An Improved Machine Learning-Based Short Message Service Spam Detection System

Odukoya Oluwatoyin, Akinyemi Bodunde, Gooding Titus and Aderounmu Ganiyu,

Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria,

Summary
The use of Short Message Services (SMS) as a mechanism of communication has resulted to loss of sensitive information such as credit card details, medical information and bank account details (user name and password). Several Machine learning-based approaches have been proposed to address this problem, but they are still unable to detect modified SMS spam messages more accurately. Thus, in this research, a stack- ensemble of four machine learning algorithms consisting of Random Forest (RF), Logistic Regression (LR), Multilayer Perceptron (MLP), and Support Vector Machine (SVM), were employed to detect more accurately SMS spams. The simulation was carried out using Python Scikit- learn tools. The performance evaluation of the proposed model was carried out by benchmarking it with an existing model. The evaluation results showed that the proposed model has an increase of 3.03% of accuracy, 8.94% of Recall, 2.17% of F-measure; and a decrease of 4.55% of Precision over the existing model. In conclusion, the ensemble method performed better than any individual algorithms and can be adopted by the Network service providers for better Quality of Service.

Keywords:
Short Message Service (SMS), Consistency-based features selection, Ensemble method, Machine Learning

1. Introduction

Advancement in technology has made end-users to access their emails, surf the World Wide Web, make video and voice calls, use text chatting, make a medical appointment, gaming and more through their smartphones. It has become a medium of advertisement and promotion of products, banking updates, agricultural information, flight updates and internet offers. SMS is also employed in direct marketing known as SMS marketing. Sometimes SMS marketing is a matter of disturbance to users. These kinds of SMSs are called spam SMS. Spam is one or more unsolicited messages, which is unwanted to the users, sent or posted as part of a larger collection of messages, all having substantially identical content. The purposes of SMS spam are advertisement and marketing of various products, sending political issues, spreading inappropriate adult content and Internet offers and phishing attack among others, hence spam SMS flooding has become a serious problem all over the world. SMS spamming gained popularity over other spamming approaches like email and Twitter, due to the increasing popularity of SMS communication. Mobile devices now contain personal and confidential information such as credit card numbers, contact lists, emails, medical records and other sensitive documents. The issue of SMS spam can be attributed to limited resources and processing power and lack of knowledge and consciousness to end-users regarding security threats. The aforementioned reasons make mobiles very eye-catching to cyber-attacks. Hackers can utilize the compromised mobiles to make calls to premium numbers without the end-users' permission, stealing contact data, or participating in botnet activities [1]. The exchange of SMS among different mobile phones is very suitable and regularly used for communication on a day to day basis. Then, the number of spam SMS messages is growing. In 2012, there were 350,000 variants of SMS spam globally [2]. SMS has been considered a serious security threat since the early 2000s [3]. For instance, hackers can send phishing attacks to collect confidential information or launch other types of attacks.

The risk of SMS spam could lead to operational or financial loss. It is getting easier to attack end-users through SMS than emails, since the mail service is more secure, and more effective email spam filter has been developed and installed by mail service providers and users. In this light, it is not the case of SMS spam. It is a challenging task since SMS messages have limited sizes (160 characters long) which means less statistically-distinguishing information. Several methods have been investigated to detect SMS spam, including H2O approaches [4]. In this light, the accuracy is still relatively low and further research is required to investigate new features and new ways of tackling this. In this research work, several feature sets and their impact on four machine learning algorithms were studied. The rest of this paper is organized as follows. Section 2 covers the literature review while section 3 describes the methodology and presents the proposed architecture, section 4 discusses the performance metrics and section5 discusses the results. Finally, Section 6 concludes the paper.

Manuscript received October 5, 2024
Manuscript revised October 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.10.22

2. Related Works

There has been quite a lot of work done in the area of SMS spam detection. A new classifier based on the H2O platform was developed in [4]. The H2O framework was used for evaluating the accuracy, precision, recall, f-measure to classify unseen data by using machine learning algorithms. The framework performed better than other frameworks such as Weka, it was used to determine the best features that will be used to improve the SMS Spam detection process. The work did not develop a system based on the framework proposed and other Machine learning (ML) algorithms needed to be added [4]. A review was done to filter SMS spam using CDA algorithms, KNN and the large cellular network method to measure accuracy, strength and weakness. A comparative review was carried out on various ML algorithms and a systematic search for a proper publisher in the field (study selection, Information Sources, data collection Process and comparison criterion). The work shows that an intensive survey was done to present an accurate result of different text classifier on different datasets for spam filtering. The research was hindered by the lack of public and real datasets, and the low number of features that were extracted per message [5]. A proposed approach to filter spam SMS using ML algorithms was proposed in [6]. Datasets were used as features of the experiments. Features were extracted from messages (HAM and SPAM) to create a feature vector. The approach performed better with Random Forest classification algorithms in terms of high accuracy. It achieved the aim of classifying spam SMS message. The research was limited to comparing ML algorithms to achieve the desired goal [6]. A systematic literature review on SMS spam detection techniques was performed in [7]. The researcher reviewed the available published research works from 2006 to 2016. Performance comparison of the studied literature was carried out. None of the studies solved the challenges of SMS spam detection of regional contents and shortcut words [7]. A proposal for an appropriate preprocessing method of vector representations and classifiers to find the best model. The proposed method was accomplished using preprocessing step, Vector Representation and Classifiers (Naïve Bayes).The system uses a simple rule to detect spam message by catching a special tag in their content. To achieve better result there is a need to enlarge the corpus to provide more data for the system. Filtering system was used to address the problems of mobile SMS spam; however performance of spam detection is very important and must reach an adequate level by making certain adaptation on filtering techniques [8]. The research in [8] founded on increasing the performance of SMS spam detection by applying the same filter used in email spam filters which achieved the highest performance. Two datasets were used for testing, one for English and another for Spanish. English dataset consists of 1002 hams and 82 spams, while Spanish contains 1157 legitimate and 199 spams. Most of the machine learning algorithms that were applied in the vector representation of messages used certain features such as words, lowercase words, bigram and trigram of characters and words bigrams. Email filtering algorithms became underperformed when used with SMS spam, this is as a result of different reasons: messages with limited features, there was no real database for SMS spam, the informal language of the messages and the short length of it [9].The survey provided a detail summary of spam SMS filtering techniques and algorithms which will help to overcome the problem of SMS spam. The research only focuses on the review of other work in the area of spam SMS filtering techniques using data mining. No system was developed to address the main purpose of the paper [10]. A hybrid system of SMS classification to detect spam or ham, using naïve Bayes classifier and apriority algorithms was proposed. It better contributed to accuracy than the state of the art method of classifying text. A categorization system that integrates association rule mining with the classification problem was built. The categorization system which was based on naïve Bayes filter SMS spam messages based on Bayesian learning and sender blacklisting mechanism was developed, it uses crowd sourcing to keep itself updated. The system was used to test the performance of the email spam filter on Korean, English and Apriority algorithms was based on logic but the result was depended on dataset, significant improvement of state of the art algorithms was depicted. Although the system outperforms the state of the art, but a part of the training of the system required little time than the state of the art. The system slightly took more time [11]. A mobile based system called SMS Assasin that can, and Spanish dataset. A set of design goal was presented, which act as guidelines to design a mobile-based SMS spam filter. A mobile-based SMS spam detection system was implemented; it was done using python S60 on a Symbian platform [12].

3. Methodology

In this study, the methodology employed to achieve the objectives are as follows:

3.1 Conceptual Model Description

The proposed system architecture as shown in Fig. 1 depicts various phases of SMS spam detection at both training and testing phases. The detailed of the phases are as follows:

- The pre-processing phases consist of steps such as tokenization, removal of stop words, stemming, lemmatization and cleansing of the dataset.

- The feature selection phase is introduced to reduce the dataset and number of features; this was achieved by using consistency-based methods.
- The dataset was then split into both training and testing datasets.
- The training dataset was subjected to classification with the ensemble of algorithms.
- The results of the classification were then validated using evaluation metrics such as accuracy, precision, f-measure and recall. The results of the classification are the output of SMS spam detection.

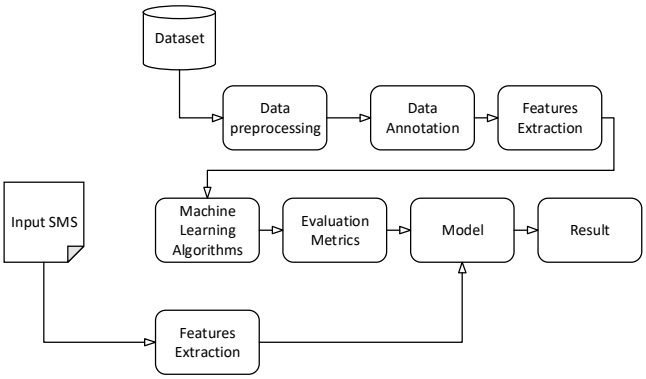

Figure. 1. SMS Spam Detection System

1) SMS Dataset collection

In this research, the University of California Irvine (UCI) Machine Learning repository dataset [15] was used. The dataset is a collection of SMS data that has been sent by users of mobile devices. The SMS corpus was marked at the sentence level. Two classes of polarity (both ham and spam) were considered in this research work. The datasets consist of 6250 text messages classified as ham and spam messages, with 1562 spam messages and 4688 ham messages. The dataset was saved in a CSV file format where each line represents one message; the line consists of the label of a message and text string. Table 1 shows examples of messages in the dataset.

2) Preprocessing

The following preprocessing techniques were employed to remove irregularities and inconsistencies from the dataset collected, to capture the problem being addressed. In the preprocessing phase, a feature engineering technique was applied to the dataset, to enable the machine learning algorithm to produce accurate results. The dataset was processed using techniques such as tokenization and stop word removal and cleaned before entering them into the

Table 1. Examples of Messages in the Dataset

Example of ham and spam messages in UCI dataset and the some of the Nigeria Mobile device user	
Ham	' Have a safe trip to Nigeria. Wish you happiness and very soon company to share moments with
Ham	' Wen did you get so spiritual and deep. That\'s great'
Ham	'U still havent got urself a jacket ah?',
Spam	"IMPORTANT - You could be entitled up to £3,160 in compensation from mis-sold PPI on a credit card or loan. Please reply PPI for info or STOP to opt out.
Spam	Dear customer your ATM ACC...NT has been block due to BVN system error, Quickly call our Customer Care line 088031314841 to reactivate Under 24 hour
Spam	U are a winner of our daily raffle to claim your prize click on the link 'XXXMobileMovieClub:

algorithm. Hence, the work engaged the use of data analysis, iterations, and examination for the performance of the classifier. Data was in a text format after processing and for this work, a CSV file format dataset was used. The python natural language processing tool was used for pre-processing the dataset. The pre-processing procedures are as follows

a. Tokenization

For the processing of the SMS dataset, words that constitute a string of character were classified. This was done because the meaning of the text depends on the relations of the word in the text. Tokenization is the process of extracting words/ data in a message[16], it also helps in the process of chopping data up into pieces, called tokens, perhaps discarding certain characters, for example, punctuation and the total number of characters. It is frequently applied in the detection of attacks. The framework used for tokenization of the SMS dataset was the Natural Language Tool Kit (NLTK), in NLTK, the method tokenized_xample (text) was used to split the sentences into words. The output of the tokenization was converted to a data frame for better text understanding in machine learning application. The tokenized data can be seen in Fig. 2.

b. Removal of Stopwords

One of the common data processing steps in NLP is stopword removal. The objective was to remove all common words that occur more frequently in the dataset [17]. Stopwords are some common words in the text messages that have no or little meaning, for example; you, me, a, him, etc. while carrying on experiment of text

Table 2. Description of features

S/N	Features description	Features criteria
#1	Message Length (ML)	The number of characters in the message
#2	Number of Words (NW)	The number of words in message, usually spam messages contains large number of words
#3	Ratio of Number of Words with length less than three (RNW3)	Number of words less than 3 (NW3) over to the total number of words (NW)
#4	Ratio of Number of Capital (RCW)	Number of Capital Words (CW) over total number of words (NW)
#5	Ratio of Alphanumeric Characters (RAC)	Number of Alphanumeric Characters over Message Length (ML)
#6	Ratio of Special Characters (RSC) such as “,*, _ ,+,=,%,$,@, , \,/,”	Number of Special Characters over Message Length (ML)
#7	Ratio of Punctuation Characters (RPC) such as “‘;?!: ( ) – “ « » <> [ ] { }”	Number of Punctuation Characters over Message Length (ML)
#8	Total number of Digit Characters (DC)	Normalized by maximum number of digit characters.
#9	The existence of word “Call” and digits together	The value of this field is either true or false
#10	The existence of URL in the message	The value of this field is either true or false

analysis, some words have features and are weighty in the text message, they are considered as features, whereas some words do not contribute at all, are considered as stopwords. Usually, articles and pronoun are normally classified as stop words. Figure 3 shows the example of stopwords in the dataset. For removal of stopwords from the SMS dataset, the natural language tool kit (NLTK) was used. In NLTK, the method stopwords_result (text), was used to remove words that do not contribute to future operation in the sentences.

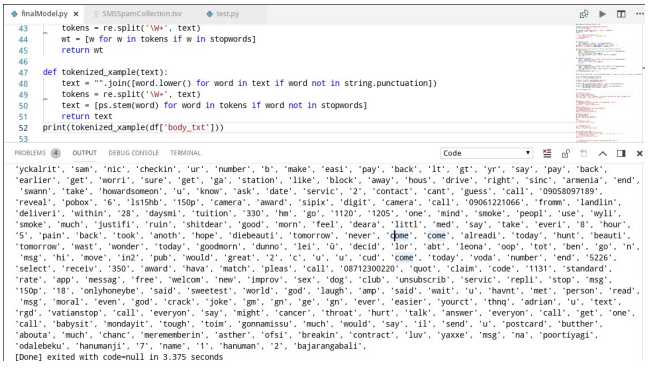

Fig 2. *Tokenized dataset*

c. Annotation of the SMS Dataset

The process of adding metadata information to the text to augment a computer capability to perform NLP is what is referred to as annotation. Features shown in Table 2[4] were used to encode the specific phenomenon to capture the desired behavior of the algorithms that were trained. Metadata was added to each text in the dataset to signify whether the message is spam or ham. In the process, 1 was added for spam and 0 for ham message.

3) Feature Extraction

In this research, in order to more concisely and accurately classify and detect SMS spam, unnecessary features were eliminated by using the consistency-based feature selection approach. In consistency-based feature selection, consistency measures were used to evaluate the importance of feature subsets. This measure is intuitively defined as a metric to measure the distance of a feature subset from the consistent state [19]. A feature set {F1... Fn} is said to be consistent, when Equation 1 holds for all c, f1,...,fn .

$$Pr(C = c \mid F_1 = f_1, \ldots, F_n = f_n) = 0 \text{ or } 1 \quad (1)$$

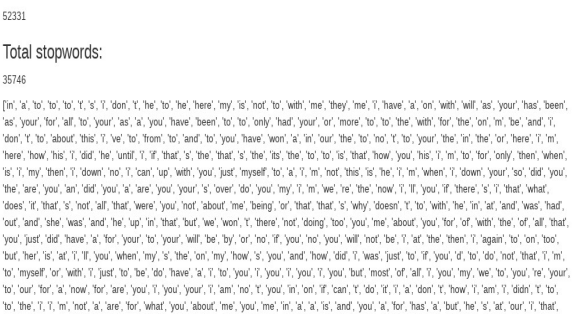

Fig 3. stopword in the dataset.

When a feature subset is consistent, the inconsistency value is 0, and as an inconsistent feature subset approaches the consistent state, the measure decreasingly approaches 0. To illustrate, {F1, F2} in our previous example is measured to be 0, whereas the measure for {F1} and {F2} should be high [19].

The total number of characters in each column of the dataset was considered, the number of rows in the dataset and the numbers of punctuation that exist in the dataset. Features engineering was considered as the technique for extracting the features. Feature engineering was performed firstly by loading the SMS dataset, Performing feature engineering on the sms data by using the function data['body_len'] = data['body_txt'].apply(lambda x : len(x) - x.count(' ')), the next was to create feature for percentage of count of punctuation in the messages, to create additional

features, data ['punct%'] = data['body_txt'].apply(lambda x : create_punc_perct(x))
was applied. There are 10 features in the SMS dataset, all of these features were not relevant in the building of the model. Algorithm 1 show the process used to extract the features from the dataset.

Algorithm 1: Feature Extraction Algorithm

```
Algorithm 1: Feature extraction
Input: text,stopwords,tokens,rw,tc,txc
Output: trainset
/* Feature extraction                                   */
Function FeatureExtraction():
    /* stopwords removal                                */
    def stopwords(text):
        stopwords=set(nltk.corpus.stopwords('english'))
        text="".join(set of words without punctuation)
        tokens=re.split(text)
        rw = set of tokens not in stopwords
        return rw, tokens
    /* create additional features                       */
    def count_punct(text):
        txc= body length of each text message
        tc = total punctuations present in each text message
        return txc, tc
    def get_featureData(c):
        /* 'c' is function that hold the data cleaning process
           */
        tfidf_vect= TfidfVectorizer(analyzer=c) /* fit the train
           data to the vectorizer and transform it          */
        /* transform the test data only                     */
        /* concat the puntuaion in % with the body length to
           the vectorized features                          */
        return trainset, testset
```

4) Ensemble Algorithm

An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. In this research work, four machine learning algorithms were used: Random Forest (RF), Logistic Regression (LR), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). They are described as follows:

(i.) Random Forest (RF) depends on a random selection of variables and data to develop a large number of decision trees. Random Forest uses decision trees to create bootstrap by selecting random features, it is considered as a special case from bagging. However, the main idea of RF is that Shallow trees which are called stumps will be pruned and tuned and the output after tuning and pruning will be aggregated, then RF will rely on such aggregation. The aggregation will lead to an accurate prediction by eliminating the error from trees.

(ii.) Logistic Regression (LR) according to [13] is one of the most popular machine learning algorithms for binary classification. LR is also known as "logit" regression that is used for estimation of discrete value such as 0 and 1, true or false, yes and no, based on a set of variables. It performs better on varies of a solution and its predictive probability lies between 0 and 1. This function is as follows in equation (2)

$$T = \frac{1}{(1+e^{-x})} \qquad (2)$$

T is the transformed, *e* is the Euler's number, *and x* is the input that will be plugged into the function.

(iii.) Multilayer perceptron (MLP) is a sub-class of an artificial neural network. An MLP comprises of, at least, three layers of nodes: an input, a hidden layer, and an output. Apart from the input nodes, every hub is a neuron that utilizes a nonlinear activation function. MLP uses a supervised learning system called back propagation for preparing. Its multiple layers and non-linear activation set MLP apart from a linear perceptron. It can distinguish data that is not linearly detachable. A multilayer perceptron is sometimes referred to as "vanilla" neural systems, particularly when they have a single hidden layer [14].

(iv.) Support Vector Machine (SVM) is a machine learning method that is widely used for data analysis and pattern recognition. SVM has been very excellent in the case of classification. Classifying data has been one of the major parts of machine learning. The idea of support vector machine is to create a hyper plane in between data sets to indicate which class it belongs to. The challenge is to train the machine to understand the structure from data and mapping with the right class label, for the best result, the hyper plane has the largest distance to the nearest training data points of any class.

3.2 Model Formulation

The process involves detecting SMS spam was implemented using the process below and a class. The SMS spam system was developed using a python

programming language in visual studio code environment, the data was preprocessed using Natural Language Tool-Kits (NLTK).

Additional features were created to improve the accuracy of the predicting machine. A total number of punctuations was calculated, and the total length of each word per records was calculated, and this step was done per each text message.

The two additional features added was "words length" and "number of punctuations" together with the vectorized text. Vectorization means turning the text into number since machine learning algorithms deal with a number and the library applied were TFidfVectorizer. The data were divided into training and test sets using the train_test_split library. All the algorithms received training sets of 80% and test sets 20%

For cross-validation the whole datasets were slot into the machine, K-Fold algorithm was used, and K=10. Therefore, the algorithm divides the dataset into 10-folds and 9-fold for training and 1-fold for testing, this was done until the whole folds pass through all phases. Each algorithm produces a model after slotting the training and test sets into the algorithms. Algorithms 2 show the process of polarity detection from the user. The process starts by entering an SMS text message into the system. If the user does not enter a message the system alert that no imputed value into the system, the error tell the user to enter a text message. The text is cleaned in features sets and is fed into the classifiers. The text is classified based on the result of all the classifiers.

Algorithm 2: SMS Polarity detection

```
Algorithm 2: SMS Polarity detection
  Input: trainset,testset, ytrain, ytest,model_1,model,in_data
  Output: ham, spam
1 Function detect_polarity():
2   def generate_model(model):
3     model_1 = model.fit(trainset, ytrain)
        dump(open('model1.sav','wb')) /* do for the rest
        algorithm...                                   */
4   def predict_model(model):
      /* general prediction-- ham=>0 && spam=>1        */
      /* use label encoder to decode the values encoded
        values                                         */
5     model = load(open('model1.sav','rb')) model_1 =
        model.predict(testset) /* do for the rest algorithm...
        */
    /* single message prediction                      */
6   def single_pediction(in_data):
      /* general prediction-- ham=>0 && spam=>1        */
      /* use label encoder to decode the values encoded
        values                                         */
7     model = load(open('model1.sav','rb'))  def getText()(text):
8       indata = {'bodytext':[text]}
9       df = pd.DataFrame(data=indata)
10      test_t = tfidf_vec.transform(df['bodytext'])
11      concatthebodylengthandthetotalpunctuationin%withdf
12    pred = model.predict(getText(in_data))
      /* decode the pred value with label encoder      */
13
14    return pred
```

4. RESULTS AND DISCUSSION

The discussion of the results from the simulation and evaluation of the proposed model are as follows:

4.1 Simulation Results

The developed system detects SMS spam messages. The user can enter an SMS message, the system analyzes the message and return spam or ham. Once the message is inputted, the text entered serves as a test to the models generated by each of the classifiers. The system takes in an SMS message from the user, the features are extracted automatically by the system, the SMS and the accuracy of the message, are predicted by the model. Figure 5 shows the screenshot for the extracted input, the extracted features show what the system uses to relate with the model for the system to detect the SMS spam.

Fig. 5. Extracted input features

The user enters an SMS message in the system for example “To get 2.50 pounds free call credit and details of great offers pls reply 2 this text with your valid name, house no and postcode”. The extracted features are fitted into the models. The result shows “Ham” which implies that the SMS message entered into the system is legitimate as shown in Fig. 6. To increase the performance of the system, the four classifiers were combined and predicted as a single result, for the inputted SMS message, the system gave a spam result as shown in Fig. 7. This implies that the sentence entered by the user is a spam message

4.2 Evaluation Results

The performances of the algorithms (i.e. classifiers) were evaluated using Receiver Operation Characteristics

(ROC) curve as the measure of text accuracy. The results in Table 3 and Fig. 8 showed that the selected classifiers perform well using the proposed model. Also, a simulation was carried out to ascertain the effectiveness of the ensemble method used. The result obtained in Table 4 and Fig. 9 established that the proposed model gives a good result with stack Generalization ensemble performing better than other ensembles such as VoltClassifier, Ada Boosting and Gradient Boosting. Performance evaluation of the proposed model was then carried out by benchmarking it with an existing model, H20 framework [4], using ROC, accuracy, precision, recall and F-measure as performance metrics. The result shown in Fig. 10 implied that the proposed model classifies accurately than the existing model. Also, the evaluation results in Table 4, shows that the proposed model gave an increase of 3.03% of accuracy, 8.94% of Recall, 2.17% of F-measure; and a decrease of 4.55% of Precision over the existing model. The result from the performance evaluation shows that there is an improvement in the accuracy of detecting SMS Spam

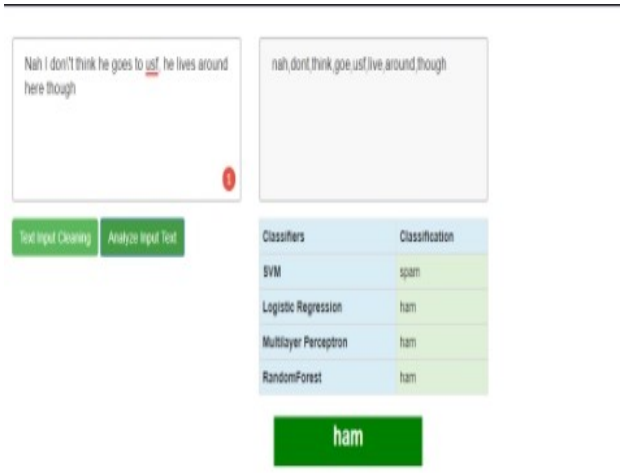

Fig. 6. SPAM detection result

Table 3: Receiver of operating characteristics for the classifiers

Table 3: Receiver of operating characteristics for the classifiers

Classifiers	Result
Multilayer Perceptron	96.89
Support Vector Machine	69.27
Logistic Regression	89.98
Random Forest	92.72

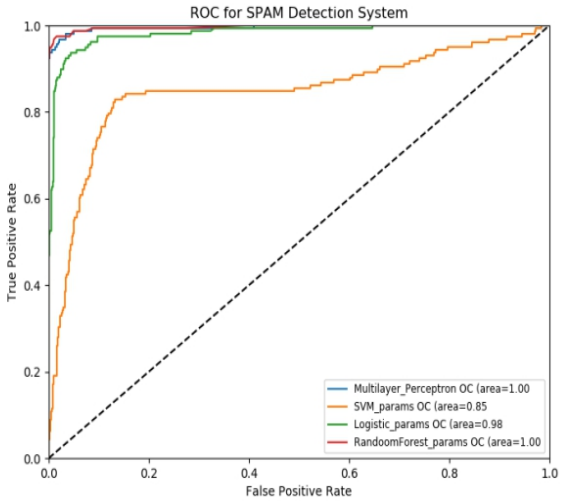

Fig 8: the ROC curve of the classifier

Table 3: Ensemble technique evaluation

Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC (%)
Stack Generalization	98.03	91.46	94.94	93.17	96.74
VoltClassifier	97.13	100	79.75	88.73	89.87
Ada Boosting	97.13	88.41	91.77	90.06	94.89
Gradient Boosting	97.04	94.96	83.54	88.89	91.41

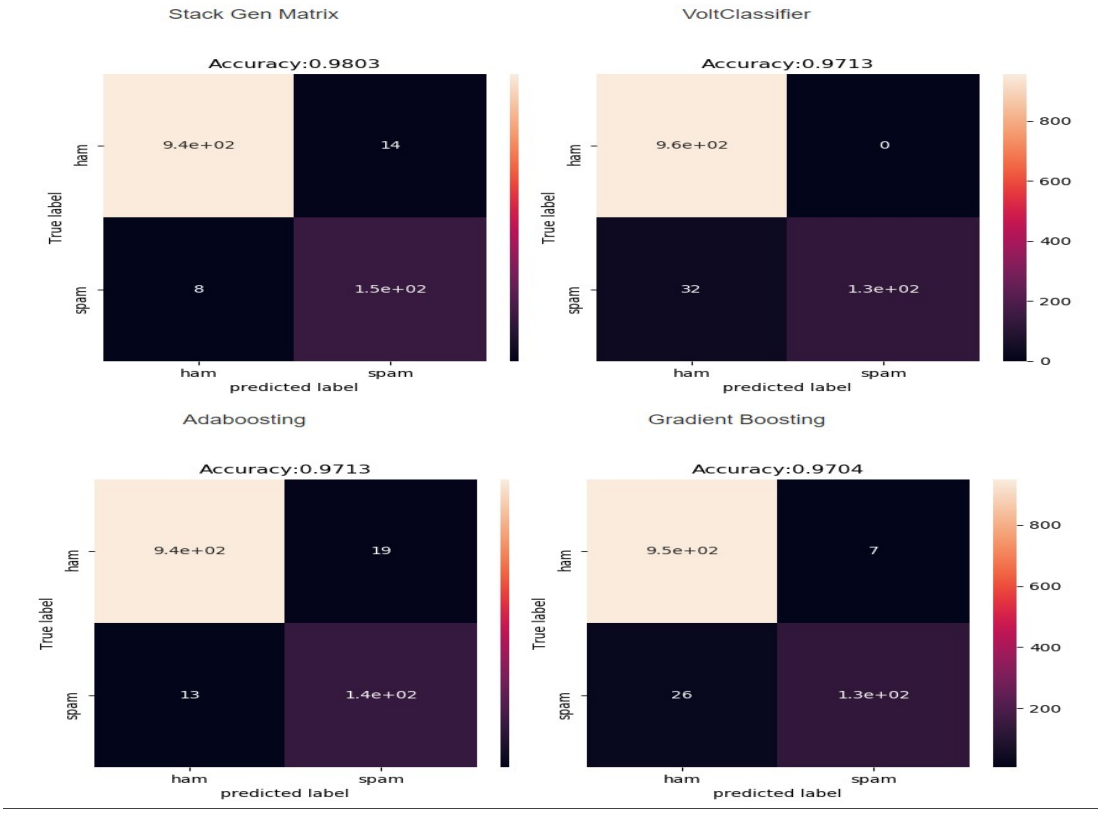

Fig 9. Confusion Metrics of the Ensemble algorithms

Table 4: Performance Evaluation results

S/N	Metrics	Existing Model	Proposed Model	Percentage difference
1	ACCURACY	95%	98.03	3.03%
2	PRECISION	96%	91.46%	4.55%
3	RECALL	86%	94.94%	8.94%
4	F-MEASURE	91%	93.17%	2.17%
5	ROC		97.74%	

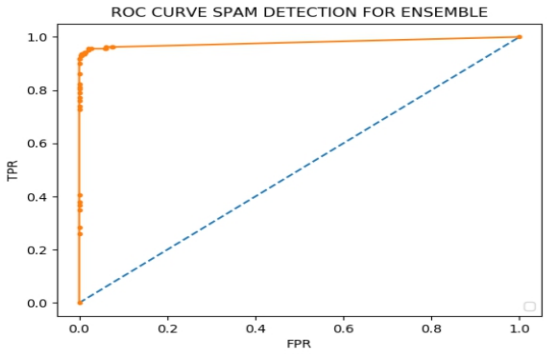

Fig 10: the ROC curve of the Existing and Proposed model

5. Conclusion

SMS is one of the best mediums of communication amongst mobile user, they are of two types: ham and spam, spam message are the annoying and unfortunate messages that must be evacuated or obstructed before the client getting them. An SMS spam filtering framework was developed using an ensemble of four machine learning algorithms namely, Random Forest (RF), Logistic Regression (LR), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). The proposed model was simulated and it was obvious in the results that the proposed model has a better

performance in detecting spam and ham messages in terms of 3.03% higher accuracy, 8.94% higher recall and 2.17% higher F-measures. Therefore, the proposed model can be adapted by the network service providers for better Quality of Service (QoS).

Acknowledgment

This Research was funded by the TETFund Research Fund" and Africa Centre of Excellence OAK-Park.

References

[1] A. Al-Hassana, E. M. El-Alfyb, "Dendritic Cell Algorithm for Mobile Phone Spam Filtering," 6th International Conference on Ambient Systems, Networks and Technologies, Procedia Computer Science, vol. 52, pp. 244 – 251, 2015.

[2] Baldwin, "350,000 different types of spam SMS messages were targeted at mobile users in 2012," Computer weekly publication [online] February 2013. Available: https://www.computerweekly.com/news/2240178681/350000-different-types-of-spam-SMS-messages-were-targeted-at-mobile-users-in-2012

[3] D.N. Sohn, J.T. Lee, K.S. Han, and H.C. Rim, "Content-based mobile spam classification using stylistically motivated features". *Pattern Recognition Letters*, vol. 33, no. 3, pp.364–369, 2012.

[4] Suleiman and G. Al-Naymat, "SMS Spam Detection Using H2O framework." *Procedia Computer Science,* vol. 113, pp 154-161, 2017.

[5] H. Sajedi, G. Z. Parast, and F. Akbari, " SMS Spam Filtering Using Machine Learning Techniques: A Survey" . *Machine Learning Research.* Vol. 1, No. 1, pp. 1-4, 2016.

[6] N. Choudhary and A.K.Jain. "Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique". *In: Singh D., Raman B., Luhach A., Lingras P. (eds) Advanced Informatics for Computing Research. Communications in Computer and Information Science, Springer, Singapore,* vol. 712, pp 18-30, 2017.

[7] L. N. Lota and B M Mainul Hossain ,"A Systematic Literature Review on SMS Spam Detection Techniques", *International Journal of Information Technology and Computer Science (IJITCS*), vol.9, no.7, pp.42-50, 2017.

[8] T.H. Pham and P. Le-Hong, "Content-based Approach for Vietna- mese Spam SMS Filtering". *In proceedings of 2016 International Conference on Asian Language Processing (IALP*), Tainan, pp. 41-44, 2016.

[9] G.V. Cormack, J.M. Gómez Hidalg, and E.P. Sánz, "Feature Engineering for mobile (SMS) spam filtering," *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, *July 23- 27, 2007, Amsterdam*, pp 871-872, 2007.

[10] N. Chaudhari, P. Jayvala, and P. Vinitashah," Survey on Spam SMS filtering using Data mining Techniques," *International Journal of Advanced Research in Computer and Communication Engineering,* Vol. 5, Issue 11, 2016

[11] I. Ahmed, D. Guan and T. C. Chung, " SMS Classification Based on Naïve Bayes Classifier and Apriori Algorithm Frequent Itemset," *International Journal of Machine Learning and Computing*, Vol. 4, No. 2, pp 184-187, 2014

[12] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta and V. Naik, "SMS Assassin: Crowdsourcing Driven Mobile-based System for SMS Spam Filtering," *in Proceedings of the 12th Workshop on Mobile Computing Systems and Applications,* pp 1-6, 2011.

[13] J. Brownlee, "*Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models and Work Projects End-To-End.,* Edition: v1.5, pp 1-24, 2016,

[14] H. Trevor, T. Robert, J. H Friedman and F. James, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction*," In proceedings of the Mathematical Intelligencer, Vol. 27*, No 2, pp 83-85, 2004.

[15] T. A. Almeida and J. M Gómez Hidalgo, "*SMS Spam Collection Data Set- UCI Machine Learning Repository,"* Available: *https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection*. 2011

[16] S. Guido and A. C. Muller, *"Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016

[17] H. Shirani-Mehr, "*SMS Spam Detection using Machine Learning Approach*," CS229 Project 2013, Stanford University, USA, pp. 1–4, 2013

[18] S. Schrauwen, "Machine learning approach to sentiment analysis using the Dutch Netlog Corpus." Computational Linguistic and Psycholingistics Research Center, pp1-78, 2010

[19] K. Shin, D. Fernandes and S. Miyazaki. "Consistency Measure for feature Selection: A formal Definition, Relative Sensitivity Comparison and a fast Algorithm". *In Proceeding of Twenty –Second International Joint Conference on Artificial Intelligence, pp* 1491-1497, 2011