Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220601

Hybrid spam message detection using convolutional neural network and long short-term memory techniques

Samuel Ibukun Olotu*[0000-0001-6864-7140] and Oladunni Abosede Daramola[000-0001-7187-9874]

Federal University of Technology Akure, Ilesa-Owo Expressway Akure, Nigeria

siolotu@futa.edu.ng

Abstract. Short Message Service (SMS) is a feature of a mobile phone that enable convenient and instant way of sending electronic messages between users. As SMS usage increases fraudulent text messages, known as spam, are becoming more common. Spam SMS may result in leaking personal information, invasion of privacy or accessing unauthorized data from mobile devices. Users of mobile phones can mistakingly give away personal information with the assumption that they are sharing it with the right recipients. This work propose a SMS spam detection method that combines convolutional neural network (CNN) and long short term memory (LSTM) deep learning algorithms. The CNN is used for feature extraction while the LSTM classifies the message. The SMS spam dataset, collected from online repository, is used to train the model. Word embeddings is used to vectorize the words in the message to make it suitable for the model. The result obtained from the implementation outperforms other machine learning algorithms with an accuracy of 99.77%.

Keywords: convolutional neural network, dense layer, SMS spam prediction, long short-term memory.

1. Introduction

Mobile phones are essential communications devices for about 5.22 billion users which make up about 66% of the world's population in 2020 [1]. The phones are usually used for both making telephone calls and messaging between users with Short Message Service (SMS) [2]. SMS is considered an affordable and instant way of sending text messaging between individual users. It is usually used for activities such as appointments, payment notifications, meeting reminders and so on [3].

As SMS usage increases, spam text messages are becoming more common. The Spam SMS contains junk messages delivered to a mobile device of an unsuspecting user and may result in leaking personal information, accessing unauthorized data from mobile devices or invasion of privacy [4]. Phone users may be further susceptible to the danger because they usually rely on SMS for important notifications such as bank transactions, job applications and promo offers.

A solution to these problems is to accurately detect spams SMS. Existing techniques for predicting spam SMS use traditional learning algorithms. These include support vector machine (SVM), naïve bayes (NB), decision tree (C4.5) and random forest. Some other works adopt convolutional neural net-

DOI: 10.54254/2755-2721/2/20220601

work (CNN) and long short-term memory (LSTM) deep learning techniques for its predictions. However, there is need for an optimized hybrid technique that achieve higher accuracy. This study aims to use a deep learning hybrid classifier to predict spam and non-spam messages.

The proposed technique use a hybrid of convolutional neural network (CNN) and long short term memory (LSTM) deep leaning algorithms. The technique is optimized by fine tuning the hyperparameter values over several experiments. The dataset used to train the model is collected from SMS Spam Corpus v.0.1 and consist of 5572 text messages.

The paper is arranged as follows. A review of related works is presented in section 2. The system design of the proposed method is explained in section 3. The implementation of the method is described in section 4. Section 5 discusses the results of the implementation. Finally, the paper is concluded in section 6.

2. Literature review

Some existing research works that applied machine learning techniques to SMS spam filtering and detection are reviewed in this section. These works fall under the major categories of supervised, unsupervised and deep learning algorithms. Akbari and Sajedi [5] proposed GentleBoost algorithm for SMS spam detection. The method combined the features of AdaBoost and LogitBoost. It reduces the word attributes and keep the accuracy at 78%. Mussa and Jameel [6] proposed extreme gradient boosting algorithm (XGBoost) to detect SMS spam. The work use sequential search algorithm to select relevant features. The method yields 98.64% of accuracy when used for handling this imbalanced dataset. Choudhary and Jain [7] propose a SMS spam filtering technique based on random forest algorithm. The work is implemented using WEKA tool and the result shows that random forest outperforms other algorithms. Sjarif et al. [8] present spam filtering technique using random forest. Term frequency-inverse document frequency (TF-IDF) feature extraction method is combined with random forest to yields the best result compared with other traditional techniques.

Random forest used with Feature extraction

Sjarif et al. [9] propose a SMS spam classification technique using Support Vector Machine (SVM) algorithm. The method outperforms other classifiers with an average an accuracy is 98.9%. Navaney et al. [10] compared the performance of support vector machines algorithm, naïve Bayes Algorithm and the maximum entropy algorithm in filtering Spam messages. The support vector machine algorithm gives the highest accuracy of 97.4%. Gupta et al. [11] present gaussian naive bayes, bernoulli naive bayes, decision tree, and multinomial naive bayes combined differently. From the result it is observed that that an ensemble of decision tree, Bernoulli Naive Bayes classifier and Gaussian Naive Bayes classifier performs best. In Tekerek [12], the SVM classifier outperforms Naive Bayes (NB), support vector machine (SVM), k-nearest neighborhood (KNN), decision tree (C4.5) and random forest (RF) in detecting SMS spam.

Jain et al. [13] present Long Short Term Memory (LSTM) deep learning technique is used to classify spam messages. The result of LSTM model outperforms SVM, naïve bayes, ANN, K-NN and random forest. Popovac et al. [14] and Huang [15] apply Convolutional Neural Network method with an accuracy of 98.4%. In Roy et al. [16] the prediction is carried using Long Short-term memory (LSTM) and Convolutional Neural Network (CNN) models. The experimental results showed that the CNN model outperformed the LSTM model. Barushka and Hajek [17] proposed a deep learning model (DBB-RDNNReL) for spam detection with good performance on strongly imbalanced and highly non-linear spam datasets. In Gomaa [18] Random Multimodel Deep Learning (RDML) algorithm outperforms other classical and deep learning methods. The RDML method combines the advantages of convolutional neural network, deep neural network and recurrent neural network.

Jain et al. [19] combines Convolutional Neural Network (CNN) and Long Short Term Neural Network (LSTM) methods for spam detection in social media. The results of the method, Sequential Stacked CNNLSTM model (SSCL), achieve an accuracy of 99.01% and 99.01% for Twitter and SMS spam respectively. Chandra and Khatri [3] utilize Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) to detect Spam to detect Spam. The result shows improvement over SVM and Naïve

DOI: 10.54254/2755-2721/2/20220601

Bayes. The hybrid technique in Baaqeel and Zagrouba [20] filter spam messages with Support Vector Machine (SVM) supervised classifier and k-means clustering unsupervised classifier.

3. Proposed model

The proposed hybrid deep learning model is made up of a word embeddings layer, long short-term memory (LSTM) layer, convolutional neural network (CNN) layer and dense neural network layer (see Figure 1).

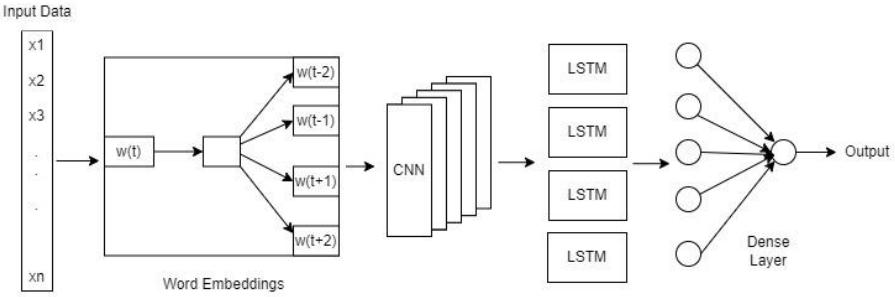

Figure 1. Proposed model architecture.

3.1. Word embeddings

The word embedding provide a way to convert the text messages to numeric vectors. A word embedding is created using the Word2vec model. The model use the skip-gram architecture to predict the given sequence from context of the word and represent the message as vectors of numbers with equal size. Each message consists of a list of the words represented as $t_1, t_2, t_3, \dots, t_n$. Unique words are selected from the message to form a vocabulary set (V) with each word assigned a unique integer value. From the embeddings created with word2vec, the word vector is extracted for words present in V as given in

$$E(m) = [e(t_1), e(t_2), e(t_3), \dots, e(t_n)] \quad (1)$$

where $e(t_1)$ is the word vector of word t_1. The word vectors are concatenated to produce the SMS word matrix, M, as shown in $M = e(t_1) \cdot e(t_2) \cdot e(t_3) \cdots e(t_n)$.

$$M = e(t_1) \cdot e(t_2) \cdot e(t_3) \cdots e(t_n) \quad (2)$$

where $\cdot$ is the sign of concatenation. The SMS message is converted to a matrix of vectors using the word vectors, number of features (see $M_{sms} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1d} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nd} \end{bmatrix}$). The word length of each SMS message represented as d and the number of messages is n.

$$M_{sms} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1d} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nd} \end{bmatrix} \quad (3)$$

3.2. Convolutional Neural Network (CNN)

The proposed model extracts the hidden features in the SMS message using 1-dimensional CNN layer with embedding layer. The convolution operation involve the SMS matrix (M) and kernel size of F to yield a feature vector according to $C = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1d} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nd} \end{bmatrix} \odot \begin{bmatrix} fm_{11} & fm_{21} \\ fm_{12} & fm_{22} \\ \vdots & \vdots \\ fm_{1d} & fm_{2d} \end{bmatrix}$:

DOI: 10.54254/2755-2721/2/20220601

$$C = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1d} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nd} \end{bmatrix} \odot \begin{bmatrix} fm_{11} & fm_{21} \\ fm_{12} & fm_{22} \\ \vdots & \vdots \\ fm_{1d} & fm_{2d} \end{bmatrix} \quad (4)$$

where $\odot$ represents the convolution operator, m is the region size of the kernel. n is the number of messages and d is the word size. Initially, the messages are of different sizes according to the number of words. The padding technique is used to make the input vector of equation length. The rectified linear unit (ReLU) function is used to change all negative values of input x to zero. A max-pooling function is used to identify important features from the text.

3.3. Long Short-Term Memory (LSTM)

The LSTM layer is used to memorize the sequence of the text data and discard unused information present in it to reduce the runtime and cost. The LSTM use the memory of the sequences of text for predicting the message as spam or not-spam. The LSTM architecture consists of forget, input and output gates. The forget gate memorizes and recognizes the information coming into the network and eliminate the information not

required for the network to learn and predict the data. The forget gate layer (ft) is defined in $f_t = \sigma(\omega[y_{(t-1)}x_t] + b_f)$:

$$f_t = \sigma(\omega[y_{(t-1)}x_t] + b_f) \quad (5)$$

where σ is a logical sigmoid function whose output is [0, 1], ω is the weight, $y_{(t-1)}$ is the previous time stamp output, x_t is the input of the current time step and b_f is the bias. LSTM uses the input gate to decide the measure of importance of the information and enable each layer to learn and eliminate unimportant information to make predictions. It is done using the input gate (i_t) sigmoid layer and tanh layer to generate the internal memory of the unit, c_t , shown in

$$c_t = tanh(\omega[y_{(t-1)}, x_t] + b_c) \quad (6)$$

Where tanh represents a hyperbolic tangent function, and its output is [−1, 1]. The output gate helps the hidden state to decide which information it should forward. The output from each step, o_t, is defined by the

$$o_t = \sigma(\omega[y_{(t-1)}, x_t] + b_o) \quad (7)$$

3.4. Dense neural network layer

The dense neural network layer is made up of fully connected hidden and output layers.

The outputs of the LSTM layer is fed into hidden layers of the dense network and finally passed to the output layer to produce the prediction. The layers use the extracted features from the LSTM layer to classify the messages into predefined spam and ham classes. The parameters specified at the dense layers are unit of the layers and activation function. The unit defines the size of the output while the activation function is used to transform the input values.

4. Experiment

4.1. Dataset description

The proposed system takes in SMS messages dataset as inputs. The dataset was obtained from the UCI Machine Learning Repository. It is made up of 5574 SMS messages in English (see Figure 2).

DOI: 10.54254/2755-2721/2/20220601

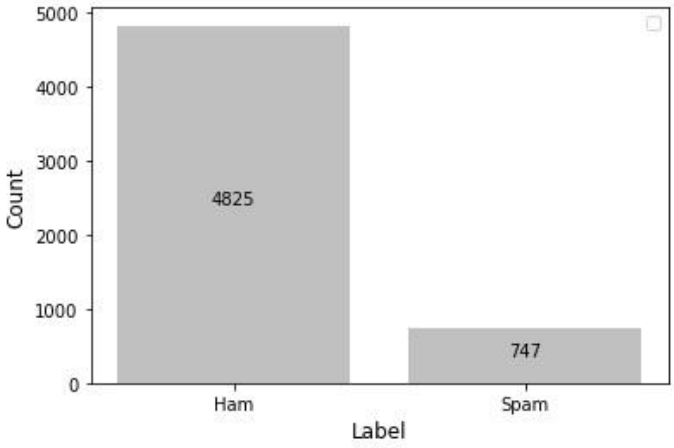

Figure 2. Dataset distribution.

Each data falls under fields, namely, label (spam or ham) and text message. The dataset is composed of 747 spam messages and 4827 and ham (non-spam) messages. The dataset is split into train and test sets. The training and testing distributions based on class labels is shown in Figure 3.

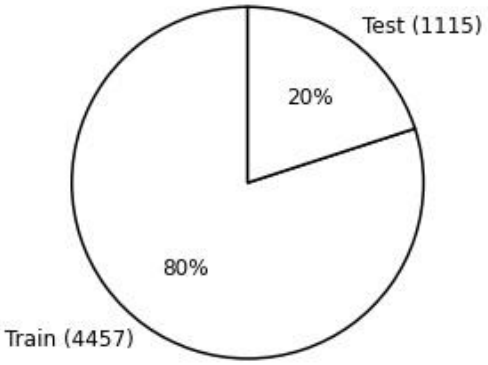

Figure 3. Data split.

4.2. Dataset preprocessing

The dataset is prepared in comma-separated values (CSV) files. On loading the data it is made to undergo stages of pre-processing in order to make it more suitable for training. The text and label are necessary columns in the dataset while the others are dropped. Other pre-processing involves converting the text messages to lower case, splitting the text into smaller pieces called tokens, removing numbers that are not relevant to the analyses and removal of punctuation symbols.

DOI: 10.54254/2755-2721/2/20220601

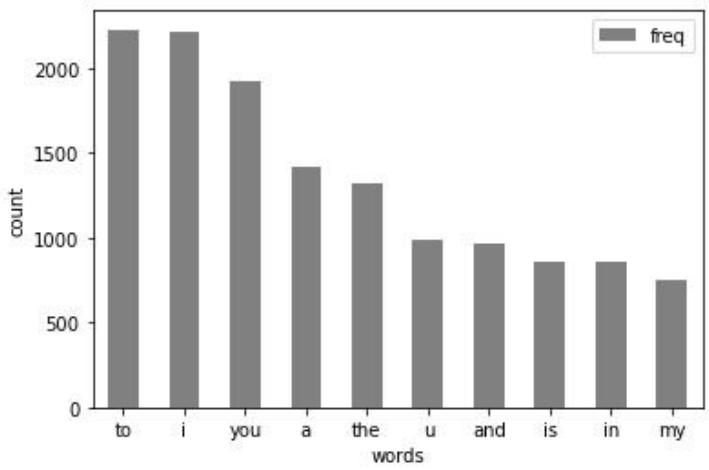

Figure 4. Some frequent words in dataset.

Before training the deep learning models stop words are removed from the text. This stop words refers to most common words that have a very little meaning, such as "to", "i", "you" and so on. Figure 4 shows the plot of 10 most frequent stopwords. The text in the label column of the dataset is changed to numeric form that is Ham and spam are transformed to 0 and 1 respectively. Stemming is used to reduced derived words to their word stem, base or root form. Lemmatization preprocessing makes sure that words that are stemmed does not lose its meaning. A predefined dictionary on the context of words is used during the lemmatization. The corpus is checked for the presence of frequent and rare words which are of not so much importance and are removed.

4.3. Experimental analysis

The proposed model use filter size and number, activation function, optimizer, dropout, number of features, LSTM units, epochs and batch size hyperparameters in the implementation. Extensive experiments were carried out to find the optimal parameter values for the dataset. The following Table 1 gives the initial values of these parameters.

Table 1. Initial parameters.

Parameter	Value
Activation function	ReLU
LSTM units	10
Optimizer	Adam
Dropout rate	0.2
Number of Features	1000
LSTM units	10
Epochs	10
Batch size	30

The convolutional neural networks use filters to improve the detection of features in the input. Different experiments of the proposed model is carried using two filter sizes and the result shows that the model performed better with size 5. Table 2 shows the results.

Table 2. Filter size.

Filter size	Accuracy
4	99.55
5	92.44

Experiments with various filter number achieves the result shown in Table 3. It is observed that a filter number of 64 give the best accuracy of model.

DOI: 10.54254/2755-2721/2/20220601

Table 3. Number of filter.

Number of filter	Accuracy
32	99.64
64	99.66
128	87.97

The activation function ensures that the model trains the dataset well. The results from three common activation functions at the hidden layers shows that Rectified linear activation function (ReLU) gives the best accuracy as shown in Table 4.

Table 4. Activation function.

Activation function	Accuracy
Tanh	99.53
ReLU	99.59
Sigmoid	99.47

The reduction of losses and improvement of accuracy of the results is the responsibility of optimization algorithms. In the analysis, several optimization parameters used are adagrad, adadelta, SGD, RMSprop and Adam. From the result is seen that the adam optimization parameter has the highest accuracy (see Table 5).

Table 5. Optimization algorithms.

Optimization	Accuracy
Adagrad	90.93
Adadelta	86.59
SGD	87.49
RMSprop	99.21
Adam	99.59

Dropout values is used to improve the neural network's performance. In the experiments, the dropout rate is adjusted from 0 to 0.9 and the results are shown in Table 6.

Table 6. Dropout.

Dropout	Accuracy
0.1	99.50
0.2	99.39
0.3	99.14
0.4	99.43
0.5	99.60
0.6	98.77
0.7	98.97
0.8	99.00
0.9	99.59

Experiments on the effect of the number of features used as inputs to the model is shown as shown in Table 7.

Table 7. Number of features.

No. of features	Accuracy
1000	99.59
2000	99.61
3000	99.78

DOI: 10.54254/2755-2721/2/20220601

Table 7. (continued).

4000	99.59
5000	99.61

The number of the LSTM unit specifies the capability of the network to memorize the information and correlate it with the past. The result of the variations of the LSTM unit gives an optimum accuracy of 99.59% at 10 units (see Table 8)

Table 8. Number of LSTM units.

No. of LSTM units	Accuracy
10	99.59
50	99.48
100	99.35
150	99.39
200	97.35

The epoch is determines the number of times the training process is repeated on the same training data. If the epoch is small, the model may not have converged and if is large, the problem of overfitting may emerge. The result of the experiments show that with an epoch of 20, the accuracy reaches the maximum value as shown in Table 9.

Table 9. Epochs.

Epochs	Accuracy
10	99.44
20	99.59
30	99.55

The number of samples used to train a model is determined by the batch size. Several batch sizes are tested in the experiments and the resulting accuracy values are shown in Table 10. It is observed that the best was obtained with the batch size of 30.

Table 10. Batch size.

Batch size	Accuracy
10	99.35
20	99.50
30	99.59
40	99.52
50	99.53

The optimized parameters are obtained for filter size/number, activation function, optimizer, dropout rate, number of features, lstm units, epochs and batch size obtained after the extensive experiments are collated as shown in Table 11. These values are used to implement the proposed model and the result of training and testing are compared with the traditional machine learning models.

Table 11. Optimized hyperparameter settings.

Parameter	Value
Word representation	word2vec
Filter size/number	4/64
Activation function	ReLU
Optimizer	Adam
Dropout rate	0.5
Number of Features	3000

DOI: 10.54254/2755-2721/2/20220601

Table 11. (continued).

LSTM units	10
Epochs	20
Batch size	30

4.4. Performance evaluation

The performance of proposed method is evaluated by carrying out training and testing process. The training and testing is set to a ratio 30% and 70% of the dataset respectively. The hybrid deep network is trained using both the features and labels of the train set. During the testing, only the features of the test set is fed into the model while the label is predicted. The metric used in the evaluation of the proposed method are accuracy, recall, precision and F-measure. True Positive (TP) is the amount of messages correctly classified as spam while True Negative (TN) are the messages accurately classified as ham. False Positive (FP) is the amount of false spam messages while False Negative (FN) are false ham. The ability of classify the message correctly is given as accuracy shown in $a\ \ ccuracy = \frac{TP+TN}{TP+TN+FP+FN}$.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

Precision is used to calculate the fraction of cases for which the accurate outcome is returned given by $p\ \ recision = \frac{TP}{TP+FP}$.

$$precision = \frac{TP}{TP+FP} \tag{7}$$

Recall is the quotient of accurate to inaccurate forecasts within real texts and is given in $r\ \ ecall = \frac{TP}{TP+FN}$.

$$recall = \frac{TP}{TP+FN} \tag{8}$$

The F-score is a valuable and efficient metric for unbalanced data given in $f\ \ score = 2 \times \frac{precision \times recall}{precision+recall}$.

$$fscore = 2 \times \frac{precision \times recall}{precision+recall} \tag{9}$$

The confusion matrix shows the values of true negative, false positive, false negative and true positive as 4824, 1, 8 and 739 respectively. The result of the implementation the proposed method is compared with K-nearest neighbour (KNN), adaboost, decision tree (C4.5), random forest, long short term neural network (LSTM) and convolutional neural network (CNN) machine learning algorithms as shown in Table 12. The result shows that the proposed model have the highest accuracy of 99.77%. The recall, precision and F-score for the proposed method shows best values over the others. In another sets of experiments the accuracy of the proposed model is also compared with algorithms from some related works. The related works used different or combinations of machine learning algorithms on the same dataset as shown in the Table 13.

Table 12. Performance of proposed and existing classifiers.

Model	Accuracy	Precision	Recall	F-score
KNN	0.878	0.586	0.371	0.455
Adaboost	0.895	0.688	0.432	0.531
C4.5	0.917	0.729	0.633	0.678
Random forest	0.93	0.884	0.563	0.688
LSTM	0.968	0.983	0.778	0.868
CNN	0.975	0.943	0.869	0.905
Proposed CNN-LSTM	0.998	0.999	0.989	0.994

DOI: 10.54254/2755-2721/2/20220601

Table 13. Performance of proposed and related works.

Author(s)	Algorithm	Accuracy (%)
Akbari and Sajedi (2015)	GentleBoost	98.30
Mussa and Jameel (2019)	XGBoost	98.64
Gupta et al. (2019)	NB	99.49
Sjarif et al. (2019)	RF	97.50
Sjarif et al. (2020)	SVM	98.90
Taheri and Javidan	RNN	98.00
Chandra and Khatri (2019)	RNN + LSTM	98.00
Baaqeell and Zagrouba (2020)	K-means + SVM	98.80
Roy et al. (2020)	CNN	99.44
Jain et al. (2019)	CNN + LSTM	99.01
Proposed model	CNN + LSTM + RNN	99.77

5. Conclusion

This work propose a technique that that combines convolutional neural network (CNN), long short term memory (LSTM) and fully-connected dense neural network techniques for spam classification. The dataset used in the model contains SMS messages in English with 747 spam messages and 4827 ham messages. The word2vec model is used to convert text messages to numeric vectors. The optimal performance of the method is obtained after several trial experiments were carried out on different values for filter size, filter number, activation function, optimization algorithm, drop out, number of features, LSTM units and batch size. The performance of proposed model was evaluated using train and test datasets. The results of the experiments of the proposed model is obtained using confusion metrics and metrics such as accuracy, precision, recall and f-score. Experiments are also performed using other machine learning models such as decision tree (C4.5), random forest (RF), k-nearest neighbor (KNN), adaboost, convolutional neural network (CNN) and long short term memory (LSTM). The proposed work also gives the highest accuracy among other related works.

References

[1] Kemp, S.: Digital 2021 - Global Overview Report. Datareportal (2021).
[2] Brown, J., Shipman, B., & Vetter, R.: SMS: The short message service. Computer, 40(12), 106-110 (2007).
[3] Chandra, A., Khatri, S. K.: Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory. International Conference on Information Systems and Computer Networks (ISCON), pp. 118–122 (2019).
[4] Gupta, M., Bakliwal, A., Agarwal, S., Mehndiratta, P.: A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers. 11th International Conference on Contemporary Computing (IC3), pp. 1-7 (2018).
[5] Akbari, F., Sajedi, H.: SMS spam detection using selected text features and Boosting Classifiers. 7th Conference on Information and Knowledge Technology (IKT), pp. 1-5 (2015).
[6] Jalal Mussa, D., M. Jameel, N. G.: Relevant SMS Spam Feature Selection Using Wrapper Approach and XGBoost Algorithm. Kurdistan Journal of Applied Research, 4(2), 110-120 (2019).
[7] Choudhary, N., Jain, A. K.: Towards filtering of SMS spam messages using machine learning based technique. In: International Conference on Advanced Informatics for Computing Research, pp. 18-30. Springer, Singapore (2017).
[8] Sjarif, N. N. A., Azmi, N. F. M., Chuprat, S., Sarkan, H. M., Yahya, Y., Sam, S. M.: SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. Procedia Computer Science. 161, 509–515 (2019).

DOI: 10.54254/2755-2721/2/20220601

[9] Sjarif, N. N. A., Yahya, Y., Chuprat, S., Azmi, N. H. F. M.: Support vector machine algorithm for SMS spam classification in the telecommunication industry. International Journal on Advanced Science Engineering Information Technology. 10, 635–639 (2020).
[10] Navaney, P., Dubey, G., Rana, A.: SMS Spam Filtering Using Supervised Machine Learning Algorithms. In: 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) pp. 43-48. IEEE (2018).
[11] Gupta, V., Mehta, A., Goel, A., Dixit, U., Pandey, A. C.: Spam detection using ensemble learning. Harmony search and nature inspired optimization algorithms, 661-668 (2019).
[12] Tekerek, A.: Support vector machine based spam SMS detection. Politeknik Dergisi, 22(3), 779-784 (2018).
[13] Jain, A. R., Joshi, K. K.: Optimal solution on vehicular Ad-hoc Network for congestion control by short message transmission. International Research Journal of Engineering and Technology (IRJET), 4, 1963–1967 (2017).
[14] Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Convolutional Neural Network Based SMS Spam Detection. 26th Telecommunications Forum (TELFOR), 1–4 (2018).
[15] Huang, T. A CNN model for SMS spam detection. 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 851–861 (2019).
[16] Roy, P. K., Singh, J. P., Banerjee, S.: Deep Learning to Filter SMS Spam. Future Generation Computer Systems 102, 524-533 (2020).
[17] Barushka, A., Hajek, P.: Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Applied Intelligence, 48(10), 3538-3556 (2018).
[18] Gomaa, W. H.: The impact of deep learning techniques on SMS spam filtering. International Journal of Advanced Computer Science and Applications, 11(1), 536-543 (2020).
[19] Jain, G., Sharma, M., Agarwal, B.: Spam detection in social media using convolutional and long short term memory neural network. Annals of Mathematics and Artificial Intelligence, 85(1), 21-44 (2019).
[20] Baaqeel, H., Zagrouba, R.: Hybrid SMS spam filtering system using machine learning techniques. In 2020 21st International Arab Conference on Information Technology (ACIT), pp. 1-8 (2020).