EPiC Series in Computing

Volume 69, 2020, Pages 170–179

Proceedings of 35th International Conference on Computers and Their Applications

A Study of Machine Learning Algorithms on Email Spam Classification

N. Sutta, Z. Liu, X. Zhang
Department of Computer Science
Southeast Missouri State University, Missouri, U.S.A.
nsattu1s@semo.edu, zliu@semo.edu, xzhang@semo.edu

Abstract

Despite the fact that different techniques have been developed to filter spam, due to the spammer's rapid adoption of new spam detection techniques, we are still overwhelmed with spam emails. Currently, machine learning techniques are the most effective ways to classify and filter spam emails. In this paper, a comprehensive comparison and analysis of the performance of various classification models on the 2007 TREC Public Spam Corpus are exhibited in various cases of without or with N-Grams as well as using separate or combined datasets. It is shown that the inclusion of the N-Grams in the pre-processing phase provides high accuracy results for classification models in most of the cases, and the models using the split approach with combined datasets give better results than models using the separate dataset.

1 Introduction

The filtering of email spam has two primary approaches and they are called knowledge engineering and machine learning. The first approach is to set up a knowledge-based system using predefined laws to decide if an incoming message is valid or not [1]. The main drawback of this approach is that a client or some other entity, such as a software vendor, must maintain and update the set of rules on an ongoing basis.

The machine learning approach, in contrast, does not require pre-defined rules, but instead, it requires messages which have been pre-classified successfully. Such messages allow sample messages to construct the training dataset used to fit the model's unique learning algorithm. Hence, the classification of spam emails can adopt a machine learning approach for classification in which the computer program learns from the input data and uses the learning to classify new observations. Machine learning algorithms such as Support Vector Machines and Naïve Bayes have been investigated on their effectiveness to successfully detect and filter spam emails [2].

Among the various researchers' work, Banday et al. [3] examined spam filter design procedures by integrating Naïve Bayes, KNN, SVM, and Bayes Additive Regression Tree and evaluated them in terms of accuracy, recall, precision, etc. Chhabra et al. [4] used the Support Vector Machine to develop spam filtering by considering nonlinear SVM classifiers with specific kernel functions over Enron Dataset. Rusland et al. [5] conducted the study using the Naïve Bayes spam filtering algorithm on two datasets that are evaluated based on accuracy, recall, precision, and F-measure. Wang [6] categorized email spams into various hierarchical folders and regulated the tasks needed to respond to an email message automatically.

The rest of the paper is structured as follows: in section 2, we summarize the work performed by various researchers using different machine learning algorithms. In section 3, we first introduce the framework of our study for email spam filtering, and then we present seven machine learning algorithms used in this study. In section 4, we describe our study of using the seven machine learning algorithms for evaluating the efficiency of spam filters and present the performance comparison and analysis of the studied machine learning techniques. Finally, in section 5, we offer the paper's conclusion.

2 Related Work

According to [7], e-mail spam and ham classification can be done using either a machine learning approach or a non-machine learning approach. Bayesian, SVM, Neural Network, Markova model, memory-based pattern discovery, etc. come under machine learning methods and Blacklist/White list, signatures, hash base, grant listing come under non-machine learning methods. To distinguish spam and ham emails, several algorithms like Naïve Bayesian (NB), Multi-layered Perceptron (MLP), J48, and Linear Discriminant Analysis (LDA) have been suggested and their performance was compared by N. Radha and R Lakshmi [8].

Mujtaba et al. reviewed Bayesian classification, k-NN, ANNs, SVMs, Artificial Immune System, and Rough sets, and compared their performance on the Spam Assassin public mail corpus [9]. Banday et al. [3] examined spam filter design procedures by integrating Naïve Bayes, KNN, SVM, and Bayes Additive Regression Tree and evaluated them in terms of accuracy, recall, precision, etc. Chhabra et al. [4] used the Support Vector Machine to develop spam filtering by considering nonlinear SVM classifiers with specific kernel functions over Enron Dataset. Rusland et al. [5] conducted the study using the Naïve Bayes spam filtering algorithm on two datasets that are evaluated based on accuracy, recall, precision, and F-measure. Wang [6] categorized email spams into various hierarchical folders and regulated the tasks needed to respond to an email message automatically.

3 Framework of the Study of Email Spam Classification

In this section, we will introduce the framework of our study for email spam filtering, as well as the seven machine learning algorithms used in this study.

3.1 Framework of Investigation of Spam Classification

Our investigation of email spam with different techniques on filtering consists of four major steps: 1. Dataset collection, 2. Pre-processing the dataset, 3. Training and Testing models and 4. Comparing and Analyzing results. The framework of our study of the email filtering process workflow is depicted in Figure-1 below.

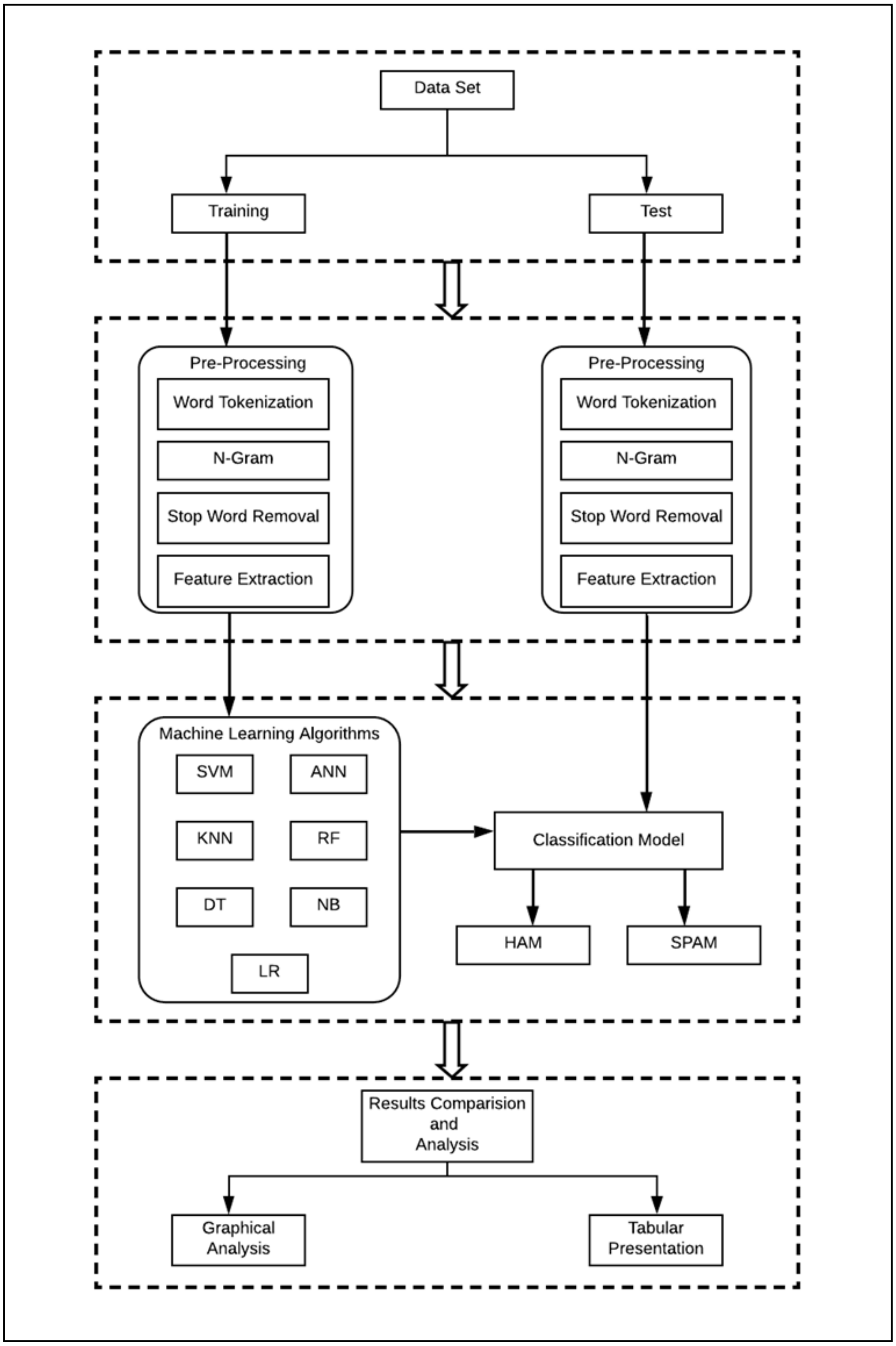

Figure 1: Framework of Investigation of Spam Filtering

3.2 Dataset Collection

The dataset used in this study is the 2007 TREC Public Spam Corpus. It is a slightly cleaned raw email message corpus with all the messages delivered between April 8, 2007 and July 6, 2007 to a particular server. The TREC 2007 public corpus consists of a total of 109,971 messages in the training dataset with 80,318 as ham messages and 29,653 as spam messages and a total of 15,084 messages in the testing dataset with 10,040 as ham messages and 5,044 as spam messages [10].

3.3 Pre-processing Dataset

As described in Figure-1, in this study we apply the data preprocessing techniques of word tokenization, N-Gram formation, stop word removal, and feature extraction to the 2007 TREC Public Spam Corpus dataset. Then we also format the datasets in a way that Machine Learning algorithms are able to execute them [11].

Words such as “and”, “in” etc. are the words that occur most frequently in a document and contain a little information that is not usually required. Those words may reduce accuracy and degrade performance when included in the processing of text. Therefore, during the preprocessing phase, it is necessary to eliminate the stop words. Tokenization is the method of splitting into words, numbers, punctuations, and other symbols a sequence of characters, and then identifying tokens that do not need to be decomposed in subsequent processing, such as punctuation marks are discarded in the tokenization process. An N-gram is a token or textual sequence made up of a number of bytes, characters or words. A token is created by moving a sliding window across a text corpus where the window size depends on the size of the token N and its displacement. Each of the N-grams is a vector coordinate representing the text being studied and the frequency with which this N-gram appears in the text can be the number of this coordinate [12]. In feature extraction, we apply Term Frequency-Inverse Document Frequency (TF-IDF) which has proved to be both simple and effective for the extraction of features in text processing.

3.4 Classification Modeling

A classification model is an algorithm that maps the input data to a particular category and tries to draw some conclusions from the values observed. The algorithms that we will present include Support Vector Machine (SVM), Artificial Neural Network (ANN), K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR).

In linear SVM, data points are separated using certain dimensional hyperplanes which maximizes the margin, i.e. the distance between the hyperplane and the nearest data point of either group. If data is not linearly separable, a non-linear kernel function such as polynomial, sigmoid and gaussian can be applied to maximum-margin hyperplanes by mapping data points in a dimensional space into a much higher-dimensional space. In an ANN, there three interconnected layers: input, hidden, which may include more than one layer and output, and each layer is made up of one or more neurons. The input layer includes neurons that send information to the hidden layer. Data is sent to the output layer by the hidden layer. Every neuron has weighted inputs (synapses), an activation function (defines the output given an input), and one output. KNN is a non-parametric instance-based learning or lazy learning technique. It is performed by comparing the test instance with K training examples and deciding as to which category it belongs to depending on the resemblance to K closest neighbors. In DT, the training dataset is split into smaller datasets until all target variables are in one category. The tree is built top-down by choosing a parameter such as entropy or Gini index that best divides the collection of items at each step. RF is an ensemble learning that uses more than one decision trees to classify data into different classes. For the given set of data, each tree votes on an overall classification and the random forest algorithm selects the most votes for the individual classification.

A NB classifier is a probabilistic classifier based on Bayes theorem with independent sound assumptions. By counting the frequency and combination of values in a given dataset, it calculates a set of probabilities. In LR, the relationship between the dependent variable like the class label and one or more independent variables like features is measured by calculating probabilities using sigmoid function. The Sigmoid-Function then maps the input into a value between the 0 and 1 for classification.

4 Performance Comparison and Analysis

In this section, we will present the results of our study using the seven machine learning algorithms for evaluating the efficiency of spam filters using the 2007 TREC Public Spam Corpus dataset in various cases of without or with N-Grams as well as using separate or combined datasets. In the first way, we executed the seven algorithms on the original provided 2007 TREC dataset which the dataset itself already splits the training and testing dataset. In the second way, the training and the testing dataset are combined first, and a training testing split is performed on them such that 70 percent of the combined data is a part of the training dataset and the remaining is a part of the testing dataset. These newly created datasets by performing the split operation are then used in the training and testing phase after they are pre-processed. These datasets are referred to as split datasets.

Our case-1 is the evaluation of models without N-Grams on separate datasets and the accuracy results of the various models are presented in figure 2. From the figure, we can find that the Random Forest is the most accurate and Decision Tree is the least accurate while the Artificial Neural Network and the Support Vector Machine give us approximately the same values. The overall performance of the models in this case is not up to the mark.

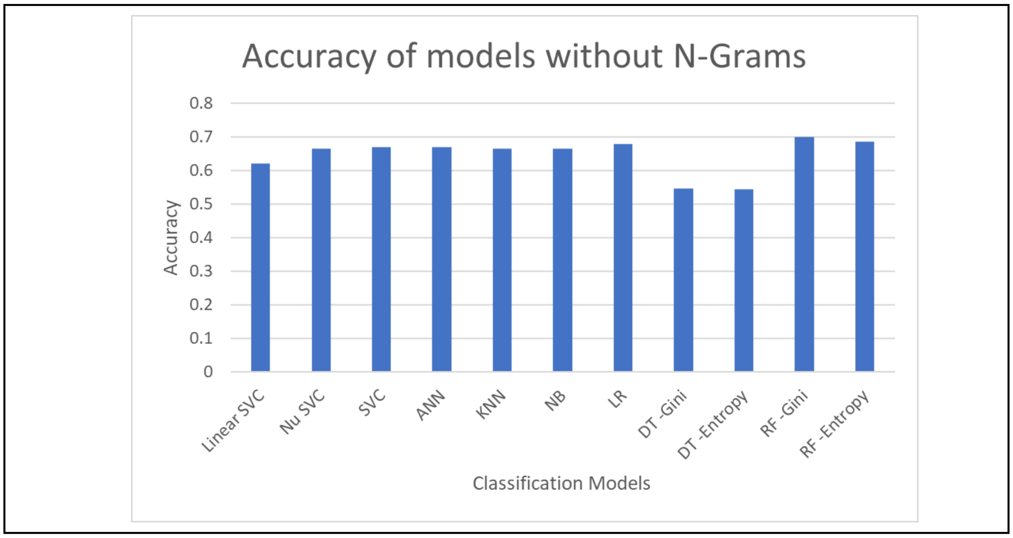

Figure 2: Accuracy of classification models without N-Gram on separate datasets

In our case-2 study, the evaluation of models without N-Grams on split datasets using accuracy is conducted and figure-3 summarize the results. Models like Logistic Regression, Decision Tree with Entropy function and Random Forest with Gini function have the highest accuracy followed by the Linear Support Vector Classification, K Nearest Neighbors and Random Forest with Entropy function having approximately the same accuracy while the Support Vector classification has the least accuracy. The Support Vector Classification has an accuracy less than 90%, Nu Support Vector Classification and Naïve Bayes fall into the accuracy range of 90% - 98%, and the remaining models have an accuracy above 98%.

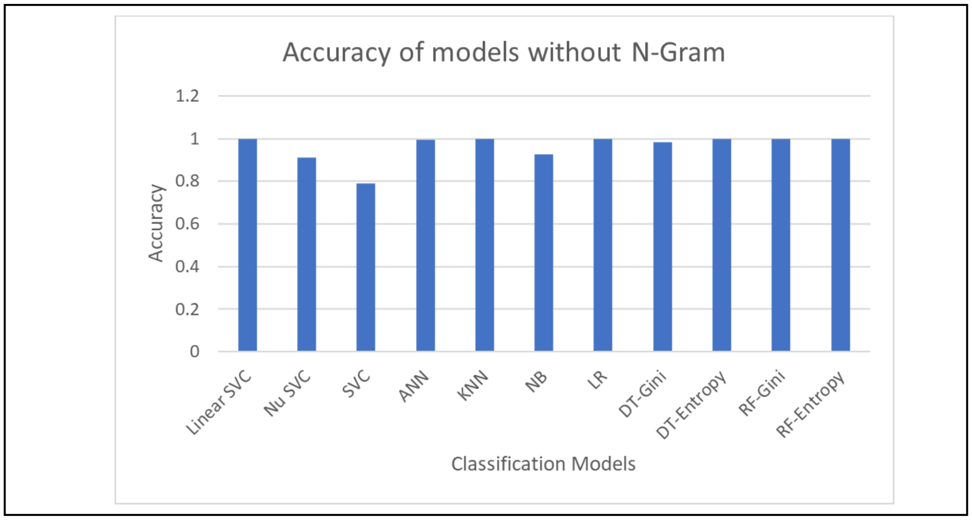

Figure 3: Accuracy of classification models without N-Gram on split datasets

After we compare the running results between case-1 and case-2, we can find that the accuracy values of models that use split datasets are higher than the models which use separate datasets. We believe the cause of the accuracy improvement is that because after combining the datasets, the training dataset has more features included (new spam words from test dataset) for training which helps in the testing phase to get better results.

In our case-3 study, the evaluation of models with N-Grams on separate datasets using accuracy is conducted and the results are presented in figure 4. According to figure 4, on separate datasets, Logistic regression and Linear Support Vector Classification with bigram features and Naïve Bayes with trigram features have the highest accuracies while Decision Tree with unigram features on both Gini and Entropy functions have the least accuracies. Models like Nu Support Vector Classification, Support Vector Classification, and Artificial Neural Network have similar accuracies for all the N-Gram features. On applying trigram features for Linear Support Vector Classification, Random Forest with Gini function, Decision Tree with Gini and Entropy function and Logistic regression; and bigram features for Random Forest with Gini function and Decision Tree with Entropy function they produce satisfactory results.

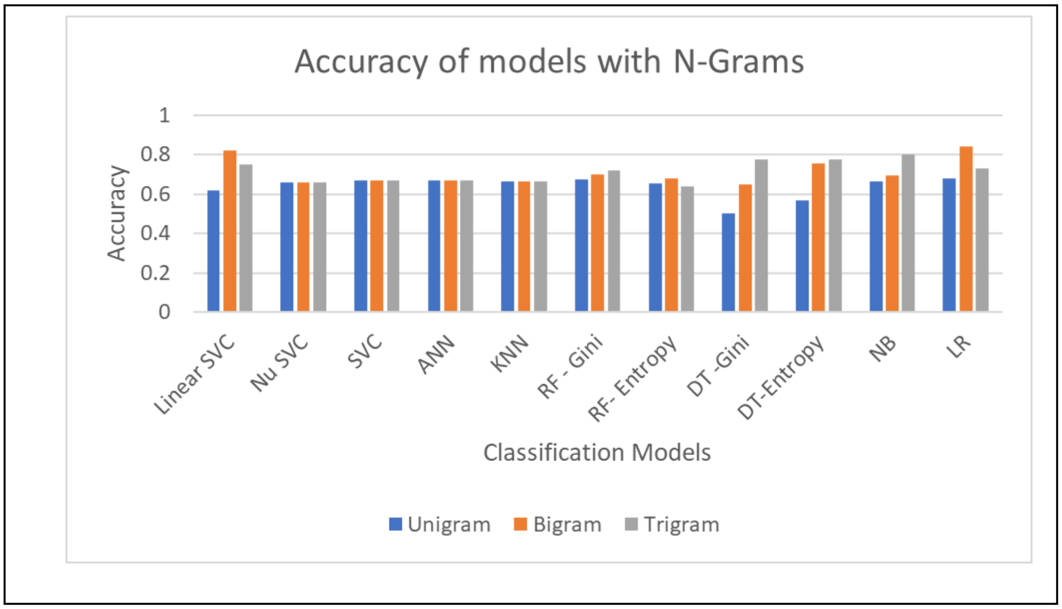

Figure 4: Accuracy of classification models with N-Gram on separate datasets

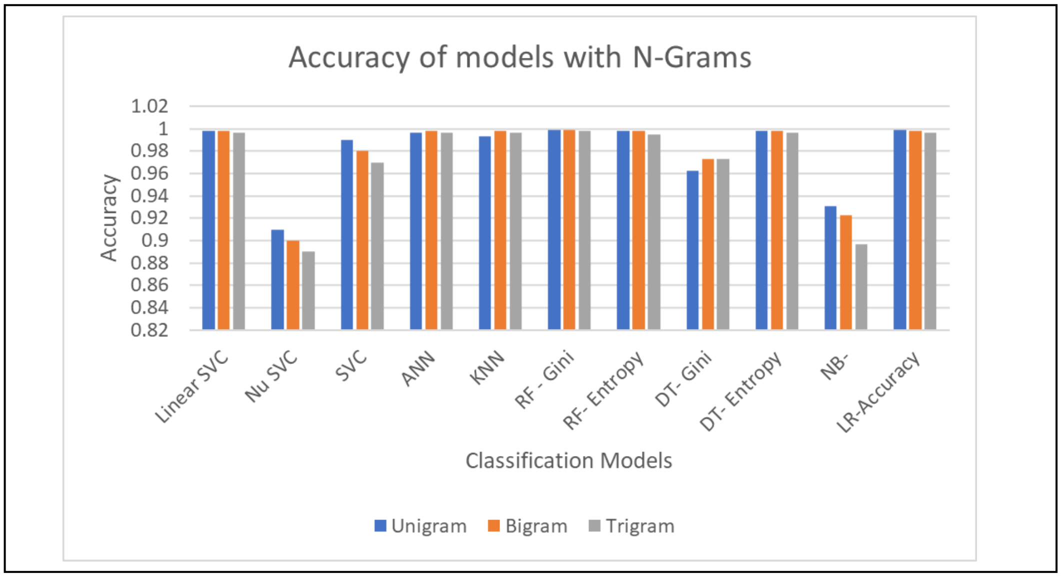

Figure 5: Accuracy of classification models with N-Gram on split datasets

In our case-4 study, the evaluation of models with N-Grams on split datasets using accuracy is conducted and the study results are presented in figure 5. According to figure 5, Random Forest has the highest accuracy for all the N-Gram features followed by Linear Support Vector Classification, Logistic Regression and Decision tree with entropy function for Unigram and bigram Features. Artificial Neural Network and K Nearest Neighbors also have high accuracy values with bigram Features while Nu Support Vector Classification and Naïve Bayes have the least accuracy values with trigram features. The Naïve Bayes and the Nu SVC with trigram have an accuracy less than 90%, Nu SVC with unigram and bigram, DT-Gini with unigram, bigram and trigram, and Naïve Bayes with unigram and bigram fall into the accuracy range of 90% - 98%, and the remaining models have an accuracy above 98%. From the comparison of the above four cases, we can find generally the models with N-Grams have given better accuracies than models without N-Grams. Furthermore, the performance of models using N-Grams on the split datasets is higher than the models without using N-Grams on the separate datasets.

Models	Confusion Matrix-Separate Dataset	Confusion Matrix-Split Dataset
Linear Support Vector Classification	[[6832 3208] [2516 2528]]	[[13559 1] [20 5169]]
Logistic Regression	[[8821 1219] [3640 1404]]	[[27202 5] [16 10294]]
Random Forest -Gini	[[9633 407] [4319 725]]	[[27202 5] [37 10273]]
Random Forest -Entropy	[[9722 318] [4411 633]]	[[27201 6] [68 10242]]

Table 1: Confusion Matrices of models on the separate and the split datasets without N-Gram

In table 1, the evaluation of selected models without N-Grams on separate datasets and split datasets using the confusion matrix is presented. Here we only select models which have good performance in terms of accuracy. The Random Forest model has the least number of false classified instances (False Negative(FN) and False Positive (FP)) and the Linear Support Vector Classification model has the highest number of false classified instances on the separate datasets. In the case of the split datasets, the Logistic regression and the Linear Support Vector Classification model have the least number of false classified instances and the Random Forest model has the highest number of false classified instances.

To improve the model's prediction accuracy, we also experimented different tuning parameter values for SVC models without N-Grams on separate and split datasets. Figure 6 is for the case of separate dataset. It shows that as the penalty parameter C of the Linear Support Vector Classification increases from 0.001 to 1, the accuracy decreases from 69% to 62%. The case of split dataset is shown in figure 7. From the figure we can see that as penalty parameter C increases from 0.001 to 1, the accuracy only increases 0.2% from 99.7% to 99.9%, which is negligible.

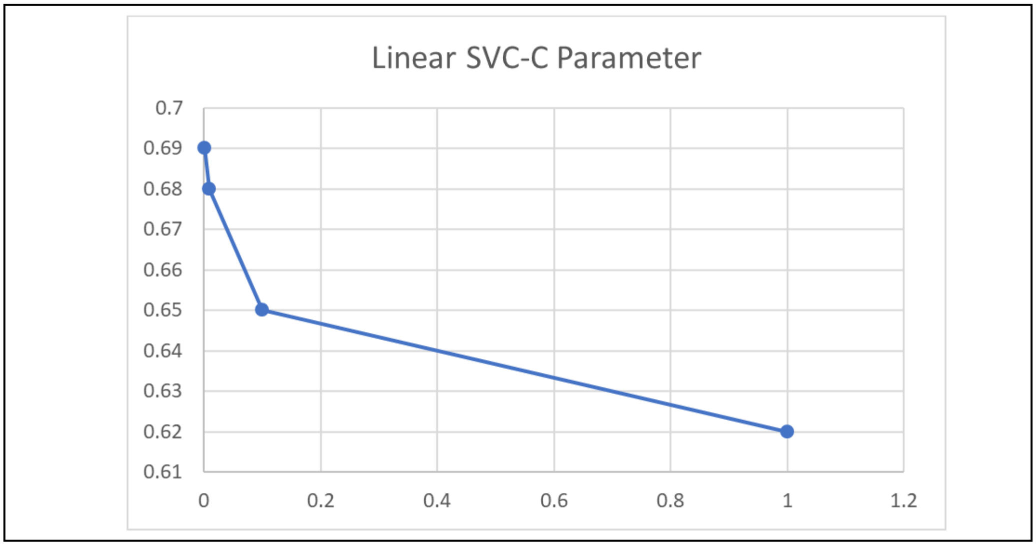

Figure 6: Accuracy versus Linear SVC's penalty parameter on the separate datasets

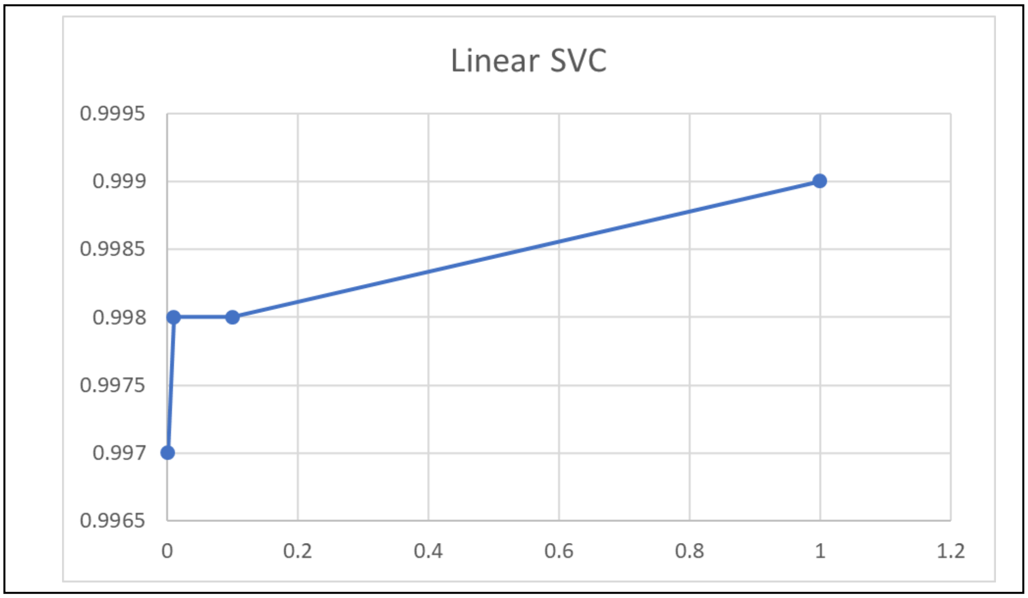

Figure 7: Accuracy versus Linear SVC's penalty parameter on the split datasets

5 Conclusion

In this paper, we investigate the applicability of seven machine learning algorithms to the spam email classification. We discuss the framework of our study which consists of data collection, dataset pre-processing, training and testing models and performance analysis. A comprehensive comparison and analysis of the performance of various classification models on the TREC Public Spam Corpus are exhibited in various cases of without or with N-Grams as well as using separate or combined datasets. The inclusion of the N-Grams in the pre-processing phase has provided high accuracy results for classification models in most of the cases. The models using the split approach with combined datasets give better results than models using the separate dataset.

References

[1] T. Guzella and W. Caminhas, "A review of machine learning approaches to Spam filtering", *Expert Systems with Applications*, 2009, 36(7), 10206–10222.

[2] W. Awad and S. ELseuof, “Machine learning methods for spam e-mail classification”, *International Journal of Computer Science and Information Technology*, 2011, 3(1), 173–184.

[3] M. Banday and R. Jan, “Effectiveness and limitations of statistical spam filters”, *Proceedings of the International Conference on New Trends in Statistics and Optimization*, 2009.

[4] P. Chhabra, R. Wadhvani and S. Shukla, “Spam filtering using support vector machine”, *International Journal of Computer & Communication Technology*, 2010, 1(2), 322-341

[5] N. F. Rusland, N. Wahid, S. Kasim and H. Hafit, “Analysis of naïve bayes algorithm for email spam filtering across multiple datasets”, *Proceedings of the IOP Conference Series: Materials Science and Engineering*, 2017

[6] X. Wang and Cloete. “Learning to classify email: a survey”, *Proceedings of the International Conference on Machine Learning and Cybernetics*, 2005

[7] S. A. Sasb, Mitri, N. Mitri and M. Awad, “Ham or spam? A comparative study for some content-based classification algorithms for email filtering”, *Proceeding of the 17th IEEE Mediterranean Electrotechnical Conference*, 2014

[8] L. Deepa and N. Radha, “Supervised learning approach for spam classification analysis using data mining tools”, *International Journal on Computer Science and Engineering*, 2010, 2(9), 2783-2789

[9] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed and M. A. Al-Garadi, “Email classification research trends: review and open issues”, *IEEE Access*, 2017, 5, 9044–9064

[10] G. Cormack, “TREC 2007 Spam track overview”, *Proceedings of Text Retrieval Conference (TREC) 2007: The Sixteenth Text Retrieval Conference*, 2007

[11] V. Gurusamy and S. Kannan, “Preprocessing techniques for text mining”, *International Journal of Computer Science & Communication Networks*

[12] C. Bouras and V. Tsogkas, “Assisting cluster coherency via n-grams and clustering as a tool to deal with the new user problem”, *International Journal of Machine Learning and Cybernetics*, 2014, 7(2)