JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

A ROBUST APPROACH FOR EFFECTIVE SPAM DETECTION USING SUPERVISED LEARNING TECHNIQUES

[1]K.A.RAHAMAN,[2]DR.S.GOVINDA RAJULU,[3]GUMMA VAJJULA SINDHUJA

[13]Assistant Professor,[2]Professor

Department of ECE

Dr. K V Subba Reddy Institute Of Technology

ABSTRACT

In this age of popular instant messaging applications, Short Message Service or SMS has lost rele-vance and has turned into the forte of service providers, business houses, and different organizations that use this service to target common users for marketing and spamming. A recent trend in spam messaging is the use of content in regional language typed in English, which makes the detection and filtering of such messages more challenging. In this work, an extended version of a standard SMS corpus containing spam and non-spam messages that is extended by the inclusion of labeled text messages in regional languages like Hindi or Bengali typed in English has been used, as gath-ered from local mobile users. Monte Carlo approach is utilized for learning and classification in a supervised approach, using a set of features and machine learning algorithms commonly used by researchers. The results illustrate how different algorithms perform in addressing the given challenge effectively.

Keywords: Spam detection, supervised learning, regional spam, Monte Carlo approach, deep learning, convolutional neural networks, SMS spam, TF-IDF vectorization

I. INTRODUCTION

Man is a social animal, and the very essence of this socializing nature lies in their ability to effectively communicate. From the cave drawings in early ages to the blazingly fast instant messaging applications prevalent in these times, the need for effective and timely commu-nication has always been a priority in human life. The basic components of a typical communication are as shown in Figure 9.1 where a communication medium is used by sender(s) to communicate with the receiver(s). This medium of communication has taken several forms over the many decades of human civi-lization. For instance, cave walls, letters (pages), and text messages are all different forms of communication medium that man has used.

Figure .1 Components of a communication.

With the onset of mobile technology in human lives, the concept of hand-written let-ters was replaced by a new form of communication, referred to as the Short Message Service or SMS. The first instance of sending a mobile device–based text message was recorded in the year 1992 [1], and it has come a long way since then. This service gained popularity at a very rapid rate, and became an integral part of technology enriched human life in the last two decades. Using the SMS, each mobile device user can compose a textual message of length up to 160 characters including alphabets, numeric values, and special symbols [2]. This constitutes the "short message" that can be sent to a recipient (another mobile device user). This mode of communication has utility especially in cases where short pieces of information need to be urgently conveyed or where attending calls is not plausible.

However, the last decade has witnessed the meteoric rise in the use of internet-based messaging services which are faster and cheaper than SMS in most cases. Also, such ser-vices are made more attractive with no message length limit, inclusion of stickers, GIFs, and other application specific enhancements to make them the primary choice of mobile-based communication. This has pushed the erstwhile default communication medium to a secondary position, and nowadays it is seldom used in day-to-day communication by general mobile users. Instead, this service has become a handy tool for different ser-vice and/or product-based companies, who use it to implement their strategy of direct marketing.

The SMS-based marketing strategy adapted by different companies provides a unique opportunity to identify and incite their potential clients by providing them attractive incen-tives and offers on chosen products or services. A recent survey revealed that 96% of the participants from India admitted they receive unwanted spam message every day, of which 42% receive almost 7 such SMS per day [3]. Despite the regulatory and preventive norms put in place by the Telecom Regulatory Authority of India (TRAI) on the broadcast of unwanted messages, only about 6% of Indian mobile users find the Do Not Disturb (DND) service useful [4].

A general understanding of spam as unwanted or unsolicited messages is essential in order to effectively prevent or detect and filter such messages at the user end. Oblivious mobile users are highly prone to signing up for such irritating SMS automatically when they are availing a service or purchasing a product of their choice. Online marketing, banking, telecom service, etc., constitute a

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

bulk of the unwanted or spam messages that Indian users usually receive. Yet more harmful is the set of fraudulent spam messages that target inno-cent users and aim to lure them and extract crucial information regarding their personal details, banking passwords, etc., as shown in Figure 9.2.

On the other hand, the desired electronic texts that a mobile user expects to receive are called ham messages. Such SMS could be bank account related updates or travel ticket-based information, etc. So, it is essential to accurately distinguish between these two types of SMS. Typically, the SMS-based communication including spam filtering may be illus-trated as represented as shown in Figure 9.3.

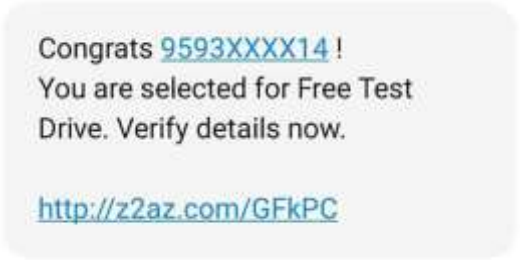

Figure: 2 An example of malicious spam message.

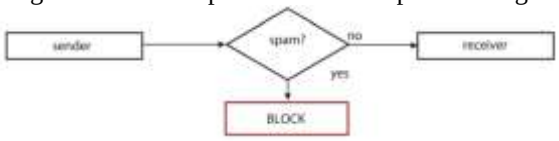

Figure.3 Flowchart of spam filtered communication system.

Over the years, there has been extensive research on different spam detection and fil-tering techniques, though not all of them have resulted in efficient and productive end user applications. The current work deals with the determination of robustness of the com-monly used classification algorithms consisting of conventional machine learning classifier models as well as contemporary Deep Neural Network architecture–based models. This is undertaken by utilizing the Monte Carlo approach by performing the training and classifi-cation tasks on different combinations of both spam and ham data for up to 100 times. As a result, the definitive performance statistics for each classification model can be realized and the best performing model may be chosen as the ideal one. The state of the art of research on spam identification has been discussed in the following Literature Review.

II. LITERATURE REVIEW

In this section, the authors have discussed some recent state of the art research works in the field of spam detection on SMS messages, going up to the last 5 years. The discussed works have proposed and implemented novel features, effective processing techniques and different advanced machine learning algorithms toward developing an efficient SMS spam recognition system.

Back in 2015, Agarwal et al. [5] utilized the comprehensive data corpus consolidated by [6] and extended it by adding a set of spam and ham SMS collected from Indian mobile users. They demonstrated how different learning algorithms like Support Vector Machine (SVM) and Multinomial Naïve Bayes (MNB) performed on the Term Frequency–Inverse

Document Frequency (TF-IDF)–based features extracted from the corpora. Starting at around this time, a plethora of research works have used the same corpus and similar set of features and learning algorithms for designing spam detection systems. In the follow-ing set of similar works, it is observed that a set of learning and classification algorithms are used for a performance comparison study. Also, there is a paradigm shift toward neu-ral network-based learning algorithms in more recent times.

In such a work in 2017, Suleiman et al. [7] demonstrated a comparative study of the performance of MNB, Random Forest, and Deep Learning algorithm–based models by using the H2O framework and a self-determined set of novel features on the same SMS corpus.

Using word embedding features, Jain et al. [8] showed in 2018 how Convolutional Neural Network (CNN) can be utilized to achieve a better performance than a number of other baseline machine learning models in determining the spam messages from the corpus of [6]. In the same year, Popovac et al. [9] illustrated how CNN algorithm performs on the same SMS corpus using TD-IDF features.

In 2019, Gupta et al. [10] proposed a voting ensemble technique on different learning algorithms, namely, MNB, Gaussian Naïve Bayes (GNB), Bernoulli Naïve Bayes (BNB), and Decision Tree (DT) for spam identification using the same corpus.

The trend of classifier performance comparison continues till recent times in 2020, where the work by Hlouli et al. [11], illustrated how Multi-Layer Perceptron (MLP), SVM, k-Nearest Neighbors (kNN), and Random Forest algorithms perform on the same SMS corpus for detecting spam and ham using Bag of Words and TF-IDF–based features. In a similar contemporary work, GuangJun et al. [12] highlighted the performance of kNN, DT, and Logistic Regression (LR) models on SMS spam corpus, though the feature extraction techniques were not discussed.

A recent but different type of work by Roy et al. [13] shows how the same SMS corpus by Hidalgo et al. [6] is classified using Long Short Term Memory (LSTM) and CNN-based machine learning models with a high accuracy. The authors also noted that dependence on manual feature selection and extraction results often influences the efficacy of the spam detection system and

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

consequently utilized the inherent features determined by the LSTM and CNN algorithms.

Another interesting observation stems from the inclusion of SMS content in languages other than English for spam and ham identification, as undertaken by Ghourabi et al. [14] in their recent work. The authors used TF-IDF and word embedding–based features for the conventional machine learning models (such as SVM, kNN, DT, and MNB) and proposed CNN-LSTM hybrid model, respectively. This is the only recent research work that intends to identify the spam content in non-English language from a multi-lingual corpus.

III. MOTIVATION

It is observed that in spite of the comparative study of classification performance under-taken by the aforementioned state-of-the-art works, none of them have attempted to deter-mine and establish the robustness of the classification techniques in spam identification. Also, the abundance of spam messages in regional language (typed in English) is largely ignored in such works.

Taking cue from the aforementioned observations, the authors have made the following contributions in the current work:

1. We introduce the novel context of identifying spam and ham SMS in regional languages that are typed in English, along with the general English corpus of spam and ham by extending it.
2. We employ a Monte Carlo approach to repeatedly perform classification using different machine learning algorithms on different combinations of spam and ham text from the extended corpus (with k-fold cross-validation for a large value of k = 100) in order to determine the efficiency of baseline learning algorithms in comparison to the CNN-based model.

IV. SYSTEM OVERVIEW

The current work follows a series of steps as illustrated in Figure 9.4. The corresponding discussions are provided in the following sections.

The SMS corpus with added Indian context, described in Section 9.5, is initially pro-cessed in a series of operations as discussed in Section 9.6. The processed text corpus is then vectorized and the TF-IDF vector is determined for the corpora as its feature. This procedure is illustrated in the Section 9.7. The different supervised learning techniques that have been trained and evaluated in this work are discussed in Section 9.8. The experimental setup and evaluation techniques have been discussed in Sections 9.9 and 9.10, respectively. The experimental results and analysis have been provided in Section 9.11. The adaptation of proposed framework in cloud architecture is discussed in brief in Section 9.12. Finally, the concluding remarks have been offered in Section 9.13.

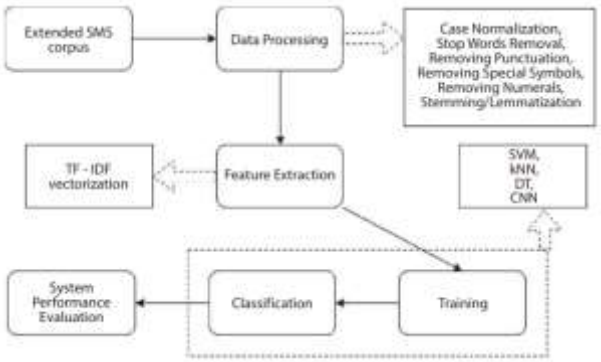

Figure.4 Overview of the system.

DATA DESCRIPTION

The authors have utilized the comprehensive and popular SMS dataset made available by [6], which has been the choice of many state-of-the-art works in this domain even in recent years. This spam and ham text corpus is compiled using free sources on the World Wide Web and corresponds to SMS messages from places like the United Kingdom (UK) and Singapore. The corpus originally consists of 5,574 texts written in English and each such message is appropriately labeled as a spam message or a ham message. The authors have further extended this data set during a period of 2 years by introducing the context of Indian spam messages. The speciality of the collected corpus is that it consists of spam SMS that the Indian companies and service providers use for marketing purposes. These mes-sages are mostly regional language-based texts that are in typed in English (shown in Figure 9.5), while some of them are written in English (shown in Figure 9.6). This set of SMSs has been collected from the students and faculties of Techno India University, West Bengal. The final corpus prepared and used in this work consists of more than 7,000 text messages. The data corpus has two columns corresponding to the message text and the label, respectively. This corpus is processed as described in the next section.

Data Processing

As evident from the illustrated spam SMS screenshots, each text contains different types of symbols, numeric values, as well as English and regional language–based words typed in English. In order to be able to extract meaningful features for the classifiers to learn, these pieces of text need to be thoroughly cleaned or normalized. Unlike usual sensory data cleaning, in such cases, the elimination of outliers and value standardization is not appro-priate. The process of text normalization or cleaning ensures that all possible "noise" is

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

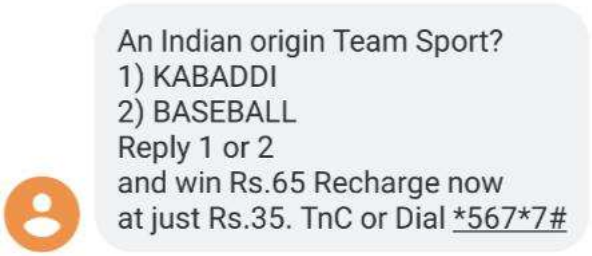

Figure .5 A sample spam message in regional language typed in English.

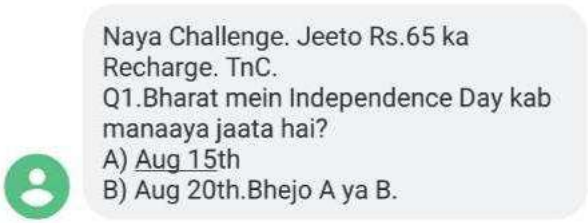

Figure.6 A sample spam message in regional language typed in English.

eliminated from the data, semantic identification and grouping is feasible, and the inherent features of the text data may be accurately represented in a simpler manner as features for the learning algorithms.

The data corpus used in this work is available in the form of a text file, which has been read and converted to a data-frame in using python-based library called pandas. The two columns of this data-frame correspond to the label and the text, and for each row in this data-frame, a set of operations is performed which are described below:

i. Case normalization: Each text message may contain different words that are capitalized, or some of whose alphabets are capitalized. For the current purpose of spam identification, there is no requirement of maintaining this distinction among words of the same meaning. For instance, different SMS texts may contain the same word "boy" written as "BOY", "Boy", "BoY", etc. In such case, it is prudent to normalize the case of each occurrence of this word such that all have the same common form "boy". In this manner, every word in every text is normalized to lower case. This is the first step that has been used to process each text in our SMS corpus.

ii. Stop-word removal: Every piece of text commonly contains a number of "stop-words", a term coined by Hans Peter Luhn which signifies their essential lack of importance in text processing. Such words are generally removed during text processing as they only contribute to computational complexity and hardly contribute to the task of learning spam or ham fea-ture, while their contribution to syntactical understanding of sentence structure is undeniable. Some common stop-words are "a", "an", "the", "I", "to", "because", "for", "in", etc. As a second step of data processing, we have completely eliminated the stop words from the text output of the previous step.

iii. Removing punctuations and special symbols: The usual messages contain a good number of punctuation marks and special symbols such as ",", "/", "?", "!", "@", "%", and "&". These symbols, though essential for sentence struc-turing, do not actually contribute much to the determination of spam or ham related characteristics. Hence, as a third step of text normalization, all such punctuation marks and special symbols are removed from each piece of text from the previous step.

iv. Removing numerals: Numbers are also widely present in SMS text mes-sages, both spam and ham. They, too, do not contribute to the training of classification algorithms and subsequent identification of spam or ham messages. Thus, we have removed all numerals present in each text message in the extended corpus used in this work.

v. Stemming or Lemmatization: In English, there is a scope of extracting the "root form" or "word of origin" for words in a particular piece of text. These "root forms" are derived using the "inflected" state of the word that is in use in the SMS text. For instance, the words "running", "ran", "runs", etc., all have the same "word of origin" that is "run". This operation has been performed for each SMS text in our corpus in order to generate an inter-mediate form of processed text. However, it is to be noted that this step is not much instrumental on SMS texts in the current work, as most of them are usually types in colloquial English, and in our case, they also consist of regional words typed in English.

After the aforementioned processing techniques have been carried out on each text in our extended corpus, it is the further used for feature extraction as discussed in the next section

Feature Extraction

The processed SMS data is to be finally used by the mathematical model–based supervised learning algorithms. These algorithms fail to deal with textual content in the data and are more comfortable with numeric values. This method of converting a text to vector rich, directly classifiable form is called vectorization. In essence, each piece of text is converted to a matrix of numbers, and every row of this matrix corresponds to a particular label which in our case is restricted to ham and spam.

Though there is a plethora of such vectorizers that may be used for transforming texts to classifiable form, not all of them are effective in every case. For the currently under-taken work, the authors have chosen to use a common vectorization technique called TF-IDF [15]. Its efficiency has been noted in state-of-the-art research works studied in the Literature Review.

In this vectorization technique, each document is transformed into a document vector, where each element

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

is the statistic derived between every term in the vocabulary and the document itself. This method of textual feature extraction makes use of the importance of each term to the document as a whole. The TF-IDF vector for a particular document j and the number of constituent terms i is calculated after the process of determining the TF and IDF is carried out, as discussed below:

• Term Frequency (TF): The individual words in a piece of text document can be individually extracted as "tokens" or "terms' in a process of "toke-nization". In this work, a word level tokenizer is utilized as the texts also contain colloquial English and regional words typed in English. The num-ber of times that each such term occurs in the document as a whole is determined as its "Term Frequency". Mathematically, TF can be defined as in Equation:

$$TF = \frac{Number\ of\ times\ a\ token\ appears\ in\ the\ corpus}{Total\ number\ of\ tokens\ in\ the\ corpus}$$

•Inverse Document Frequency (IDF): The measure of the importance of any term or token in a particular document can be analyzed using a simple logic. In natural language processing, this importance is said to be decreasing with the increasing frequency of the term in a particular corpus. This same logic can be mathematically expressed as in Equation:

$$Importance\ of\ token\ i \propto \frac{1}{frequency\ of\ token\ i\ in\ the\ document}$$

It follows that the IDF can be determined as in Equation:

$$IDF = log_{10} \frac{Total\ number\ of\ documents}{Number\ of\ documents\ in\ which\ token\ i\ appears}$$

• Calculation of TF-IDF: Finally, the consolidated TF-IDF value needs to be determined. It signifies the weightage of the TF of every token i in the doc-ument j corresponding to the IDF value of the token i. This ensures that the least common words in the document possess more weightage and vice versa. The mathematical expression of TF-IDF score for every word in the docu-ment is expressed as in Equation:

$$TF - IDF = TF\ x\ IDF$$

The vectorized data corpus is further used during experimentation by the different mathematical learning algorithms for system performance evaluation, as discussed in the succeeding sections.

Learning Techniques Used

The duly processed and vectorized, feature-rich, labeled text corpus is used for training the following classification algorithms in a supervised approach. The notion of supervised classification is that an algorithm will attempt to learn from a given set of labeled inputs and then use that knowledge to further determine the class of a new set of observed data. In the current work, the problem of determining spam and ham SMS has been addressed in this manner.

Support Vector Machine

The SVM algorithm [16] trains a model using labeled data to find an optimal plane of separation that can be used to classify the new test data. In our case of a binary prob-lem, the objective is to identify a hyperplane which has maximum distance from data points of both the classes during training. This hyperplane ensures the presence of a max-imum number of possible points of a class on one side, given two separate classes of data. Mathematically, the hyperplane can be represented by Equation where the value of ||w|| is to be minimized:

$$w^T x + b = 0$$

where, w is the weight vector, x is input vector, and b is the bias.

Specific data points that lie closest to the hyperplane act as support vectors and help in determining the performance of the classifier. These support vectors are determined by Lagrangian multiplier method. In the current work, the SVM algorithm with a linear kernel has been employed, as the text corpus is feature-rich and the data is linearly separable.

k-Nearest Neighbors

The kNN classification technique [17] attempts to learn the nearness or proximity between a set of points, in order to determine their individual class as class label. This class label is chosen by a simple voting mechanism where the class with maximum number of votes in the defined neighborhood is chosen as the class of the element vector. Certain stan-dard distance metrics are popularly used for determining this proximity, for example, the Manhattan distance, which is calculated as in Equation:

$$f = \sum_{i=1}^{k} |x_i - y_i|$$

In the processed and labeled corpus, the kNN algorithm is used to learn and find the nearest class label for every vectorized text during experimentation.

Decision Tree

The DT classifier is based on a cascading tree structure, where the whole corpus is broken down into smaller subsets, with increasing depth. The leaf nodes correspond to the class labels, and all internal decision nodes represent the tests on attributes. Beginning with the entire data at the root node, the decision rules are applied at each node, to predict the out-come and generate the next node.

The current work uses Classification and Regression Tree (CART)–based [18] DT algo-rithm with Gini index

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

as cost function, which is given by the formula in Equation:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

Convolutional Neural Network

Mostly used with two-dimensional vectors such as images, the Convolutional Neural Network [19] learning mechanism can also be utilized in training a classifier to recognize spam and ham text messages in a corpus. Inherently, every neural network has a set of lay-ers which can be configured as required, and the layers are capable of extracting features from the input data and map them to the respective classes/labels.

In any neural network, generally, the layers are fully connected with corresponding activation functions, whereas in CNNs, there are convolutional layers, pooling layers, etc., along with fully connected layers in the end. The convolutional layers repeatedly use filters on the input data to generate feature-based maps for the specific training data. In all the experiments performed in this work, the CNN classifier utilizes the concept of early stop-ping, where incremental weight updation and testing is performed until there is no loss minimization for a limit of 10 epochs. This helps to eliminate over-fitting and subsequent poor performance of the classifier.

The architecture of CNN-based models can be customized as per requirement of the given problem statement, and the CNN architecture used in this work is as described below.

The first two sections consist of convolutional layers, each of which uses the given ker-nel values to convolve the input (or previous layer's output stream). Also, in each layer, the output is mapped by Rectified Linear Unit (ReLu) activation function [20], given by Equation:

$$relu(x) = \max(0, x)$$

This activation function re-enforces non-linearity in the feature set after the convolution operations.

• First Layer: A 1D convolutional layer that takes the vectorized spam and ham texts as input streams of data. A total of 32 feature detectors or filters have been used here, each with a kernel size of 3. This allows the layer to learn 32 different basic features from the input data.

• Second Layer: The neuron matrix from the first layer is fed in to this CNN layer, and 64 new filters are used for training, with 3 kernels in use.

• Third Layer: After two convolutional layers, a max-pooling layer [21] is used, which helps to eliminate the dependence on the feature position in the vec-tor. This is obtained by down-sampling the data from the last layer while retaining the effective features. The down-sampling is done with a sliding window of height 2 across the data, such that it is replaced by the maximum value. In the process, 50% of the data in the neuron matrix is discarded.

• Fourth Layer: The next layer is dropout [22], which makes our CNN model ignore certain neurons randomly during training by assigning them zero weight. This ensures that there is a proper learning of the model, less sensi-tivity to minute variations in data or specific neuron weights, thus preventing over-fitting. With the use of dropout, the classifier is also able to perform bet-ter when tested with new data. The authors have experimentally determined a dropout rate of 0.4, meaning 40% of data values are given a zero-weight value.

• Fifth Layer: The flatten layer is used next to convert the previous output to a 1D vector such that it can be directly fed to the succeeding fully connected network.

• Sixth Layer: The final section consists of two fully connected (dense) layers with an intermediate normalization layer. The first dense layer takes the flat-tened feature vector as input and applies the ReLu activation function on this input, with 100 neurons as output. The result is normalized by scaling it to a mean of zero and unit standard deviation in the intermediate layer. As the addressed challenge is a two-class problem, the final dense layer takes the normalized feature vector and applies the Sigmoid activation function to predict the probability as output (between 0 and 1). These probabilistic values along with the true labels are then used by the cost function for model performance evaluation.

The architecture of the CNN classifier designed for this spam classification problem is illustrated in Figure 9.7

Experimental Setup

The labeled and vectorized spam and ham SMS texts are used in the experiment where the previously discussed classifiers are trained on the data and then their classification per-formance is recorded. In each case, a k-fold cross-validation technique is used to split the complete feature vector into training and testing sets randomly. This ensures that there is no bias in the trained models, no dependence on the particular splits of data, and no persisting holdout problem.

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

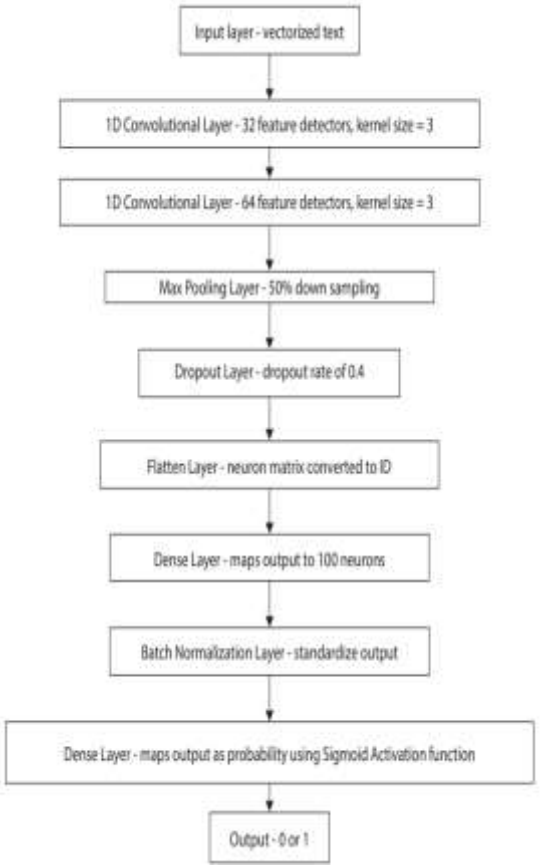

Figure .7 Architecture of designed CNN classifier.

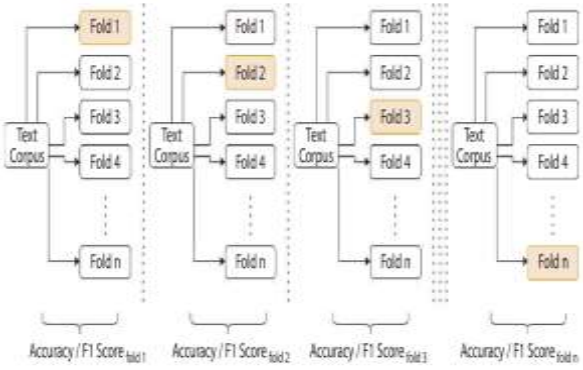

Figure .8 Illustrative example of the k-fold cross-validation technique.

The k-fold cross-validation technique may be better realized with the help of a simple example, as shown in Figure 9.8. Assuming the determined number of folds is k, the com-plete text corpus will be broken into k equal parts. From this point onward, an iterative process will follow. In the first iteration, all but the first fold of data will be utilized for training the model, while the first fold will serve the purpose of testing model performance. Similarly, in the second iteration, all but the second fold of data will be used for model train-ing and the second fold itself will be utilized for model evaluation via testing. This iterative process continues for all the k-folds of data, and the model performance is expressed as the mean of model performance in all k cases.

In this work, the classification and evaluation are designed in a Monte Carlo [23] approach where the concept of repeated random sampling is used for training and testing the classifiers. As already stated, the k-fold cross-validation is a standard technique that is very popularly used with the supervised learning techniques. However, a low standard value of 5 or 10 folds is employed in the majority of research works. This minimum number of folds of training and evaluation, while computationally easy, is not appropriate to provide a proper idea regarding the robustness of the classification model in use, i.e., they cannot be judged properly.

In order to address this issue, the authors have performed the experiments using the set of previously discussed classifiers, and by training and testing them for a high number of randomly sampled sets of spam and ham data. For this purpose, a substantially high value of k is used in k-fold cross-validation, which has been set at a maximum of 100. Each classi-fier is thus trained and tested on random splits using the cross-validation technique where the value of k is kept between 10 and 100, with intervals of 10 folds. Also, the evaluation of classifier performance has been done with the help of a set of standard evaluation metrics, as discussed in the next section.

Evaluation Metrics

To serve the purpose of determining classifier performance, a set of standard metrics need to be utilized such that the results may be considered as valid. It enables a comparison of the system performance with that of the state-of-the-art in SMS message-based spam recogni-tion. The set of such metrics used in this work, are discussed below:

• Accuracy denotes the proper and accurate representation of each event detec-tion, given by Equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

•Precision denotes the correctly classified events out of all detected events, given by Equation:

$$Precision = \frac{TP}{TP + FP}$$

It is defined as ratio of all true positives to all events flagged as positives by the classifier.

• Recall denotes the ratio of correctly detected events of all true events, using Equation:

$$Recall = \frac{TP}{TP + FN}$$

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

It is defined as ratio of all true positives to all events flagged as positives by the classifier.

• F1 score: this is a measure of the harmonic mean of the precision and recall determined in the previous steps, calculated using Equation:

$$F1\ score = 2x\frac{Precision\ x\ Recall}{Precision + Recall}$$

In each of the above metrics, the terms TP, TN, FP, and FN have been repeatedly used. These can be interpreted as discussed here:

• True Positive (TP) denotes the case where the model has correctly assigned the data to a particular class, i.e., the model has correctly determined a spam message as spam.

• True Negative (TN) means the model has correctly determined that the data does not belong to the class, i.e., the model has correctly determined a text as not spam.

• False Positive (FP) means the model has wrongly assigned the data to a class, i.e., the model has wrongly determined a spam message as a ham message.

• False Negative (FN) means the model has incorrectly determined that the data does not belong to the class, i.e., the model has incorrectly determined a spam message as not belonging to the class of spam messages.

The results of the experiments are expressed in terms of these metrics and illustrated in the next section.

V. EXPERIMENTAL RESULTS

The classifiers that have been previously discussed, namely, CNN, SVM, kNN and DT, are used for the system experiments. Each classifier is trained on k − 1 out of k randomly generated folds of data and evaluated on the remaining data by test-ing. Also, as discussed, the k-fold cross-validation–based evaluation is repeated for values of the fold k, where $10 \leq k \leq 100$, $k = k + 10$. The motive of this methodology is to gain clarity about the robustness of the classifier performance based on repeated random sampling using Monte Carlo approach.

The processed and vectorized corpus of spam and ham SMS is learned and classified by each classifier.

The mean of mean accuracies for all the determined folds of evaluation up to 100 is illus-trated in Table 9.1. The classification accuracies for each of the 10 steps of experiments are also visualized in the accuracy plot in Figure.

Similarly, in Table 9.2, the mean of mean F1 scores achieved by the four different classi-fication models has been illustrated. The same has been graphically demonstrated in Figure 9.10. From the figures and the tables listed above, it is obvious that CNN has the best per-formance in accurately classifying the extended spam and ham message corpus used in this work. This is true in both the cases of accuracy and F1 scores. The maximum system performance is indeed achieved by the deployed CNN classifier, with an accuracy as high as 99.85% and a mean accuracy of 99.48%. Similarly, a maximum F1 score of 99.83% is

Table.1 Mean of mean accuracies of different classifiers in k-fold cross-validation.

Classifier	k = 10	k = 20	k = 30	k = 40	k = 50	k = 60	k = 70	k = 80	k = 90	k = 100	Mean	Std. Dev.
SVM	98.08	98.31	98.34	98.39	98.54	98.35	98.42	98.56	98.48	98.51	98.42	0.15
kNN	94.71	94.87	94.62	94.84	94.72	94.91	94.78	94.77	94.48	94.88	94.76	0.13
DT	96.11	97.12	97.24	97.11	97.29	97.17	97.03	97.11	97.25	97.25	97.17	0.08
CNN	99.35	99.27	99.39	99.38	99.83	99.41	99.28	99.77	99.71	99.64	99.50	0.2

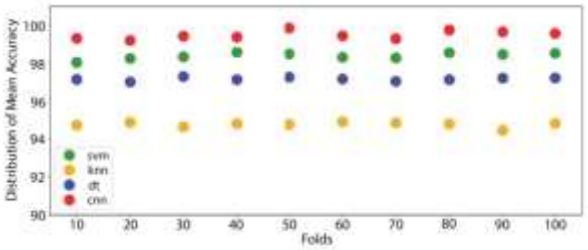

Figure.9 Mean of mean accuracies of the different classification models.

Table.2 Mean of mean F1 scores of different classifiers in k-fold cross-validation.

Classifier	k = 10	k = 20	k = 30	k = 40	k = 50	k = 60	k = 70	k = 80	k = 90	k = 100	Mean	Std. Dev.
SVM	98.04	98.25	98.34	98.37	98.49	98.31	98.28	98.55	98.46	98.52	98.38	0.16
kNN	94.73	94.88	94.65	94.82	94.77	94.91	94.86	94.79	94.45	94.83	94.77	0.13
DT	96.94	97.01	97.29	97.13	97.26	97.16	97.04	97.13	97.21	97.22	97.16	0.08
CNN	99.31	99.20	99.42	99.37	99.85	99.44	99.28	99.75	99.63	99.57	99.48	0.2

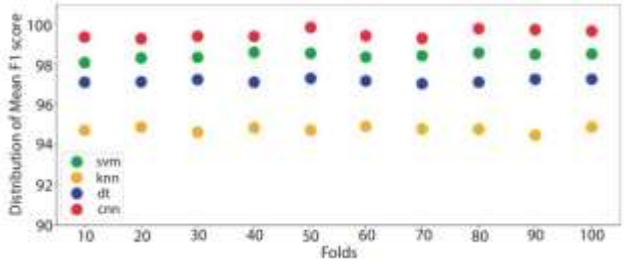

Figure .10 Mean of mean F1 scores of the different classification models.

exhibited by this classification model with a mean score of this observed performance is indeed remarkable and can be attributed to the convolution-based learning, deep and con-nected networks, and inherent error minimization capacity of the CNN model.

In contrast, the conventional machine learning algorithms fail to perform as good as the CNN model. However, it is observed that SVM performs the best among all the machine learning algorithms both in terms of accuracy and F1 scores, with mean of mean values of 98.38% and 98.42%, respectively. Thus, we can state that SVM performs robustly among all the baseline machine learning algorithms that have been tested in this work.

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

Analysis of the DT-based classification results show that it closely follows SVM in terms of F1 score and accuracy measures. This classifier model fairs well with mean accuracy and F1 scores above 97%. Consequently, the worse performance is noted on the part of the kNN algorithm–based classifier which performs with a high accuracy and F1 score of about 94% in all cases but is comparatively poorer in spam detection from the extended SMS corpus used in this work.

A statistical analysis of classifier performance in the overall experiment in terms of the mean and standard deviation of the mean accuracy and F1 score values are illustrated in Figures 9.11 and 9.12. It is evident from these figures that the CNN learning model has the best mean performance and the maximum standard deviation among all classifiers used, though the standard deviations in all cases in only fractional. The standard deviation for SVM is lower, followed by that of kNN. Notably, the standard deviation for DT is the lowest with a good classification performance. Very similar statistical values are also noted in the case of F1 score of the classifiers. The standard deviation is the same in case of mean F1 scores for all classifiers except SVM where the value is fractionally reduced.

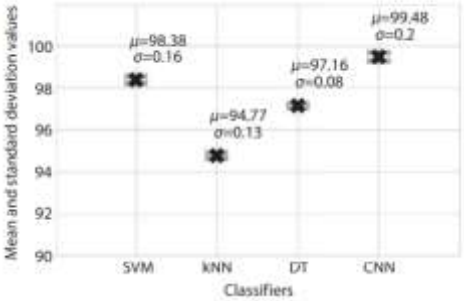

Figure.11 Statistical distribution of classifier performance in terms of mean accuracies.

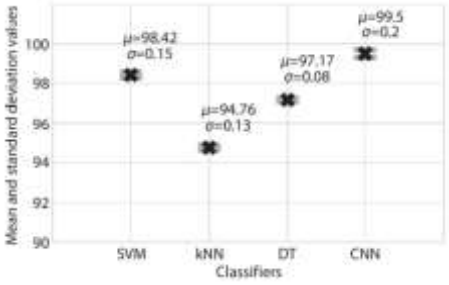

Figure .12 Statistical distribution of classifier performance in terms of mean F1 score.

Remarkably, the illustrations in Figures 9.9 and 9.10 show the efficiency and capability of the developed spam detection framework. The graphical representations are very much alike, which highlights that the adopted technique of k-fold cross-validation for high value of k (=100) has helped in eliminating the inconsistency in system performance graphs for all the classifiers, in terms of both accuracy and F1 score. Indeed, the statistical analysis in Figures 9.11 and 9.12 is evidence of the success of our implemented Monte Carlo–based testing mechanism that has resulted in a stable performance for all the classifiers, with a very irrelevant and small standard deviation in each case.

Thus, on the basis of classification performance in terms of accuracy and F1 score, we determine that the developed system is indeed robust to the underlying patterns and fea-tures of spam and ham messages of both the types—those that are originally in English and typed in English, and the ones that are originally in regional languages but typed in English.

Observations in Comparison With State-of-the-Art

A few observations can be highlighted based on results of the experiments in relation with the state-of-the-art research works conducted in the domain of SMS spam classification, as follows:

Table.3 Performance comparison with contemporary works.

Article	Performance achieved	Dataset used	Additional data (If any)
Ghourabi *et al.* [14]	Accuracy: 98.32%	Almeida *et al.* [6]	Arabic spam and ham messages
Roy *et al.* [13]	Accuracy: 99.44%		None
Popovac *et al.* [9]	Accuracy: 98.60 %		None
Jain *et al.* [8]	Accuracy: 98.65%		None
Barushka *et al.* [24]	Accuracy: 98.51%		None
Our work	**Accuracy: 99.85%**		**Regional messages typed in English**

• The superior performance of the CNN classifier is established in our exper-iments, and it can be deduced that CNN is the best candidate for building a robust spam identification system. This corroborates with the recent works [8, 9, 13, 14] that have been studied in our Literature Review. Table 9.3 high-lights the efficient performance and provides a comparison of our proposed system with these works.

• Among the conventional learning techniques, SVM shows the best and most robust performance in comparison to the other models based on algorithms of kNN and DT. This is in keeping with the observations of state-of-the-art works by Agarwal et al. [5], Jain et al. [8], El Hlouli et al. [11], and Ghourabi et al. [14]. In each of these works, it is noted that the SVM classifier model achieves the highest accuracy of classification among all other learning algorithms.

• The poor performance of DT classifier model is also substantiated by state-of-the-art research works [10, 12, 14]. Thus, it is deduced that the DT clas-sifier is not suitable for deploying a robust spam detection system. It needs to be mentioned that this is not because of poor performance of DT, but the superior learning and classification capability of such data as exhibited by CNN and other baseline machine learning algorithms.

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

Application in Cloud Architecture

The proposed system determines that the CNN-based trained classifier performs more robustly than the conventionally used machine learning algorithms, even when trained with the set of spam messages including regional texts typed in English. When adapted in the cloud, the overall system may be represented as in Figure 9.13. This architecture is loosely based on the recent patent by Skudlard et al. [25].

As seen in the figure, the mobile or portable devices DEVICE 1 and DEVICE 2 (which may be smartphones or tablets capable of sending and receiving SMS messages) are

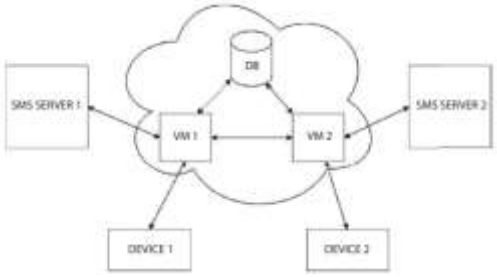

Figure .13 Illustration of the proposed CNN-based model in cloud architecture.

in communication with the virtual machines VIRTUAL MACHINE 1 and VIRTUAL MACHINE 2, respectively. Each of these VMs may be sharing the same hardware or may be based on separate pieces of hardware with the capacity for intercommunication using all standard protocols. Also, each VM maintains the trained CNN-based classifier with the data from the shared database DB.

All SMS messages from SMS SERVER 1 and SMS SERVER 2 to the user devices (DEVICE 1 and DEVICE 2) are intercepted by VIRTUAL MACHINE 1 and VIRTUAL MACHINE 2, respectively. The text is processed and classified using the trained models in the VM before transmission to the user devices. If a particular piece of SMS text is flagged as spam by the respective VIRTUAL MACHINE, then the flagged message is forwarded to the user for confirmation. Once the user confirms that the SMS is indeed spam, the shared database DB is updated accordingly. On the other hand, if in case the text is not classified as spam, then it is directly forwarded to the user. Given that the system is not ideal, i.e., it may not be 100% accurate, there may arise a case where a particular message that the user feels is spam, it is not flagged accurately. In any such case, the user has the option to report the text as spam and the bi-directional feedback system ensures that the database DB is updated accordingly. This, in turn, results in an updated classifier in the VMs, which learns the new user prefer-ence and works more efficiently thereafter.

VI. CONCLUSION

Effective spam detection and filtering is a very well visited field of research, and there is a wide variety of feasible solutions that have been proposed. It is obvious from a review of relevant, recent state-of-the-art literature that the most distinct progress is in the use of newer, advanced algorithms that are capable of learning more about the inherent patterns of different spam and ham messages in a text corpus. Such algorithms are mostly based on Neural Networks and variants of Deep Neural Networks, such as CNN and LSTM. In the current work, a spam detection system that takes as input a comprehensive and well tested SMS corpus, which has been extended by including the context of regional messages typed in English, has been designed and evaluated. The system employs a Monte Carlo approach to determine which of the supervised classification algorithms among CNN and other con-ventional machine learning algorithms like SVM, kNN, and DT and is the most robust in detecting the spam messages accurately. For this purpose, k-fold cross-validation has been utilized with a high value of k = 100, at intervals of 10 folds. It has been determined experimentally that the proposed approach results in consistent performance in case of all the classifiers and that CNN emerges as the most robust classification technique with an accuracy and F1 score about 99.5%. Also, among the conventional learning algorithms, SVM is the most robust with standard evaluation metric values of above 98%. Thus, the given novel text corpus has been effectively classified by the designed system and CNN can be utilized as a robust learning and classification technique. A cloud-based framework for implementing the proposed classifier is also discussed. In future, this work can be used as a reference for building robust, real-time spam detection and filtering systems that need to work on SMS corpora that is challenging and contains novel contexts.

REFERENCES

1. Hppy bthdy txt!, BBC, BBC News World Edition, UK, 3 December 2002, [Online]. Available: http://news.bbc.co.uk/2/hi/uk_news/2538083.stm. [Accessed October 2020].

2. Short Message Service (SMS) Message Format, Sustainability of Digital Formats, United States of America, September 2002, [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000431.shtml. [Accessed, October 2020].

3. India's Spam SMS Problem: Are These Smart SMS Blocking Apps the Solution?, Dazeinfo, India, August 2020, [Online]. Available: https://dazeinfo.com/2020/08/24/indias-spam-sms-problem-are-these-smart-sms-blocking-apps-the-solution/. [Accessed October 2020].

4. The SMS inbox on Indian smartphones is now just a spam bin, Quartz India, India, March 2019, [Online].

ISSN- 2394-5125 VOL 07, ISSUE 05, 2020

Available: https://qz.com/india/1573148/telecom-realty-firms-banks-send-most-sms-spam-in-india/. [Accessed October 2020].
5. Agarwal, S., Kaur, S., Garhwal, S., SMS spam detection for Indian messages, in: 1st International Conference on Next Generation Computing Technologies (NGCT) 2015, UCI Machine Learning Repository, United States of America, IEEE, pp. 634–638, 2015.
6. Almeida, T.A. and Gómez, J.M., SMS Spam Collection v. 1, UCI Machine Learning Repository, United States of America, 2012. [Online]. Available: http://www.dt.fee.unicamp.br/~tiago/sms-spamcollection/, [Accessed October 2020].
7. Suleiman, D. and Al-Naymat, G., SMS spam detection using H2O framework. Proc. Comput. Sci., 113, 154–161, 2017.
8. Jain, G., Sharma, M., Agarwal, B., Spam detection on social media using semantic convolu-tional neural network. Int. J. Knowl. Discovery Bioinf. (IJKDB), IGI Global, 8, 12–26, 2018.
9. Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A., Convolutional neu-ral network based SMS spam detection, in: 2018 26th Telecommunications Forum (TELFOR), Serbia, 2018.
10. Gupta, V., Mehta, A., Goel, A., Dixit, U., Pandey, A.C., Spam detection using ensemble learning, in: Harmony Search and Nature Inspired Optimization Algorithms, pp. 661–668, 2019.
11. El Hlouli, F.Z., Riffi, J., Mahraz, M.A., El Yahyaouy, A., Tairi, H., Detection of SMS Spam Using Machine-Learning Algorithms, Embedded Systems and Artificial Intelligence: Proceedings of ESAI 2019, Fez, Morocco, 1076, 429, Springer Nature, Singapore, 2020.
12. GuangJun, L., Nazir, S., Khan, H.U., Haq, A.U., Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms. Secur. Commun. Netw., Hindawi, 2020, 1–6, 2020.
13. Roy, P.K., Singh, J.P., Banerjee, S., Deep learning to filter SMS spam. Future Gener. Comput. Syst., 102, 524–533, 2020.
14. Ghourabi, A., Mahmood, M.A., Alzubi, Q.M., A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. Future Internet, 12, 156, 2020.
15. Sammut, C. and Webb, G.I., TF-IDF, in: Encyclopedia of Machine Learning, pp. 986–987, Springer, US, 2010.
16. Gunn, S.R. and others, Support vector machines for classification and regression, ISIS technical report, vol. 14, pp. 5–16, University of Southampton, UK, 1998.
17. Altman, N.S., An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat., 46, 175–185, 1992.
18. Loh, W.-Y., Classification and regression trees, in: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, pp. 14–23, 2011.
19. Goodfellow, I., Bengio, Y., Courville, A., Deep learning, vol. 1, MIT press, Cambridge, 2016.
20. Krizhevsky, A., Sutskever, I., Hinton, G.E., Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp. 1097–1105, 2012.
21. Yamaguchi, K., Sakamoto, K., Akabane, T., Fujimoto, Y., A neural network for speaker-independent isolated word recognition, in: First International Conference on Spoken Language Processing, 1990.
22. G.E. Hinton, A. Krizhevsky, I. Sutskever, N. Srivastva, System and method for addressing over-fitting in a neural network. USA Patent US Patent 9, 406, 017, 2016.
23. Kalos, M.H. and Whitlock, P.A., Monte carlo methods, John Wiley & Sons, New York, USA, 2009.
24. Barushka, A. and Hajek, P., Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Appl. Intell., 48, 1–19, Springer, 2018.
25. A.E. Skudlark, L.K. Tran, Y. Jin, Cloud-Based Spam Detection. USA Patent US Patent App. 16/901,056, 2020.