

An intelligent auto-response short message service categorization model using semantic index

Budi Padmaja¹, Myneni Madhu Bala², Epili Krishna Rao Patro¹, Adiraju Chaya Srikruthi¹,
Vytila Avinash¹, Chenumalla Sudheshna¹

¹Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India

²Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India

Article Info

Article history:

Received Apr 3, 2023

Revised Jul 22, 2023

Accepted Aug 1, 2023

Keywords:

Auto-response

Multi-class classification

Multi-layer

Perceptron

Semantic

Short message service

Spam

ABSTRACT

Short message service (SMS) is one of the quickest and easiest ways used for communication, used by businesses, government organizations, and banks to send short messages to large groups of people. Categorization of SMS under different message types in their inboxes will provide a concise view for receivers. Former studies on the said problem are at the binary level as ham or spam which triggered the masking of specific messages that were useful to the end user but were treated as spam. Further, it is extended with multi labels such as ham, spam, and others which is not sufficient to meet all the necessities of end users. Hence, a multi-class SMS categorization is needed based on the semantics (information) embedded in it. This paper introduces an intelligent auto-response model using a semantic index for categorizing SMS messages into 5 categories: ham, spam, info, transactions, and one time password's, using the multi-layer perceptron (MLP) algorithm. In this approach, each SMS is classified into one of the predefined categories. This experiment was conducted on the "multi-class SMS dataset" with 7,398 messages, which are differentiated into 5 classes. The accuracy obtained from the experiment was 97%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Budi Padmaja

Department of Computer Science and Engineering, Institute of Aeronautical Engineering

Hyderabad, Telangana, India

Email: b.padmaja@gmail.com

1. INTRODUCTION

The advancement of communication technology and extensive use of mobile phones has brought instant messaging; short message service (SMS) to take the top place among the various other modes of communication because of its low cost, easy usage, and fast messaging rate. To send and receive SMS, both the sender and the receiver must be within a proper network and no internet connection is required. Owing to its popularity and affordable service of telecommunication, the service of SMS is widely used by government organizations, banks, and businesses to communicate information or notify their users or clients. According to the research work carried out earlier, the categorization of emails and SMS are not alike. Emails are provisioned with sophisticated techniques [1] to differentiate between messages by using content and metadata. On the other hand, SMS spam filtering is yet not potent as most techniques that label SMS [2]–[4]. They carry the limitation of performing feature engineering manually; this manual feature extraction is considered to be an inefficient approach. In an attempt to use traditional machine learning techniques, one has to have good knowledge of the problem working area or domain. Without this prior knowledge, designing and selecting features is considered a difficult task [5].

The standard of the machine learning model depends not only on the dataset but also on how efficiently the patterns are encoded in the features of the data. Identification of appropriate features demands a good grip over that domain and requires prior knowledge of expertise. The components selected must be re-examined based on the feature importance graph and information gain. Only after this it is achievable to filter out the features that are useful for the classification and those that are not. It is an arduous and time-wasting process to iteratively examine the features by this trial-and-error method [6]. One of the possible ways to eliminate the inefficiency of feature extraction can be done using automated feature extraction.

Most of the research work carried out for the classification of SMS was binary classification (i.e., into ham and spam). Spam refers to the irrelevant messages [7]. Ham is labeled for general communication among people and contains the desired information. This categorization is further extended by classifying SMS data into multiple classes; most are based on machine learning algorithms. The classification of SMS is a very old problem, and numerous research has been performed to solve this problem using various algorithms and datasets.

Various machine learning algorithms [8], [9] are used to build a SMS classification model. They discussed in depth different classifiers viz support vector machine (SVM), naive Bayes (NB), decision tree (DT), logistic regression (LR), random forest (RF), AdaBoost, artificial neural network (ANN), and convolutional neural network (CNN). The experimentation was done on two different datasets like “SMS spam collection V.1”, “SMS spam collection database” and “Spam SMS Dataset 2011-12” from Kaggle for training and testing the model. In [9], the total work is defined with 4 phases: experimental database, feature creation, feature selection (CV), and comparison of algorithms. A deep learning model is used for addressing the same problem [10]. They used convolutional neural network classifier, to train and test the model with “Tiago’s dataset”. Their model consisted of two convolutional layers each with 32 filters, a kernel of size 3, the activation function used was rectified linear unit (ReLU) and the max pooling had a pool size of 2. They suggested future research with the improvement of the architecture of the CNN network. A new method GentleBoost classifier (boosting classifier) [11] is introduced to overcome the issues of computation and space complexities. They trained and tested the model with “Tiago’s dataset”. The model consisted of several steps which include tokenizing the message followed by feature extraction. On comparing their model with naïve Bayes, they concluded that GentleBoost took less space and computational complexity. In [12], a model is built with different machine learning algorithms such as naive bayes, support vector machine, and maximum entropy classifier on the “SMS spam collection” dataset. Their model mainly consists of two steps, tokenization, and formation of DocumentTermMatrix. A deep-learning model on “UCI SMS Spam” using the recurrent neural network (RNN) algorithm is proposed [13] for identifying ham and spam messages. They considered different parameters for their model like batch size, max sequence length, RNN size, embedding size, min word frequency, learning rate, training data, and test data. They examined a particular training algorithm, and a cycle, or period of applying the considered algorithm on all training vectors which is known as an epoch. A model on “UCI SMS Spam” dataset is built and validated with different deep learning and machine learning algorithms [14]. They used naive bayes, logistic regression, k-nearest neighbors, decision tree classification, support vector machine, random forest, and long short-term memory (LSTM) classifiers. Artificial neural network classifier [15]. The model accuracy is validated on “UCI benchmark dataset” with deep learning and machine learning algorithms [16]. They used naive naïve Bayes (NB), random forest (RF), gradient boosting (GB), logistic regression (LR), stochastic gradient descent (SGD), convolutional neural network, and long short-term memory. Their model includes multiple steps for the creation of a word matrix, feature extraction, and classification. The problem is addressed with multinomial naïve Bayes (MNB), support vector machine (SVM), random forest (RF), and Adaboost classifiers [17]. The message dataset is collected from volunteers, and students attending Thapar University [18]. A deep analysis of machine learning and deep learning algorithms used to address the same problem, and concluded that SVM and NB are mostly used, but SVMs are directly proportional to the size of the dataset, SVMs are not based on probability hence they are not suitable for the classification of labels. The performance of the model is improved with optimization techniques like particle swarm optimization [19], artificial bee colony [20], bat algorithm [21], bees algorithm [22], cat swarm algorithm [23], approximate muscle guided beam search [24] are some of the recently best-proven algorithms for classification problems. A step-by-step process of SMS classification model on “UCI SMS Spam” with a word embedding technique term frequency and inverse document frequency (TF-IDF) [25]. They have used 5 different machine learning algorithms in this model-multinomial naïve Bayes (MNB), k-nearest neighbor (KNN), SVM, DT, and RF. The random forest algorithm has achieved the highest accuracy among all the algorithms. The only drawback of this method is that the dataset is imbalanced, so it is not enough to evaluate the performance based on accuracy. A variety of deep learning algorithms like LSTM and gated recurrent unit (GRU) [26] are used and compared the results between the proposed model and machine learning algorithms like SVM and NB proposed by Almeida and Hidalgo. The analysis of the performance of model on “SMS spam collection v.1” dataset using naïve Bayes classifier [27] is outperformed to classify spam and ham messages with other classification algorithms. A response model for SMS [28] was developed with alert 1 (does not need immediate attention), alert 2 (requires attention within the day), and alert 3 (requires immediate attention ASAP). They used

the NB algorithm for the classification of SMS. A binary classification of SMS into spam and ham messages [29], by SVM algorithm on the “UCI SMS SPAM dataset” is proposed with fusion of natural language processing (NLP) techniques for pre-processing. An extended multi-class model by using the logistic regression algorithm for classifying SMS into regular, info, ads, and fraud is developed [30]–[34]. It is implemented on the real-time dataset by collecting messages from multiple volunteers. The performance is compared with different algorithms like KNN, DT, and NB. Table 1 shows the summary of the related works discussed above.

In summary, the manuscript is organized into the following sections: section 2 explains the design and implementation of the proposed SMS auto-response classifier with model refinement parameters. Section 3 depicts the methods involved in the auto-response SMS categorization model focusing on content preprocessing and feature extraction methods content-based and semantic-based. Section 4 illustrates the results and discussion with model performance and validation of real-time data.

Table 1. Summary of related works

Author Names	Dataset considered	Algorithms used	Accuracy
Gupta <i>et al.</i> [8]	SMS spam collection V.1, Spam SMS dataset 2011-12	SVM, NB, DT, LR, RF, AdaBoost, ANN, CNN	99.10% (CNN)
Sethi <i>et al.</i> [9]	SMS spam collection database	NB, RF, LR	98.445% (NB)
Popovac <i>et al.</i> [10]	Tiago’s dataset	CNN	98.4%
Akbari and Sajedi. [11]	Tiago’s dataset	GentleBoost	98.30%
Navaney <i>et al.</i> [12]	SMS Spam Collection	NB, SVM, MAXENT	97.4% (NB)
Taheri and Javidan [13]	UCI SMS Spam	RNN	98.11%
Gadde <i>et al.</i> [14]	UCI SMS Spam	NB, LR, KNN, DT, SVM, RF, LSTM	98.5%
Sheikhi <i>et al.</i> [15]	UCI SMS Spam	ANN	98.8%
Roy <i>et al.</i> [16]	UCI Benchmark Dataset	NB, RF, GB, LR, SGD, CNN, LSTM	99.44% (CNN)
Agarwal <i>et al.</i> [17]	Messages collected from volunteers, students attending Thapar University	MNB, SVM, RF, AdaBoost	98.23% (SVM)
Abdulhamid <i>et al.</i> [18]	NA	Survey of multiple datasets and multiple algorithms	NA
Sarif <i>et al.</i> [25]	UCI SMS Spam	MNB, SVM, RF, KNN	97.5% (RF)
Poomka <i>et al.</i> [26]	an SMS spam dataset proposed by Almeida and Hidalgo	LSTM, GRU	98.18 (LSTM)
Mathew and Issac [27]	SMS Spam Collection v.1	NB	97.15%
Ordoñez <i>et al.</i> [28]	NA	NB	89%
Jain <i>et al.</i> [29]	UCI SMS Spam	SVM	98.7%
Dewi <i>et al.</i> [30]	Dataset from seven volunteers	LR, NB, KNN, DT	97.5% (LR)

2. SMS AUTO-RESPONSE CLASSIFIER

The auto-response classifier uses a multi-layer perceptron (MLP) a feedforward neural network (FNN) model. The MLP model is built from three layers- an input layer, and hidden layer(s) and followed by an output layer. The first layer; the input layer is responsible for acquiring input data (SMS) that must be classified. After the input layer comes an arbitrary number of hidden layers that make up the computational engine of the multi-layer perceptron model. The output layer is the final layer which performs the classification and labeling of the SMS into its respective class/category. The data flow is from the input layer to the output layer (i.e., forward propagation), which is why the MLP model is considered to follow the feed-forward networking method. The nodes (neurons) of the model are capable of analyzing continuous-natured functions and then performing tasks such as pattern recognition, prediction, and classification. Next on testing and training datasets, feature extraction is done. The word vector is created based on the training dataset, after which two-word count matrices are created by transforming the training and testing parts of the dataset. The next step is to train the MLP model; a multi-layer perceptron has 3 different parts and for each input, there is one neuron (or node) associated. It also has one output layer with a single node. For each output, it can have any number of hidden layers, and each hidden layer can have any number of nodes. When it comes to the classification of SMS into multiple classes, the word count matrix obtained from the feature extraction step is passed to the MLP model as an input, each cell in this matrix is considered as a node of the input layer, and whether each node of the input layer is provided with a random weight. These weights with some bias values help in determining the output value from each node. The output of the node is generally some integer value but for classification, we need binary results (i.e., whether to consider the output of node 1 or not to consider 0) for this purpose we use the activation function. The activation function decides whether to activate a neuron or not. Here the bias is used to shift the curve of the activation function from right to left or left to right based on which the neuron gets activated or deactivated. Figure 1 represents the complete process that occurs within a single node to produce an output. The individual node mathematical function is represented in (1).

$$\sum_{i=1}^n (w_i x_i) + Bias + Activation Function \tag{1}$$

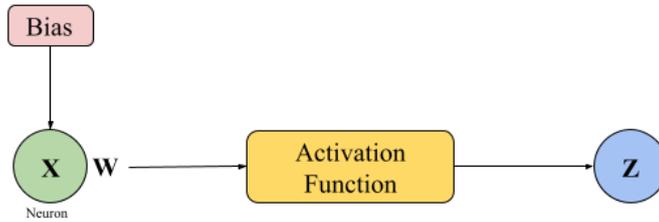


Figure 1. Structure of neuron

In Figure 2, Z indicates the output passed to the next hidden layer. N is the number of node in that hidden layer. W indicates the random weights assigned to each neuron. X is the value of the node/neuron. From each node, the output can be expected based on the node value obtained from $Y = WX + Bias$. This is similar to the straight-line equation, $y = mx + c$. Based on the node value, the value of Y is defined and mapped onto the activation function curve, which decides to activate or deactivate a node.

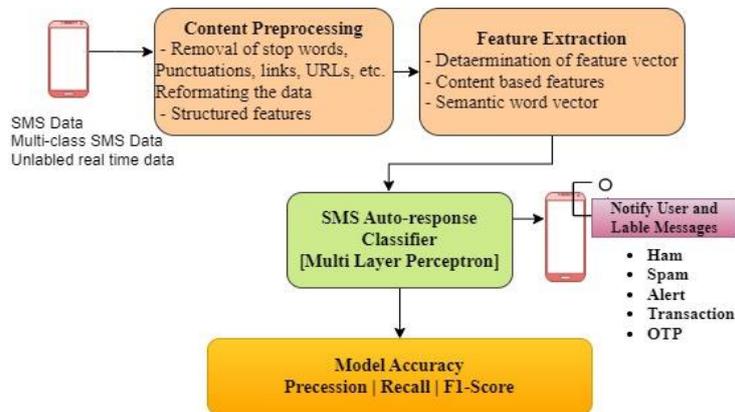


Figure 2. Schematic diagram of the proposed auto-response SMS categorization model

The various mathematical equations used in this experimentation as activation functions for impacting the dimensions of the features. The identity activation function is a simple linear function that returns the same value as its input. The mathematical representation for the identity activation function is given in (2).

$$f(x) = x \tag{2}$$

The ReLU activation function is a commonly used activation function in neural network models. It is a nonlinear function that applies a threshold operation to the input, setting any negative values to zero and leaving positive values unchanged. The mathematical equation of ReLU is given in (3).

$$f(x) = \max(0, x) \tag{3}$$

The logistic activation function is commonly used in classification problems. It is a sigmoid function that maps any real-valued input to the range [0, 1]. The mathematical equation for the logistic activation function is given in (4).

$$f(x) = 1/(1 + \exp(-x)) \tag{4}$$

The hyperbolic tangent (tanh) activation function is a commonly used activation function in neural networks. It is a sigmoid function that maps any real-valued input to the range [-1, 1]. The mathematical formula for the tanh activation function is given in (5).

$$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x)) \quad (5)$$

The results from the classifier vary based on the number of hidden layers used and the size of each hidden layer. By default, only one hidden layer is used with size 100 (max_iter), it is the epoch for the classifier (by default the value is 200) and the default activation function is rectified linear activation function (ReLU [21]). A detailed view of the different accuracies based on certain parameters and changing the hyperparameters are discussed in further sections.

3. METHOD

The proposed auto-response SMS categorization model has been built to predict multiple classes based on the semantics (information) embedded in it. In this approach, each SMS message is classified into one of the pre-defined categories -ham, spam, info, transactions, and one-time password (OTPs). The process includes identification of the SMS dataset, formatting and preprocessing the dataset, semantic-based feature extraction, and Auto-response SMS classification using MLP on both training and testing data. The model outputs a SMS class after performing validation with unlabeled and real time data. The model is built by using the MLP algorithm. The MLP model is built with at least three layers namely the input layer, hidden layer(s), and the output layer. Each layer is defined as a collection of perceptron's (nodes). Every node is associated with an activation function that determines the node to which the output of the current node must be transferred. The entire dataset is first preprocessed and formatted by editing the column names and adding extra columns if necessary. The formatted data is then fractionated as training data and testing data. As the name defines, training data is used to train the MLP model, whereas the testing data is used to examine/test the model that is trained. After this comes the feature extraction. In this, the strings (SMS text) are converted into vector format which is used as an input to the MLP model. The MLP model applies bias values and activation functions on the nodes to find out the class to which the SMS belongs. There are five main classes into which each SMS can be uniquely classified, they are ham, spam, info, transactions, and OTPs. The entire process flow is shown in Figure 2.

3.1. Dataset

To perform this experiment the dataset used is multi-class SMS dataset [20]. The data set consists of 5 categories of messages, they are ham (1,729), spam (1,772), alerts (465), OTP's (1,625), and transactions (1,807), a total of 7,398 messages. The unlabeled data used in this study consists of a collection of real-time messages obtained from students at the Institute of Aeronautical Engineering [IARE]. This data set comprises 100 messages, with 20 messages from each class of SMS, including personal messages, multiple transactions, OTPs for various purposes, spam messages, and SMS from various shopping sites. The unlabeled data is utilized to test the model's performance.

3.2. Data preprocessing

In this phase, the original data has been cleaned and reformatted with the selected feature set. In data cleaning, punctuations like !"#\$%&'()*+,-./:;<=>?@[\] ^ ` { } are removed and then all the stop words are removed from the messages. After removing stop words and punctuations, all the messages in the dataset are converted to lowercase to ensure that all the words in a message fall under the same letter case. Once the data cleaning process is completed a clean and noise-free dataset is available. Now data formatting is performed on cleaned data with transformation, renaming of the existing columns, and newly derived features. The message "type" column is renamed as "label No" and the "message" column as "text". And newly derived features are the "category" and "length" of each message. The "category" is derived by the type of the message based on the "label No" value i.e., 0: ham, 1: spam, 2: alerts, 3: transactions, 4: OTPs. The "length"s column consists of the length of each message in the dataset. The columns that are newly created are used for analysis, creating training vectors, and visualizations. Table 2 shows the dataset after data formatting.

Table 2. Structure features

label_num	Text	Category	Length
0	Go until Jurong point, crazy . . Available on...	ham	116
0	Ok lar ... Joking wif u oni ...	ham	31
1	Free entry in {}NUMBER{} a wkly comp to win {}MIS...	spam	210
0	U dun say so early hor ... U c already then sa...	ham	51
0	Nah I do n't think he goes to usf , he lives a...	ham	63

3.3. Feature extraction

The process of converting raw data into numerical values that can be utilized/processed by a classifier is called feature extraction. This feature extraction process ensures that there is no loss of information from the raw data. In our case, the raw data is the available dataset consisting of the messages which are in the form of strings. This raw data is not accepted by the ML model, it cannot be used directly for training and testing of the model. Hence there is a need to extract the numerical features from the raw data. The numerical features from the messages will be the frequency count of each word in the dataset and the label num values (0: ham, 1: spam, 2: alerts, 3: transactions, 4: OTPs). The feature extraction process is done in the following two steps: Step 1 is creation of semantic-word vector, semantic-based word vector and the transformation of the text document. The creation of a semantics-based word vector is a map of words where each word in the dataset or the text document is considered as a key, and the index of each word is based on the ordering of the words is assigned as the value, before preparing this word vector, stop words are removed from the dataset or the text document. Step 2 is transformation of text document, semantic-word vector obtained from the above step is used to transform the text document or the dataset into the matrix where each column of the matrix is a word from the text document or the dataset, each row represents a particular sentence or text sample in the text document and each cell represents the count of each word in each sentence of the text document. Now these obtained matrices or vectors are provided to the model for training and classification as shown in Figure 2.

3.4. SMS categorization model

The SMS categorization model is constructed with the help of a multi-layer perceptron by considering both content-based and semantic-based features. The response to the model is labeled as 5 classes viz. Ham, spam, alert, transaction, and OTP. The evaluation of the proposed auto-response SMS categorization model is observed with the help of precision, recall, and F1 score.

4. RESULT AND DISCUSSION

The proposed SMS Auto-response model is built using an MLP classifier on multi-class SMS dataset. The model is constructed with a minimal number of layers. The activation function used is ReLU and up to 6,000 epochs model is iterated to find the optimal response.

4.1. Dataset statistics

The dataset comprises a total of 7,398 messages, including 1,729 ham messages, 1,772 spam messages, 465 alert messages, 1,807 transaction messages, and 1,625 OTP messages. The testing and training data are detailed in Table 3. The formatted SMS data is split into two segments- testing and training. In this experiment, the test size parameter is observed to be 0.2 which indicates 20% of the dataset messages are used for testing and the remaining 80% of messages are used for training the MLP model.

Table 3. Training and testing dataset details

Training dataset		Testing dataset	
Category	Count	Category	Count
Transaction (3)	1,440	Transaction (3)	367
Spam (1)	1,429	Spam (1)	343
Ham (0)	1,382	Ham (0)	347
OTP (4)	1,298	OTP (4)	327
Alerts (2)	369	Alerts (2)	96
Total	5,918	Total	1,480

Figure 3 shows the visualization of semantics involved in various categories of SMS messages in the form of the word cloud which depicts the most repeated words of each class. The analysis of the word cloud reveals that certain words, such as “number” and “person”, are predominantly repeated in the ham class, this class mainly consists of messages related to general communication. In contrast, the spam class contains numerous words that are repeated with similar frequency, such as “date” and “organization” and many more which are generally fraud messages. The alerts class consists of words like “dear”, “customer”, and “time”, which are related to providing information to customers, while the transaction class is associated with words like “duration”, “ac”, “number”, and “debited”, these messages are related to financial transactions. Finally, the OTP class is distinguished by the frequent repetition of words like “OTP”, “code”, and “verification”. More details about prominent semantics involved in each category are given in Table 4.

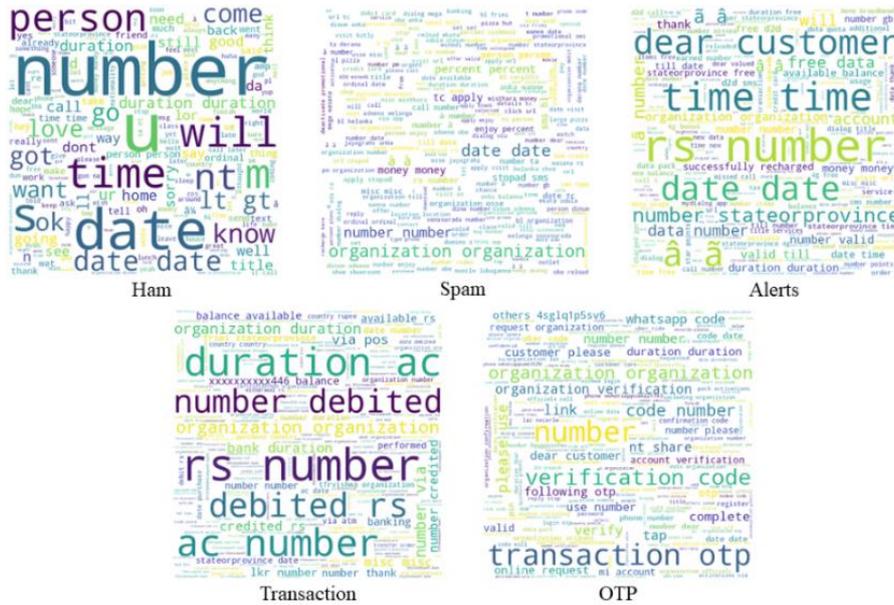


Figure 3. Illustration of category-wise semantics visualization in word cloud

Table 4. Category wise semantic information

SMS category	Semantic information	Prominent semantics involved
Ham	General communication-various words commonly used in daily conversations	“person,” “come,” “number,” “ok,” “date,” and “time”
Spam	Does not serve any meaning	“organization,” “tc apply,” “money,” and “date”
Alert	Sent by companies to contact their customers	“dear,” “customer,” “number,” “recharge,” “successfully,” and “time”
Transaction	Involve bank transactions where money is debited or credited	“ac,” “number,” “debited,” “credited,” “balance,” and “available”
OTP	Used for verification or validation purposes	“otp,” “code,” “verification,” “number,” “organization,” “valid,” and “verify”

4.2. SMS auto-response model accuracy

MLP Classifier has achieved the highest accuracy of 97% when max iter is 3,000, activation is “ReLU”, and hidden layer sizes are 150, 100, 50, 25. Later a confusion matrix is plotted, and the below confusion matrix in Figure 4 shows the number of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) produced by the classification model. These values are used for calculating the accuracy of the model.

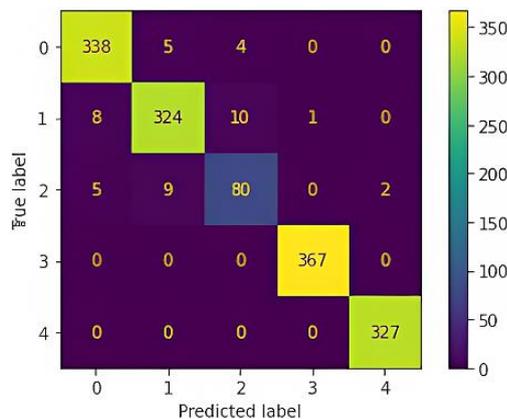


Figure 4. Confusion matrix

From Figure 4, it is identified that out of 347 ham messages 338 are identified as ham, 5 are identified as spam, 4 are identified as alerts, and 0 are identified as transactions and OTPs. Out of 343 spam messages 324 are identified as spam, 8 are identified as ham, 10 are identified as alerts, 1 is identified as transactions, and 0 as OTPs. Out of 96 alerts, 80 are identified as alerts, 5 are identified as ham, 9 are identified as spam, 0 are identified as transactions, and 2 are identified as OTPs. Out of 367 transactions, all are accurately identified as a transaction. Out of 327 OTPs all are accurately identified as OTPs.

Table 5 shows precision and recall values for each category. Transactions and OTPs are SMS of high priority which are perfectly classified by the model. Ham and spam are personal messages whose precision and recall values are greater than 0.9. Alerts are considered to be least priority messages as they contain advertisements. The precision and recall of alerts are around 0.8.

Next several accuracy comparisons have been performed which are mentioned below. Figure 5 shows a comparison between accuracy and hidden layer sizes when tested with different activation functions like ReLU, logistic, Tanh, and identity, the results of each activation function are represented with different colors. It is observed that the highest accuracy of 97% is achieved by ReLU activation function when hidden layer sizes are fixed to {150, 100, 50, 25}.

From Figure 5 it is observed that the highest accuracy is achieved when the activation function is set as RELU and hidden layer sizes are set as 150, 100, 50, 25. Hence these values are considered and kept constant for comparison between Epoch and accuracy. Table 6 shows the accuracy trend with respect to number of Epochs, by keeping the activation function and hidden layer sizes constant.

Table 5. Classification report

Category	Precision	Recall	F1-Score
Ham (0)	0.96	0.97	0.97
Spam (1)	0.96	0.94	0.95
Alerts (2)	0.85	0.83	0.84
Transactions (3)	1.0	1.0	1.0
OTP's (4)	0.99	1.0	1.0

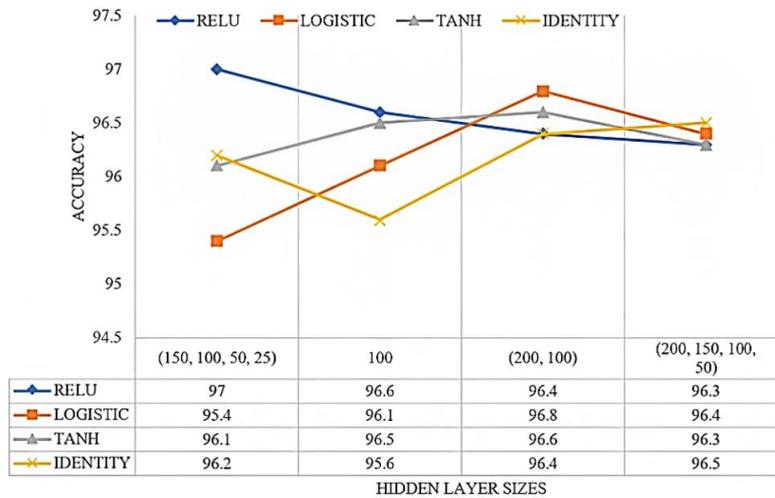


Figure 5. Accuracy vs hidden layer sizes

Table 6. Epoch vs accuracy

Max Iterations (Epochs)	Accuracy (%)
500	96.5
1,000	96.6
2,000	96.6
3,000	97.0
5,000	96.4
6,000	96.6

Figure 6 shows a comparison between the accuracies of algorithms used in various papers. From the research work that is done, it is identified that multi-class SMS datasets related to the existing research either

are not provided, or the dataset is constructed by researchers themselves which is not publicly available. Hence the comparison is done with the algorithms used in previous works based on the existing MULTI-CLASS SMS DATASET. The highest accuracy (97%) was achieved by multi-layer perceptron (MLP) and linear regression (LR). Among MLP and LR, MLP model is given more preference due to the drawbacks of linear regression in terms of classification problems. Linear regression works only on continuous values, which is not the case for classification (i.e., classification works on discrete values). Another drawback is that the threshold value must be shifted each time new data points are incorporated into the graph. The next highest accuracy was achieved by SVM at 96.8%. Although the classification accuracy of SVM is high, it is not suitable for large datasets due to its training complexity. The next algorithm compared is NB which has an accuracy of 96.4%. Regardless of its accuracy, NB follows bad estimation as it works on the assumption of independence between predictors.

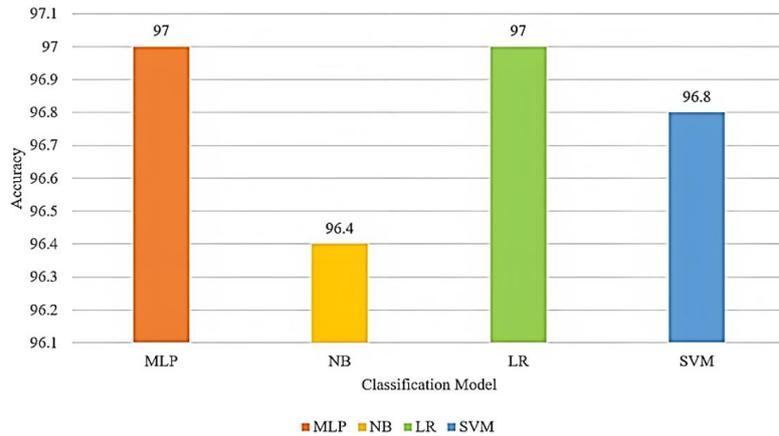


Figure 6. Accuracy comparison

4.3. Model validation-accuracy of unlabeled data

The unlabeled data set comprises 100 SMS messages, with 20 messages from each of the ham, spam, alerts, transactions, and OTPs classes. These messages were obtained from students at the Institute of Aeronautical Engineering (IAE) and are not pre-classified into specific SMS categories. Instead, the messages were directly fed to the model for classification, and the results were compared to manual classification. Figure 7 depicts the accuracy of the model with unlabeled data illustrated in the confusion matrix. In this, 0 indicates ham class; 1 indicates spam class; 2 indicates alerts class; 3 indicates transactions class; and 4 indicates OTPs class. The results indicate that out of the 20 ham messages, 17 were correctly classified as ham, while 2 were misclassified as spam and 1 as alerts. Of the 20 spam messages, 15 were accurately classified as spam, 4 were misclassified as ham, and 1 was misclassified as alerts. Out of the 20 alerts, 12 were correctly classified as alerts, 3 were misclassified as ham, 4 were misclassified as spam, and one was misclassified as OTP. For the 20 transactions, 16 were classified correctly as transactions, 2 were misclassified as ham, 1 was misclassified as alerts, and 1 was misclassified as OTP. Finally, all 20 OTP messages were correctly classified as OTPs.

	0	1	2	3	4
0	17	2	1	0	0
1	4	15	1	0	0
2	3	2	15	0	0
3	2	0	0	18	0
4	0	0	0	0	20
True Label	0	1	2	3	4
	Predicted Label				

Figure 7. Illustration of accuracy of SMS auto-response model with unlabeled data

5. CONCLUSION

The SMS auto-response model helps the users to distinguish between the various messages received over their mobile phones. This model categorizes SMS messages into five main classes-ham, spam, info, Transactions, and OTPs based on the semantics involved in the messages. A neural network model: multi-layer perceptron is implemented on the multi-class SMS dataset to address the task of message categorization. The five-class text message classification runs successfully with an accuracy of 97% with MLP. The limitation of the work is that it is mainly dependent on the text messages received in the English language only. Thus, this paper could extend the future study to incorporate similar neural network algorithms to classify SMS written in various languages. This paper also can be extended by employing various deep-learning techniques for text message classification.

ACKNOWLEDGEMENT

The authors would like to extend their sincerest gratitude to all those who have supported and contributed to the development of this research paper. This research has been supported by the Computer Science and Engineering Department of the Institute of Aeronautical Engineering (IARE).

REFERENCES

- [1] H. Drucker, Donghui Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999, doi: 10.1109/72.788645.
- [2] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, "Spam detection using ensemble learning," in *Advances in Intelligent Systems and Computing*, vol. 741, Springer Singapore, 2019, pp. 661–668.
- [3] Z. Chen *et al.*, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, vol. 433–434, pp. 346–364, Apr. 2018, doi: 10.1016/j.ins.2017.04.044.
- [4] L. Chen, Z. Yan, W. Zhang, and R. Kantola, "TruSMS: a trustworthy SMS spam control system based on trust management," *Future Generation Computer Systems*, vol. 49, pp. 77–93, Aug. 2015, doi: 10.1016/j.future.2014.06.010.
- [5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [6] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic LSTM for spam detection," *International Journal of Information Technology*, vol. 11, no. 2, pp. 239–250, Jun. 2019, doi: 10.1007/s41870-018-0157-5.
- [7] A. Kumar, J. P. Singh, and N. P. Rana, "Authenticity of geo-location and place name in tweets," *AMCIS 2017 - America's Conference on Information Systems: A Tradition of Innovation*, 2017.
- [8] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A comparative study of spam SMS detection using machine learning classifiers," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Aug. 2018, pp. 1–7, doi: 10.1109/IC3.2018.8530469.
- [9] P. Sethi, V. Bhandari, and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," in *2017 International Conference on Computing and Communication Technologies for Smart Nation*, Oct. 2018, pp. 28–31, doi: 10.1109/IC3TSN.2017.8284445.
- [10] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Convolutional neural network based SMS spam detection," in *2018 26th Telecommunications Forum (TELFOR)*, Nov. 2018, pp. 1–4, doi: 10.1109/TELFOR.2018.8611916.
- [11] F. Akbari and H. Sajedi, "SMS spam detection using selected text features and boosting classifiers," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, May 2015, pp. 1–5, doi: 10.1109/IKT.2015.7288782.
- [12] P. Navaney, G. Dubey, and A. Rana, "SMS spam filtering using supervised machine learning algorithms," in *2018 8th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, Jan. 2018, pp. 43–48, doi: 10.1109/CONFLUENCE.2018.8442564.
- [13] R. Taheri and R. Javidan, "Spam filtering in SMS using recurrent neural networks," in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, Oct. 2017, pp. 331–336, doi: 10.1109/AISP.2017.8515158.
- [14] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS spam detection using machine learning and deep learning techniques," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2021, pp. 358–362, doi: 10.1109/ICACCS51430.2021.9441783.
- [15] S. Sheikhi, M. T. Kheirabadi, and A. Bazzazi, "An effective model for SMS spam detection using content-based features and averaged neural network," *International Journal of Engineering*, vol. 33, no. 2, pp. 221–228, Feb. 2020, doi: 10.5829/ije.2020.33.02b.06.
- [16] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS Spam," *Future Generation Computer Systems*, vol. 102, pp. 524–533, Jan. 2020, doi: 10.1016/j.future.2019.09.001.
- [17] S. Agarwal, S. Kaur, and S. Garhwal, "SMS spam detection for Indian messages," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, Sep. 2015, pp. 634–638, doi: 10.1109/NGCT.2015.7375198.
- [18] S. M. Abdulhamid *et al.*, "A review on mobile SMS spam filtering techniques," *IEEE Access*, vol. 5, pp. 15650–15666, 2017, doi: 10.1109/ACCESS.2017.2666785.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [20] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: 10.1007/s10898-007-9149-x.
- [21] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Studies in Computational Intelligence*, vol. 284, Springer Berlin Heidelberg, 2010, pp. 65–74.
- [22] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm-a novel tool for complex optimisation problems," in *Intelligent Production Machines and Systems*, Elsevier, 2006, pp. 454–459.
- [23] S.-C. Chu, P. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Pacific RIM International Conference on Artificial Intelligence*, 2006, pp. 854–858, doi: 10.1007/978-3-540-36668-3_94.

- [24] H. Jiang, S. Zhang, Z. Ren, X. Lai, and Y. Piao, "Approximate muscle guided beam search for three-index assignment problem," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8794, Springer International Publishing, 2014, pp. 44–52.
- [25] N. N. A. Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, 2019, doi: 10.1016/j.procs.2019.11.150.
- [26] P. Poomka, W. Pongsena, N. Kerdprasop, and K. Kerdprasop, "SMS spam detection based on long short-term memory and gated recurrent unit," *International Journal of Future Computer and Communication*, vol. 8, no. 1, pp. 11–15, Mar. 2019, doi: 10.18178/ijfcc.2019.8.1.532.
- [27] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Dec. 2011, vol. 1, pp. 101–105, doi: 10.1109/ICCSNT.2011.6181918.
- [28] A. Ordonez, R. E. Paje, and R. Naz, "SMS classification method for disaster response using naïve Bayes algorithm," in *2018 International Symposium on Computer, Consumer and Control (IS3C)*, Dec. 2018, pp. 233–236, doi: 10.1109/IS3C.2018.00066.
- [29] T. Jain, P. Garg, N. Chalil, A. Sinha, V. K. Verma, and R. Gupta, "SMS spam classification using machine learning techniques," in *2022 12th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, Jan. 2022, pp. 273–279, doi: 10.1109/Confluence52989.2022.9734128.
- [30] F. K. Dewi, M. M. R. Fadhlurrahman, M. D. Rahmaniato, and R. Mahendra, "Multiclass SMS message categorization: Beyond spam binary classification," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2017, pp. 210–215, doi: 10.1109/ICACSIS.2017.8355035.
- [31] S. Merugula, B. Padmaja, and R. Veerubommu, "RideNN-OptDRN: heart disease detection using RideNN based feature fusion and optimized deep residual network," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 28, Dec. 2022, doi: 10.1002/cpe.7355.
- [32] B. Padmaja, M. B. Myneni, and E. Krishna Rao Patro, "A comparison on visual prediction models for MAMO (multi activity-multi object) recognition using deep learning," *Journal of Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00296-8.
- [33] P. B. M. M. N. M, N. V. Somalaraju, M. Kovuri, and K. Sriramwar, "Chest X-Ray image analysis for respiratory disease prediction using Grad-CAM," in *2023 2nd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, Feb. 2023, pp. 1–7, doi: 10.1109/DELCON57910.2023.10127464.
- [34] B. Padmaja, C. H. V. K. N. S. N. Moorthy, N. Venkateswarulu, and M. M. Bala, "Exploration of issues, challenges and latest developments in autonomous cars," *Journal of Big Data*, vol. 10, no. 1, May 2023, doi: 10.1186/s40537-023-00701-y.

BIOGRAPHIES OF AUTHORS



Budi Padmaja    is currently working as an associate professor and head, CSE (Artificial Intelligence and Machine Learning), Institute of Aeronautical Engineering, Hyderabad, Telangana, India. She has received her B.Tech. from North Eastern Regional Institute of Science and Technology (NERIST), Arunachal Pradesh, India in May 2001. She completed her M.Tech. from School of Information Technology, Jawaharlal Nehru Technological University Hyderabad (JNTUH), and Hyderabad, India in 2010. She was awarded the Ph.D. degree in computer science and engineering in 2021 by JNTUH, Hyderabad. She has vast teaching and research experience of 20 years. She has published more than 25 research papers in various international journals and presented more than 15 papers in various international conferences. She is also a reviewer for 08 International Journals. Her current areas of research interest include machine learning, deep learning, computer vision, and social network analysis. She is a life member of ISTE, CSI, IAENG and CSTA. She can be contacted at email: b.padmaja@gmail.com.



Myneni Madhu Bala    working as a professor of computer science and engineering at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad. She received her Ph.D. in computer science and engineering from Jawaharlal Nehru Technological University Hyderabad (JNTUH). She has twenty-one years of academic and research experience. Her research interests are data science frameworks, image mining, text mining, machine learning, artificial intelligence, deep learning, and data analytics. She has published 57 articles in reputed Journals indexed in SCOPUS, and SCI. She has published 2 patents. She is the Principal Investigator of DST funded project on sustainable smart city development. And has received various grants from AICTE for organizing short term training programs, infrastructure development, and Faculty Development Programs. And selected a part of AICTE national mission programs such as student learning outcomes assessment (SLA) and technical book writing (TBW). She is a reviewer for Elsevier, Springer and more indexed journals. She acted as session chair, organizing member, and advisory member for various international conferences. She delivered various invited talks on data modelling, data science, and analytics. She is a life member of professional bodies like CSI and ISTE, Sr. Member for IEEE, WIE and international association IAENG, ICST, SDIWC. She can be contacted at email: baladandamudi@gmail.com.



Epili Krishna Rao Patro    is a faculty member in the Department of Computer Science, Institute of Aeronautical Engineering, Hyderabad, Telangana, India. He has received his MCA from Anna University, Chennai and M.Tech. in Computer Science and Engineering from JNTUH, Hyderabad, India. Currently he is pursuing his Ph.D. in intrusion detection system using machine learning from VelTech University, Chennai. He has more than 22 years of teaching experience and published 10 research papers in various international journals and conferences. He can be contacted at email: patrosir@gmail.com.



Adiraju Chaya Srikruthi    is a B.Tech. student in the Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India. Her research interests are artificial intelligence, machine learning and deep learning. She can be contacted at email: chayasri32@gmail.com.



Vytila Avinash    is a B.Tech. student in the Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India. His research interests are artificial intelligence, machine learning, and deep learning. He can be contacted at email: avinashvytila351@gmail.com.



Chenumalla Sudheshna    is a B.Tech. student in the Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India. Her research interests are artificial intelligence, machine learning, and deep learning. She can be contacted at email: Sudheshna1503@gmail.com.