IJCAT - International Journal of Computing and Technology, Volume 7, Issue 6, June 2020
ISSN (Online) : 2348-6090
www.IJCAT.org
Impact Factor: 0.835

# SMS Spam Detection System Usinag Effective One-Dimensional Ternary Pattern (1D-TP)

**[1] Taofeek-Ibrahim Fatimoh Abidemi; [2] Oluwakemi Christiana Abikoye; [3] Toye Nike Toyin**

[1,3] Department of Computer Science, Federal Polytechnic Offa
Kwara State, Nigeria

[2] Department of Computer Science, University of Ilorin
Ilorin, Ilorin, Nigeria

**Abstract -** Short Message Service (SMS) have been identified to be a fast communication approach due to its low cost and stress-free nature. The general acceptability of SMS has exposed it to many threats and one of these is spamming. In SMS spam detection, the feature extraction plays a vital role, the extraction of SMS features involves a process of decreasing an initial set of raw features into more measurable forms before the classification. This paper considered the improvement of One Dimensional Ternary Pattern (1D-TP) by the introduction of nature-inspired optimization algorithm known as simulated annealing to obtain optimized features. Seven machine learning algorithms; Bayesian Network (BN), Naïve Bayes (NB), Radial Basic Artificial Neural Network (RBFN), Random Forest (RF), K-nearest neighbours (KNN), Logistic Regression (LR), and Support Vector Machine (SVM) were employed for classification. The developed was evaluated using Kaggle SMS Spam Dataset. Experimental results showed that the highest accuracy of 86.56% was obtained in LR for non-optimized upper features and the highest accuracy of 92.56% was recorded in RF for optimized upper features. The highest precision of 0.95 was achieved in BN for non-optimized upper features of 1D-TP and the highest precision of 0.94 was obtained in optimized upper features of 1D-TP. The highest recall of 1.00 was obtained in NB, SVM and LR for non-optimized lower/upper features and the highest recall of 1.00 was recorded in NB and RF for lower /upper features of 1D-TP.

**Keywords:** *Accuracy, Feature Extraction, Short Message Service, Spam SMS, One Dimensional Ternary Pattern*

.

## 1. Introduction

Short Message Service (SMS) is one of the common communication services used in disseminating information to a large number of people through mobile phones or internet-connected computers, where messages transmission happen through communication standard protocols [1]. SMS is beyond the conventional method of testing, it is now being applied in different domains for authentication like mobile banking, one-time password delivery, information retrieval systems, smartphone configuration, Over-The-Air (OTA) configuration and alerts of social web site [2]. SMS technology was developed out of the Global System for Mobile communication GSM, the communication standard, an internationally accepted cell phone network specification [3]. SMS is employed globally, because of the high response rate, cheapness and personal service [4]. The high demand in the use of SMS has unnecessarily attracted spammers to send spam SMS called a junk SMS and which also leads to SMS spam message problem [5]. SMS problem is increasing every day with a high increase in the use of text messaging. In the telecommunication, smartphone devices era, users now have personal and confidential information such as contact lists, credit card numbers, photographs, password and much more stored in their devices making them prone to cyber-attacks through SMS spam [6]. This allows hackers to perform unethical activities to access smartphone data without the knowledge of end-user, thus compromising the user's privacy. Spam messages appear to be rising and cause not only user's annoyance but critical data loss for some users. Apart from this, SMS spam can also act as a driving force for malware and key-loggers. Several problems have been traceable to SMS in the world of communication despite the diverse advantages connected with SMS [7]. SMS Spam is referred to as an undesirable SMS usually carried to a large number of recipients [8]. A Spam is usually sent in bulk for business advertisements and other determinations [9].

Nowadays, the bulk of SMS messages received by mobile phone users are unfortunately the upsetting spam messages such as bank's credits, new tariffs of communications service providers and promotion and discount announcements of stores [10]. SMS spamming has gained much attention over other spamming methods such as email due to the increasing demand for communication

using SMS [11]. It has instituted great trouble to the mobile phone subscribers given its pervasive nature. It causes loss of productivity subject to substantial cost, usage of high bandwidth and invasion of personal privacy. SMS spam results to the frustration of the mobile phone users just like email spam, which leads to new societal frictions to mobile handset devices.

In SMS spam detection, the feature extraction remains one of the predominant phase [12]. The extraction of feature extraction involves a process of reducing the initial set of original features. [13]. The reduction produces approximate to original feature in fewer dimensions, while still retaining the same structure of original features [14]. If SMS spam features are not adequately represented, it may result in poor accuracy and misclassification during detection. Several feature types in SMS spam filtering have been encountered in different approaches, which have resulted at a different level of accuracies.

This paper developed SMS spam detection system by improving upon one-dimensional ternary pattern feature extraction algorithm through the introduction of simulated annealing as a meta-heuristic optimization algorithm. Also, seven different classifiers; Bayesian Network (BN), Naïve Bayes (NB), Radial Basic Artificial Neural Network (RBFN), Random Forest (RF), K-nearest neighbours (KNN), Logistic Regression (LR), and Support Vector Machine (SVM) were used to classify the SMS to either Ham or Spam.

## 2 Related Work

[15] proposed a novel classifier which depended majorly on $H_20$ as a platform to conduct comparisons between different machine learning algorithms. The machine learning algorithms used for comparisons were Random Forest, Deep Learning and Naïve Bayes. Results showed that the important features that could affect the detection of SMS spam were the number of digits and existing URL in SMS text. The dataset used for the evaluation of the models was obtained from UCI Machine Learning Repositories. The experiment analyses showed that Naïve Bayes achieved faster runtime of 0.6 seconds with high performance. When compared with deep learning and random forest it has the lowest precision, recall, f-measure and accuracy. In term of accuracy, the random forest was considered the best with 50 trees and 20 maximum depths, where 96% of precision, 86% of recall, 91% off-measure and 0.977% of accuracy respectively.

[16] introduced a model to handle the classification problem of messages as either spam or ham by experimenting and analyzing the relative strengths of some machine learning algorithms; K-Nearest Neighbours (KNN), Decision Tree Classifier, Random Forest Classifier, Logistic Regression, SGD Classifier, Multinomial Naive Bayes (NB), Support Vector Machine (SVM) to have a logical comparison of the performance measures of the methods. The proposed model achieved an average accuracy of 98.49% with SVM model on 'SMS Spam Collection' dataset.

[6] evaluated machine learning techniques for spam SMS detection. A comparative study was conducted among eight different classifiers; Artificial Neural Network (ANN), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Naïve Bayes (NB), Logistic Regression (LR), Random Forest (RF) and AdaBoost. The two datasets; SMS Spam collection v.1 and Spam SMS Dataset were used to test the efficiency of the developed model. Results obtained from the evaluation of the classifiers showed that CNN achieved the highest accuracy of 99.19% and 98.25% and AR value of 0.9926 and 0.9994 for the two datasets.

[17] came up with a spam filter for Arabic and English languages by using two filters to detect SMS spam efficiently. The study applied a content-based approach to building a spam filter for English and Arabic languages. Several steps were considered which included; Read English and Arabic dataset, Preprocessing phase, Feature Extraction and Classification. The features were pre-processed, then the extraction of features was performed. Eight features were extracted from English messages and six features from Arabic messages. The features of messages for English and Arabic languages were divided into two sets: a training set and testing set. The training set was used to train the algorithms while the test set was applied to evaluate the performance of proposed Spam filter for the English and Arabic language. Two classifiers; Naive Bayes and Neural Network were employed for classification.

The results of the proposed system recorded an accuracy of 97% for the English language when using eight features and 80% from the dataset for training. Accuracy of 95% was obtained for the Arabic language with six features. [18] enhanced the accuracy by the extraction of more header features. The adaptive and collaborative was conducted using machine learning and cluster computing for fast classification of emails to either "Spam" or "Ham". The adaptive technique was used to create new rules for classification and cluster approach for parallel computing power to increase computational speed. The evaluation of the developed detection model was done

using Spam Assassin dataset. The collaborative approach produced a parallel environment where multiple anti-spam techniques and divided test corpora were used as input. The false-positive and false-negative percentage were obtained and accuracy was computed. [19] presented a general model that differentiated and filtered the spam messages applying some existing classification algorithms in machine learning. The model built a generalized SMS spam-filtering model, which filtered messages from various backgrounds (Singapore, American, Indian English and so on). In the model, preliminary results were mentioned based on Singapore and Indian English based publicly available datasets. The proposed approach gave a high precision utilizing Indian English SMS large datasets and other background's datasets too. [20] developed a novel approach for feature extraction in SMS spam filtering.

One-dimensional ternary patterns were applied to extract features from SMS messages. In the first phase, the text message was transformed into UTF-8 values. In the second phase, each character (its UTF-8 value) in the message NB), Radial Basis Feed-forward Neural Network (RBFNN), K-Nearest Neighbours(KNN), and RF were used for classification. The system was evaluated with three different SMS corpora datasets. SMS Spam Corpus v.0.1 (DS1), British English SMS Corpora (DS2) and DS3. The achieved accuracies and other employee performance measures showed that the proposed approach, one-dimensional ternary patterns can be effectively employed in SMS spam filtering. [21] reviewed critically the existing SMS spam filters by identifying and analyzing their problems. Some of these problems are adaptability to spammers' concept drift, SMS flooding on the network, overhead during training and testing; memory and computational robustness. A taxonomy for existing SMS Spam filtering methods was constructed. The study concluded by recommending the use of an adaptive and collaborative SMS spam filtering system. The study concluded by proposing an adaptive and collaborative server-side SMS spam filtering solution with an intrusion detection mechanism to ensure the privacy and security of was compared with its neighbours. Five machine learning techniques; Bayesian Network(BN), Naïve Bayes(the users. The adaptive nature is to enable it to adapt to spammer's concept drift such as misspellings, compounding, hyphenation, toggles

## 3. Methodology

The SMS Spam detection model was developed to improve upon the 1D-TP feature extraction approach. The Kaggle SMS Spam dataset a publicly available online dataset was employed to evaluate the SMS Spam detection model. The data was pre-processed to remove unwanted characters by stemming and conversion of a message to UTF-8 values of characters in the text using python function. The pre-processed data was passed into ID-TP feature extraction algorithm. Simulated Annealing was applied to optimize the ID-TP extracted features through parameters setting. Optimized features then passed to seven different machine learning algorithms for classification. The framework of the detection model is shown in Figure 1.

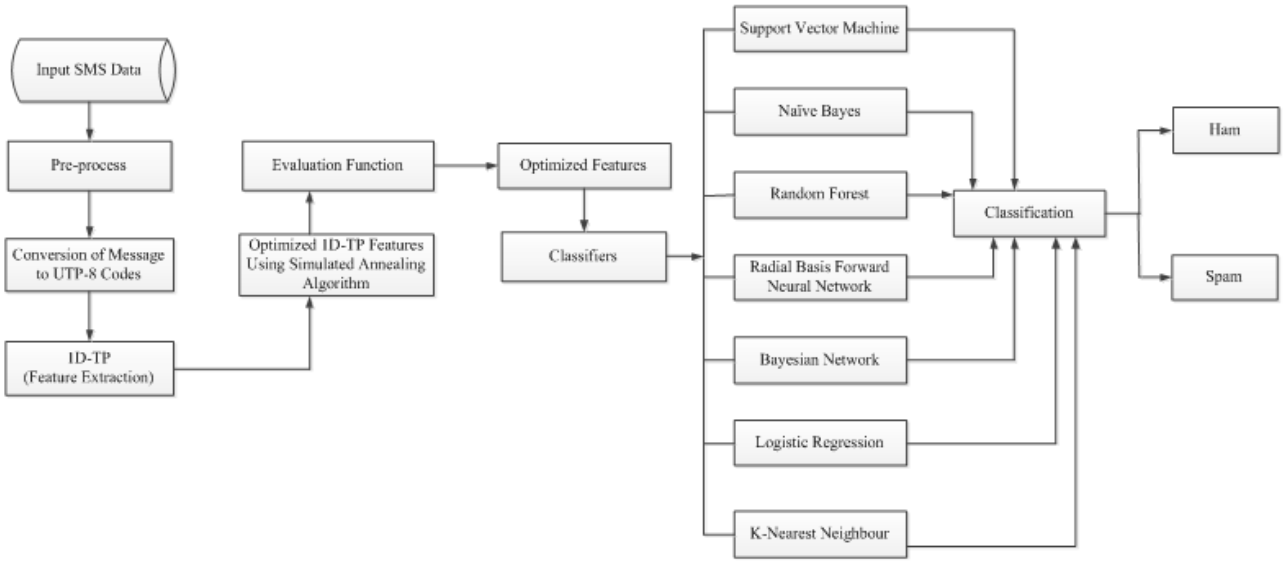


Figure 1: Frame of the developed SMS Spam Detection System

### 3.1 Dataset Collection

The dataset comprises of SMS messages in English of 5,574 messages, labelled to be either legitimate (ham) or spam. The files contain one message per line. Each line is composed of two columns: v1 contains the label (ham or spam) and v2 contains the raw text. The messages are classified as Spam or Ham using NLTK and Scikit-learn.

### 3.2 Data Pre-processing

The pre-processing transformed messages in SMS into a uniform format so that it can be understood by the learning algorithm. Removing of stop words, words lesser than or equal to two, performing stemming to reduce the vocabulary and converting the remaining part of the message to UTF-8 values of characters in the text, are the basic preprocessing steps were carried out in this study. Stemming was done using the Porter Stemming Algorithm. The algorithm of the Pre-process phase is shown in Figure 2.

**Step 1:** Start
**Step 2:** Perform stemming

```
/*Porter's algorithm*/
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
ps = PorterStemmer()
```

**Step 3:** Convert the message to UTF-8 values of the characters in the text
**Step 4:** Output the UTF-8 values of the SMS message
**Step 5:** End

Figure 2: Algorithm of the Pre-process

### 3.4 One Dimensional-Ternary Pattern (1D-TP) Algorithm

In 1D-TP, the patterns are formed from the comparisons of Unicode values of the characters in SMS messages with the Unicode values of their neighbours. The value of each member of the 1-D series was compared with its neighbours, and the result of these comparisons was expressed as a decimal number. For text messages, the comparisons were carried out after converting the characters to their UTF-8 values. The algorithm of 1D-TP is shown in Figure 3.

**Step 1:** Start
**Step 2:** Determine the number of neighbours of a character by the parameter P
**Step 3:** Assign P/2 characters previously and after of central character, $Pc$ as neighbours of that character
**Step 4:** Compare the value of each member of the 1-D series with its neighbour using the following equation:

$$TP = \begin{cases} 1 & Pc > Pi + \beta \\ 0 & Pc \leq Pc + \beta \text{and} Pc \geq Pi + \beta \\ -1 & P < Pi - \beta \end{cases}$$

Where pc is the central character, $\beta$ threshold parameter, which is a user-defined parameter. The value of $Pi, local\ changes,$ is within $Pc_{\pm}\beta$
**Step 5:** The value of the central character (Pc) is adjusted according to ±β value.
/* Therefore, for each Pc, its neighbours are filtered depending on β*/
**Step 6:** Two different decimal numbers are generated from the comparison results for each $Pc$ Negative ones $(Pc < Pi - \beta)$ are employed to generate low features Positive ones $(Pc < Pi + \beta)$ are used to extract up features
**Step 7:** Obtained decimal numbers (up-low) are used instead of the UTF-8 values of he text Messages.
**Step 8:** Process continues for each data in the message
**Step 9:** Two different 1D signals will be obtained from the up and low values
**Step 10:** Histograms which are upper and lower histograms are formed from the 1D signals
**Step 11:** End

Figure 3: Algorithm of 1D-TP

### 3.5 Simulated Annealing Optimization

This stage involves the action of finding the optimal (best or most effective) values of an objective function (1D-TP transformation process on an SMS) using Simulated Annealing Algorithm. This algorithm allows a variation in both ρ and β to detect different patterns (1D-TP) on the SMS characters from which the optimal values for them are being determined after undergoing several trials (iterations); it searches for the best solution by generating a random initial solution and "exploring" the area nearby. If a neighbouring solution is better than the current one, then it moves to it. If not, then the algorithm stays put. The algorithm of simulated annealing is shown in Figure 4.

**Step 1:** Start
**Step 2:** Set initial value for the maximum possible parameter called the "temperature", as Tmax
**Step 3:** Set initial value for the minimum possible parameter called the "temperature", as Tmin
**Step 4:** Set initial value for the maximum possible Iteration, as MaxIt
**Step 5**: Set the initial value for the total number of utilized neighbours of characters, i.e. P as X1
**Step 6**: Set the initial value for the threshold parameter, i.e. Beta (B) as X2
**Step 7**: Start a global Iteration, at T = Tmax and alpha = random(0, 1)
**Step 8**: Start a new Local Iteration, at i = 1
Step 9: Compute the cost function (1D-TP transformation of a given SMS message with the current values of parameters X1 and X2) as, E = Cost(X1, X2)
**Step 10:** Generate a random neighbouring solution of X1 and X2 as Next_X1 and Next_X2 respectively
**Step 11:** Compute the cost function (1D-TP transformation of a given SMS message with the next values of parameters Next_X1 and Next_X2) as, E_Next = Cost(Next_X1, Next_X2)
**Step 12:** Evaluate the change in cost as delta_E = E_Next – E
**Step 13:** if (delta_E < acceptance_treshold_value) then, move to the next solution by accepting the new values of X1 and X2 as, set X1 = Next_X1 and set X2 = Next_X2, goto step 14 otherwise goto step 13 if (Exp(delta_E, T) > random(0,1)) then, move to the next solution by accepting the new values of X1 and X2 as, set X1 = Next_X1 and set X2 = Next_X2
**Step 14:** Increment the value of i by 1
**Step 15:** Check if the current iteration is the last iteration, i.e. is i <= MaxIt then goto step 8 otherwise continue on the next step
**Step16:** Update the value of T as, set T = alpha * T
**Step 17:** Check if the current T is out of the region (Tmax, Tmin), i.e. is T >= Tmin then goto step 7 otherwise continue on the next step
**Step 18:** Output the best solution obtained for X1 and X2 as P and B respectively
**Step 19:** End

Figure 4: Simulated Annealing Algorithm

## 4. Results and Discussion

The results of the evaluation parameters; accuracy, recall and precision were conducted using Kaggle SMS dataset as shown in the following subsections:

### 4.1 Results of Accuracy

The different results were obtained for without optimization algorithm and with optimized 1D-TP features using simulated annealing as shown in Table 1 and 2.

Table 1: 1D-TP without Optimization Algorithm (Accuracy)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 85.79 | 84.54 | 84.81 | 85.32 | 85.25 |
| | BN | 74.14 | 75.45 | 73.60 | 80.15 | 71.91 |
| | RBFN | 81.64 | 81.71 | 82.25 | 82.89 | 84.15 |
| | KNN | 83.25 | 83.94 | 83.97 | 84.22 | 83.60 |
| | RF | 85.63 | 85.32 | 85.73 | 85.49 | 86.38 |
| | SVM | 86.24 | 85.18 | 86.06 | 85.63 | 86.24 |
| | LR | 85.58 | 85.46 | 86.20 | 86.06 | 86.38 |
| Upper Features | NB | 85.58 | 85.32 | 86.30 | 86.15 | 86.19 |
| | BN | 80.36 | 79.29 | 81.11 | 80.15 | 79.69 |
| | RBFN | 84.25 | 82.99 | 84.97 | 85.73 | 85.57 |
| | KNN | 84.30 | 85.13 | 85.87 | 85.16 | 85.43 |
| | RF | 85.44 | 84.89 | 85.25 | 85.87 | 85.86 |
| | SVM | 85.67 | 85.27 | 86.20 | 86.06 | 86.33 |
| | LR | 85.82 | 86.56 | 86.35 | 86.06 | 86.33 |

From Table 1, the highest accuracy of 86.56% was recorded in LR for non-optimized upper features of 1 D-TP. The lowest accuracy of 71.91% was obtained in BN for non-optimized lower features of 1D-TP.

Table 2: 1D-TP with Simulated Annealing Optimization Algorithm (Accuracy)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 86.46 | 86.46 | 86.46 | 86.49 | 86.52 |
| | BN | 83.62 | 83.95 | 85.01 | 84.58 | 84.10 |
| | RBFN | 85.97 | 85.28 | 84.64 | 86.76 | 86.46 |
| | KNN | 79.59 | 81.02 | 83.40 | 84.10 | 85.04 |
| | RF | 90.72 | 90.57 | 90.45 | 90.30 | 90.15 |
| | SVM | 89.57 | 89.57 | 89.60 | 90.33 | 90.33 |
| | LR | 87.30 | 87.88 | 87.55 | 87.85 | 87.79 |
| Upper Features | NB | 86.94 | 86.94 | 86.94 | 86.94 | 86.94 |
| | BN | 90.16 | 89.75 | 90.61 | 89.89 | 88.39 |
| | RBFN | 90.07 | 89.89 | 89.39 | 89.89 | 90.20 |
| | KNN | 87.17 | 89.02 | 89.25 | 90.52 | 90.79 |
| | RF | 92.56 | 92.02 | 91.61 | 91,52 | 91.70 |
| | SVM | 90.07 | 89.89 | 90.52 | 89.93 | 90.02 |
| | LR | 89.57 | 89.75 | 90.07 | 89.71 | 90.29 |

From Table 2, the highest accuracy of 92.56% was recorded in RF for optimized upper features of 1D-TP. The lowest accuracy of 79.59% was obtained in KNN for optimized lower features of 1D-TP.

## 4.2 Results of Precision

The different precisions were obtained for without optimization algorithm and with optimized 1D-TP features using simulated annealing as shown in Table 3 and 4

Table 3: 1D-TP without Optimization Algorithm (Precision)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 0.86 | 0.86 | 0.86 | 0.86 | 0.86 |
| | BN | 0.90 | 0.92 | 0.92 | 0.92 | 0.92 |
| | RBFN | 0.89 | 0.91 | 0.92 | 0.91 | 0.91 |
| | KNN | 0.87 | 0.88 | 0.88 | 0.88 | 0.88 |
| | RF | 0.91 | 0.90 | 0.90 | 0.90 | 0.90 |
| | SVM | 0.90 | 0.91 | 0.91 | 0.91 | 0.92 |
| | LR | 0.89 | 0.90 | 0.90 | 0.90 | 0.90 |
| Upper Features | NB | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| | BN | 0.94 | 0.94 | 0.95 | 0.94 | 0.94 |
| | RBFN | 0.93 | 0.92 | 0.94 | 0.92 | 0.92 |
| | KNN | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | RF | 0.92 | 0.92 | 0.91 | 0.91 | 0.91 |
| | SVM | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | LR | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 |

In Table 3, the highest precision of 0.95 was recorded in BN for non-optimized upper features of 1D-TP. The lowest precision of 0.86 was obtained in NB for non-optimized lower features of 1D-TP.

Table 4: 1D-TP with Simulated Annealing Optimization Algorithm (Precision)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 0.86 | 0.86 | 0.86 | 0.86 | 0.87 |
| | BN | 0.90 | 0.92 | 0.92 | 0.92 | 0.92 |
| | RBFN | 0.89 | 0.91 | 0.92 | 0.91 | 0.91 |
| | KNN | 0.87 | 0.88 | 0.88 | 0.88 | 0.88 |
| | RF | 0.91 | 0.90 | 0.90 | 0.90 | 0.90 |
| | SVM | 0.91 | 0.90 | 0.91 | 0.91 | 0.92 |
| | LR | 0.89 | 0.90 | 0.90 | 0.90 | 0.90 |
| Upper Features | NB | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| | BN | 0.94 | 0.94 | 0.95 | 0.94 | 0.94 |
| | RBFN | 0.93 | 0.92 | 0.94 | 0.92 | 0.92 |
| | KNN | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | RF | 0.92 | 0.92 | 0.91 | 0.91 | 0.91 |
| | SVM | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | LR | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 |

In Table 4, the highest precision of 0.94 was recorded in BN and RBFN for optimized upper features. The lowest precision of 0.86 was obtained in NB for optimized lower features.

### 4.3 Results of Recall

The different results were obtained for without optimization algorithm and with optimized 1D-TP using simulated annealing as shown in Table 5 and 6.

Table 5: 1D-TP without Optimization Algorithm (Recall)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | BN | 0.83 | 0.75 | 0.86 | 0.82 | 0.83 |
| | RBFN | 0.93 | 0.94 | 0.94 | 0.96 | 0.96 |
| | KNN | 0.97 | 0.98 | 0.97 | 0.98 | 0.96 |
| | RF | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 |
| | SVM | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 |
| | LR | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Upper Features | NB | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | BN | 0.90 | 0.89 | 0.90 | 0.89 | 0.88 |
| | RBFN | 0.89 | 0.94 | 0.97 | 0.98 | 0.98 |
| | KNN | 0.98 | 0.99 | 0.99 | 0.98 | 0.98 |
| | RF | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | LR | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

In Table 5, the highest recall of 1.00 was recorded in NB, SVM and LR for non-optimized lower and upper features. The lowest recall of 0.75 was obtained in BN for non-optimized lower features.

Table 6: 1D-TP with Simulated Annealing Optimization Algorithm (Recall)

| Features | Model | $\beta = 0$ | $\beta = 1$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ |
|---|---|---|---|---|---|---|
| Lower Features | NB | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | BN | 0.91 | 0.90 | 0.90 | 0.90 | 0.89 |
| | RBFN | 0.95 | 0.92 | 0.90 | 0.93 | 0.93 |
| | KNN | 0.89 | 0.91 | 0.93 | 0.95 | 0.96 |
| | RF | 1.00 | 1.00 | 1.00 | 0.95 | 1.00 |
| | SVM | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 |
| | LR | 0.97 | 0.97 | 0.96 | 0.97 | 0.97 |
| Upper Features | NB | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | BN | 0.94 | 0.94 | 0.94 | 0.94 | 0.93 |
| | RBFN | 0.97 | 0.96 | 0.94 | 0.97 | 0.97 |
| | KNN | 0.94 | 0.95 | 0.96 | 0.97 | 0.98 |
| | RF | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SVM | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| | LR | 0.98 | 0.98 | 0.98 | 0.98 | 0.99 |

From Table 6, the highest recall of 1.00 was recorded in NB and RF for optimized lower features and upper features of 1D-TP. The lowest recall of 0.89 was obtained in KNN for optimized lower features of 1D-TP.

## 5. Conclusion

The upturn in mobile communication technology produces numerous effects on people. One of the usefulness is communication through SMS, which is one of the most prevalent communication methods nowadays. Similar to other general communication techniques, SMS also suffers from spam messages. In SMS Spam detection, the feature extraction is a core step which precedes the classification stage. This paper optimized 1D-TP feature extraction for SMS Spam filtering system by the application of nature-inspired optimization algorithm known as simulated annealing. The results obtained from the optimized 1D-TP features outperformed the non-optimized 1D-TP features for SMS spam detection. Thus showed that the proposed system is an effective approach for SMS Spam filtering.

## References

[1] P. Nivaashini, M., Soundariya, R. S, Kodiieswari, A. & Thangaraj, "SMS Spam Detection Using Neural Network Classifier," *Int. J. Pure Appl. Math.*, vol. 119, no. 18, pp. 2425–2436, 2018.

[2] A. Mizuki, T. Matsumoto, T. Uemura, and S. Kichimi, "Improving SMS Processing Power for the Increasing Smartphone Demand," *NTT DOCOMO Tech. J.*, vol. 14, no. 4, pp. 60–62, 2013.

[3] V. K. Katankar, "Short Message Service using SMS Gateway," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 5, pp. 1487–1491, 2010.

[4] N. Choudhary and A. K. Jain, "Towards filtering of SMS Spam Messages using Machine Learning-Based Technique," in *Communications in Computer and Information Science*, 2017, vol. 712, pp. 18–30.

[5] W. N. Gansterer, A. G. K. Janecek, and R. Neumayer, "Spam filtering based on latent semantic indexing," in *Survey of Text Mining II: Clustering, Classification, and Retrieval*, 2008, pp. 165–183.

[6] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers," in *2018 11th International Conference on Contemporary Computing, IC3 2018*, 2018, pp. 1–7.

[7] M. Abdulhamid, M. Shafie, A. Latiff, H. Chiroma, and O. Osho, "A Review on Mobile SMS Spam Filtering Techniques A Review on Mobile SMS Spam Filtering Techniques," no. February, 2017.

[8] T. M. & Mahmoud and A. M. Mahfouz, "SMS Spam Filtering Technique Based on Artificial Immune System," *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 589–597, 2012.

[9] N. Chaudhari, P. Jayvala, and P. Vinitashah, "Survey on Spam SMS filtering using Data mining Techniques," *Ijarcce*, vol. 5, no. 11, pp. 193–195, 2016.

[10] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "The Impact of Feature Extraction and Selection on SMS Spam Filtering," *Elektron. IR ELEKTROTECHNIKA*, vol. 19, no. 5, pp. 67–72, 2014.

[11] T. Subramaniam, H. A. Jalab, and A. Y. Taqa, "Overview of textual anti-spam filtering techniques," *Int. J. Phys. Sci.*, vol. 5, no. 12, pp. 1869–1882, 2010.

[12] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "The impact of feature extraction and selection on SMS spam filtering," *Elektron. ir Elektrotechnika*, vol. 19, no. 5, pp. 67–72, 2013.

[13] R. Kaur and R. Rajput, "Face recognition and its various techniques : a review," *Int. J. Sci. Eng. Technol. Res.*, vol. 2, no. 3, pp. 670–675, 2013.

[14] S. Telgaonkar, A. H & Deshmukh, "Dimensionality Reduction and Classification through PCA and LDA," *Int. J. Comput. Appl.*, vol. 122, no. 17, pp. 4–8, 2015.

[15] D. Suleiman and G. Al-Naymat, "SMS Spam Detection using H2O Framework SMS Spam Detection using H2O Framework," in *Procedia Computer Science*, 2017, vol. 113, pp. 154–161.

[16] M. Ramabai, "Spam Detection using NLP Techniques," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 2423–2426, 2019.

[17] H. H. Mansoor and S. H. Shaker, "Using classification techniques to SMS spam filter," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 1734–1739, 2019.

[18] A. S. Rajput, J. S. Sohal, and V. Athavale, "Email Header Feature Extraction using Adaptive and Collaborative approach for Email Classification," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 158–164, 2019.

[19] R. K. Kaliyar, P. Narang, and A. Goswami, "SMS Spam Filtering on Multiple Background Datasets Using Machine Learning Techniques: A Novel Approach," in *Proceedings of the 8th International Advance Computing Conference, IACC 2018*, 2018, pp. 59–65.

[20] O. F. Kaya, Y., & Ertugrul, "A Novel Feature Extraction Approach in SMS Spam Filtering for Mobile Communication: One-Dimensional Ternary Patterns," *Secure. Commun. Networks*, vol. 9, pp. 4680–4690, 2016.

[21] O. O. Abayomi-alli, S. A. Onashoga, and A. S. Sodiya, "A Critical Analysis of Existing SMS SPAM Filtering Approaches," in *1st International Conference on Applied Information Technology*, 2015, pp. 211–220.