A Comparison of Naive Bayesian and Deep Learning Methods for Spam-Filtering

Kiana Vang
Department of Computer Science and Engineering
University of Minnesota - Twin Cities, Minneapolis, MN
May 2021

***Abstract*— With the increase in advanced technologies, it has revolutionized how people communicate. Communication done via electronic mail (e-mail) has become easier and faster. However, it has also contributed to rapid dissemination of unwanted messages, e.g. spam emails. Spam emails pose many threats, such as harm to personal devices and or theft of personal information. Inspired by Paul Graham's work, this paper addresses this issue by creating a spam filter using state-of-the-art methods in machine learning and artificial intelligence: Naive Bayes and Deep Learning. Two Naive Bayesian classifiers are generated. The sensitivity and specificity scores for the Naive Bayesian classifier using CounterVectorizer is 90.6 percent and 96.8 percent, respectively. Its false positive rate is 12.5 percent. For the Naive Bayesian classifier using TfidfVectorizer, sensitivity and specificity scores are 86.9 percent and 98.2 percent, respectively with a false positive rate of 7.6 percent. The deep learning model achieved 95.2 percent and 99.6 percent, respectively, for sensitivity and specificity. It has a false positive rate of 1.4 percent. In comparison, deep learning model performed the best. While Naive Bayesian and deep learning methods explored in this paper did not outperform Graham's metrics, they still performed generally well (>80 percent). Testing on Graham's data, expanding training data, and exploration of other machine learning methods can improve future work and progress.**

I. Introduction

Unwelcome messages whether received through telemarketing or postal mail can be bothersome. Spam emails are no different. The dissemination of unwanted emails can lead to other harmful issues, such as theft of personal information and computer malware. Inspired by Paul Graham's "Plan for Spam" and his latest approach "Better Bayesian Filtering," I addressed this issue by applying Graham's spam filtering method using Naïve Bayesian approach to observe if the results turn out the same. For further assessment of this approach, I designed a deep learning neural network model and compared the results. More information regarding Graham's work can be accessed at this website: *http://www.paulgraham.com/spam.html*.

II. Background and Related Work

A. A study of spam filtering using support vector machines

Electronic mail (e-mail) is a major avenue for communication in current times. It is fast, easy, and economical to use. However, a major fallback of using this form of communication is the dissemination of unwanted mail known as spam emails. Amayri et al. present a solution to this issue using support vector machines (SVM), which are a set of supervised learning methods used for classification. The critical component when using SVMs is knowing how to select the right kernel, a mathematical

formula, that can precisely manipulate and transform data. The authors investigate several distance-based kernels and specify spam-filtering behaviors using SVM. They propose and show the effective use of string kernels for spam-filtering. A string kernel differs from a distance-based kernel by its ability to define similarity between pair of documents. This makes a string kernel most suitable for text classification. It measures the total occurrence of shared substrings of a certain length in a feature space. To cope with realtime scenarios, the authors present empirical results from a broad study of online, transductive, and online active models. An online model is a classifier where the order of filtering is determined by design. The order is either chronological or random. The transductive model estimates the value of a classification function at given points; thus, constructing a maximum margin and improving overall generalization performance. Online active models are classifiers that filter in a stream one message at a time. Their concluding results show that active online method using string kernels achieves higher precision and recall rates.

B. Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets

In many previous research on spam-filtering, different machine learning techniques are used, such as Random Forest, Naive Bayesian, Support Vector Machines, and Neural Network. In this study, Rusland et al. present and test a Naive Bayes algorithm and its performance on e-mail spam filtering using two different datasets. One containing spam messages, and the other nonspam. Naive Bayes is a popular algorithm used for spam filters because of its simplicity–it is easy to implement and does not require a lot of time to train and evaluate. Performance on these datasets is measured based on overall accuracy, recall, precision, and F-measure. The authors use WEKA, a tool for evaluating their Naive Bayes algorithm. Their results show that the type of e-mail and number of instances of the dataset influence overall performance of Naive Bayes.

C. Applicability of machine learning in spam and phishing email filtering: Review and approaches

Due to increasing technological advancements and simplicity of communication through emails, unsolicited bulk emails (UBEs) have become a severe issue. Spam emails not only waste time or take up network bandwidth, but may also contain malware that threatens security. This motivates the authors to develop more robust and dependable UBE filters that can detect such emails.

In this study, Gangavarapu et al. focus on finding the most discriminating set that contains behavior-based features that are appropriate in detection of UBEs. They also perform an exhaustive comparative study using several state-of-the-art machine learning algorithms, such as Naive Bayes (NB), Support Vector Machines (SVM), Bagged Decision Trees (BDT), Random Forest (RF), Extra Trees (ET), AdaBoost (AB), Stochastic Gradient Boosting (SGB), and Voting Ensemble (VE). BDT, RF, ET, AB, SGB, and VE are known as ensemble classifiers. Ensemble learning is an approach that groups several classifiers together. Doing so pools the advantages of these classifiers, and thus, results in an improved classification performance. Their proposed models produced an overall 99 percent accuracy in UBEs classification.

D. Machine Learning for E-mail Spam Filtering: Review,Techniques and Trends.

In this paper, Bhowmick et al. present a comprehensive review of the most effective content-based e-mail spam filtering techniques.

Their study is multi-fold and contains the following:

- an exploration of spam characteristics, trends and evasion techniques
- a discussion of feature engineering and extraction
- a taxonomy presentation of content-based spam filtering
- a report of new findings and suggestions for future investigation

Their paper summarizes the strengths and limitations of various machine learning techniques, such as Naive Bays, k-Nearest Neighbor, Artificial Neural Network, Support Vector Machines, LogitBoost, Regression, and Ensembles.

E. Machine learning for email spam filtering: Review, approaches and open research problems

An intense and increasing need for more dependable and robust anti-spam filters is important now more than ever. In their study, Dada et al. present background information that include important concepts, attempts, efficiency, and research trends regarding this topic. They evaluate spam filterings processes of leading internet service providers (ISPs), such as Gmail, Yahoo, and Outlook. Their thorough analysis show the strengths and drawbacks of existing machine learning technique used for spam filtering. Their study suggests deep learning and deep adversarial learning to be highly effective and recommends these methods to handle spam-filtering problems. Deep learning methods exploit both artificial intelligence and machine learning to learn features directly from data using multi-layer perceptrons.

F. Robust personalizable spam filtering via local and global discrimination modeling

Joint distribution of e-mails and labels changes from user to user. Therefore, sample training data do not represent a true distribution. E-mail service providers provide two filtering options, global or personalized, for all users. However, usefulness of these options depend on the robustness and scalability of the filter. In this study, Junejo et al. seek to address these challenges and present a robust personalizable filter based on local and global discrimination modeling. Borrowed from Junejo and Karim, their algorithm, Discriminating Term Weighting for Text Classification (DTWC), uses relative and log relative risks to quantify discrimination information. This information is then used to construct a two-dimensional space for linear classification. They evaluate their algorithm under varying distribution shift, on gray e-mails, unseen emails, and under varying filter size. The authors show that their personalized service-sid spam filter (PSSF) is scalable for personalized filtering.

G. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks

The accelerated growth of unwanted emails has inspired machine learning researchers to develop various anti-spam methods. However, this has become increasingly challenging as spammers are finding unique ways to make their emails seem more legitimate, thus, reducing the probability of detecting their emails as spam. To solve this, a more complex model is required.

In this study, Barushka et al. introduce a novel spam filter this issue. This filter integrates an N-gram *tf.idf* feature selection, modified distribution-based balancing algorithm and a regularized deep multi-layer perceptron neural network (NN) model with rectified linear units (DBB-RDNN-ReL). This ensemble can capture more complex features from data in higher dimensional space. The authors compare and show that their novel spam filter outperformed other machine learning methods,

such as Minimum Description Length, Factorial Design using Support Vector Machines and Naive Bayes, Incremental Learning C4.5, and Random Forest, Voting and Convolutional Neural Netwok in terms of classification accuracy.

H. Spam filtering: How the dimensionality reduction affects the accuracy of Naive Bayes classifiers

The best known techniques to filter spam emails are based on Bayes decision theory. However, these techniques tend to fail when dealing with high dimensions of feature space. This is known as the curse of dimensionality. Even if this is the case, it is unclear if Naive Bayes spam filters' performance depend on the scheme applied for reducing dimensionality of feature space. Almeida et al. seek to address this issue by presenting a performance evaluation. In this study, they examine the performance of many term-selection techniques with seven different models of Naive Bayes spam filters. Their results show that among all classifiers, Boolean Naive Bayes and Basic Naive Bayes achieved the best individual and average rank performance.

III. Methodology

In statistics, Bayes' Theorem is a method used to describe the probability (i.e. posterior) of an event given prior knowledge, likelihood, and evidence. The probability result is then used to determine the classification of the event. Naïve Bayes' is an extension of this theorem which assumes that all inputs are independent of each other.

Using Naïve Bayes', Graham's spam filter in "A Plan for Spam" caught 99.5 percent of spam with less than 0.03 percent of false positives. After discovering CRM114, an effective text classifier, his updated approach used a dataset that includes 1750 spams and achieved a slightly higher filtering rate of 99.75 percent.

Graham's algorithm is written in Lisp. However, the algorithm in this paper is written in Python language (version 3.9.1) using software application Spyder (version 5.0.0) as its integrated development environment (IDE). Graham used his own dataset, which includes a corpus of spam and another corpus of nonspam mail where each contains 4000 messages in it. This dataset is not available to me. Therefore, this paper uses a free and opensource dataset from SpamAssassin, which includes non-spam and spam emails. This dataset has a total of 5000 messages all of which comes with various level of difficulty. A message is considered "easy" if it is easy to differentiate as spam or non-spam, and "hard" otherwise. It is important to note that the dataset did not come with labels; therefore, prior to any data processing, I assigned emails their respective labels. An email is labeled as '0' if it is non-spam, and '1' if it is spam.

To evaluate the credibility of the solution to this problem, a confusion matrix is generated. The matrix contains the total number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) predicted by the algorithm. Sensitivity and specificity metrics are then used to evaluate the strength of the solutions. Sensitivity captures the proportion of positives that is correctly identified, and specificity captures the proportion of negatives that are correctly identified. The higher the results of the two metrics, the better the algorithm's predictability and classification strength.

Moreover, false positive rate (FPR) captures the proportion of negatives that is categorized as positives. In this case, the lower the rate, the better.

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

$$Specificity = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{TN + FP}$$

To see how the Naïve Bayesian classifier performs relative to other state-of-the-art machine learning methods, I designed a deep learning (i.e. artificial neural network) model using the same dataset as the Naïve Bayesian classifier described previously. Deep learning is part of Natural Language Processing (NLP) techniques that are used for text generation, classification, translation, semantic analysis, etc. One of the benefits of deep learning is that a model can be trained to learn feature levels of increasing abstraction with little human contribution. In other words, it learns directly from input data without feature extraction and assumptions. Starting from raw input, each hidden layer of the model combines values in its preceding layer and learns more complicated functions of the input. Once it finishes learning, the model can be tested using never-before-seen data and labels. Similarly, to evaluate the credibility of the deep learning model, a confusion matrix is generated. The design and results of the Naïve Bayesian classifier and deep learning model is shown in the following section. For information on spam filtering using deep learning, this method is further described in Sie Huai Gan's article "Spam Filtering System with Deep Learning," which can be accessed at this website: *https://towardsdatascience.com/spam-filtering-system-with-deep-learning-b8070b28f9e0*

The methodology in this paper is summarized as follows:

1) obtain sufficient data
2) process data
3) visualize data
4) build Naive Bayesian classifier
5) build Deep Learning model
6) display results and compare

Data obtained from SpamAssassin is processed using procedure described in Algorithm 1. Algorithm 2 describes the general procedure for building a spam-filtering classifier or model.

Algorithm 1: Process Data

Result: Processed Spam and Non-spam Data
load libraries;
open raw data;
for *an email in raw data* **do**
    remove hyperlinks;
    change to lower case;
    remove digits;
    remove special characters;
    remove single characters;
    remove stop words;
    perform lemmatization;
end

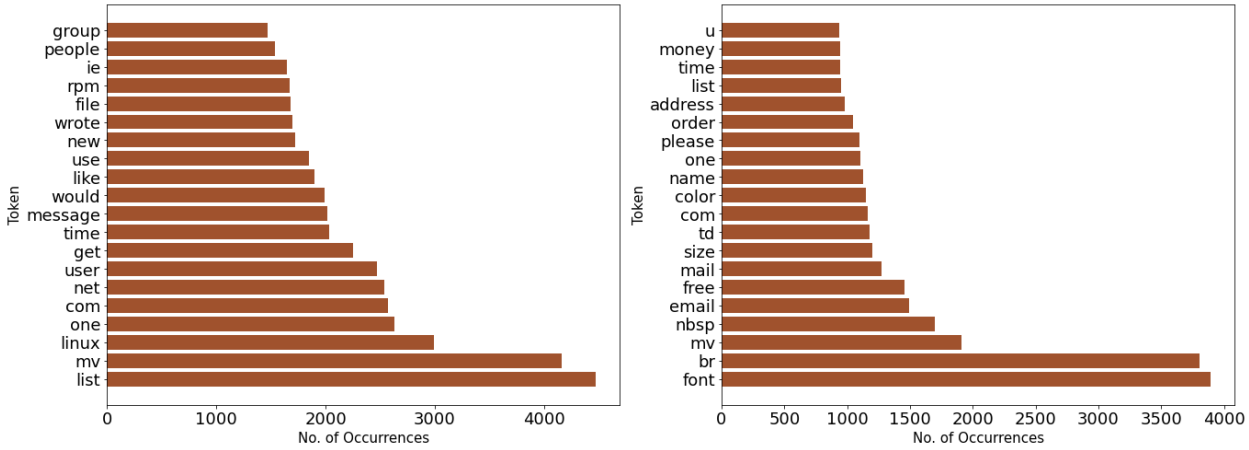
Algorithm 2: General Spam-Filtering Classifier or Model

Result: A trained classifier or model with prediction results
load libraries and processed data;
initialization;
perform tokenization;
perform padding (if deep learning);
split data into training and test sets;
fit classifier to training data;
evaluate classifier's prediction ability;
using classifier, predict on test data;
plot confusion matrix;

IV. Design and Results

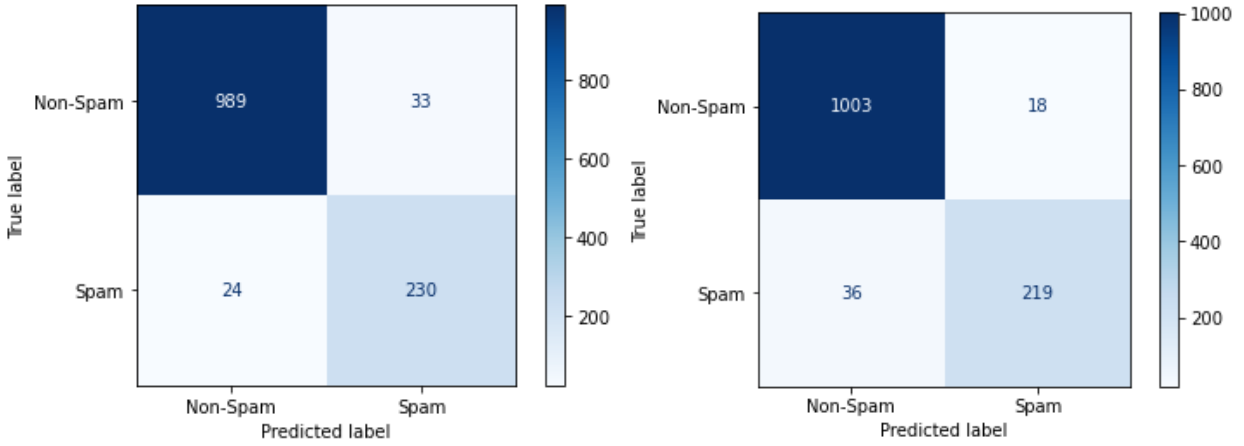
After data processing, the most common words appearing in spam and non-spam emails is shown in Figure 1. A Naive Bayesian classifier and deep learning model are then built using procedures in Algorithm 2 and built-in libraries, such as Sklearn and Tensorflow.

For the Naive Bayesian method, there are two ways to tokenize and convert text data



(a) Non-spam Emails. (b) Spam Emails

Fig. 1: Most common tokens among emails.

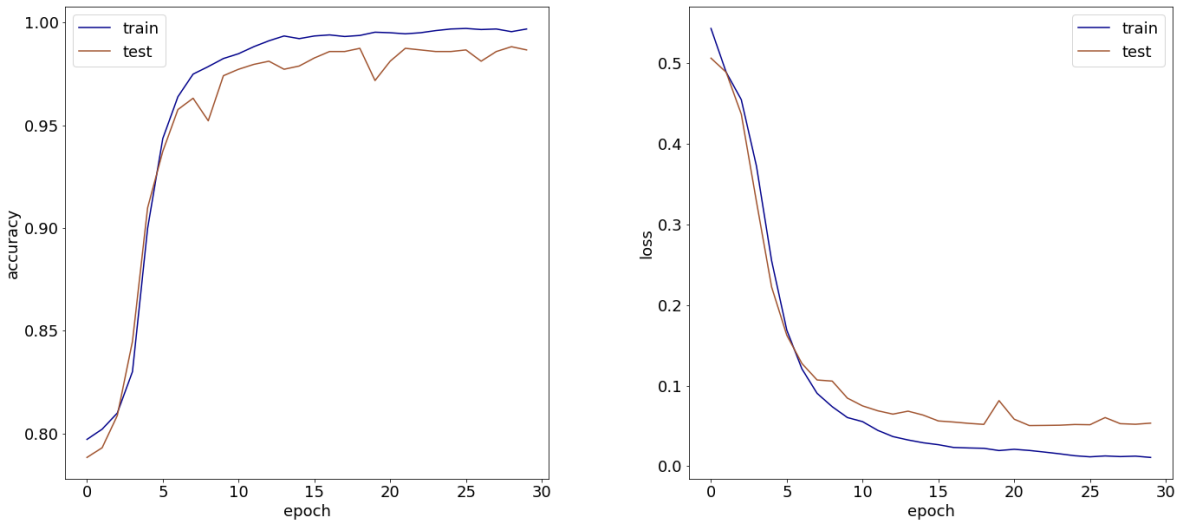


(a) Using CounterVectorizer (b) Using TfidfVectorizer

Fig. 2: Confusion Matrix - Naive Bayesian Method

Type	Sensitivity	Specificity
Naive Bayesian - CountVectorizer	90.6	96.8
Naive Bayesian - TfidfVectorizer	86.9	98.2
Deep Learning	95.2	99.6

TABLE I: Sensitivity and Specificity Metrics



(a) Model Accuracy

(b) Model Loss

Fig. 3: Evaluation of Deep Learning Model

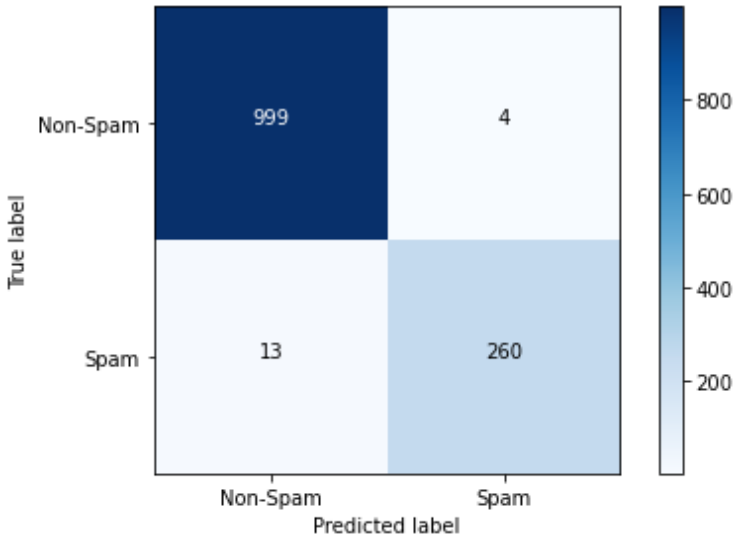


Fig. 4: Deep Learning Confusion Matrix

into vectors for numerical analysis. One can use CountVectorizer or TfidfVectorizer. By using CountVectorizer, a word is counted every time it appears in the message. The major disadvantage of this approach is biasing in favor of most frequent words and ignoring rare words that could be useful. On the other hand, TfidfVectorizer deals with this issue of biasing by considering the weightage of a word. It weights word counts by how often they appear in messages.

I pursued both vectorization options to observe their difference and effectiveness. Confusion matrices for the Naive Bayesian and deep learning methods are shown in Figure 2 and 4, respectively.

Similarly for the deep learning method, tokenization of the data is performed. After tokenization, padding is done to have all samples in the dataset to be of equal length. The model is then fit to the data and trained for 30 epochs. The model's accuracy and loss rates over 30 epochs are shown in Figure 3. 'Binary Classification Loss Function' is the method used to compute loss (i.e. error) for this model because there are only two classes (spam and non-spam) the model is dealing with.

Sensitivity and specificity metrics for the Naive Bayesian and deep learning approaches are compared and shown in Table 1.

V. Discussion

From Figure 1, out of the top 20 most common tokens, the top three of those that appear in non-spam emails are: 'list,' 'mv,' and 'linux' where 'list' appeared 4467 times, 'mv' appeared 4159 times, and 'linux' appeared 2987 times. For spam emails, the top three tokens are 'font,' 'br,' and 'mv.' 'Font' appeared 3889 times, 'br' appeared 3805 times, and 'mv' appeared 1906 times. Observing the top three tokens does not provide any useful information in between spam and non-spam emails. However, better intuition can be made by looking at top 20 tokens in its entirety. From human experience, words such as 'free,' 'please,' 'order,' 'time,' and 'money' are common words found in personal junk e-mails.

Observing the confusion matrices for Naive Bayesian classifier in Figure 2, both appear to perform very similarly despite CountVectorizer favoring most frequent words. Recall that sensitivity captures the proportion of positives that is correctly identified whereas specificity captures the proportion of negatives that are correctly identified. The sensitivity and specificity scores for the Naive Bayesian classifier using CounterVectorizer is 90.6 percent and 96.8 percent, respectively. Its false positive rate is 12.5 percent. For the Naive Bayesian classifier using TfidfVectorizer, sensitivity and specificity scores are 86.9 percent and 98.2 percent, respectively with a false positive rate of 7.6 percent. From this experiment, the Naive Bayesian classifier using CounterVectorizer and TfidfVectorizer did not perform the same or better than Graham's classifier.

A deep learning model is developed to see if its results can match or exceed Graham's results. The plot in Figure 3a is used to determine if there exists any over-fitting. Overfitting is an issue where the model accurately learns the features of the dataset but when tested, the model performs poorly on never-before-seen data. Here, accuracy of 'test' follows a similar trend as 'train.' The gap between the two trendlines are not too drastic, and therefore the model is not really overfitting the data. Figure 3b shows the loss calculations for each epoch. While training the model, the goal is to minimize the loss function as much as possible. Loss calculations are used to determine how off predictions are from the actual. The lower the loss value, the better the predictions are. In Figure 3b, 'test' starts off with a loss value of 0.5 and converges to a value slightly below 0.1.

A confusion matrix for the deep learning

model is shown in Figure 4. The sensitivity and specificity scores for the deep learning model is 95.2 percent and 99.6 percent, respectively. It has a false positive rate of 1.4 percent.

A comparison of the Naive Bayesian classifiers and deep learning model and their results is shown in Table 1. Overall, the deep learning model performed better than the Naive Bayesian classifiers in regards to sensitivity and specificity. However, in comparison to Graham's classifier, the deep learning model's results are slightly lower by 4.5 percent and 1.4 percent respectively in terms of sensitivity and false positive rates.

VI. Conclusion

In the current digital era, the ways in which people communicate have evolved with increasing and advancing technologies. Sending messages via e-mails has become a preferred method; it is one of the most convenient and efficient ways to communicate. However, with these benefits also come many disadvantages. Unwanted messages have found their ways into personal inboxes, and the need for a robust spam filter is born.

Following Graham's guidance, this paper presented two versions of a building a spam filter; one using Naive Bayes' and another using deep learning, all of which are powerful state-of-the-art techniques used in machine learning and artificial intelligence.

While both classifiers did not outperform Graham's metrics, with 99.75 percent accuracy and 0.03 false positive rate, both performed generally well. Out of the two classifiers, the deep learning model generated the best results. It achieved 95.2 percent accuracy rate with 1.4 percent false positive rate.

For future work, it would be interesting to observe how the two classifiers perform on Graham's dataset. Moreover, finding additional datasets could help improve training and validation scores, especially for deep learning. Another area for future progress is to expand and experiment with other machine learning methods to see which method(s) is/are best for filtering spam overall.

References

[1] Amayri, Ola, & Bouguila, Nizar. (2010). A study of spam filtering using support vector machines. The Artificial Intelligence Review, 34(1), 73-108.

[2] Rusland, Nurul Fitriah, Wahid, Norfaradilla, Kasim, Shahreen, & Hafit, Hanayanti. (2017). Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. IOP Conference Series. Materials Science and Engineering, 226(1), 12091.

[3] Gangavarapu, Tushaar, Jaidhar, C. D, & Chanduka, Bhabesh. (2020). Applicability of machine learning in spam and phishing email filtering: Review and approaches. The Artificial Intelligence Review, 53(7), 5019-5081.

[4] Bhowmick, Alexy, & Hazarika, Shyamanta M. (2016). Machine Learning for E-mail Spam Filtering: Review,Techniques and Trends. http://arxiv.org/licenses/nonexclusive-distrib/1.0.

[5] Dada, Emmanuel Gbenga, Bassi, Joseph Stephen, Chiroma, Haruna, Abdulhamid, Shafi'i Muhammad, Adetunmbi, Adebayo Olusola, & Ajibuwa, Opeyemi Emmanuel. (2019). Machine learning for email spam filtering: Review, approaches and open research problems. Heliyon, 5(6), E01802.

[6] Junejo, Khurum Nazir, Junejo, Khurum Nazir, Karim, Asim, & Karim, Asim. (2013). Robust personalizable spam filtering via local and global discrimination modeling. Knowledge and Information Systems, 34(2), 299-334.

[7] Barushka, Aliaksandr, & Hajek, Petr. (2018). Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Applied Intelligence (Dordrecht, Netherlands), 48(10), 3538-3556.

[8] Almeida, Tiago A, Almeida, Tiago A, Almeida, Jurandy, Almeida, Jurandy, Yamakami, Akebo, & Yamakami, Akebo. (2011). Spam filtering: How the dimensionality reduction affects the accuracy of Naive Bayes classifiers. Journal of Internet Services and Applications, 1(3), 183-200.