Multimedia Tools and Applications (2023) 82:31069–31099
https://doi.org/10.1007/s11042-023-14641-5

Machine intelligence based hybrid classifier for spam detection and sentiment analysis of SMS messages

Ulligaddala Srinivasarao[1] · Aakanksha Sharaff[1]

Received: 29 January 2022 / Revised: 10 June 2022 / Accepted: 4 February 2023 /
Published online: 23 February 2023

Abstract

Short Message Service (SMS) on mobile phones has improved because of technological advancements and increases in content-based marketing where smart phones are frequently overburden with spam SMS. Spam messages are not important since they include virus and spyware. Several text classification methods have been suggested to address spam. However, none of these methods can guarantee a full spam-free solution since each filtering and modeling methodology has its own set of strengths and weaknesses. This paper suggests a hybrid classifier based on SMS spam classification and sentiment analysis. The datasets are pre-processed and Word2vec data augmentation is used to extract the features. Then, the features are fed to six various feature selection methods and equilibrium optimization (EO). Optimum components are then fed into a hybrid K-Nearest Neighbors (KNN) and support vector machine (SVM) classifier is to classify SMS messages. Further, to optimize the parameters of the network and to improve the accuracy, the optimization algorithm Rat Swarm Optimization (RSO) is used. Then, AFINN and SentiWordNet are used for sentiment analysis. This framework is evaluated on the three benchmark datasets; when comparing the performance of proposed method on the three dataset, spam assassin dataset achieves better spam detection accuracy of 99.82%.

Keywords Spam detection · Short message service · Sentiment analysis · Data augmentation · And feature selection

✉ 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, Raipur, Chhattisgarh, India

1 Introduction

The usage of short message services (SMS) on mobile phones has been improved unpredictably to a significant level due to the properties like independence on internet services and no need for frequent updates. Many companies use this as an advertisement platform [1]. Also, some malicious attackers use this area for illegal activities and security risks like SMS spam. Commercial, unsolicited, and bulk electronic messages are commonly known as spam messages. This is used to transmit viruses, arrogant adverts, or malware to the mobile phones of targeted consumers [34]. SMS, email, Internet telephony, social network are the different platforms used to transmit spam messages to the consumers. The spam messages affect the users and cost both customers and mobile network operators (MNO). Also, it affects the different areas of life such as financial, education, privacy, health, security, etc. [37]. The bulk nature of spam messages is very annoying to the users. More SMS is spam messages such as promotions, discount offers, credit opportunities, and fake lottery notifications. These spam messages use the device memory and transmit apps, including malicious information, to identify the private information or to cause financial loss. Due to these problems, the MNO take some privacy policies to protect their customers from such type of malicious attacks [35].

The process of constructing and extracting attributes from the text is called feature extraction [29, 39]. In SMS spam detection, features are divided into two groups. Also, spam review detection methods are classified into two categories: supervised technique and unsupervised technique [30]. Unseen data reviews can be detected using a labeled dataset in the supervised approach. The unsupervised method can identify the hidden patterns using an unlabeled dataset. Feature selection is the selection of a subset, including features used for the classification problem [7, 8]. Sentiment analysis is a technique for recognizing positive or negative emotions in text. It aids in investigating the role of polarity in short-message spam filtering and the evaluation of whether sentiment classification can help with this aim [38]. It also seeks to give away to verify the idea that short message sentiment functions will improve the outcomes of typical short message screening classifiers [27]. In recent years, various methods aimed at assisting with sentiment classification have been proposed. Lexicon-based approaches are valuable tools in our research. These techniques are used to determine the polarity of a word or expression [24, 31, 32].

Motivation Mainly, spam detection identifies the spam messages and spam emails. It is difficult to determine boundaries for spam detection; one must ensure that the identified SMS are spam pages only and not misclassified as legitimate messages. Millions of spam SMS are sent every day, promoting pornographic websites, drugs or software, or fraud. Spam SMS has significant financial consequences for both end-users and service providers. Because of the growing significance of this problem, a new classification technique has been developed.

Moreover, Sentiment Analysis is concerned with determining whether a bit of text is objective or subjective, and if personal, whether it is positive or negative. It can lead to more accurate tools for extracting semantic information. Hence, in this research work, a Hybrid KNN-SVM classifier and RSO algorithm for classification purposes and AFINN and SentiWordNet techniques are used for sentiment analysis.

1.1 Contribution

- A hybrid classifier with effective optimization is used to obtain an accurate classification.
- To find efficient and optimal values for accurate prediction of spam SMS messages, six methods are used for feature selection with EO to select optimal features, enhancing the classification accuracy.
- A hybrid KNN-SVM classifier is developed along with the RSO algorithm to enhance the overall performance of spam SMS prediction. This technique helps to improve the classification accuracy, and the messages are classified into ham and spam. RSO was used to improve the classification accuracy.
- The sentiment analysis is done by AFINN and SentiWordNet techniques to enhance the classification accuracy. This helps to classify the text into positive and negative.

The remaining paper is arranged as follows. The works connected to our suggested algorithm are discussed in Section 2. In Section 3, the presented algorithm is briefly presented. Using some simulation parameters, the simulation results and performance are deliberated in Section 4. Section 5 includes the conclusion.

2 Related work

2.1 Spam message classification from SMS

The comparison of different classification techniques was proposed by [12]. The appearance of known phrases, words, idioms, and abbreviations affected spam SMS classification. This comparison was done between deep learning (DL) methods and the traditional machine learning techniques. Ordonez et al. [25] proposed a Naïve Bayes algorithm (NB) -based method to classify the SMS into Spam, Invalid, Alert 1, Alert 2, and Alert 3. This model obtained higher accuracy of 89%; however, in this model, the misclassification was occur. Roy et al. [28] introduced an efficient way of filtering the spam SMS. A DL algorithms Long-short term memory (LSTM) and Convolutional neural networks (CNN) models were used to classify spam messages and legitimate messages. This model obtained better accuracy of 99.4%; however this process was applicable for English languages only.

Classification of spam and ham messages using different supervised machine learning algorithms was proposed by [23]. The performance evaluation of supervised machine learning algorithms such as NB Algorithm, maximum entropy algorithm, and SVM algorithm were compared. In that comparison, the SVM model obtained better accuracy of 97.4% on the real time dataset. However, this model required more memory for processing. A novel method for spam SMS filtering based on LSTM and recurrent neural network (RNN) was proposed by [7, 8]. The proposed Keras models and Tensor Flow backend models classify spam and ham messages. This model obtained better accuracy of 98% on UCI dataset. However, the system has high complexity due to the complex structure. Lee and Kang [18], developed a spam SMS message filtering technique based on CBOW based word embedding process. CBOW technique used for the word embedding technique and feed-forward neural network was applied for the classification. But, the accuracy was not improved when the hidden layers were increased. Hence, there was a need of optimized hidden layers.

2.2 Techniques for feature selection

A novel technique for selecting features on new semantics was proposed by [22]. The major aim of this work was to group the features based on word based into semantic topics and makes feature vectors. This work used three feature selection models and the performance was compared with the nine machine learning approaches. In this work, for some classifiers the error was increased slightly. Cekik and Uysal [6] suggested a unique feature selection technique for short text categorization based on rough set theory. A rough set was used to calculate the sparsity effect. The experimentation was carried by varying the sizes of features for four datasets. Finally, the proposed approach performed better in terms of Macro-F1 scores. A novel method was proposed for the classification of text, named Multivariate Relative Discrimination Criterion (MRDC), and was proposed by [16]. The filter and supervised feature selection methods were assigned to the MRDC technique. This strategy focused on reducing duplicate features. This model overcomes the other univariate and multivariate models.

[13] developed a method for classifying text with a small database. The evaluation of feature selection considered some criteria on classification performance, efficiency, and stability called Multiple Criteria Decision Making Problem (MCDM); a comparison of five MCDM based methods was also developed. In some cases, this model obtained poor results than the existing feature selection models. A cost-sensitive feature collection was designed by [5]. This model has two phases. In the initial phase, a multi-objective evolutionary feature selection reduces misclassification cost and minimizes the number of attributes used for spam classification. Then, cost sensitive ensemble model was used. This model was evaluated on two datasets and obtained better performance. However, the results may degrade for large datasets.

2.3 Sentiment Analysis from SMS

An opinion mining technique was developed for detecting the polarity from the text was proposed by [11]. This work introduced a modern form of sentiment approach, named sentiment phrase pattern matching. It was a technique that determines the sentiments from the response text. Deep learning models used for the sentiment classification were discussed [4]. The Deep Learning models were tested on movie reviews in the Turkish language. The impact of pre-word embeddings on the proposed model was discovered. The identification of the need to improve the sentiment analysis was presented by [15]. It combined categorization with domain-specific contextual analysis and domain-adopted lexicons to improve knowledge. To keep track of keywords and their sentiment levels, sentiment lexicons were employed. A deep network model for paraphrase detection for short text messages was proposed by [2].

A unique deep neural network-based strategy was developed, relying on coarse-grained sentence modeling with a convolutional neural network (CNN), a recurrent neural network (RNN) model, and a fine-grained word-level similarity matching model. Pong-Inwong et al. [26] introduced Sentiment phrase pattern matching is a new sentiment analysis method (SPPM). The suggested approach is divided into three phases. It extracts reactions and opinions from dialogues in a teaching evaluation process as open-ended queries, allowing students to submit comments to their educators on elements that influence coaching and learning in the institution. A new mechanism was proposed by [36] to improve sentiment analysis accuracy. The new method was a hybrid method that combines Bi-LSTM and CNN to form the LMAEB-CNN technique. It reduces the overfitting problems and improves the classification accuracy. Kumar and Kurhekar [14] introduced a digit locker in cloud as the sentiment analyzer. Various ML classifiers were used for text classification. This model enhanced the analyzer accuracy by measuring the features with large information relevant to the specific class. Finally, the classifier logistic regression achieved better accuracy of 84.8% on NLTK dataset. Sharma et al. [31–33] presented a sentiment analysis on the basis of lexicon for human emotion analysis. The fuzzy set function was utilized for complementing the emotional values of the negated world. Finally, this algorithm proved that this model has better emotion analyzing capacity when compared to other approaches**.** The existing research works focused only on automatic classification of text and doesn't consider the messages. Even though, some of the research works were carried out on SMS classification, the approaches doesn't obtained better results. Sometimes misclassification occurs, that is spam messages are classified as Ham. Further, the accuracy of sentiment analysis is low. None of the research works addressed the issues of spam messages and sentiment analysis within a single frame work. In the last two decades, machine learning models obtained better results in automatic classification. Hence, to overcome these issues hybrid KNN-SVM classifier is introduced. Further, to optimize the parameters of KNN-SVM is optimized by RSO algorithm which is used to increase the classification accuracy.

3 Proposed methodology

The difficulty of detecting spam SMS messages is a subset of seeing spam e-mails. An SMS message is restricted to 160 characters and can only include text, hyperlinks, graphics, and attachments. As a result, spam message detection is a two-class text classification issue, with the classes "spam" and "ham. "This section contains subsections that discuss the proposed system's architectural overview, preprocessing, feature extraction, feature selection, classification, and sentiment analysis.

The structure of the suggested method is represented in Fig. 1. The block diagram shows the dataset undergoing preprocessing, data augmentation, feature selection, classification, and sentiment analysis. The dataset consists of ham and spam data used for testing and training functions. Preprocessing is a technique for cleaning and preparing data for the following stage. It is done by steaming, tokenization, stop word removal. Word2vec augmentation is utilized to minimize the feature space by extracting relevant features from the dataset. After the augmentation, the feature selection is carried out by six various techniques such as Proportional Rough Feature Selector (PRFS), Pearson Correlation coefficient (PCC), Least Absolute Shrinkage and Selection Operator (LASSO), GSS coefficient, Multivariate Relative Discrimination Criterion (MRDC), and Copula based feature selection (CBFS) along with the EO algorithm used to select an optimal

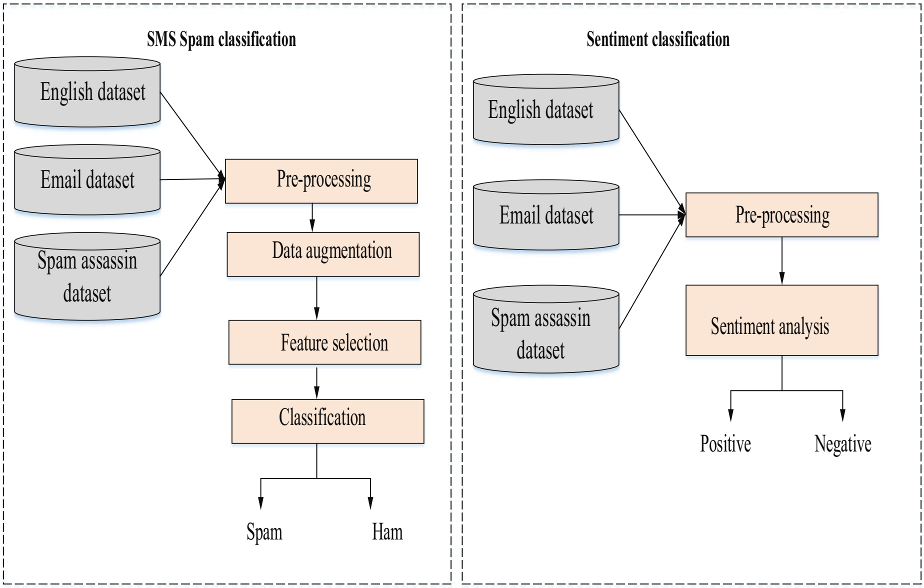


Fig. 1 Proposed system architecture

feature from these feature set. The classification includes the training phase, and the data is classified into spam and ham messages. A hybrid KNN-SVM classifier is used for classification, and the RSO algorithm is used along with this classifier to increase the classification accuracy. The sentiment analysis is done on the dataset for analyzing the polarity. AFINN lexicon-based approach and SentiWordNet is used for the sentiment analysis. It is used to classify the dataset into positive and negative.

3.1 Preprocessing

Text is a form of data that is a sequence of words or characters. Data pre-processing is a crucial step in ML technology. Stemming, stop word removal, and tokenization is some of the steps involved. Stop words such as 'the', 'an', 'a', 'in' need to be ignored while processing. Stop words are removed using a list of words already considered as stop words. Natural Language Toolkit (NLT) has a stop word list that consists of nearly 16 different languages. Tokenization is a process that splits the sentence, paragraph, or text into smaller units. Stemming is used to reduce words into their stem by chopping off the ends of words and often by removing derivational affixes.

Stemming In practically all Natural Language Processing (NLP) projects, stemming is the most used data pre-processing process. Stemming is the process of reducing a word to its word stem, which attaches to suffixes and prefixes or to the roots of words called a lemma.

Stop word removal Stop word removal is one of the most used pre-processing processes in various NLP applications. The concept is to remove words that appear in all of the documents in the corpus. Articles and pronouns are typically categorized as stop words. These words have no meaning in some NLP tasks, such as information retrieval and classification, implying that they are not discriminative.

Tokenization Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into smaller components like individual words or phrases. Tokens are the names given to each of these smaller units. Words, numerals, or punctuation marks could be used as tokens. By finding word boundaries, tokenization creates smaller units.

3.2 Data augmentation using word2vec augmentation

After the preprocessing, the features are moved to the data augmentation process. The data augmentation process works on the exact principle of feature extraction. The proper data augmentation technique is used to achieve better performance. Thus proper data augmentation should help to enhance the performance of the model. Word2vec based augmentation is used for the augmentation process. Word2vec is a reliable augmentation method that locates the most relative terms for an input word using a word embedding pre-trained model on a publicly available dataset.

Using Gensim, the SMS data is converted to a Word2vec format. The modified models were then used to supplement data by choosing a word from a sentence at random and using cosine similarity to discover similar words. We utilize the cosine similarity as a weighting factor to find a replacement for the input word to locate a similar term. Word2vec has the advantage of generating vectors that are more topically connected, or words with similar meanings are represented similarly. Here augmentation is done in three steps.

Synonym augmentation The best name classes for having synonyms in many situations are verbs and nouns. It organizes verbs, nouns, adverbs, and adjectives into synsets, collections of cognitive synonyms that express a different term. It also offers brief descriptions and usage examples and a variety of relationships between these synonym sets. It connects word forms and letter strings specific to word senses; as a result, in the network, semantically disambiguated words are discovered near each other.

Semantic similarity augmentation Using distributed word representation, one may distinguish semantically related terms. This approach needs either a pre-trained text embedding structure for the target language or enough information from the target system to generate the embedding model. This method does not necessitate access to a language's dictionary to locate synonyms. This will aid languages where such tools are more difficult to come by but with sufficient unsupervised text information to create embedding models.

Round-trip translation It is also called as recursive, rear, or bi-directional localization. It's the method of transforming a word, phrase, or sentence from one language to another and back again. RTT may be used as a supplement to increase the amount of training data. This approach combines the source and aim sentences to form a new pair that retains the original meaning.

3.3 Feature selection

The extracted features are the next move to the feature selection process. Due to filters, wrappers, and embedded feature selection approaches, researchers recommend filters to pick different features, particularly in text classifications issues, because filters are classifier independent and have a rapid computation time. Here, six techniques such as PRFS, PCC, LASSO, GSS, MRDC, and CBFS are used for the effective comparison. Calculate the result of each method with equilibrium optimization.

3.3.1 Feature selection techniques

Proportional rough feature selector (PRFS) The sparsity issue is a sort of issue that arises in brief writings with a small number of words. RST could be used for an efficient and effective solution to this challenge. RST uncovers hidden patterns in data and has a high success rate in exposing redundant and nonsensical data, leading to inconsistencies in the computer system. Using RST, a unique feature selection approach based on the filter is suggested, namely PRFS [6]. A filter feature selection method should provide high scores to highly relevant features and lower grades to less relevant ones in theory. The significant value of a term can be evaluated by using the succeeding formula.

$$PRFS\,(t) = \sum_{i=1}^{M} \frac{|Lower| + \alpha * |MS|}{|NMS|/(|SP_0| + |NEG| + 1)} \tag{1}$$

Here, $|NMS|$ and $|MS|$ are the total elements of sets and $|SP_0|$ denotes the entire count of items in the collection SP_0 and $|NEG|$ is the total count of elements in the collection NEG.

Pearson correlation coefficient (PCC) The standard metric is used in machine learning [20]. It’s a metric for expressing the power of a linear relationship between two data variables, with values ranging from 1 to −1.

A positive correlation is shown by the number 1: A positive correlation is when the values of one variable rise in tandem with the values of another. A negative correlation is shown by the number − 1: One variable’s value falls as another rise. 0 represents no linear correlation between two variables. A PCC-based strategy is used to pick the optimized features by deleting the redundant functions. The PCC-based feature selection approach tests various subsets of features based on strongly correlated characteristics. The following equation is used to estimate the value of the Pearson correlation coefficient using two parameters m_i and n_i.

$$P_{mn} = \frac{\sum_{i=1}^{t} \left(m_i - \overline{m}\right) \left(n_i - \overline{n}\right)}{\sqrt{\sum_{i=1}^{t} \left(m_i - \overline{m}\right)^2} \sqrt{\sum_{i=1}^{n} \left(n_i - \overline{n}\right)^2}} \tag{2}$$

Where $\overline{m}$ and $\overline{n}$ are the mean values of two parameters.

The Pearson correlation coefficient is based on the following assumptions:

- All variables must have a natural distribution.
- The two factors have a straight line relationship.
- The data is spread evenly along the regression axis.

Least absolute shrinkage and selection operator (LASSO) It's an efficient tool that does two things: regularizes and selects features. The LASSO method [19] limits the number of the absolute values of the model parameters: it must be less than a predetermined amount (upper bound). The approach employs a shrinkage (regularization) procedure in which the regression parameters' coefficients are punished, with some being lowered to zero. During the features selection phase, variables with a non-zero coefficient following the shrinking procedure are selected for the model. The purpose of this procedure is to reduce the prediction error as much as possible.

In tuning parameter λ, that controls the penalty's power, is essential in practice. Indeed, when it λ is large enough, the dimensionality is reduced by forcing the variables to be precisely equal to zero. The larger the parameter λ, the more coefficients would be reduced to zero. If λ= 0, however, we have an OLS (Ordinary Least Squares) regression. The LASSO approach has a variety of advantages. It is feasible to minimize variance without significantly increasing bias, for example, by lowering and deleting variables; this is especially successful when there are a small count of instances and a large count of variables. When it comes to the tuning parameter λ, we know that as λ increases, bias rises, variance falls; therefore, a trade-off between bias and variance must be found.

Furthermore, the LASSO helps to improve model interpretability by removing unnecessary variables that aren't related to the answer variable, reducing overfitting. Since the emphasis of this paper is on the feature selection task, this is the point where we are most interested.

GSS coefficient (GSS) The GSS coefficient [21] is a condensed version of the statistics X^2. They absolutely eliminate the $\sqrt{N}$ function and the denominator. It has less value for features which are limited but have more correlation coefficient. It is computed by:

$$GSS(f) = \text{maximum}\, GSS\left(f, c_j\right) \tag{3}$$

$$GSS\left(f, c_j\right) = \left(N_{f,c_j} \times N_{\overline{f},c_j}\right) - \left(N_{f,\overline{c_j}} \times N_{\overline{f},c_j}\right) \tag{4}$$

where N_f and $N_{\overline{f}}$ are the document frequency with and without features. N_{c_j} and $N_{\overline{c_j}}$ are the document frequency belongs and not belongs to c_j.

Multivariate relative discrimination criterion (MRDC) MRDC [16] analyses attribute in two phases. In the first phase, features are evaluated with the help of conventional RDC parameters, and redundancy with them is explored in the second stage to pick a last group of elements. The initial stage aims to identify the most critical qualities conceivable, while the second step evaluates their relationship. In this stage of the suggested method, attributes are first sorted in falling order by their weight values. A function with the greatest significance is detected and included in the subset that was chosen at the end (represented by S). S is then combined with a function with the lowest association with S. To put it another way; each stage determines the correlation between non-selected and selected attributes. The function with both the highest applicability and most minor correlation is picked. This procedure is repeated until the chosen features (S) volume exceeds k. The following equation is used to measure the MRDC coefficient.

$$MRDC\,(f_m) = RDC\,(f_m) - \sum_{f_m \neq f_n, f_n \in S} correlation\,(f_m, f_n) \tag{5}$$

Where $RDC(f_m)$ is the importance of a characteristic f_m, and the relationship between two characteristics f_m and f_n is denoted by *correlation* (f_m, f_n), Their relationship value determines this. The correlation value is calculated using the PCC.

$$correlation\,(f_m, f_n) = \left| \frac{\sum_{q\in|docs|}\left(f_{m,q}-\overline{f}_m\right)\left(f_{n,q}-\overline{f}_n\right)}{\sqrt{\sum_{q\in|docs|}\left(f_{m,q}-\overline{f}_m\right)^2}\sqrt{\sum_{q\in|docs|}\left(f_{n,q}-\overline{f}_n\right)^2}} \right| \tag{6}$$

Where the mean values of f_m and f_n vectors are represented by $\overline{f}_m$ and $\overline{f}_n$ respectively. $f_{m,q}$ and $f_{n,q}$ are the worth of features m and n for q^{th} document respectively. A perfect positive correlation has a value of 1, while a perfect negative correlation has a value of −1. The suitable values between attributes in datasets can be negative in some situations, which can cause problems when computing the MRDC criterion's second factor. To answer this condition, the eq. (7) is utilized to recalculate the range values from [1 1] to [0 1].

$$normalize\,(x_m) = \frac{x_m - x_{m,\min}}{x_{i,\max} - x_{i,\min}} \tag{7}$$

Where $x_{m,\,max}$, $x_{m,\,min}$ are the highest and lowest values of x_m respectively.

Copula based feature section (CBFS) The copula-based attribute selection technique [17] is utilized to optimize redundancy and relevancy, which is more durable than previous methods. Additionally, the copula mutual information between and is minimized, decreasing redundancy between them while simultaneously increasing the mutual knowledge of copula with a class label. As a result, while we're utilizing a first-order incremental hunt to choose one characteristic at a time in each stage, we suggest that instead of using the traditional knowledge metric, we use (empirical) copula-based shared information to achieve more stability. In addition, multivariate mutual knowledge is used rather than using the average after choosing multiple features. It can be expressed mathematically as follows.

After the selection of features $f_1, \cdots, f_n \in S$, then select next feature ($f_{n+1} = f_{CBFS}$) by using

$$\begin{aligned} f_{CBFS} &= \arg \max_{f_m \in (F-S)} \left[T_C(f_m; G) - T_C(f_m; f_1; f_2; \cdots, f_s)\right] \\ &= \arg \max_{f_m \in (F-S)} \Big[-H\Big(C\,(P(f_m), P(G)) + H(C\,(P\,(f_m), P(f_1), \cdots, P(f_n)))\Big] \end{aligned} \tag{8}$$

Here $H(f_m)$ represents non-selected characteristics entropy, $H(f_n)$ represents the entropy of a set of features, and $H(G)$ indicates the target class's entropy.

3.3.2 Equilibrium optimization (EO)

Equilibrium Optimization [10] is done to select optimal features from the six different characteristics. It's a new optimization approach that uses control volume mass balance methods to assess equilibrium and dynamic phases. Each particle (solution) acts as a search agent in EO, with its concentration (position). The following three steps discuss the mathematical modeling of EO.

Step1: *Initialization:* The optimization process is initiated by the initial population of EO; these initial constraints are selected based on several particles and dimensions with uniform initialization. Random generation of the initial concentration vector is according to the following equation.

$$D_i^{initial} = D_{\min} + (D_{\max} - D_{\min}) * X_i \tag{9}$$

Where $i = 0, 1, 2, \cdots, n$. The concentration vector is represented by $D_i^{initial}$. The upper bound dimension in the problem is determined by $D_{\min}$, whereas the lower bound dimension is determined by $D_{\max}$. In eq. 9, X_i indicates a random number and n is the total number of particles present inside the group. The value of X_i is in between [0, 1].

Step 2: *equilibrium pool and candidates:* Like all optimization algorithms, EO is also trying to achieve a better optimization result. It continuously searches for the system's equilibrium state. After attaining the state of equilibrium, it forces to move towards the near-optimal solution of the optimization problem. During optimization, EO doesn't know the concentration level to attain the equilibrium state. Therefore, it is forced to assign five particles. Among the five particles, four particles are the best ones in the population, and the extra one is the average of these four particles. Further exploitation and operator exploration is carried out with the help of these five equilibrium particles. The selected five particles are stored as vectors, generally known as the equilibrium pool. The following equation indicates the typical representation of equilibrium pool.

$$\overrightarrow{D}_{eq,pool} = \left\{ \overrightarrow{D}_{eq\,(1)}, \overrightarrow{D}_{eq\,(2)}, \overrightarrow{D}_{eq\,(3)}, \overrightarrow{D}_{eq\,(4)}, \overrightarrow{D}_{eq\,(avg)} \right\} \tag{10}$$

Step 3: *Updating the concentration:* EO usually having a balance between diversification and intensification. Let $\overrightarrow{\gamma}$ represents a random vector which lies in the interval [0, 1]. Then the expression for fitness function can be explained as follows.

$$\overrightarrow{B}_f = e^{-\overrightarrow{\gamma}(\tau - \tau_0)} \tag{11}$$

Where τ represents the current iteration and the initial value is represented by τ_0. $\overrightarrow{B}_f$ represents an exponential term. According to the following equation, the value of decreases as the number of iterations grows.

$$\tau = \left(1 - \frac{n_{it}}{\tau_{\max}}\right)^{\left(d_0 * \frac{n_{it}}{\tau_{\max}}\right)} \tag{12}$$

Where n_{it} denotes the total count of iteration, τ and $\tau_{\max}$ denotes the current and maximum value of iteration respectively. To control the capability of intensification a constant parameter d_0 is used.

Step 4: *Optimization of parameter:* Parameter optimization is achieved by updating the concentration. The following equation describes the parameter optimization process.

$$H_{opt} = \overrightarrow{E}_{pc} + \left(\overrightarrow{E} - \overrightarrow{E}_{pc}\right) * \overrightarrow{B}_f + \left(\frac{1}{\gamma * V}\right) * \left(1 - \overrightarrow{B}_f\right) \tag{13}$$

Where γ is a random vector in between [0, 1]. $V = 1$ and F is an exponential term. As a result of EO, the features from copula based feature selection are selected as the optimal feature set.

3.4 SMS classification using hybrid KNN-SVM with RSO

The selected feature is transferred to the classification process. The classification process is done by a hybrid KNN-SVM method and the RSO algorithm. The k-nearest neighbor (KNN) and support vector machine (SVM) pattern classification algorithms are used. Instead of looking at the nearest K occurrences to the unclassified case, the K-Nearest Neighbour approach looks at the closest K instances. The new instance class is decided by the class that appears the most frequently among those K instances. We use the trial-and-error method to select K, achieving the best outcome. The Euclidean distance is used to estimate proximity SVMs (support vector machines) are efficient data classification methods.

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{14}$$

where x_i and y_i are the two points in Euclidean distance. They assign two-category points to two disjoint half-spaces in either the linear classifier's original input space or a higher-dimensional feature space for nonlinear classifiers. An SVM aims to produce a decision surface with a hyperplane that optimizes the difference between true and false examples. An effective approach in computational learning theory is used to achieve this beneficial property. It employs a process of systemic risk minimization in particular. According to the theory, the mathematical definition of Vapnik-Chervonenk encircles the error rate generalized is (VC) dimensionality. Kernels are used to do all necessary computations in the input space directly by implicitly mapping into a high-dimensional dot product feature map as an input vector.

The KNN method classifies component vectors based on the closest training instances in the feature space. The class with the most KNN receives a hidden function vector, where k is a positive integer. The k value is determined empirically, for example, by considering the training dataset's classification error. The class of the function vectors for another neighbor is simply assigned to it in the case where k = 1. On the other hand, SVM is a state-of-the-art pattern classification algorithm that uses the kernel trick to find the maximum-margin hyperplane in a transformed feature space. Although there are many kernel forms, the linear kernel was chosen for this investigation because of its previous effectiveness in text categorization research. The SVM classifying algorithm is effective for samples far from the separating hyperplane, whereas the KNN categorizing technique is appropriate for data close to the hyperplane. The distance formula used here is dependent on the kernel function, and it goes like this.

$$||\phi(x) - \phi(x_i)||^2 = k(x,x) - 2k(x,x_i) + k(x_i,x_i) \tag{15}$$

The distance threshold β should satisfy $0 < \beta < 1$. This value is optimized using RSO algorithm. By optimizing the distance threshold, it helps to improve the classification accuracy. The use of this optimization algorithm can enhance the accuracy of the classification. It is a bio-inspired optimization algorithm that can resolve complex optimization issues. The chasing and attacking actions of rats in nature are essential for this optimizer. The mathematical model of the RSO algorithm [9] is described in two steps. The behavior of rats can be classified into two phases. They were chasing and fighting.

Chasing the prey Rats are generally social animals who engage in social agonistic activity to catch prey in groups. To mathematically characterize this action, we conclude that the best search agent is aware of the position of the prey. The other search agents will adjust their locations about the best search agent found so far. This mechanism is based on the following equations.

$$\overrightarrow{G} = M*\overrightarrow{G}_i(x) + N*\left(\overrightarrow{G}_r(x)-\overrightarrow{G}_i(x)\right) \tag{15}$$

Where the positions of rats defined by $\overrightarrow{G}_i(x)$ and best optimal solution is defined by $\overrightarrow{G}_r(x)M$ and N parameters were calculated using following equations

$$M = B - x \times \left(\frac{B}{Max_{Iteration}}\right) \tag{16}$$

Where, $x = 0, 1, 2, \ldots, Max_{Iteration}$

$$N = rand()*2 \tag{17}$$

As a consequence, B and N are both random numbers ranging from [1, 5] and [0, 2] respectively. Over the course of iterations, the parameters M and N are responsible for improved exploration and exploitation.

Fighting with prey The following equation is used to describe the fighting process of rats with prey.

$$\overrightarrow{G}_i(x+1) = \left|\overrightarrow{G}_r(x)-\overrightarrow{G}\right| \tag{18}$$

Next position of rat was defined by $\overrightarrow{G}_i(x+1)$. It preserves the best option and keeps track of other search agents' locations about the best search agent. The modified values of parameters A and C ensure exploration and exploitation. The RSO algorithm saves the best solution for the fewest operators.

3.5 Sentiment analysis

The whole dataset is transferred for sentiment analysis. Here, the sentiment analysis is done using AFINN and SentiWordNet. The method of recognizing positive and negative opinions about a subject or issue from a text is known as sentiment analysis. This section's primary purpose is to apply each message's polarity to the original dataset to conduct the tests. There are three options for identifying text sentiment. The first is to manually mark text, which needs a lot of effort and time. The second choice is to use an NLP, lexicon, or ML solution. The third type is hybrid, which uses human experts or crowd sourcing to provide input on sentiment analysis results or mark training data sets. AFINN and SentiWordNet are two popular Lexicons.

3.5.1 AFINN lexicon approach

The AFINN lexicon is one of the most basic and widely used lexicons for sentiment analysis. The AFINN lexicon allows a value to each text ranging from −5 to 5, with lower values indicating negative sentiment and higher numbers indicating positive view. Finn Arup Nielsen manually labelled the words in AFINN from 2009 to 2011. The AFINN lexicon has the absolute values and the most positive values.

3.5.2 SentiWordNet approach

The SentiWordNet approach makes use of SentiWordNet's freely accessible library. Each term t found in WordNet is assigned one of three numerical values: obj(t), pos(t), and neg(t), which describe the term's objective, positive, and negative polarities, respectively. The outcomes of eight ternary classifiers are combined to generate these three ratings. To use SentiWordNet, we must first extract specific little words and then check the SentiWordNet ratings. Adjectives are commonly used in an opinionated manner in the English language, whereas adverbs are generally used as modifiers or complements.

Enhanced Variable Scoring and Adjective Importance Scoring algorithms are employed in SentiWordNet. The Adjective priority scoring scheme allows you to score an adjective+adverb combination by giving the significance of adverbs a constant weight. Still, the variable scoring scheme will enable you to change the adjective scores. The Variable Scoring and Adjective Priority Scoring techniques have been modified to simplify and increase accuracy. Rather than restricting the values of adverbs to 0 and adjectives to −1 and + 1, we made one easy alteration, and we used the SentiWordNet's original scores. Adjectives and adverbs are then given different weights based on the scoring system.

The workflow of the suggested algorithm is represented in Table 1. The input dataset is pre-processed using stemming, tokenization, and stop word removal. Then the pre-processed features are augmented using word2vector augmentation. After that different feature selection techniques were used along with EO to identify the optimum features. The selected features are transmitted to classification. The classification can be achieved by using hybrid KNN-SVM with RSO. Then the result evaluation can be done in terms of precision, accuracy, f measure, recall, MAE, RMSE, and kappa statistics metrics.

4 Simulation results

The experiment is done on the PYTHON tool. The experimental analysis is carried out in two phases: classification-based results and sentiment-based results. The different performance metrics like accuracy, precision, recall, f-measure, RMSE, MAE, and kappa statistic matrix are determined to estimate the effectiveness of the proposed approach. To determine the effectiveness of the proposed technique, the evaluation metrics of the suggested methods are compared to current methods. The simulation is done on the English SMS, Email, and spam

Table 1 Algorithm of the proposed work

```
Step 1: Collecting the info
        (a) collect the information from the data resources to collect the input data
        (b) SMS dataset, Email dataset, and spam assassin dataset are used as the input data
Step 2: Getting ready the data
        (a) Apply pre-processing approaches such as stemming, tokenization, stop word removal
Step 3: Data augmentation
        Apply word-2-vec augmentation to the preprocessed data
Step 4: Feature selection
        Apply the six feature selection techniques such as PRFS, PCC, LASSO, GSS, MRDC,
        and CBFS along with EO algorithm to select the attributes
Input: Augmented attributes
Output: Optimum feature with high fitness value
Initialize, the population size $n$
Calculate the entropy value for each feature selection
Then,
Evaluate fitness for equilibrium optimization
Parameter optimization using equation 13
Return the best direction with the most carefully chosen features
Step 5: Classification
        Apply the hybrid KNN-SVM with RSO to classify messages into spam and ham
Calculate the Euclidean distance
The distance is calculated using equation 15
This function is optimized using RSO
Initialize the parameters A,C and R
Calculate the fitness value for search agent
While ($x < Max_{iteration}$) do
For each search agent do
        Update the position of current search agent using equation 18
        $x \leftarrow x+1$
End while
Step 6: Sentiment analysis
        Apply AFINN Lexicon approach, and SentiWordNet approach for the sentiment
        classification.
Step 7: Evaluation of result
        Analyze the outcomes using performance metrics like accuracy, precision, F-measure,
        Recall, Kappa statistics, MAE, and RMSE
End
```

assassin datasets. The proposed method (Hybrid KNN-SVM with RSO) has been compared with techniques like Naïve Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Artificial Neural Network (ANN), Random Forest (RF), and Convolutional Neural Network (CNN) [29].

4.1 Dataset

Even though many email databases have been made available to researchers, there are just a handful of free SMS collections in the research. As a result of this research, a novel SMS message set in English, an email dataset, and a spam assassin dataset are identified Table 2.

Figure 2 depicts a word cloud for both spam and ham messages. A dataset of publically available SMS was used to construct the hybrid KNN-SVM classifier for spam and ham

Table 2 Dataset description

Dataset		Before classification	After classification
SMS dataset	Total messages	5574	5574
	Spam	747	769
	Ham	4827	4805
Email dataset	Total messages	5172	5172
	Spam	1500	1740
	Ham	3672	3432
Spam assassin	Total messages	3252	3252
	Spam	501	529
	Ham	2751	2723

classification. The SMS messages in the database have been classified as either ham or spam. The genuine messages are identified as Ham, while the spam messages are identified as spam. Figure 2 shows a sample word cloud for both ham and spam messages.

4.2 Performance metrics

The proposed terms for performance evaluation are precision, recall, accuracy, and AR value. The outcome shows that the proposed method provides high performance than any other approach now in use. The performance metrics are explained as follows.

- Precision: It can be defined as the number-to-number ratio of positive samples that is classified into total number of samples. It is based on percentage of cases that are wrongly categorized.

$$precision = \frac{TP}{TP + FP} \tag{19}$$

- Recall: It can be defined as the ratio of number of positive samples classified as positive to total number of positive samples. It is based on percentage of cases that are rightly categorized.

$$recall = \frac{TP}{TP + FN} \tag{20}$$

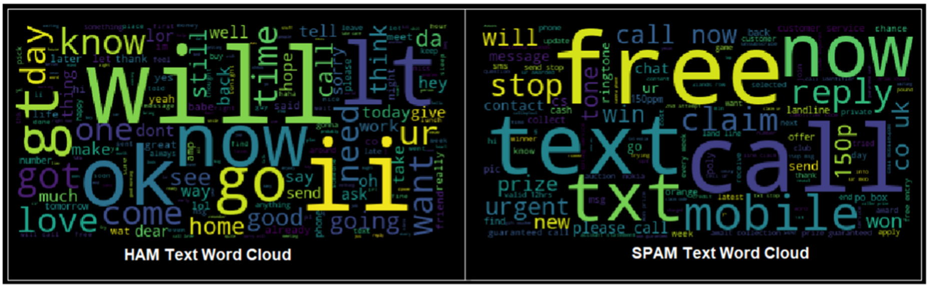


Fig. 2 Word cloud (**a**) ham, (**b**) spam messages

- Accuracy: The value close to the true value is defined as the accuracy.

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

- F-measure: It is a measure of accuracy of test

$$F - measure = \frac{2 \times precision \times recall}{recall + precision} \tag{22}$$

- Kappa Statistics: It's a metric for how closely the instances identified by the machine learning classifier matched the data labeled as ground truth, while accounting for the accuracy of a random classifier as assessed by the predicted accuracy.

$$Kappastatistics = \frac{p(a) - p(e)}{1 - p(e)} \tag{23}$$

Where $p(e)$ denotes the predicted agreement between the classifier and the genuine values, while $p(a)$ denotes the fraction of actual agreement.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(\widehat{x}_i - x_i\right)^2}{n}} \tag{24}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| x_i - \widehat{x}_i \right| \tag{25}$$

Where, $\widehat{x}_i$ denotes the predicted value, x_i characterizes the actual value, and n symbolizes the total observations.

4.3 Performance evaluation

The following section shows the performance for spam and ham classification evaluated by the benchmark datasets English SMS message, Email, and spam assassin.

4.3.1 English SMS message dataset

Tables 3 and 4 show each feature selection technique's mean and variance value. From Tables 3 and 4 analysis, the CBFS with EO produce better results when compared with existing feature selection approaches. Tables 5 and 6 given below provides the comparison of the performance metrics. F-measure and kappa statistics metrics are compared with existing techniques such as NB, SVM, LR, DT, and KNN [3].The performance analysis on SMS dataset for accuracy, recall, precision, f-measure, kappa statistics, MAE, and RMSE are represented in Table 5 and 6.

Figure 3 shows the performance evaluation of the accuracy, precision, recall, MAE, RMSE, kappa static matrix, and F-measure value. The performance of the suggested method is compared with existing techniques such as NB, DT, LR, RF, ANN, and CNN. The proposed method achieved an accuracy of about 99.69%, higher than that of other existing algorithms. The accuracy of the proposed method is improved due to the use of an improved classification

Table 3 Mean and Variance of feature selection techniques

Feature selection method	Accuracy		Precision		F-measure		Recall	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
PRFS-EO	94.98	91.32	81.18	78.06	81.45	78.32	81.72	78.58
PCC-EO	95.29	91.62	82.66	79.48	82.52	79.35	82.39	79.22
LASSO-EO	95.47	91.79	83.33	80.12	83.19	79.99	83.05	79.86
GSS-EO	95.74	92.05	84.33	81.08	84.19	80.95	84.05	80.82
MRDC-EO	96.28	92.57	86.82	83.48	86.09	82.78	85.38	82.09
CBFS-EO	96.95	93.22	90.59	87.10	88.43	85.03	86.37	83.05

Table 4 Mean and variance of feature selection techniques

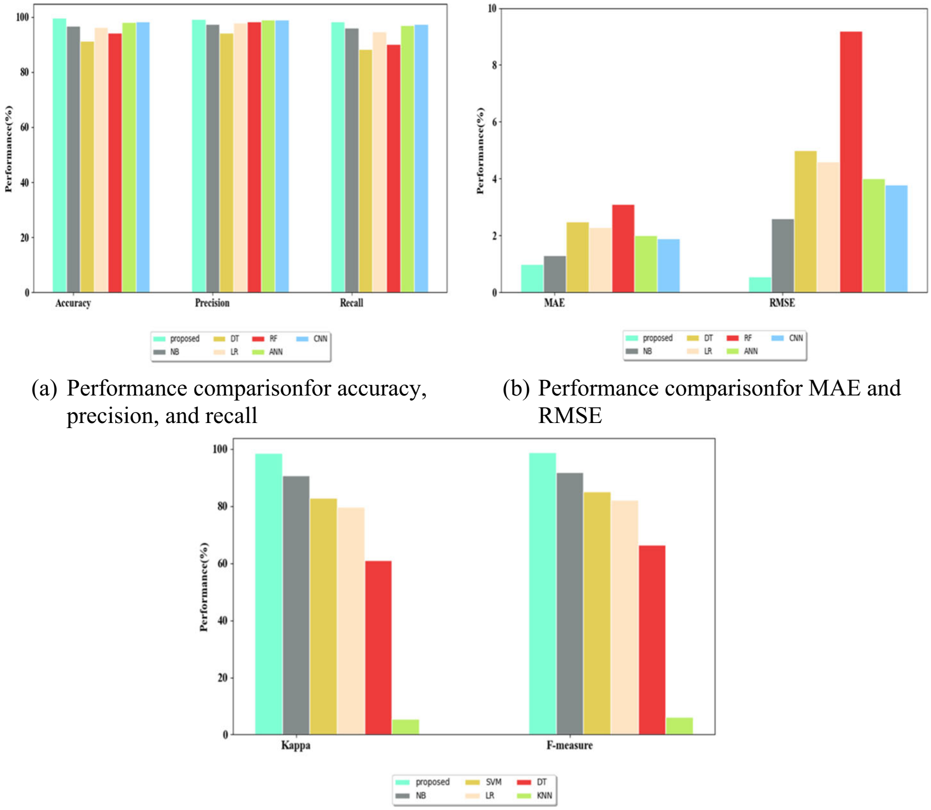
Feature selection method	Kappa statistics		MAE		RMSE	
	Mean	Variance	Mean	Variance	Mean	Variance
PRFS-EO	78.55	75.52	1.58	1.79	1.68	1.95
PCC-EO	79.80	76.73	1.34	1.73	1.63	1.92
LASSO-EO	80.57	77.47	1.29	1.62	1.61	1.86
GSS-EO	81.72	78.58	1.19	1.53	1.58	1.79
MRDC-EO	83.94	80.71	1.04	0.98	1.37	1.42
CBFS-EO	86.67	83.34	0.97	0.52	1.21	1.04

Table 5 Comparison of performance metrics

Method	Accuracy (%)	Precision (%)	Recall (%)	MAE	RMSE
Proposed	99.69	99.32	98.33	0.99	0.56
NB	96.75	0.975	0.961	1.3	2.6
DT	91.25	0.943	0.883	2.5	5
LR	96.25	0.979	0.946	2.3	4.6
RF	94.25	0.984	0.902	3.1	9.2
ANN	98	0.989	0.970	2	4
CNN	98.25	0.989	0.975	1.9	3.8

Table 6 Comparison of performance metrics

Method	F-Measure	Kappa Statistics
Proposed	98.83	98.64
NB	91.9	90.7
SVM	85.1	82.9
LR	82.1	79.8
DT	66.4	61.0
KNN	62.98	54.43



(a) Performance comparisonfor accuracy, precision, and recall

(b) Performance comparisonfor MAE and RMSE

(c) Performance comparison in terms of kappa statics and f-measure

Fig. 3 Overall Performance of hybrid KNN-SVM with RSO classifier

technique. The NB, DT, LR, RF, ANN, and CNN techniques obtained an accuracy of about 96.75%, 91.25%, 96.25%, 94.25%, 98%, and 98.25%, respectively. The proposed method showed a precision of 99.6%, which is higher than that of other techniques. As the accuracy increases, the precision also increases, which is due to the use of the hybrid classifier technique used in the prediction. The precision of NB, DT, LR, RF, ANN, and CNN are 97.5%, 94.3%, 97.9%, 98.4%, 98.9%, and 98.9% respectively. The proposed method showed a recall value of 98.6%, and the techniques like NB, DT, LR, RF, ANN, and CNN showed 96.1%, 88.3%, 94.6%, 90.2%, 97.0%, and 97.5%, respectively. The value of recall is increased because of the increase in accuracy and precision. The proposed method showed 0.927 of F-measure value and the existing techniques NB, SVM, LR, DT, and KNN showed values such as 0.919%, 0.851%, 0.821%, 0.664%, and 0.062% respectively. The presented method showed a kappa statistics value of 0.912, and the techniques like NB, SVM, LR, DT, and KNN showed 0.907, 0.829, 0.798, 0.610, and 0.054, respectively.

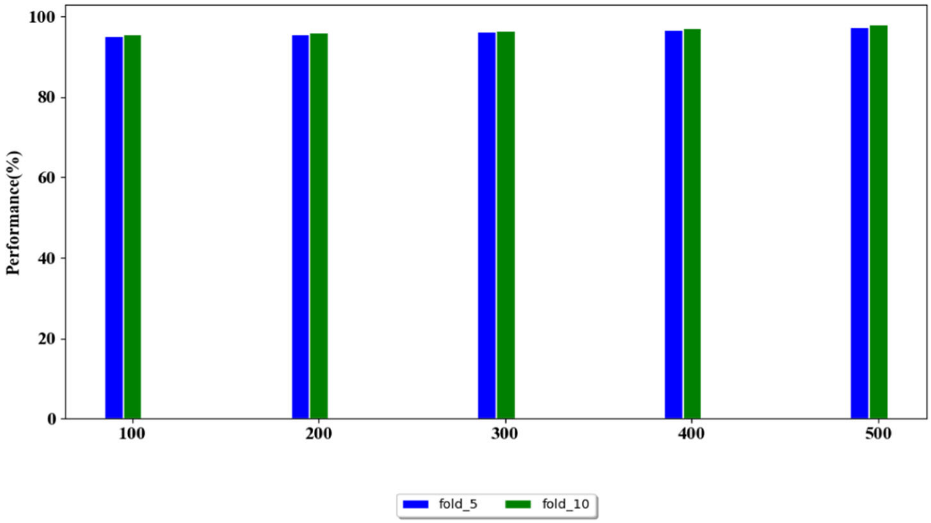


Fig. 4 Hidden neuron vs accuracy for 10-fold and 5-fold cross-validation

The cross-validation result attained for the different numbers of hidden neurons are shown in Fig. 4. We have used 500 neurons in this work, and the cross-validation accuracy for 100, 200, 300, 400, and 500 neurons is evaluated. The 5-fold and 10-fold cross-validation is done in

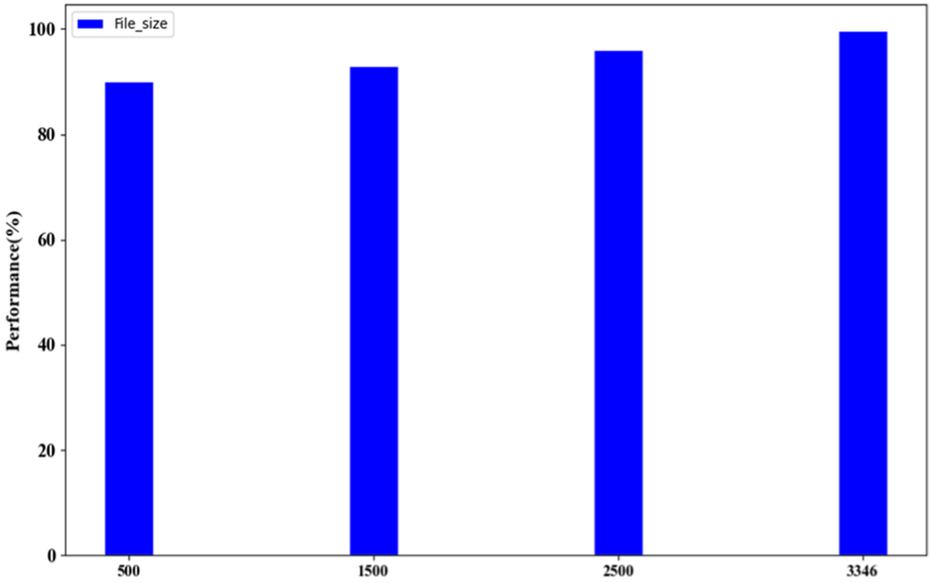
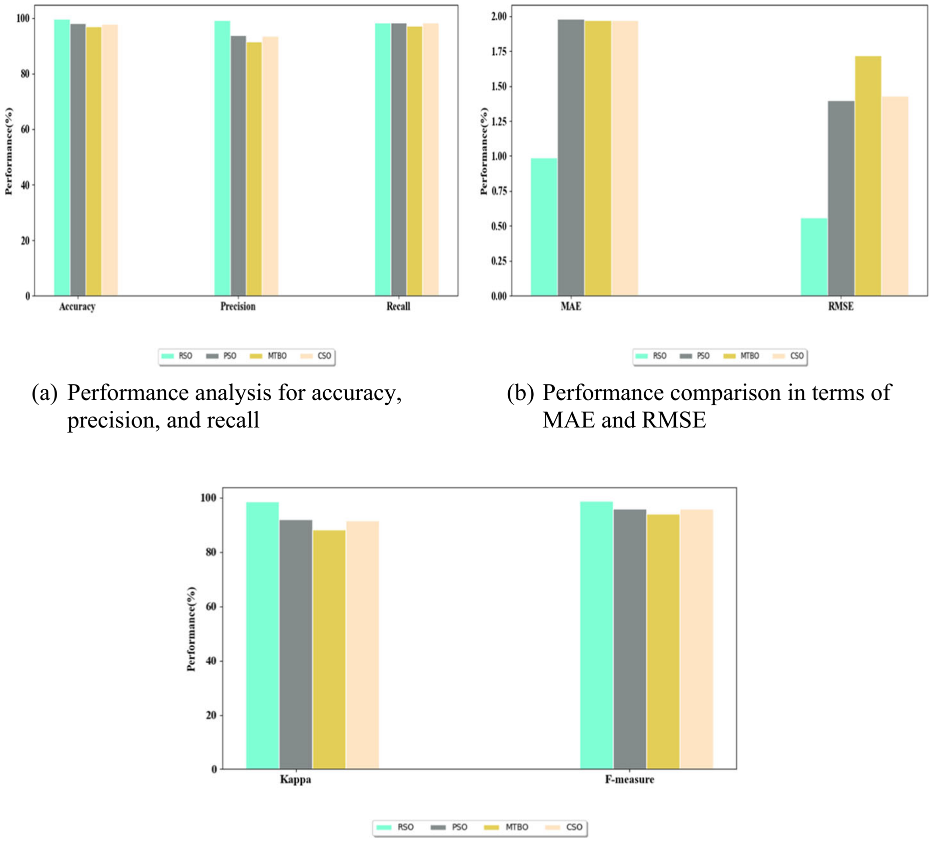


Fig. 5 Accuracy comparison for hybrid KNN-SVM with RSO classifier with different file size

this method. The accuracy attained at fold-5 for 100 neurons is higher than other neurons. For 10-fold, the 400 hidden neurons have achieved higher accuracy results.

Figure 5 represents the accuracy attained for different file sizes. The accuracy of the proposed hybrid KNN-SVM with RSO classification gets increased with an increase in file size. This is mainly due to the efficient performance of the proposed feature selection and classifier techniques. The total length of the file used in our work is 5572. The features selected by the proposed feature selection techniques show promising results in classification. Figure 6 compares different optimization algorithms with accuracy, recall, precision, and f-measure. The proposed model is compared with three different optimizations like Particle Swarm Optimization (PSO), Magnetotactic Bacteria Optimization (MTBO), and Crow search optimization (CSO). From the analysis it is proved that the proposed RSO based classification obtained better results.

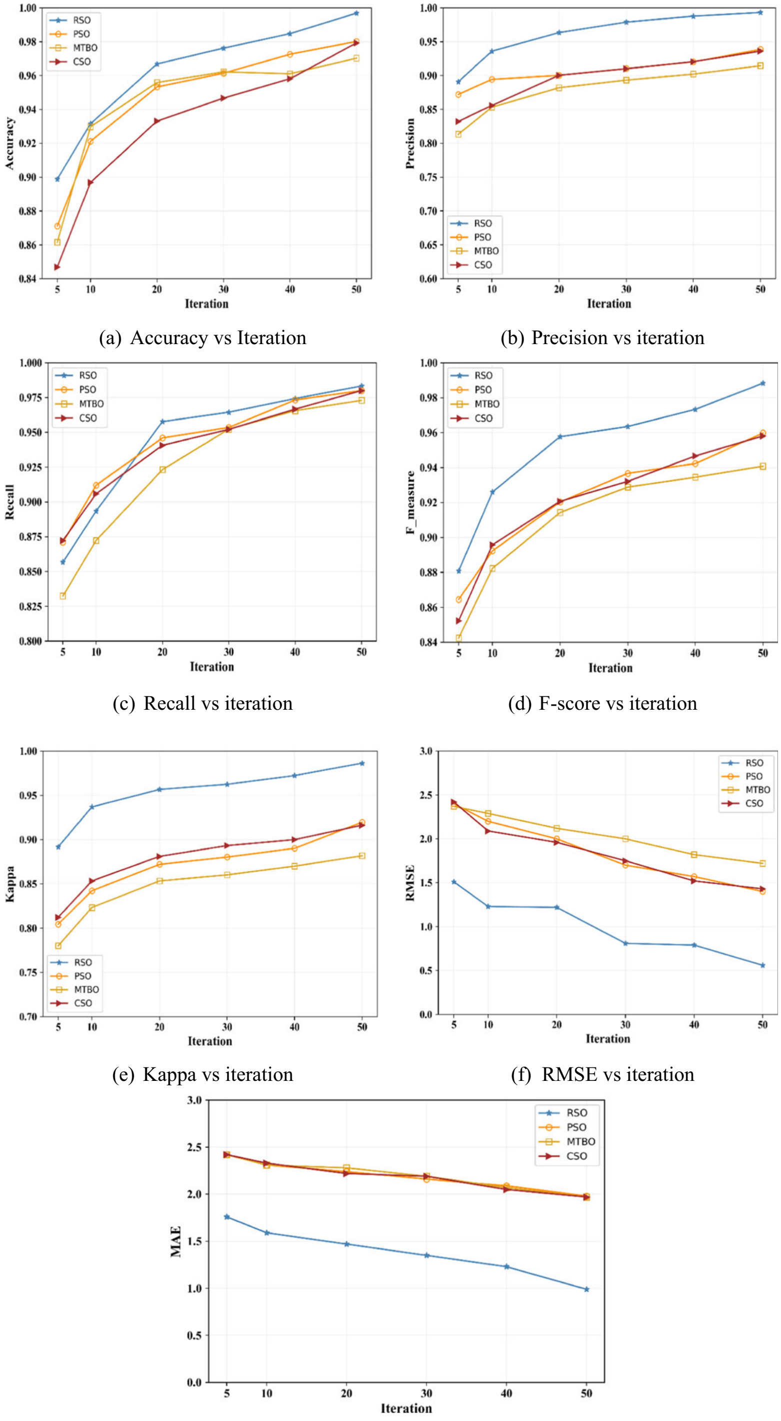


(a) Performance analysis for accuracy, precision, and recall

(b) Performance comparison in terms of MAE and RMSE

(c) Performance comparison for of kappa statics and f-measure

Fig. 6 Comparison of different optimization algorithms



(a) Accuracy vs Iteration

(b) Precision vs iteration

(c) Recall vs iteration

(d) F-score vs iteration

(e) Kappa vs iteration

(f) RMSE vs iteration

(g) MAE vs iteration

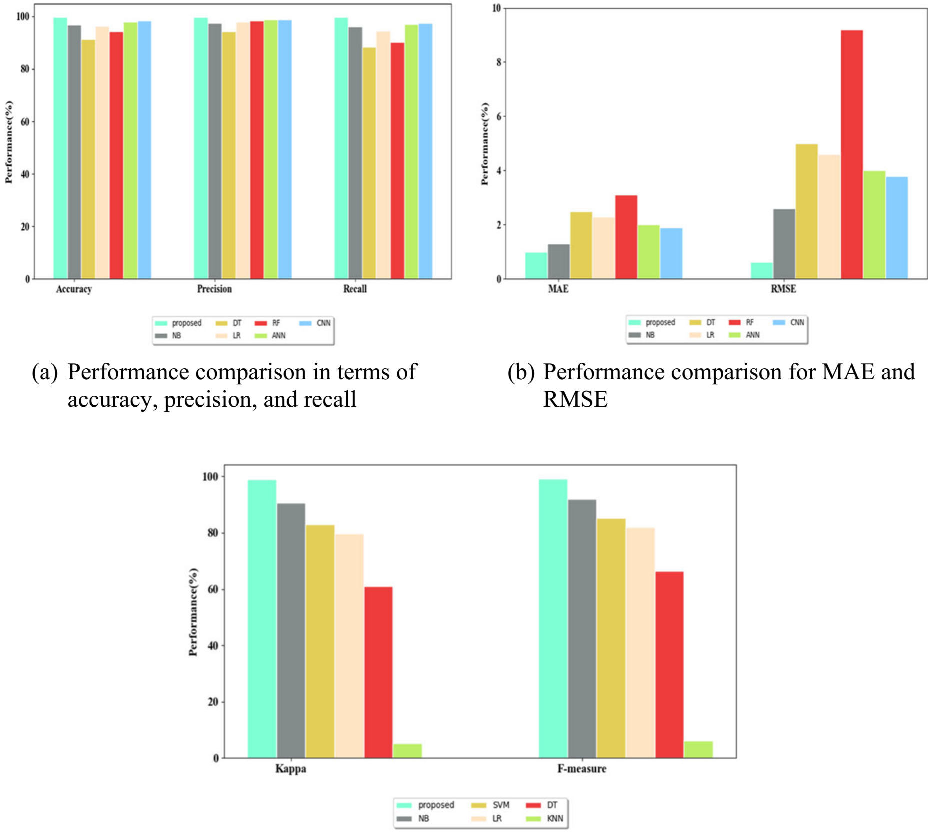
Fig. 7 Comparison of different performance metrics for various optimization with iteration

The optimization-based results for different iteration values for the SMS dataset are shown in Fig. 7. To verify the effectiveness of the proposed optimization, the experimentation is conducted for different optimizations. The proposed RSO optimization algorithm is compared with three various optimizations: MTBO, PSO, and CSO. From these results, the proposed RSO is better than other optimization algorithms.

4.3.2 Email dataset

Figure 8 shows the performance evaluation of the accuracy, precision, recall, MAE, RMSE, kappa statics matrix, and F-measure value for the email dataset. The performance of the presented method is compared with existing techniques such as NB, DT, LR, RF, ANN, and CNN. The proposed method achieved an accuracy of about 99.76%, higher than that of other existing algorithms.

Figure 9 represents the metrics like accuracy, precision, recall, MAE, RMSE, kappa and f-measure. From the comparison, it is observed that the performances based on RSO algorithm attained better results. The optimization-based results for different iteration values for the email dataset are shown in Fig. 10. The proposed RSO optimization algorithm is compared with

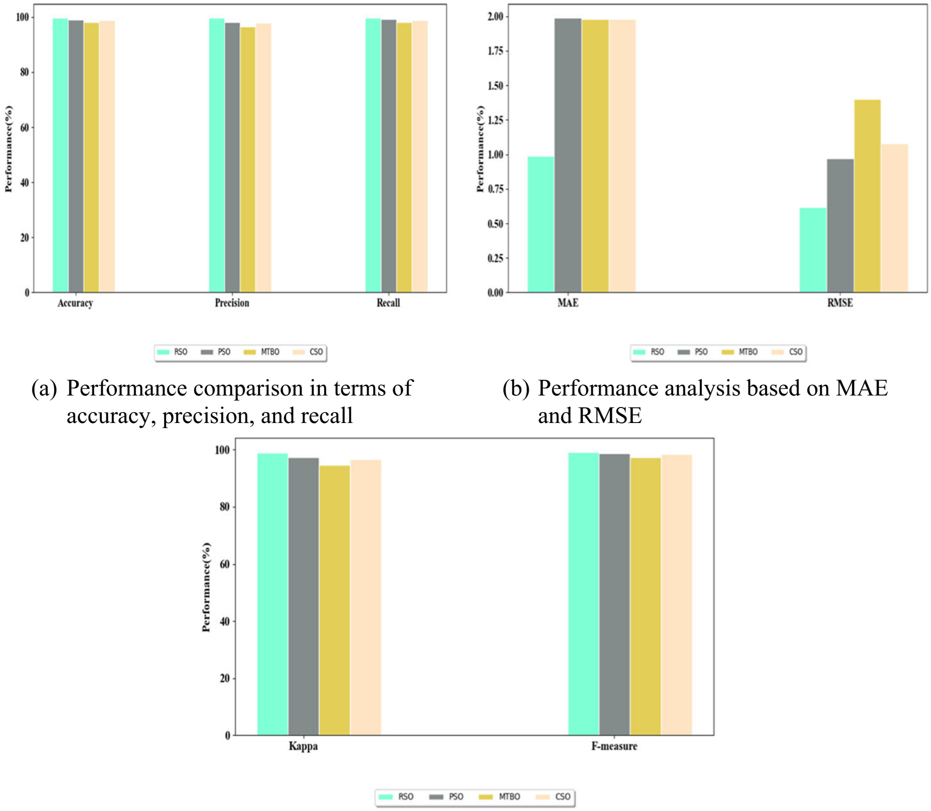


(a) Performance comparison in terms of accuracy, precision, and recall

(b) Performance comparison for MAE and RMSE

(c) Performance analysis in terms of kappa statics and f-measure

Fig. 8 Overall Performance of hybrid KNN-SVM with RSO classifier with email dataset



(a) Performance comparison in terms of accuracy, precision, and recall

(b) Performance analysis based on MAE and RMSE

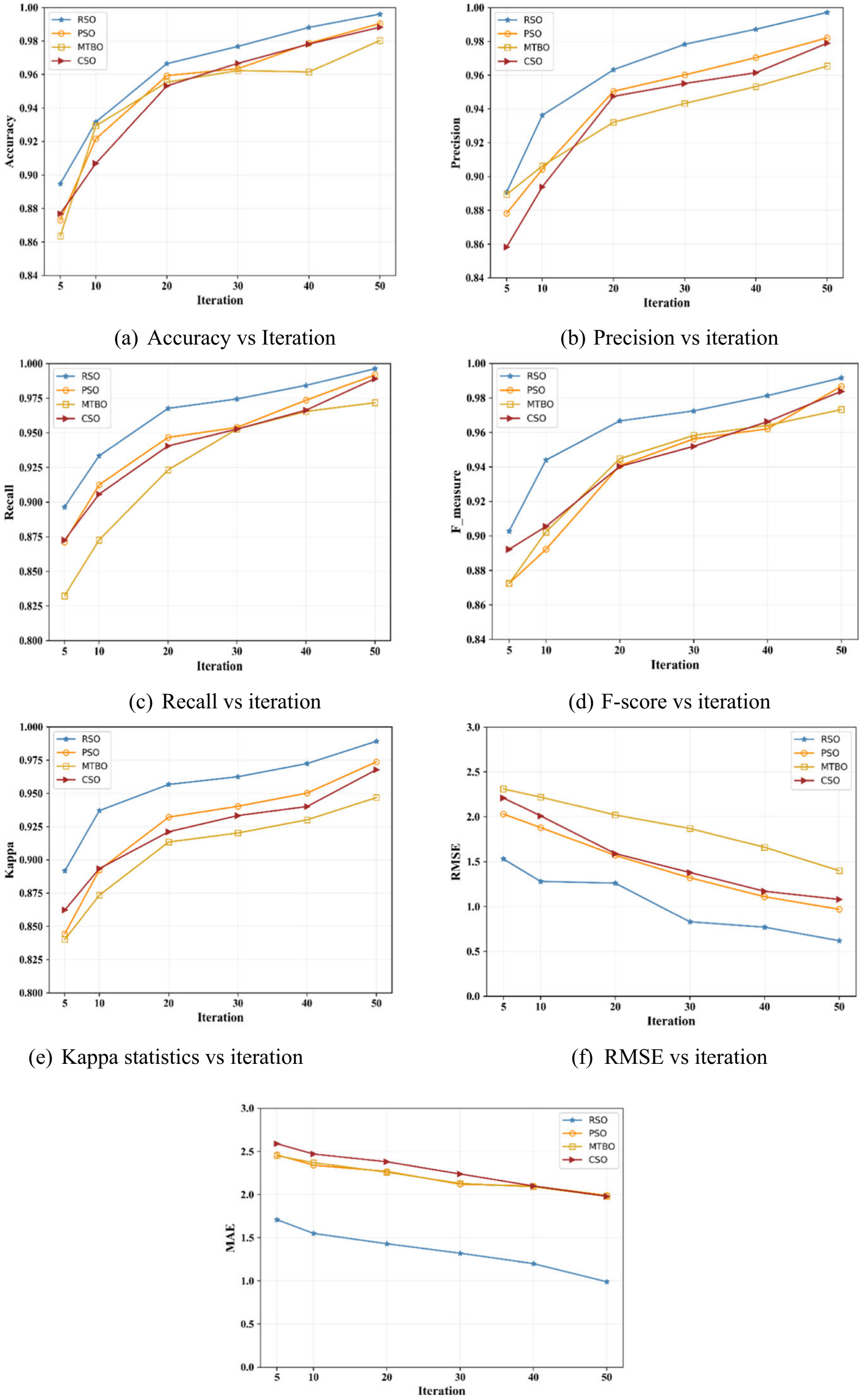
(c) Performance evaluation based on kappa statics and f-measure

Fig. 9 Comparison of different optimization algorithms

three different optimizations: PSO, MTBO, and CSO. The proposed RSO is found better than other optimization algorithms from these results. It is also observed that when the iteration is increased, the performance is also increased.

Figure 11 represents the performance evaluation of the spam assassin dataset in terms of accuracy, precision, recall, MAE, RMSE, kappa statics matrix, and F-measure value. The proposed method's result is evaluated to existing strategies such as NB, DT, LR, RF, ANN, and CNN. The proposed method achieved an accuracy of about 99.82%, higher than that of other existing algorithms.

Figure 12 represents the metrics like accuracy, precision, recall, MAE, RMSE, kappa and f-measure on Spam Assassin dataset. In the comparison, the performances like accuracy, precision, recall, kappa and f-measure are higher for proposed algorithm. Further, the proposed model attained less error values for the metrics like MAE and RMSE. The optimization-based results for different iteration values for the spam assassin dataset are shown in Fig. 13. Three other optimization algorithms, such as PSO, MTBO, and CSO, are compared to the suggested RSO optimization algorithm. Based on these findings, the suggested RSO outperforms other optimization methods.



(a) Accuracy vs Iteration

(b) Precision vs iteration

(c) Recall vs iteration

(d) F-score vs iteration

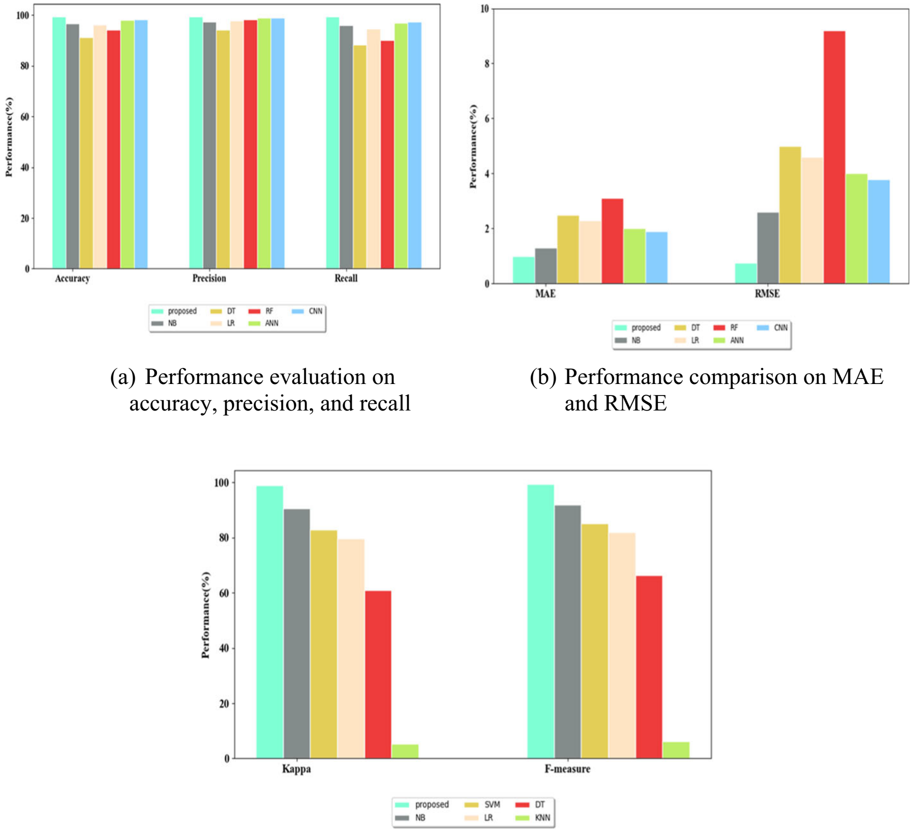
(e) Kappa statistics vs iteration

(f) RMSE vs iteration

(g) MAE vs iteration

Fig. 10 Comparison of different performance metrics for various optimization with iteration

Spam Assassin dataset



(a) Performance evaluation on accuracy, precision, and recall

(b) Performance comparison on MAE and RMSE

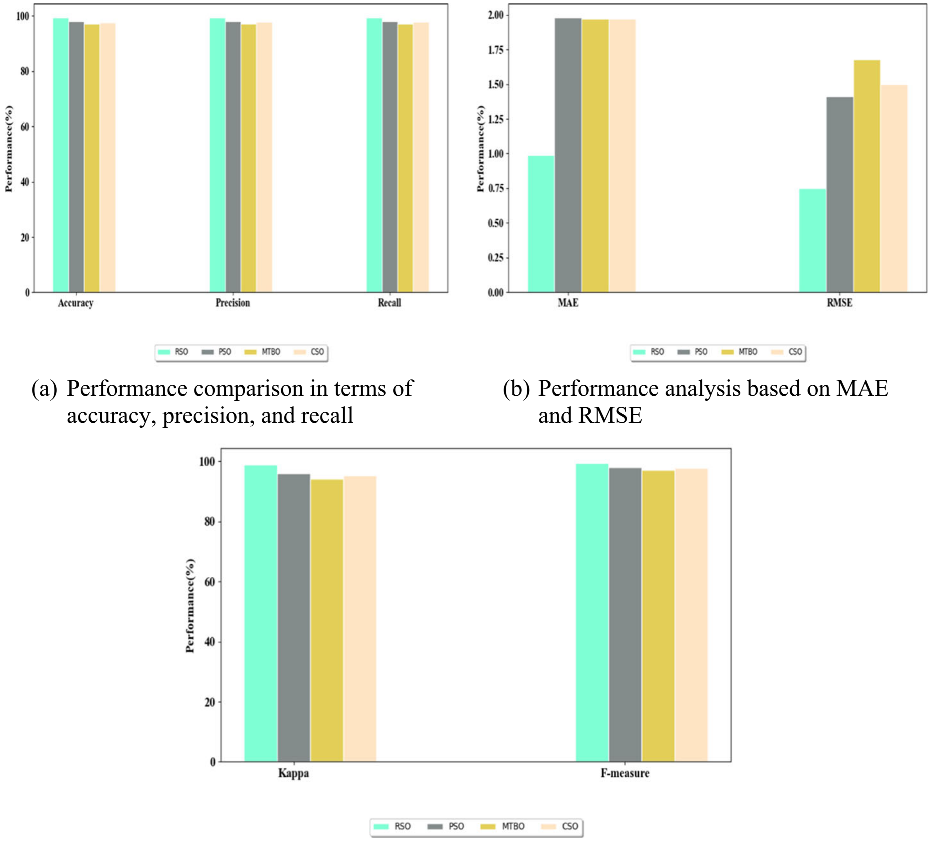
(c) Performance analysis on kappa statics and f-measure

Fig. 11 Overall Performance of hybrid KNN-SVM with RSO classifier with email dataset

4.3.3 Performance analysis for sentiment classification

Figure 14 represents the sentiment analysis on a different dataset. AFINN based and SentiWordNet is used for sentiment analysis. The accuracy vs feature size is shown in the above figure. Figure 14 **a**, **b**, and **c** represent the SMS dataset, Email dataset, and Spam assassin dataset, respectively. As the feature size increases, the sentiment analysis's accuracy also increases.

Table 7 presents the results for assessing normality of data for the hybrid KNN-SVM with RSO on the three datasets. In this work two tests like KS (Kolmogorov- Smirnov) and SW (Shapiro-wilk) are conducted. From this statistical test, it is observed that the data was distributed normally with the significance value of 0.0 for both KS test and SW test. Finally,



(a) Performance comparison in terms of accuracy, precision, and recall

(b) Performance analysis based on MAE and RMSE

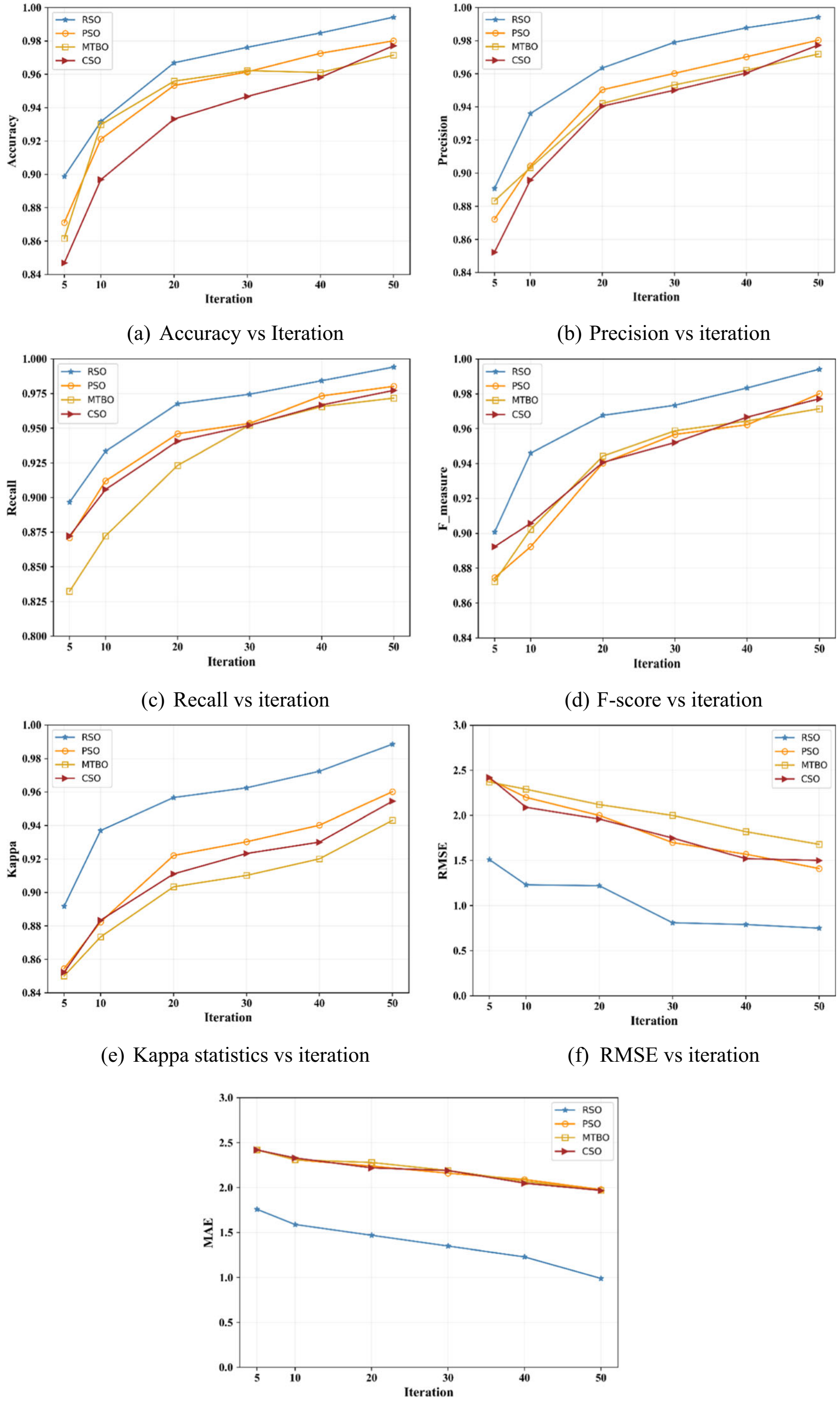
(c) Performance evaluation or kappa statics and f-measure

Fig. 12 Comparison with existing optimization algorithms

it is proved that this model is more accurate and suitable for spam SMS and sentiment classifications.

5 Conclusion

In this paper, effective SMS classification and sentiment analysis has been proposed hybrid SVM and KNN classifier with an acceptable optimization algorithm to make this procedure a fact. Spam filtering is a critical problem for safe SMS communication. The feature selection is achieved by using six techniques and EO to select optimal features. And the classification is done using a hybrid KNN-SVM classifier with an RSO algorithm. The sentiment analysis was achieved using the AFINN lexicon method and SentiWordNet. This helps to identify the



(a) Accuracy vs Iteration

(b) Precision vs iteration

(c) Recall vs iteration

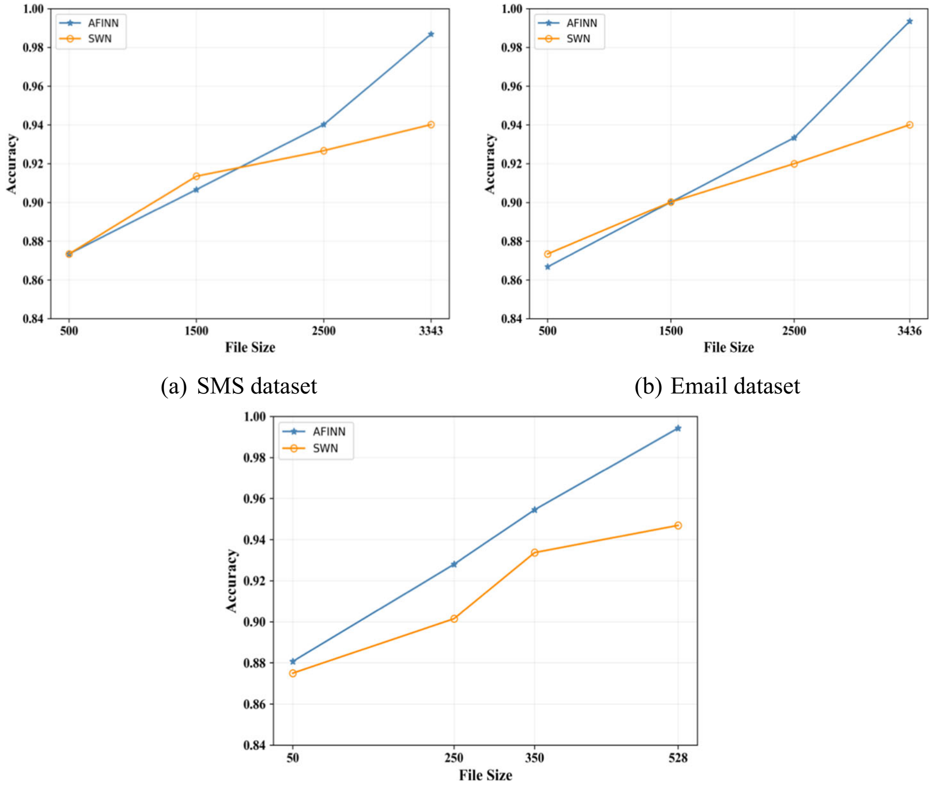
(d) F-score vs iteration

(e) Kappa statistics vs iteration

(f) RMSE vs iteration

(g) MAE vs iteration

Fig. 13 Comparison of different performance metrics for various optimization with iteration



(a) SMS dataset

(b) Email dataset

(c) Spam assassin dataset

Fig. 14 Comparison on sentiment analysis

positive and negative nature of the text. It helps to increase the accuracy. The proposed method is analyzed in the SMS, email, and spam assassin datasets. PYTHON tool is used for the implementation. The proposed model outperformed current classifiers in terms of accuracy, f-measure, precision, kappa statistics, recall, and AR value. When comparing the three datasets the spam assassin dataset achieved better spam detection accuracy of 99.82%. In the future, to improve the performance of the system the deep learning models will be utilized on large datasets.

Table 7 Test for assessing normality of data for the hybrid KNN-SVM with RSO

Datasets	KS-Test			SW-test		
	Statistic	Degree of freedom	Significance	Statistic	Degree of freedom	Significance
Spam assassin	0.5	352	0.0	0.63	352	0.0
Email	0.5	739	0.0	0.52	739	0.0
SMS	0.5	739	0.0	0.40	739	0.0

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, Odusami M (2019) A review of soft techniques for SMS spam classification: methods, approaches and applications. Eng Appl Artif Intell 86:197–212
2. Agarwal B, Ramampiaro H, Langseth H, Ruocco M (2018) A deep network model for paraphrase detection in short text messages. Inf Process Manag 54(6):922–937
3. Arivoli PV, Chakravarthy T, Kumaravelan G (2017) Empirical evaluation of machine learning algorithms for automatic document classification. Int J Adv Res Comput Sci 8(8):299–302
4. Ay Karakuş B, Talo M, Hallaç İR, Aydin G (2018) Evaluating deep learning models for sentiment classification. Concurr Comput: Prac Exp 30(21):e4783
5. Barushka A, Hajek P (2020) Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. Neural Comput Applic 32(9):4239–4257
6. Cekik R, Uysal AK (2020) A novel filter feature selection method using rough set for short text data. Expert Syst Appl 160:113691
7. Chandra A, Khatri SK (2019a) Spam SMS filtering using recurrent neural network and long short term memory. In 2019 4th international conference on information systems and computer networks (ISCON) (pp. 118-122). IEEE
8. Chandra A, Khatri SK (2019b) Spam SMS filtering using recurrent neural network and long short term memory. In 2019 4th international conference on information systems and computer networks (ISCON) (pp. 118-122). IEEE
9. Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2020) A novel algorithm for global optimization: rat swarm optimizer. Journal of ambient intelligence and humanized computing, pp.1-26
10. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:105190
11. Federici M, Dragoni M (2016) A knowledge-based approach for aspect-based opinion mining. In semantic web evaluation challenge (pp. 141-152). Springer, Cham
12. Gupta M, Bakliwal A, Agarwal S, Mehndiratta P (2018) A comparative study of spam SMS detection using machine learning classifiers. In 2018 eleventh international conference on contemporary computing (IC3) (pp. 1-7). IEEE
13. Kou G, Yang P, Peng Y, Xiao F, Chen Y, Alsaadi FE (2020) Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. Appl Soft Comput 86:105836
14. Kumar K, Kurhekar M (2017) Sentimentalizer: Docker container utility over cloud. In 2017 ninth international conference on advances in pattern recognition (ICAPR) (pp. 1-6). IEEE
15. Kumar KN, 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
16. Labani M, Moradi P, Ahmadizar F, Jalili M (2018) A novel multivariate filter method for feature selection in text classification problems. Eng Appl Artif Intell 70:25–37
17. Lall S, Sinha D, Ghosh A, Sengupta D, Bandyopadhyay S (2021) Stable feature selection using copula based mutual information. Pattern Recogn 112:107697
18. Lee HY, Kang SS (2019) Word embedding method of sms messages for spam message filtering. In 2019 IEEE international conference on big data and smart computing (BigComp) (pp. 1-4). IEEE
19. Li F, Lai L, Cui S (2020) On the adversarial robustness of feature selection using LASSO. In 2020 IEEE 30th international workshop on machine learning for signal processing (MLSP) (pp. 1-6). IEEE
20. Liu Y, Mu Y, Chen K, Li Y, Guo J (2020) Daily activity feature selection in smart homes based on Pearson correlation coefficient. Neural processing letters, pp.1-17

21. Madasu A, Elango S (2020) Efficient feature selection techniques for sentiment analysis. Multimed Tools Appl 79(9):6313–6335
22. Mendez JR, Cotos-Yanez TR, Ruano-Ordas D (2019) A new semantic-based feature selection method for spam filtering. Appl Soft Comput 76:89–104
23. Navaney P, Dubey G, Rana A (2018) SMS spam filtering using supervised machine learning algorithms. In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (confluence) (pp. 43-48). IEEE
24. Negi A, Kumar K, Chauhan P (2021) Deep neural network-based multi-class image classification for plant diseases. Agricultural Informatics: Automation Using the IoT and Machine Learning, pp.117–129
25. Ordonez A, Paje RE, Naz R (2018) SMS classification method for disaster response using Naïve Bayes algorithm. In 2018 International Symposium on Computer, Consumer and Control (IS3C) (pp. 233-236). IEEE
26. 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
27. Popovac M, Karanovic M, Sladojevic S, Arsenovic M, Anderla A (2018) Convolutional neural network based SMS spam detection. In 2018 26th telecommunications forum (TELFOR) (pp. 1-4). IEEE
28. Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter SMS spam. Futur Gener Comput Syst 102: 524–533
29. Shafi'I MA, AbdLatiff MS, Chiroma H, Osho O, Abdul-Salaam G, Abubakar AI, Herawan T (2017) A review on mobile SMS spam filtering techniques. IEEE Access 5:15650–15666
30. Sharaff A (2019) Spam detection in SMS based on feature selection techniques. In Emerging Technologies in Data Mining and Information Security (pp. 555-563). Springer, Singapore
31. Sharma S, Kumar P, Kumar K (2017a) LEXER: lexicon based emotion analyzer. In International Conference on Pattern Recognition and Machine Intelligence (pp. 373-379). Springer, Cham
32. Sharma S, Kumar K, Singh N (2017b) D-FES: deep facial expression recognition system. In 2017 Conference on Information and Communication Technology (CICT) (pp. 1-6). IEEE
33. Sharma S, Shivhare SN, Singh N, Kumar K (2019) Computationally efficient ann model for small-scale problems. In Machine Intelligence and Signal Analysis (pp. 423-435). Springer, Singapore
34. Sisodia DS, Mahapatra S, Sharma A (2020) Automated SMS classification and spam analysis using topic modeling. In 2nd International Conference on data, Engineering and Applications (IDEA) (pp. 1-6). IEEE
35. Sjarif NNA, Azmi NFM, Chuprat S, Sarkan HM, Yahya Y, Sam SM (2019) SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. Procedia Comput Sci 161: 509–515
36. Su YJ, Hu WC, Jiang JH, Su RY (2020) A novel LMAEB-CNN model for Chinese microblog sentiment analysis. J Supercomput:1–15
37. Suleiman D, Al-Naymat G (2017) SMS spam detection using H2O framework. Procedia Comput Sci 113: 154–161
38. Xia T (2020) A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems. IEEE Access 8:82653–82661
39. Zainal K, Jali MZ (2016) A review of feature extraction optimization in SMS spam messages classification. In: International Conference on Soft Computing in data Science (pp. 158-170). Springer, Singapor.

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