International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

SMS Spam Detection using RNN

Dr. K. Sree Ram Murthy[1], Dr.K.Kranthi Kumar[2], K.Srikar[3], CH.Nithya[4], K.Mahender Reddy[5], K.Sai Kumar[6]

[1, 2]*Associate Professor, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*
[3,4,,5,6]*B. Tech Student, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

Abstract - U*sing cell phones and smart systems has grown more and more in recent years, and short message service has become one of the most significant communication means. Eightieth of the world's active users use mobile phone sms as a communication method. Unwanted SMS messages made for the following purposes are among some of this large variety of short messages. Sending low quality SMS and plenty of incredibly low value cell SMS kit operators. Quick message has developed into a business-trade of many billion bucks. The creation of inappropriate messages for the purpose of advertisement is harassment and the source of these messages on SMS has become the main challenge during this service. Especially SMS spam is more irritating then email spam. We are inclined to use RNN to distinguish ham and spam sequences of variable length. Several tracked methods, such as the Bayes classifier, nearest neighbour, SVM, and neural networks, are used to counter this disadvantage. Additional generic techniques for classifying unwanted Text messages include SVM and Bayesian. The most effective result was ninety-two related precision. By RNN we've got acquire higher accuracy. RNN really runs computationally smart*.

Key Words: Machine learning, RNN, Spam Detection,

1. INTRODUCTION

The SPAM DETECTION IN SMS application RECURRENT NEURAL NETWORK project is used to distinguish spam and ham messages as Short Message Service (SMS) has developed into a multi-billion dollar industrial company and generates unwanted messages for advertising purposes, some messages are distracting and we waste your time seeing them[1]. To reduce this limitation we tend to build associated application which distinguishes RNN algorithm exploitation of unwanted messages with the help of keywords.

The goal in this project is to use completely different machine learning algorithms get the drawback for the classification of SMS spam, compare their performance to gain insight and explore the problem more, and style an application supported by for each of these algorithms that will filter SMS spams with high precision. We use information from the UCI machine learning archive of 5574 instant messages obtained in 2012. It includes a set of 425 physical SMS spam messages derived from the Grumble text information processing network, a subset of 3,375 indiscriminately selected NUS SMS Corpus (NSC) non-spam (ham) messages, a list of 450 non-spam SMS messages obtained from caroline Tag's Philosophy Thesis Professor, and the SMS Spam Corpus[5]. The dataset may be a large text file, during which each line begins with the message name, followed by the text message string. Once the data is preprocessed and the options extracted, machine learning techniques such as RNN, naive Bayes, SVM and various ways are applied to the samples and their performance is compared. Finally, the execution of the undertaking's best classifier is analyzed against the performance of the classifiers applied to this dataset in the first paper.

1.1 Support Vector Machine(SVM)

Support vector machine is another simple algorithmic norm which every talented AI should have in their arsenal. Help vector machine is strongly favored by many, because it achieves critical precision with less computing power. Support Vector Machine, abbreviated as SVM, is used for every task of regression and classification. But, it's commonly used in goals for classification. The aim of the support vector machine algorithm is to search for a hyperplane in the associated N-dimensional space (N — the quantity of features) which separately classifies the information points. There are several potential hyperplanes which would be chosen to separate the 2 categories of information points. Our goal is to search for a plane with the utmost margin, that is, the utmost distance between the points of knowledge of each category. Increasing the margin gap should provide some reinforcement so that future data points can be identified with extra confidence.

1.2 Random Forests

Random forests are associated with the overall methodology for classification. The set could be a combination of decision trees made from a bootstrap sample of the coaching set[5]. In addition, when designing a chosen tree, the split that is chosen when

splitting a node is that the best partition is only between a random set of features. This may increase the bias of one model, but the average reduces the variance and may also make adjustments to increase the bias. As a result, a higher model is built. In this work, the creation of random forests in the scikitlearn python library is employed, that averages the probabilistic predictions.

1.3 AdaBoost

Adaboost may be a technique that encourages ensemble building classifiers that are modified by previous classifiers in favor of misclassified instances. The classifiers that it uses will be as weak as only slightly higher than the random estimate, and will still improve the final model[5]. This technique is also used to enhance the ultimate ensemble model, in combination with alternative strategies. Sure weights are added to the coaching samples in every iteration of Adaboost. Such units of area weights distributed evenly prior to initial iteration. Then, when each iteration increases the weights for incorrectly labeled labels by current model, and weights are shriveled for properly sorted samples. This means that the current predictor focuses on previous classificator vulnerabilities. We tried Adaboost implementation using scikit-learn python library with decision trees.

1.4 KNN

The k-nearest neighbors (KNN) algorithmic rule could be a easy, easy to-implement supervised machine learning which will that may be wont to solve each classification and regression issues. A supervised machine learning algorithm rule (as unkind associate unsupervised machine learning algorithm) is one that depends on labeled input file to be told a function that produces associate applicable output once given new unlabelled information. K-nearest neighbor is applied as a basic instance-based learning algorithmic rule to classification problems. The label for a check sample is expected in this method to support the majority vote of its K-nearest neighbors [5].

1.5 Naive Bayes (NB)

On the final extracted features, the NB algorithmic rule applies. The speed and ease of this algorithmic rule along with its high accuracy make it a fascinating classifier for spam detection problems. In the form of the naive Bayes algorithmic rule with the multinomial event model, entering the message length function corresponds to the presumptuous independent associate variable Bernouli for writing every character within the text message in spam or ham messages[5].

1.6 RNN

As recurrent Neural Network (RNN) may be a kind of Neural Network where the output is taken from the preceding step is fed as input to the present step[2]. Some keywords are used to distinguish message from ham and spam. First, we need to train the dataset using keywords and then we need to upload the file and then we get the output as expected and the accuracy as well. By using RNN the precision is greater as compared with various methods such as SVM, Navie bayes, Multi NB, KNN.

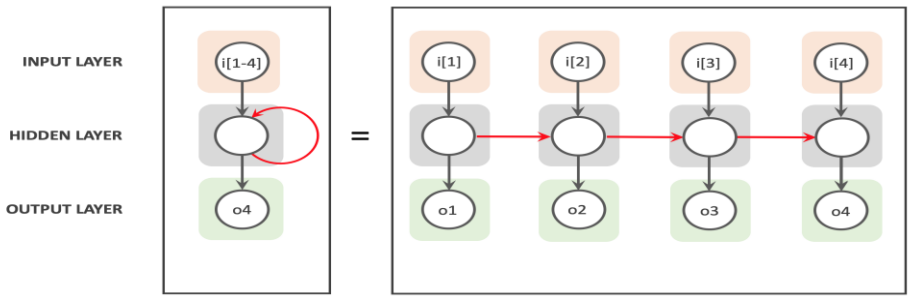

Fig-1: Architectural Design

2. EXPERMENTAL ANALYSIS

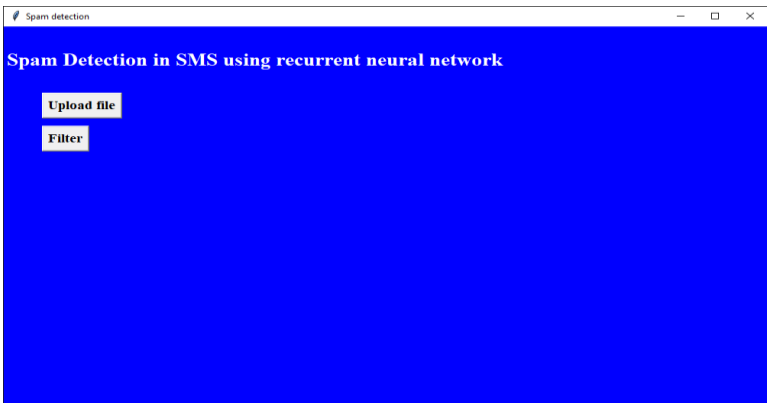

Fig -2.1: Home Page of Sms Spam Detection

After building the sequential model we get GUI page with two buttons are provided where upload file is responsible for taking the input csv file and the second button filter is used for filtering the csv file to get output. Here the capable of handling this uploading and filtering there is no necessary of any logins.

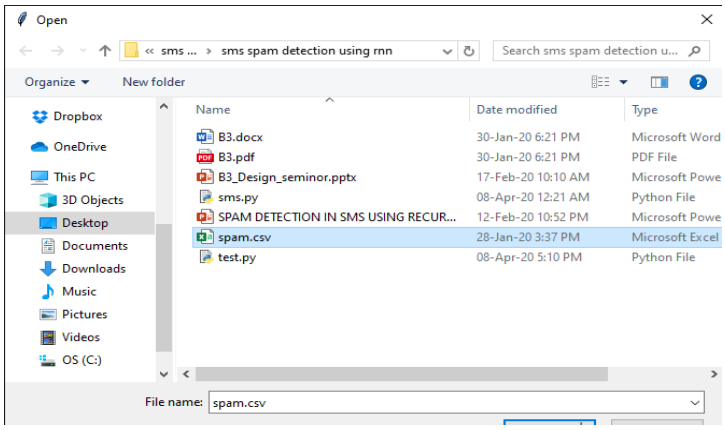

Fig -2.2: Selection of File in Sms Spam Detection

After clicking the upload file we can choose the path where the file is actually stored and upload it directly to the page. This direct uploading requires the actual file to be saved in the system before it is fed to the project.

Fig -2.3: After Selecting the Path in Sms Spam Detection

After uploading the path the file is taken and given to a variable called file name. After feeding the input location of the path is also displayed it is displayed beside the button upload file.

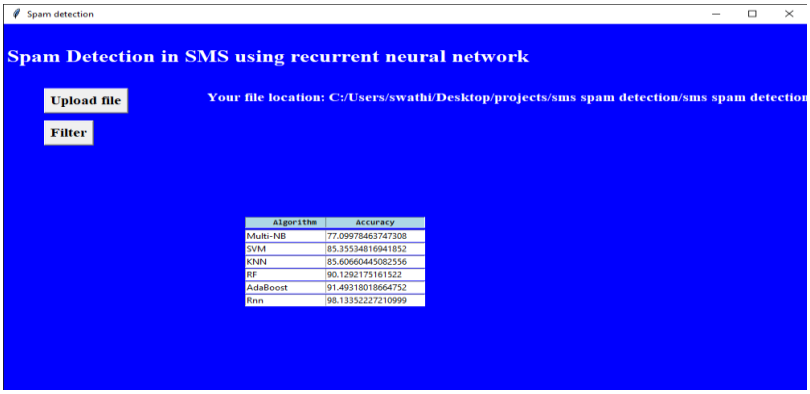

Fig -2.4: After Clicking the Filter Button

After clicking the filter button it shows the accuracy of the file in the form of table and comparison is also given in the format of table there are also 5 more algorithms which is compared with they are multi NB, SVM, KNN, RF, AdaBoost. This is tabulated and show the user which algorithm is the best.

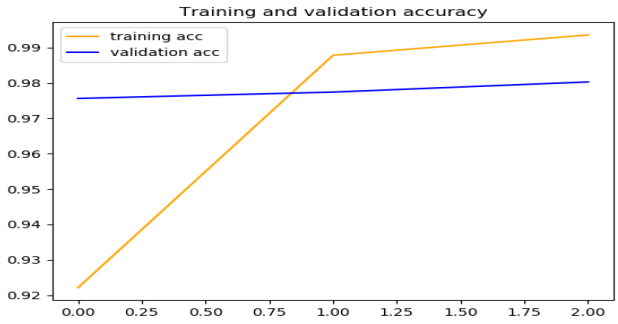

Chart -1: Rnn Training and Validation Accuracy Graph

Graph is plot Between Training and Validation Accuracy from the list of data acquired From Rnn model.

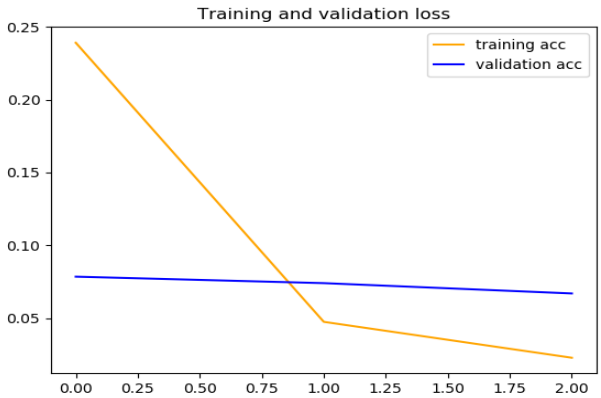

Chart -2: Rnn Training and Validation Loss Graph

Graph is plot Between Training and Validation Loss from the list of data acquired From Rnn model

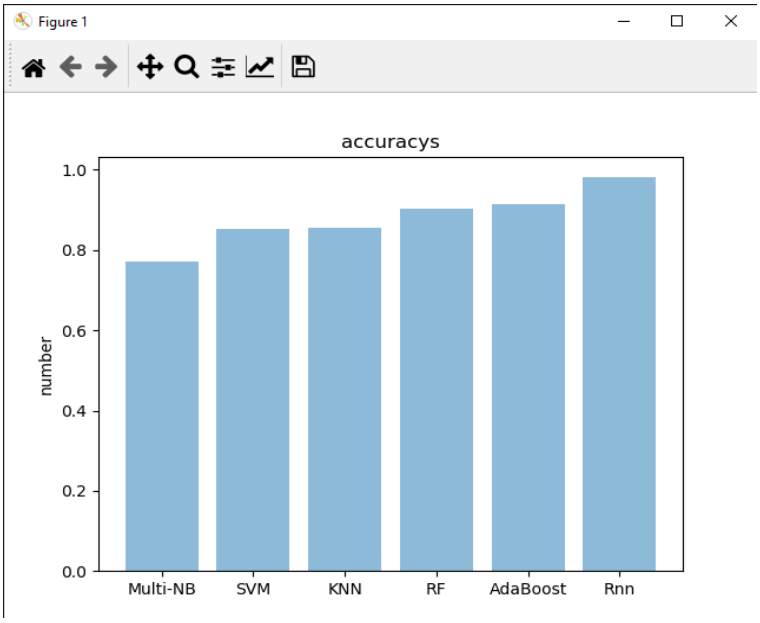

Chart -3: Bar graph Of Different Algorithms

The output is detected and graph is predicted to show the visualization view of table using this matplotlib library. This shows the detailed accuracy percentage when different algorithms are used.

Table -1: Results

Algorithm	Accuracy
Multi-NB	77.09
SVM	85.35
KNN	85.60
RF	90.12
AdaBoost	91.49
RNN	99.13

From the above table we can conclude that RNN gives best Accuracy when compared to the other algorithms.

3. CONCLUSION

Previous studies shows that there is an accuracy of below 92% by using different algorithms. Now this algorithm (Recurrent neural network) shows a high accuracy of 98%. This is also useful in detecting which message is spam and which is not spam and helps the customer to believe only ham messages and not spam. This helps users to filter their spam messages and personal messages

REFERENCES

[1] https://www.researchgate.net/publication/328759146_Spam_filtering_in_SMS_using_recurrent_neural_networks.
[2] https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network.
[3] https://www.kaggle.com/kentata/rnn-for-spam-detection.
[4] https://ieeexplore.ieee.org/abstract/document/7727636.
[5] http://cs229.stanford.edu/proj2013/ShiraniMehrSMSSpamDetectionUsingMachineLearningApproach.pdf.