spam/ham classification is less.

The accuracy, precision, recall, f-measure, and RMSE of the proposed KELM classifier are compared with three different existing techniques they are CNN-LSTM, SVM, and random forest. The comparison result for different performance metrics is given in Table 4. Among all these techniques, the proposed KELM has attained higher accuracy results for spam/ham classification. This is because the proposed approach has included three different feature selection techniques before classification. The accuracy attained by KELM is 98.61%, however, the precision, recall, RMSE, and f-measure of KELM are 99.73%, 98.13%, 0.073, and 98.92%, respectively. The kernel function usage has improved the classification process's overall performance. Existing CNN-LSTM has shown effective performance in classification. Still, the usage of LSTM has reduced the overall performance of CNN because the LSTM has been subjected to overfitting in large amounts, which automatically reduces the performance of CNN.

4.4 Performance analysis for sentiment classification

The sentiment analysis from classified inbox messages is a demanding topic, as it enhances the overall efficiency of spam identification. Therefore, to motivate this, a hybrid fuzzy based network is introduced in the proposed architecture, which uses fuzzy concepts and obtains a crisp output during sentiment analysis. In this, the hybrid fuzzy and RNN is considered the major advantage because the fuzzy greatly impacts processing text during detection and classification. Moreover, the RNN can analyze a huge amount of text data with less error rate. These two merits are combined in this proposed architecture and have obtained an efficient performance than other existing architectures. The existing methods that are considered for comparison are RNN, DBN, and ANN. These three techniques have shown efficient performance in text classification but haven't shown better performance than the proposed FRNN-HHO architecture.

4.4.1 RNN

RNN has a great impact on handling the machine translation process. Unlike the Feed-forward network (FFN), the RNN can handle large input sequences with variable lengths. The activation of each recurrent hidden unit during each iteration is based on the previous unit. In this, the unrolled network is used, which unrolls a 4-layer network for a 4-word sentence (i.e. each layer corresponds to each word).

Table 4 Comparison of classification-based results

Methods	Metrics				
	Accuracy	Precision	Recall	F-measure	RMSE
KELM (Proposed)	98.61	99.73	98.13	98.92	0.073
CNN-LSTM [40]	98.374	95.39	87.88	91.482	0.093
SVM [24]	97.832	97.77	80	88	0.89
Random forest [32]	93.2	93.2	93.2	93.2	0.983

4.4.2 DBN

DBN takes the extracted features as input. The input's low-level features are extracted using the first few layers. Then, the upper layers are used to obtain the learned feature, and the complex features are obtained from these layers. Finally, the sentiment is predicted at the output layers.

4.4.3 ANN

ANN trains the network by performing back propagation (BP), during which the neuron activation happens on the hidden layer. Forward and backward passes are included in the BP algorithm. Forward pass performs activation value detection, whereas backward pass performs weight and bias adjustment. The weight and bias adjustment are performed by determining the difference between the actual and desired network outputs. The forward and backward pass happens iteratively till the network gets converges.

The classified (spam and ham) messages are then input to FRNN for sentiment analysis. The sentiment analysis performance of the proposed FRNN is improved using the HHO algorithm. HHO is an efficient and rapid optimization process that increases the training performance of FRNN. The results obtained due to this hybrid classification algorithm are discussed below. The optimization-based and classification-based results are briefly discussed in this section. The metrics that are determined to show the effectiveness of the proposed sentiment analysis algorithm are accuracy, RMSE, precision, F-measure, recall, and MAE. The cross-fold validation for the different numbers of neurons is also analyzed in this method. The performance of the proposed sentiment analysis classifier is compared with three different neural networks they are RNN, DBN, and ANN. The comparison result of proposed and existing sentiment analysis techniques are shown in Figs. 9, 10, and 11. Among all these techniques, the proposed FRNN has attained better results than other existing techniques.

The ROC curve for ham and spam messages of three different datasets is shown in Fig. 12. The AUC value attained for ROC is also shown in Figs. 12 (1, 2, and 3). Among these three datasets, the AUC attained by SMS datasets (0.9699) is high then the other two datasets. The ROC attained for ham messages of three datasets is shown in Fig. 12 (1a, 2a, and 3a). This ROC is obtained by estimating the comparison between the FPR

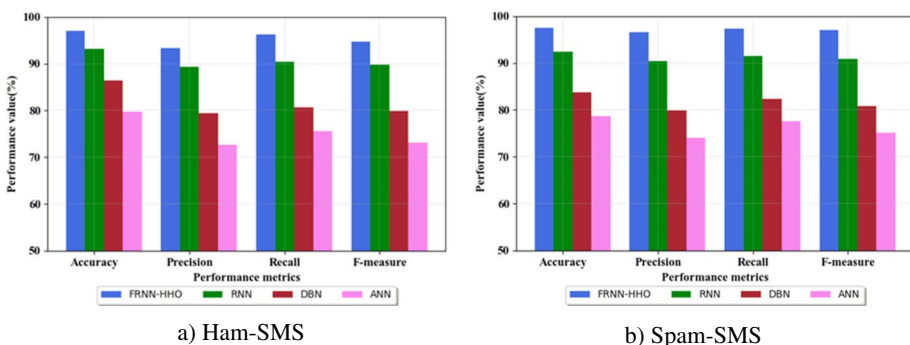


Fig. 9 Sentiment analysis of different classifiers for spam and ham messages from SMS dataset

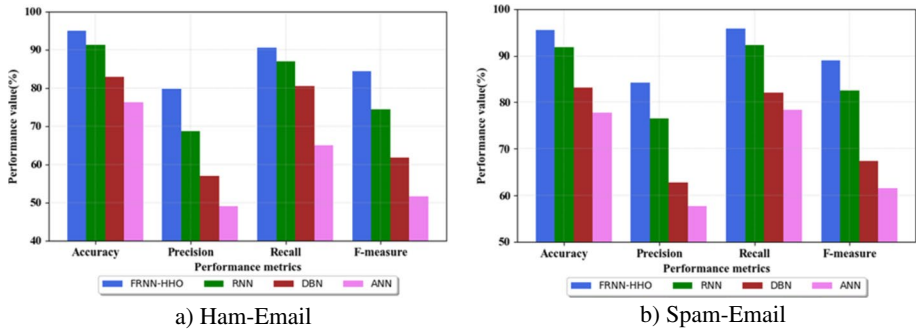


Fig. 10 Sentiment analysis of different classifiers for spam and ham messages from Email dataset

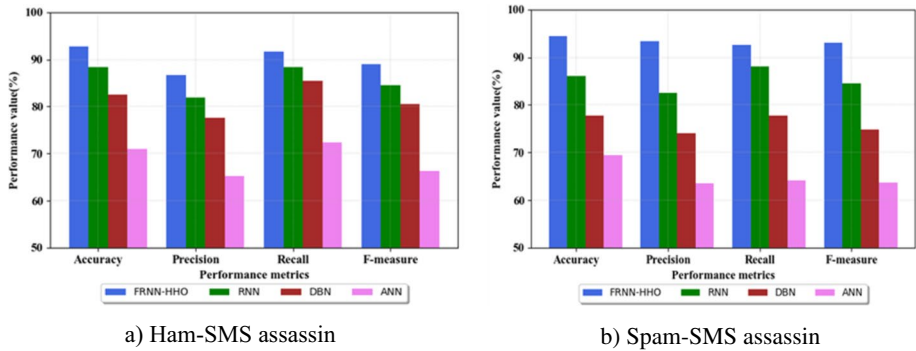


Fig. 11 Sentiment analysis of different classifiers for spam and ham messages from Spam-assassin dataset

and TPR. Moreover, the ROC for existing techniques like RNN, DBN, and ANN for ham and spam of three different datasets is also depicted in Fig. 12.

The optimization-based results for different datasets are shown in Table 5. The proposed HHO optimization algorithm is compared with 3 different and recently evolved optimization algorithms they are SSD (Social ski-driver optimization), SFO (Sunflower optimization), and SSO (Social spider optimization). Among all these approaches, the performance shown by the proposed HHO is found to be better than other optimization algorithms. As it includes only a few steps for identifying the optimal solutions, the optimal weight selection performance of FRNN gets improved.

The comparative analysis between proposed and existing techniques is shown in Table 6. Few works have concentrated on solving both classification and sentiment analysis, whereas few works have concentrated on any one of these two tasks. A comparative analysis is done between all these approaches. Among all techniques, the proposed architecture has achieved

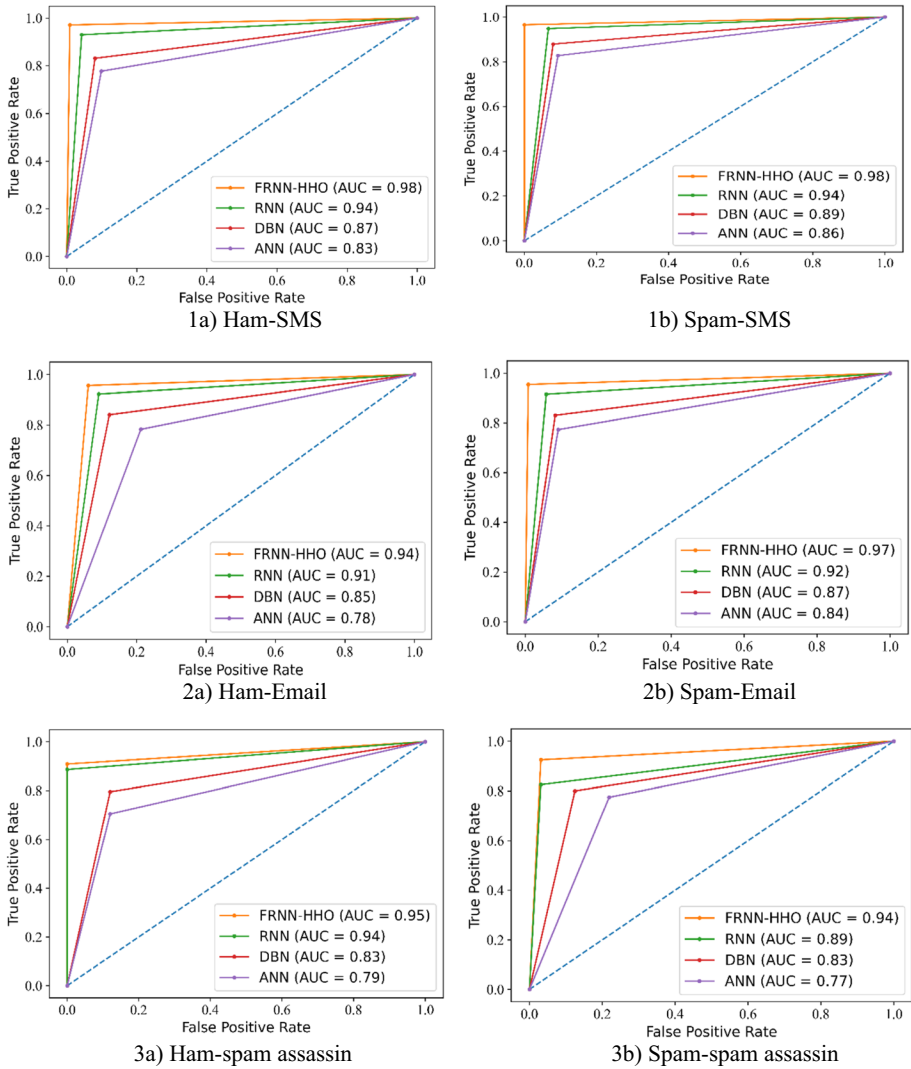


Fig. 12 ROC curve for ham and spam messages [1) SMS dataset, 2) Email dataset, and 3) Spam-assassin dataset]

more efficiency than other existing algorithms. This is because the proposed approach has used fuzzy based classifier for sentiment analysis which has obtained a crisp output by reducing the error and processing time. This hybrid architecture has utilized the advantages of fuzzy and RNN to achieve better accuracy in text processing. The proposed approach has attained less error and better accuracy. Moreover, three different and standard datasets are used to evaluate the proposed performance. The proposed approach has shown less satisfactory time complexity compared to other algorithms.

Table 5 Optimization-based comparison for different datasets

Optimization algorithm	Metrics	Dataset					
		SMS		Email		Spam-assassin	
		Spam	Ham	Spam	Ham	Spam	Ham
HHO	Accuracy	97.49	96.99	95.56	94.99	94.44	92.75
	Precision	96.6	93.3	84.16	79.875	93.38	88.38
	Recall	97.39	96.2	95.86	90.526	95.83	93.63
	F-measure	97	94.7	89.08	84.49	94.51	90.62
	RMSE	0.060	0.0526	0.0841	0.0875	0.1805	0.1594
	MAE	0.036	0.0375	0.0576	0.0625	0.0972	0.1014
SSD	Accuracy	92	90.22	84.78	89.58	83.33	82.6
	Precision	89.5	85.1	64.739	66.34	80.39	77.03
	Recall	91.26	83.51	83.57	85.21	84.72	85.4
	F-measure	90.3	83.97	69.78	71.03	82.12	80
	RMSE	0.1778	0.187	0.3825	0.2541	0.4583	0.4347
	MAE	0.1119	0.127	0.228	0.1541	0.2638	0.2608
SFO	Accuracy	83.39	88.721	79.91	82	73.6	81.15
	Precision	79.089	87.83	59.49	55.54	68.7	74.85
	Recall	82.105	82.36	79.84	79.09	71.38	82
	F-measure	80.183	84.14	64.03	60.12	69.57	77.48
	RMSE	0.3636	0.2932	0.4579	0.3541	0.555	0.449
	MAE	0.2318	0.1729	0.286	0.2375	0.361	0.2753
SSO	Accuracy	80.632	83.39	73.85	73	75.36	68.05
	Precision	75.88	79.08	54.96	47.44	68.9	62
	Recall	79.094	82.105	74.74	62.77	76.1	62.5
	F-measure	76.89	80.18	57.58	49.03	71.29	62.02
	RMSE	0.4071	0.363	0.6203	0.5791	0.637	0.6527
	MAE	0.2648	0.231	0.381	0.37	0.376	0.4305

5 Conclusion

Sentiment classification from the classified messages by identifying the sentiment polarity from the inbox messages is essential. Polarity identification improves the overall classification accuracy of spam/ham messages. Initially, the spam and ham messages are classified using the KELM classifier. To achieve efficient classification, the input data is pre-processed, and the features are required for classification. The sentiments present in the classified messages are then determined using Textblog. Based on the Textblog result and extracted features, the sentiments from the classified messages are determined using the FRNN-HHO classifier. The HHO is a hybrid along with FRNN to improve the overall classification accuracy. The effectiveness of proposed classification and optimization-based techniques are evaluated using different performance metrics: accuracy, recall, precision, RMSE, MAE, and f-measure. Three different datasets are used to evaluate the proposed approach performance: SMS, Email, and spam-assassin datasets. The proposed approach has shown a satisfactory result in classifying the sentiments from the classified SMS. The accuracy attained by KELM is 98.61%, however, the precision, recall, RMSE, and

Table 6 Comparison between proposed and existing techniques

Ref no	Technique	Spam/ham classification	Sentiment analysis	Performance metrics
Proposed	KELM, and FRNN-HHO	Y	Y	Accuracy –98.61% (KELM), AUC—0.9699 (FRNN-HHO)
[29]	LSTM	Y	Y	Accuracy – 98% (Spam classification), accuracy – 73.81% (sentiment classification)
[46]	HMM	Y	N	AUC – 0.920, accuracy – 0.96,
[1]	Bi-LSTM	Y	N	Accuracy – 93.4%, Recall – 91.7%, Precision – 96.9%, and F-measure – 94.23%
[31]	DCF	Y	N	Accuracy – 98.38%, AUC – 0.955
[44]	RCNN	Y	Y	Accuracy – 96.13%, Recall – 96.11%, precision – 96.25%, and f-measure – 96.13%
[34]	k-means, one-vs-rest classifier	Y	Y	Accuracy – 97%

f-measure of KELM are 99.73%, 98.13%, 0.073, and 98.92%, respectively. The AUC of the proposed approach is found to be 0.9699 (SMS dataset), 0.958 (Email dataset), and 0.95 (spam assassin). However, the execution time taken by the proposed approach is not satisfactory; this is because the attacking strategies performed by the HHO algorithm to reach the optimal solution have consumed more time than other optimization techniques to show the promising result. This is considered the major limitation of our proposed approach. Therefore in future, planned to develop the architecture (hybridizing the fuzzy logic with an optimization algorithm) that attains better classification results with less execution time and error.

Acknowledgements The authors would like to thank the National Institute of Technology Raipur, Chhattisgarh, India for providing infrastructure and facilities to carry out this research work.

Data availability The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Declarations

Conflict of interest The authors declare no potential conflict of interest.

References

1. Abayomi-Alli O, Misra S, Abayomi-Alli A (2022) A deep learning method for automatic SMS spam classification: performance of learning algorithms on indigenous dataset. *Concurr Comput Practice Exper* 34(17):e6989. <https://doi.org/10.1002/cpe.6989>
2. Agarwal B, Ramampiaro H, Langseth H, Ruocco M (2018) A deep network model for paraphrase detection in short text messages. *Inf Process Manage* 54(6):922–937
3. Albalawi R, Yeap TH, Benyoucef M (2020) Using topic modeling methods for short-text data: A comparative analysis. *Front Artif Intell* 3:42
4. Alsmadi I, Gan KH (2019) Review of short-text classification. *Int J Web Inf Syst* 15(2):155–182. <https://doi.org/10.1108/IJWIS-12-2017-0083>
5. Arora M, Kansal V (2019) Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. *Soc Netw Anal Min* 9(1):1–14
6. Ay Karakuş B, Talo M, Hallaç İR, Aydin G (2018) Evaluating deep learning models for sentiment classification. *Concurr Comput Practice Exp* 30(21):e4783
7. Bandhakavi A, Wiratunga N, Padmanabhan D, Massie S (2017) Lexicon based feature extraction for emotion text classification. *Pattern Recogn Lett* 93:133–142
8. Barushka A, Hajek P (2020) Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Comput Appl* 32(9):4239–4257
9. Chalothom T, Ellman J (2015) Simple approaches of sentiment analysis via ensemble learning In *information science and applications* (pp. 631–639). Springer, Berlin, Heidelberg
10. Chérif F, Abdelaziz M (2020) Stepanov-like pseudo almost periodic solution of quaternion-valued for fuzzy recurrent neural networks with mixed delays. *Neural Process Lett* 51(3):2211–2243
11. DeepaLakshmi S, Velmurugan T (2016) Empirical study of feature selection methods for high dimensional data. *Indian J Sci Technol* 9(39):1–6
12. Dragoni M, Federici M, Rexha A (2019) An unsupervised aspect extraction strategy for monitoring real-time reviews stream. *Inf Process Manage* 56(3):1103–1118
13. Ghaleb, SA, Mohamad, M, Abdullah, EFHS, Ghanem, WA (2020) December. Spam classification based on supervised learning using grasshopper optimization algorithm and artificial neural network. In *International Conference on Advances in Cyber Security* (pp. 420–434). Springer, Singapore
14. Guelilil I, Azouaou F, Benali F, Hachani AE, Mendoza M (2020) The role of transliteration in the process of arabizi translation/sentiment analysis In *Recent Advances in NLP: The Case of Arabic Language* (pp. 101–128). Springer, Cham
15. Guo S, Zhang X, Yang X, Chen R, Guo C, Li H, Li T (2020) Developer activity motivated bug triaging: via convolutional neural network. *Neural Process Lett* 51(3):2589–2606

16. Hung C, Chen SJ (2016) Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowl-Based Syst* 110:224–232
17. Islam MR, Liu S, Wang X, Xu G (2020) Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Soc Netw Anal Min* 10(1):1–20
18. Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. *Ann Math Artif Intell* 85(1):21–44
19. Khiari W, Roche M, Hafsia AB (2016) Integration of lexical and semantic knowledge for sentiment analysis in sms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp 1185–1189)
20. Lee HY, Kang SS (2018) SMS text messages filtering using word embedding and deep learning techniques. *Smart Med J* 7(4):24–29
21. Li Y, Nie X, Huang R (2018) Web spam classification method based on deep belief networks. *Expert Syst Appl* 96:261–270
22. Li J, Xi B, Du Q, Song R, Li Y, Ren G (2018) Deep kernel extreme-learning machine for the spectral-spatial classification of hyperspectral imagery. *Remote Sens* 10(12):2036
23. Moayed H, Gör M, Lyu Z, Bui DT (2020) Herding Behaviors of grasshopper and Harris hawk for hybridizing the neural network in predicting the soil compression coefficient. *Measurement* 152
24. Nagwani NK, Sharaff A (2017) SMS spam filtering and thread identification using bi-level text classification and clustering techniques. *J Inf Sci* 43(1):75–87
25. Naresh Kumar KE, Uma V (2020) Need for Hybrid Lexicon Based Context Aware Sentiment Analysis for Handling Uncertainty—An Experimental Study In *Emerging Trends in Electrical, Communications, and Information Technologies* (pp. 117–124). Springer, Singapore
26. Pettersson K (2018) Android Messaging Investigator: forensic text analysis tool for SMS, Kik and Hangouts conversations
27. Pong-Inwong C, Songpan W (2019) Sentiment analysis in teaching evaluations using sentiment phrase pattern matching (SPPM) based on association mining. *Int J Mach Learn Cybern* 10(8):2177–2186
28. Rehman A, Javed K, Babri HA (2017) Feature selection based on a normalized difference measure for text classification. *Inf Process Manage* 53(2):473–489. <https://doi.org/10.1016/j.ipm.2016.12.004>
29. Rodrigues AP, Fernandes R, Shetty A, Lakshmana K, Shafi RM (2022) Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques. *Comput Intell Neurosci* 2022:Article ID 5211949, 14 pages. <https://doi.org/10.1155/2022/5211949>
30. Sahoo, D, Liu, C, Hoi, SC (2017) Malicious URL detection using machine learning: A survey. arXiv preprint arXiv:1701.07179
31. Shaaban MA, Hassan YF, Guirguis SK (2022) Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. *Complex Intell Syst* 8:4897–4909. <https://doi.org/10.1007/s40747-022-00741-6>
32. Sharaff A, Nagwani NK (2016) Email thread identification using latent Dirichlet allocation and non-negative matrix factorization based clustering techniques. *J Inf Sci* 42(2):200–212
33. Sharaff, A, Srinivasarao, U (2020) Towards classification of email through selection of informative features. In *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)* (pp. 316–320). IEEE
34. Sharaff A, Kamal C, Porwal S, Bhatia S, Kaur K, Hassan MM (2021) Spam message detection using Danger theory and Krill herd optimization. *Comput Networks* 199(9, November 2021):108453
35. Srinivasan S, Ravi V, Alazab M, Ketha S, Al-Zoubi AM, KottiPadannayil S (2021) Spam emails detection based on distributed word embedding with deep learning In *Machine intelligence and big data analytics for cybersecurity applications* (pp 161–189). Springer, Cham
36. Srinivasarao U, Sharaff A (2021) Sentiment analysis from email pattern using feature selection algorithm. *Expert Syst*: e12867. <https://doi.org/10.1111/exsy.12867>
37. Srinivasarao U, Sharaff A (2021) Email sentiment classification using lexicon-based opinion labeling In *Intelligent computing and communication systems* (pp 211–218). Springer: Singapore. https://doi.org/10.1007/978-981-16-1295-4_22
38. Srinivasarao U, Sharaff A (2022) Email thread sentiment sequence identification using PLSA clustering algorithm. *Expert Syst Appl* 193, 1 May 2022. <https://doi.org/10.1016/j.eswa.2021.116475>
39. Stone JV (2002) Independent component analysis: an introduction. *Trends Cogn Sci* 6(2):59–64
40. Su YJ, Hu WC, Jiang JH, Su RY (2020) A novel LMAEB-CNN model for Chinese microblog sentiment analysis. *J Supercomput* 76:9127–9141. <https://doi.org/10.1007/s11227-020-03198-x>
41. Su YJ, Chen CH, Chen TY, Cheng CC (2020) Chinese microblog sentiment analysis by adding emoticons to attention-based CNN. *J Int Technol* 21(3):821–829
42. Ullah F, Wang J, Farhan M, Jabbar S, Naseer MK, Asif M (2020) LSA based smart assessment methodology for SDN infrastructure in IoT environment. *Int J Parallel Prog* 48(2):162–177

43. Waheeb W, Ghazali R (2017) Content-based SMS classification: statistical analysis for the relationship between number of features and classification performance. *Computación y Sistemas* 21(4):771–785
44. Waheed A, Salam A, Bangash JI, Bangash M (2021) Lexicon and learn-based sentiment analysis for web spam detection. *IEEE-SEM* 9(3):ISSN 2320-9151
45. Wan C, Jiang S, Wang C, Yuan Y, Wang C (2020) A novel sentence embedding based topic detection method for microblogs. *IEEE Access* 8:202980–202992
46. Xia T, Chen X (2021) A weighted feature enhanced Hidden Markov Model for spam SMS filtering. *Neurocomputing* 444:48–58

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.