SPAM Detection in SMS

Prateek Somani [1]

Abstract

In a world full of social media, messaging is very popular. Nowadays, messages and Emails are the easiest way of communication, and they both contain SPAM messages. So an accurate and precise method for SPAM detection is needed. In cryptography, most of the messages are encrypted on the network, but when they are on the device, we can decrypt the message and classify it as SPAM and non-SPAM. I have used different kinds of algorithms for this problem and compared them to various hyper-parameters.

1. INTRODUCTION

Currently, 85% of emails and messages received by mobile users are spam. Spam has a lot of definitions, Spam is a junk mail/message, or an unsolicited mail/message. Spam e-mails/message are also those unwanted, unsolicited e-mails/message that are not intended for a specific receiver. This has become a severe problem. Sending inappropriate messages to many recipients indiscriminately has resulted in users' anger but large profits for spammers. In my study, many different classifiers of machine learning and deep learning were used for the classification. The comparison of the various classifiers on various hyper-parameter is performed, and the most suitable hyperparameter is used.

In my final project, I am working on the Orignator tracing of messages in WhatsApp. We can also add SPAM and Non-SPAM classification of messages in WhatsApp. Nowadays, many business accounts of WhatsApp are used for SPAM messages to the user. We can use the algorithms studied in the paper to classify messages as SPAM and Non-SPAM. This will add a new direction to the problem.

In the next sections we will see the data set used and result of all methods used for classification.

2. Related Work

There are many papers on spam E-mail classification. There is also much literature available for spam SMS classification. But there is some difference between SMS and E-mail classification. Usually, SMS are smaller, and E-mails are larger compared to SMS. Sharaff, Nagwani, and Dhadse (2016) used a machine-learning algorithm to classify ham and spam E-mails. They applied four experiments for classification using WEKA software (Hall et al., 2009). The dataset used for testing the four classifiers was based on Enron. Those classifiers are (Iterative Dichotomiser) ID3, Decision Tree (J48), Simple Cart, and Active directory Tree (AD tree). The accuracies of each classifier were (92.7%) for J48, (89.1%) for ID3, (90.9%) for AD Tree, and (92.6%) for simple cart.

Trivedi and Dey (2013) consider two probabilistic algorithms based on Bayesian and NB and three boosting algorithms: Bagging, Boosting with Resampling, and AdaBoost. Initially, the Probabilistic classifiers were tested on the Enron Dataset without Boosting and, thereafter, with the help of Boosting algorithms. The Genetic Search Method was used for selecting the most informative 375 features out of 1,359 features created at the outset. The results showed that in identifying complex Spam messages, the Bayesian classifier performs better than NB with or without boosting.

3. Data set

”SMS Spam Collection Data Set” of USI is used for the study. The dataset contains 5572 instances, and the dataset has a label for each message. Messages are either spam or non-spam (ham). There are 4900 non-spam and 672 spam messages in the dataset.

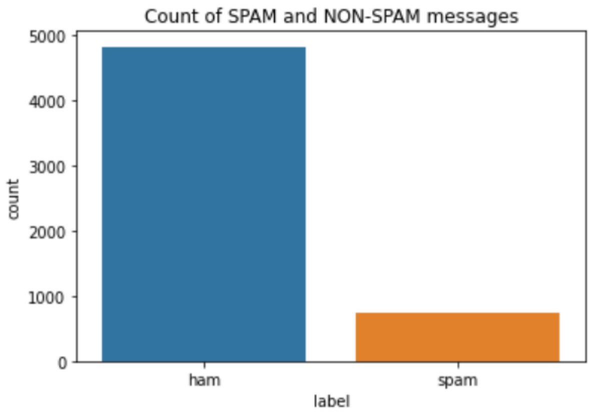

Figure 1. spam and non spam message count

In cryptography frequency of each word in a message plays a significant role. Through machine learning, we can study and use the frequency of words very easily. Frequency of word help to decrypt the message without any key. We have analyzed the frequency of the word in spam and non-spam messages. This will help us to understand which word occurs more in spam messages. We have preprocessed data to get more accurate result.

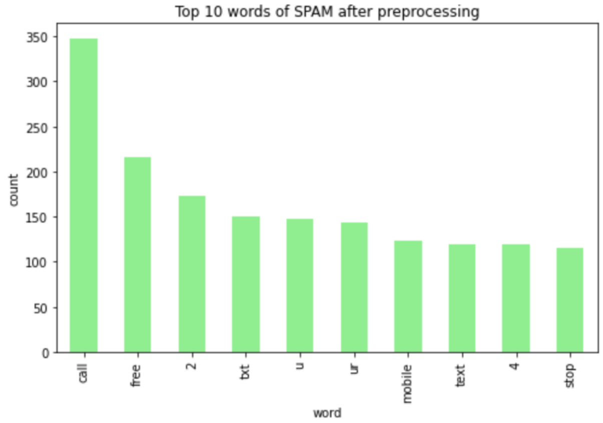

Figure 2. Word count of SPAM messages

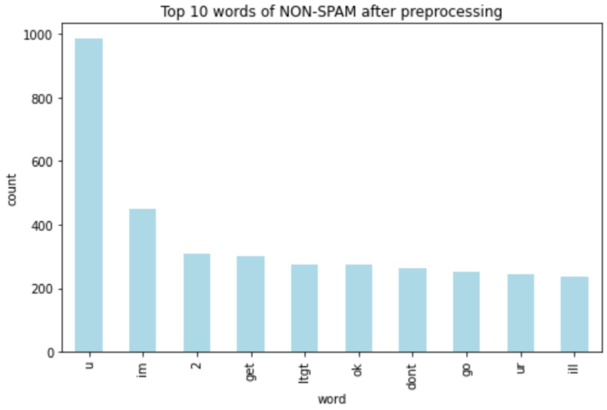

Figure 3. Word count of NON-SPAM messages

I have used 80% data as training and 20% as testing. So total 4457 messages are used for training and 1115 messages are used for testing.

4. Preprocessing

Preprocessing plays an important role in any analysis. Most of the real-world data is inadequate. So preprocessing is one of the major tasks. In preprocessing, all the punctuation and stop words are removed from the dataset. Removing stop words helps to remove words like 'a', 'an', 'the' etc. All the words are converted to lower case so that words like 'the' and 'The' are not treated differently. We can remove the numbers from the message, but numbers can play an important role in detecting SPAM messages because they may represent some amount/reward or phone number.

5. Experiments and Results

I have used different kind of algorithms for classification. Both Machine Learning and Deep Learning algorithms are used for classification purpose. Some of the algorithm provide very similar result as their accuracy is almost same. Some of the algorithms used are :

5.1. Machine Learning Algorithms:

5.1.1. Support Vector Machine

Support Vector Machines are supervised machine learning algorithms used for classification and regression. SVM works well for both linearly separable features and nonlinearly separable features. Our problem is a classification-based problem, so I have used SVM with different hyper-parameter. SVM uses different kinds of kernel functions which are linear, poly, rbf, etc. We can change the regularization parameter (c). I have compared all the kernel functions with different 'c' values. **Accuracy of SVM is 98.57%.**

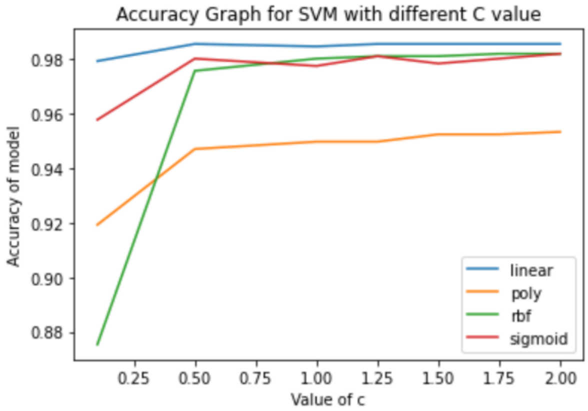

Figure 4. Accuracy of SVM

5.1.2. Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. There are different Naive Bayes classifiers such as Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, etc. This paper used Multinomial Naive Bayes because it is one of the classic naive Bayes variants used in text classification. MultinomialNB uses additive smoothing parameter (alpha) as hyper-parameter. **Accuracy of Naive Bayes is 98.65%.**

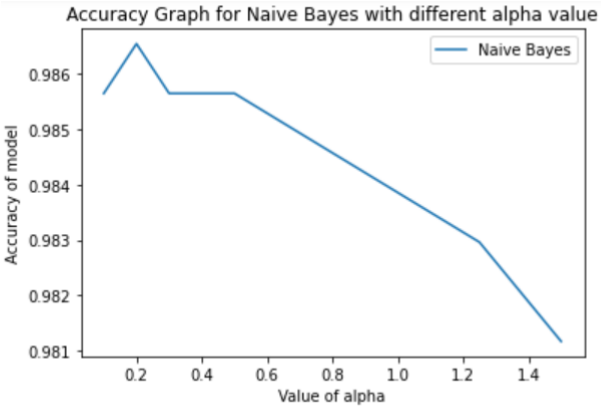

Figure 5. Accuracy of Naive Bayes

5.1.3. K-NEIGHBORS CLASSIFIER

K-Neighbors Classifier is a type of Nearest Neighbors Classification. In this classification is computed from a simple majority vote of the nearest neighbors of each point. K-Neighbors Classifier implements learning based on the nearest neighbors of each query point, where k is an integer value specified by the user. The optimal choice of the k is data-dependent. A larger k suppresses the effects of noise, but makes the classification boundaries less distinct. **Accuracy of K-Neighbors Classifier is 95.52%.**

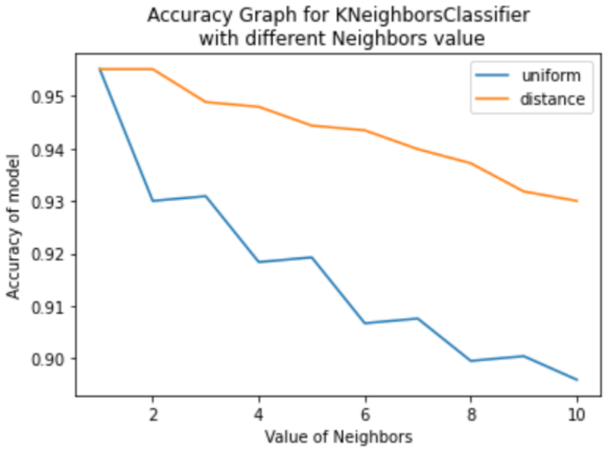

Figure 6. Accuracy of K-Neighbors Classifier

5.1.4. RANDOM FOREST CLASSIFIER

Random forest, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes model's prediction. Gini and entropy are function to measure the quality of a split and we can change the number of trees in the forest with the help of n_estimators. **Accuracy of Random Forest Classifier is 97.67%.**

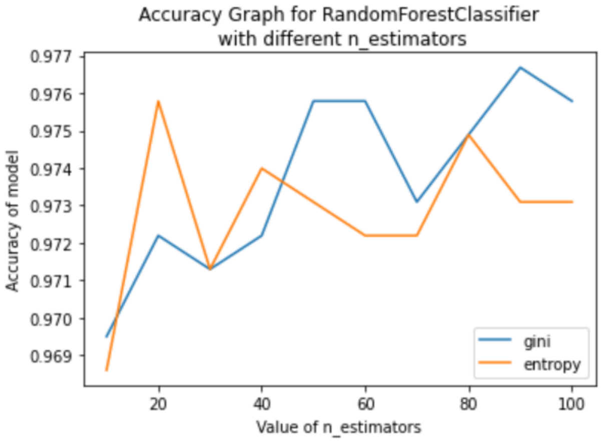

Figure 7. Accuracy of Random Forest Classifier

5.1.5. DECISION TREE CLASSIFIER

Decision Tree Classifier is a class capable of performing multi-class classification on a dataset. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Different kind of strategy are used to choose the split at each node. Supported strategies are "best" and "random". We can change the minimum number of samples required to split an internal node by "min_samples_split". **Accuracy of Decision Tree Classifier is 97.49%.**

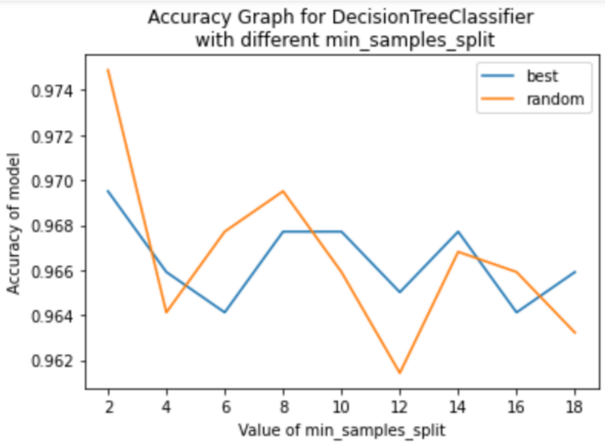

Figure 8. Accuracy of Decision Tree Classifier

5.1.6. Extra Tree Classifier

This implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. **Accuracy of Extra Tree Classifier is 98.12%.**

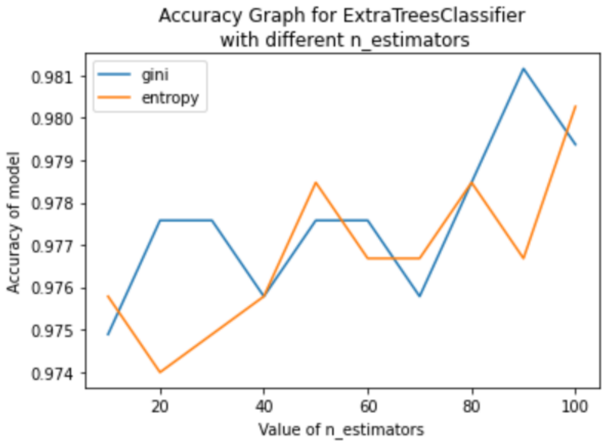

Figure 9. Accuracy of Extra Tree Classifier

5.1.7. AdaBoost Classifier

This is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. **Accuracy of AdaBoost Classifier is 97.49%.**

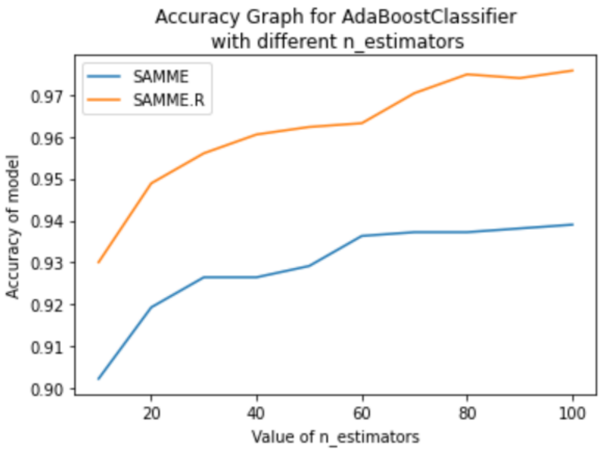

Figure 10. Accuracy of AdaBoost Classifier

5.1.8. Bagging Classifier

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. **Accuracy of Bagging Classifier is 97.85%.**

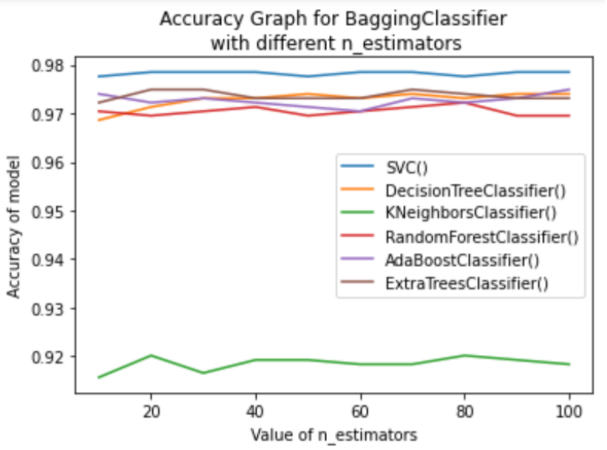

Figure 11. Accuracy of Bagging Classifier

5.2. Deep Learning Algorithms

5.2.1. Dense Model

Sequential model is used from Keras. The embedding layer is used as the first layer, and it will map each word to an N-dimensional vector of real numbers. The second layer is the pooling layer which helps to reduce the number of the parameter in the model to avoid overfitting. The third layer is the Dense layer with the activation function ”relu” and to avoid overfitting, I have added a dropout of 0.1. As this is a binary classification, the last layer will have only one output neuron. The last layer/ final output layer is the Dense layer with a sigmoid activation function. The last layer will give the probability that a particular message belongs to SPAM or Non-SPAM. Model is trained for 20 epoch. **Accuracy of Dense Classifier is 97.85%.**

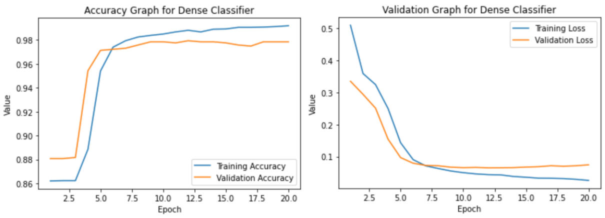

Figure 12. Accuracy and Validation of Dense Classifier

5.2.2. LSTM Model

LSTM are special kind of RNN, capable of learning long-term dependencies. LSTMs also have this chain like structure, but the repeating module has a different structure than RNN. Sequential model is used from Keras. We have used two LSTM layer in model with dropout of 0.1. Model is trained for 20 epoch. **Accuracy of LSTM Classifier is 90.4%.**

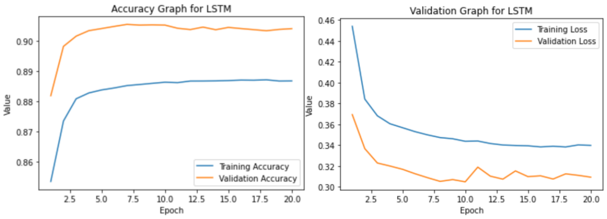

Figure 13. Accuracy and Validation of LSTM Classifier

5.2.3. Bi-directional LSTM Model

This look exactly the same as its unidirectional counterpart. The difference is that the network is not just connected to the past, but also to the future. It is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. Model is trained for 20 epoch. **Accuracy of Bi-directional LSTM Classifier is 97.45%.**

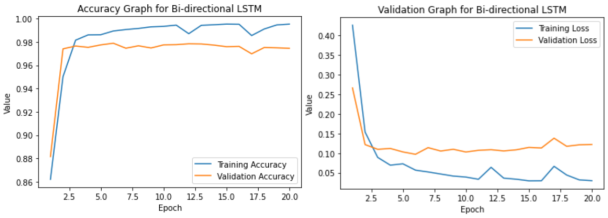

Figure 14. Accuracy and Validation of Bi-directional LSTM Classifier

6. Conclusion

There are many possible algorithms for spam detection. Some of them may perform well, while some of them do not perform well. Our study found that Naive Bayes and SVM performed almost similar, and their accuracy is around 98.5%. All other machine learning algorithm accuracy is also around 95% to 98%. We have tried Deep learning algorithms hoping that they may perform well, but this was not the case. Infect LSTM performed very bad its accuracy is around 90.4%. This result may be specific to the dataset, but we can say that Naive Bayes and SVM performed very well. Figure 15 shows comparative study of accuracy of all the algorithm.

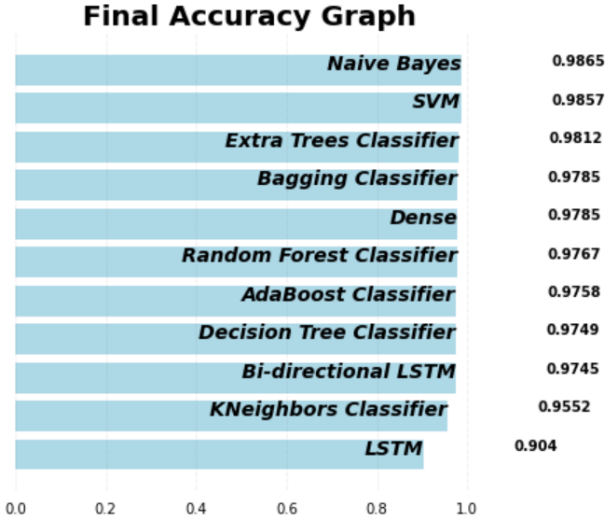

Figure 15. Accuracy of Different Algorithms

Git Hub Link of implementation

7. References

1. M. Bassiouni, M. Ali, E. A. El-Dahshan - Ham and Spam E-Mails Classification Using Machine Learning Techniques, 2018.
2. Luo GuangJun, S. Nazir, H. Ullah Khan, A. Ul Haq - Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms, 2020.
3. E. G. Dada, J. S. Bassi, H. Chiroma - Machine learning for email spam filtering: review, approaches and open research problems, 2019.
4. The Ultimate Guide To SMS: Spam or Ham Classifier Using Python, 2020.
5. NLP: Spam Detection in SMS (text) data using Deep Learning, 2020.
6. A. Sharaff, N. K. Nagwani, A. Dhadse - Comparative Study of Classification Algorithms for Spam Email Detection, 2016.
7. S. K. Trivedi and S. Dey - Interplay between Probabilistic Classifiers and Boosting Algorithms for Detecting Complex Unsolicited Emails, 2013.