JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

AI-based Feature Selection with Unsupervised Learning for Efficient Spam and Phishing Email Classification

Prsanya Thotha[1], Shreya Reddy Pundru[1], vamshika Mashetty[1], P. Supriya[2]

[1]UG Student, [2]Assistant Professor, [1,2]Department of Computer Science and Engineering

[1,2]Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

ABSTRACT

Email has become one of the most important forms of communication. In 2014, there are estimated to be 4.1 billion email accounts worldwide, and about 196 billion emails are sent each day worldwide. Spam is one of the major threats posed to email users. In 2013, 69.6% of all email flows were spam. Links in spam emails may lead to users to websites with malware or phishing schemes, which can access and disrupt the receiver's computer system. These sites can also gather sensitive information from. Additionally, spam costs businesses around $2000 per employee per year due to decreased productivity. Therefore, an effective spam filtering technology is a significant contribution to the sustainability of the cyberspace and to our society. Current spam techniques could be paired with content-based spam filtering methods to increase effectiveness. Content-based methods analyze the content of the email to determine if the email is spam. Therefore, this project employs artificial neural networks to detect SPAM, HAM, and Phishing emails by applying features selection algorithm called PCA (principal component analysis). All existing algorithms detected only SPAM and HAM emails, but proposed algorithm designed to detect 3 different classes called SPAM, HAM, and Phishing. To implement this project, we have combined three different datasets called UCI, CSDMC and SPAM ASSASSIN dataset, where UCI and CSDMC datasets provided SPAM and HAM emails and Spam Assassin dataset provided Phishing emails. All these emails were processed to extract important features used in spam and phishing emails such as JAVA SCRIPTS, HTML tags and other alluring URLS to attract users.

1. INTRODUCTION

The rapid development of Internet technologies has immensely changed on-line users' experience, while security issues are also getting more overwhelming. The current situation is that new threats may not only cause severe damage to customers' computers but also aim to steal their money and identity. Among these threats, phishing is a noteworthy one and is a criminal activity that uses social engineering and technology to steal a victim's identity data and account information. According to a report from the Anti-Phishing Working Group (APWG), the number of phishing detections in the first quarter of 2018 increased by 46% compared with the fourth quarter of 2017 [1]. According to the striking data, phishing has shown an apparent upward trend in recent years. Similarly, the harm caused by phishing can be imagined as well.

For phishing, the most widely used and influential mean is the phishing email. Phishing email refers to an attacker using a fake email to trick the recipient into returning information such as an account password to a designated recipient. Additionally, it may be used to trick recipients into entering special web pages, which are usually disguised as real web pages, such as a bank's web page, to convince users to enter sensitive information such as a credit card or bank card number and password. Although the attack of phishing email seems simple, its harm is immense. In the United States alone, phishing emails are expected to bring a loss of 500 million dollars per year [2]. According to the

APWG, the number of phishing emails increased from 68,270 in 2014 to 106,421 in 2015, and the number of different phishing emails reported from January to June 2017 was approximately 100,000. In addition, Gartner's report notes that the number of users who have ever received phishing emails has reached a total of 109 billion. Microsoft analyzes and scans over 470 billion emails in Office 365 every month to find phishing and malware. From January to December 2018, the proportion of inbound emails that were phishing emails increased by 250%. Great harm and strong growth momentum have forced people to pay attention to phishing emails. Therefore, many detection methods for phishing emails have been proposed.

Various techniques for detecting phishing emails are mentioned in the literature. In the entire technology development process, there are mainly three types of technical methods including blacklist mechanisms, classification algorithms based on machine learning and based on deep learning. From previous work, the existing detection methods based on the blacklist mechanism mainly rely on people's identification and reporting of phishing links requiring a large amount of manpower and time. However, applying artificial intelligence (AI) to the detection method based on a machine learning classification algorithm requires feature engineering to manually find representative features that are not conducive to the migration of application scenarios. Moreover, the current detection method based on deep learning is limited to word embedding in the content representation of the email. These methods directly transferred natural language processing (NLP) and deep learning technology, ignoring the specificity of phishing email detection so that the results were not ideal [3], [4].

2. LITERATURE SURVEY

Gangavarapu et al. [5] aimed at elucidated on the way of extracting email content and behavior-based features, what features are appropriate in the detection of UBEs, and the selection of the most discriminating feature set. Furthermore, to accurately handle the menace of UBEs, this work facilitated an exhaustive comparative study using several state-of-the-art machine learning algorithms. This proposed model resulted in an overall accuracy of 99% in the classification of UBEs. The text is accompanied by snippets of Python code, to enable the reader to implement the approaches elucidated in this paper.

Srinivasan et al. [6] presented a new methodology for detecting spam emails based on deep learning architectures in the context of natural language processing (NLP). Past works on classical machine learning based spam email detection has relied on various feature engineering methods. This proposed method leveraged the text representation of NLP and map towards spam email detection task. Various email representation methods are utilized to transform emails into email word vectors, as an essential step for machine learning algorithms. Moreover, optimal parameters are identified for many deep learning architectures and email representation by following the hyper-parameter tuning approach. The performance of many classical machine learning classifiers and deep learning architectures with various text representations are evaluated based on publicly available three email corpora.

AbdulNabi et al. [7] introduced the effectiveness of word embedding in classifying spam emails. Pre-trained transformer model BERT (Bidirectional Encoder Representations from Transformers) is fine-tuned to execute the task of detecting spam emails from non-spam (HAM). BERT uses attention layers to take the context of the text into its perspective. Results are compared to a baseline DNN (deep neural network) model that contains a BiLSTM (bidirectional Long Short-Term Memory) layer and two stacked Dense layers. In addition, results are compared to a set of classic classifiers k-NN (k-nearest neighbors) and NB (Naive Bayes). Two open-source data sets are used, one to train the model

and the other to test the persistence and robustness of the model against unseen data. The proposed approach attained the highest accuracy of 98.67% and 98.66% F1 score. Alam et al. [8] developed a model to detect the phishing attacks using machine learning (ML) algorithms like random forest (RF) and decision tree (DT). A standard legitimate dataset of phishing attacks from Kaggle was aided for ML processing. To analyze the attributes of the dataset, the proposed model has used feature selection algorithms like principal component analysis (PCA). Finally, a maximum accuracy of 97% was achieved through the random forest algorithm.

Hassanpour et al. [9] presented some of the early results on the classification of spam email using deep learning and machine methods. This work utilized word2vec to represent emails instead of using the popular keyword or other rule-based methods. Vector representations are then fed into a neural network to create a learning model. This work has tested our method on an open dataset and found over 96% accuracy levels with the deep learning classification methods in comparison to the standard machine learning algorithms. Kumar et al. [10] discussed the machine learning algorithms and applied all these algorithms on this data sets and best algorithm is selected for the email spam detection having best precision and accuracy.

3. EXISTING SYSTEM

3.1 Support Vector Machine Algorithm (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

3.1.2 Disadvantages of SVM

- Support vector machine algorithm is not acceptable for large data sets.
- It does not execute very well when the data set has more sound i.e. target classes are overlapping.
- In cases where the number of properties for each data point outstrips the number of training data specimens, the support vector machine will underperform.
- As the support vector classifier works by placing data points, above and below the classifying hyperplane there is no probabilistic clarification for the classification.

3.2 Naïve bayes

Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle, and that is each feature being classified is not related to any other feature. The presence or absence of a feature does not affect the presence or absence of the other feature. Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. ... Naïve Bayes

Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes. To understand Naive Bayes theorem's working, it is important to understand the Bayes theorem concept first as it is based on the latter.

Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event. It is based on the following formula:

$$P(A|B) = P(A) * P(B|A)/P(B)$$

Where we are calculating the probability of class A when predictor B is already provided.

P(B) = prior probability of B

P(A) = prior probability of class A

P(B|A) = occurrence of predictor B given class A probability

3.2.1 Disadvantages of Naïve bayes

The Naive Bayes algorithm has the following disadvantages:

- The prediction accuracy of this algorithm is lower than the other probability algorithms.
- It is not suitable for regression. Naive Bayes algorithm is only used for textual data classification and cannot be used to predict numeric values.

3.3 AdaBoost Algorithm

What is the AdaBoost Algorithm?

AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps.

It is a one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. Adaboost should meet two conditions:

1. The classifier should be trained interactively on various weighed training examples.
2. In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.

How does the AdaBoost algorithm work?

It works in the following steps:

- Initially, Adaboost selects a training subset randomly.

- It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.
- It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.
- Also, It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.
- This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.
- To classify, perform a "vote" across all the learning algorithms you built.

3.3.1 Disadvantages of Adaboost

- AdaBoost is sensitive to noise data. It is highly affected by outliers because it tries to fit each point perfectly.

4. PROPOSED SYSTEM

This project employs artificial neural networks to detect SPAM, HAM, and Phishing emails by applying features selection algorithm called PCA (principal component analysis). To implement this project, we have combined three different datasets called UCI, CSDMC and SPAM ASSASSIN dataset, where UCI and CSDMC datasets provided SPAM and HAM emails and Spam Assassin dataset provided Phishing emails. All these emails were processed to extract important features used in spam and phishing emails such as JAVA SCRIPTS, HTML tags and other alluring URLS to attract users.

4.1 Pre-processing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

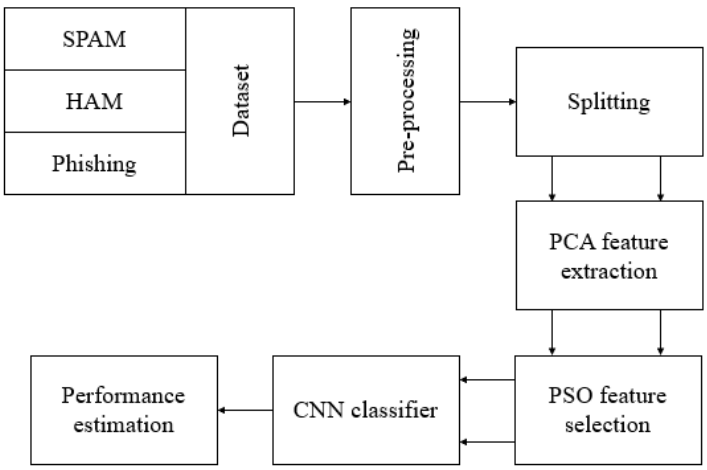
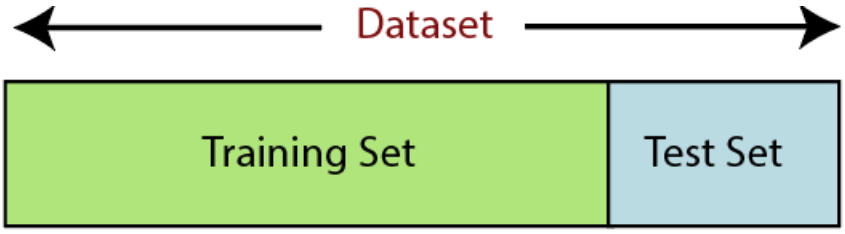


Fig. 4.1: Block diagram of proposed system.

4.1.1 Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

4.2 Principal Component Analysis (PCA)

Principal component analysis is an approach of machine learning which is utilized to reduce the dimensionality. It utilizes simple operations of matrices from statistics and linear algebra to compute a projection of source data into the similar count or lesser dimensions. PCA can be thought of a projection approach where data with m-columns or features are projected into a subspace by m or even lesser columns while preserving the most vital part of source data. Let I be a source image matrix with a size of n * m and results in J which is a projection of I. The primary step is to compute the value of mean for every column. Next, the values in every column are centered by subtracting the value of mean column. Now, covariance of the centered matrix is computed. At last, compute the eigenvalue decomposition of every covariance matrix, which gives the list of eigenvalues or eigenvectors. These eigenvectors constitute the directions or components for the reduced subspace of J, whereas the peak amplitudes for the directions are represented by these eigenvectors. Now, these vectors can be sorted by the eigenvalues in descending order to render a ranking of elements or axes of the new subspace for I. Generally, k eigenvectors will be selected which are referred principal components or features

4.3 Particle Swarm Optimization (PSO)

Feature selection method is used for generating an optimal number of features to be used for a certain task like classification. Particle Swarm Optimization (PSO) is an algorithm influenced by the habit of bird flocking or fish schooling. PSO is best used to find the maximum or minimum of a function defined on a multidimensional vector space. PSO has a main advantage of having fewer parameters to tune. PSO obtains the best solution from particles' interaction, but through high-dimensional search space, it converges at a very slow speed towards the global optimum. Moreover, regarding complex and large datasets, it shows poor-quality results. This algorithm is that it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process.

4.4 CNN Classifier

According to the facts, training and testing of CNN involves in allowing every source data via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from.

Advantages of proposed system

- CNNs do not require human supervision for the task of identifying important features.
- They are very accurate at image recognition and classification.
- Weight sharing is another major advantage of CNNs.
- Convolutional neural networks also minimize computation in comparison with a regular neural network.
- CNNs make use of the same knowledge across all image locations.

5. RESULTS AND DISCUSSION

Sample dataset

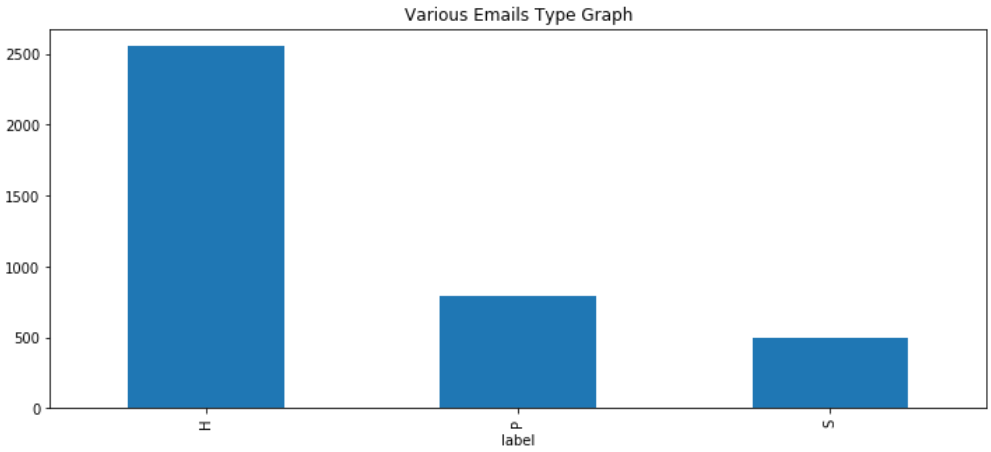
	body_forms	body_html	body_noCharacters	body_noDistinctWords	body_noFunctionWords	body_noWords	body_richness	body_suspension	body_verif
0	False	False	257	40	2	46	0.178988	False	
1	False	False	579	66	3	77	0.132988	False	
2	False	True	14972	1146	7	2395	0.159965	False	
3	False	True	1042	127	3	174	0.166987	False	
4	True	True	9205	427	4	968	0.105160	False	
...	...	...	...	...	...	...	...	...	...
3839	False	True	36506	820	9	1296	0.033655	True	
3840	False	True	10936	395	15	1185	0.106509	False	
3841	False	True	1181	80	2	115	0.097375	False	
3842	False	True	93089	1211	0	1215	0.013052	False	
3843	False	True	10383	144	0	177	0.017047	False	

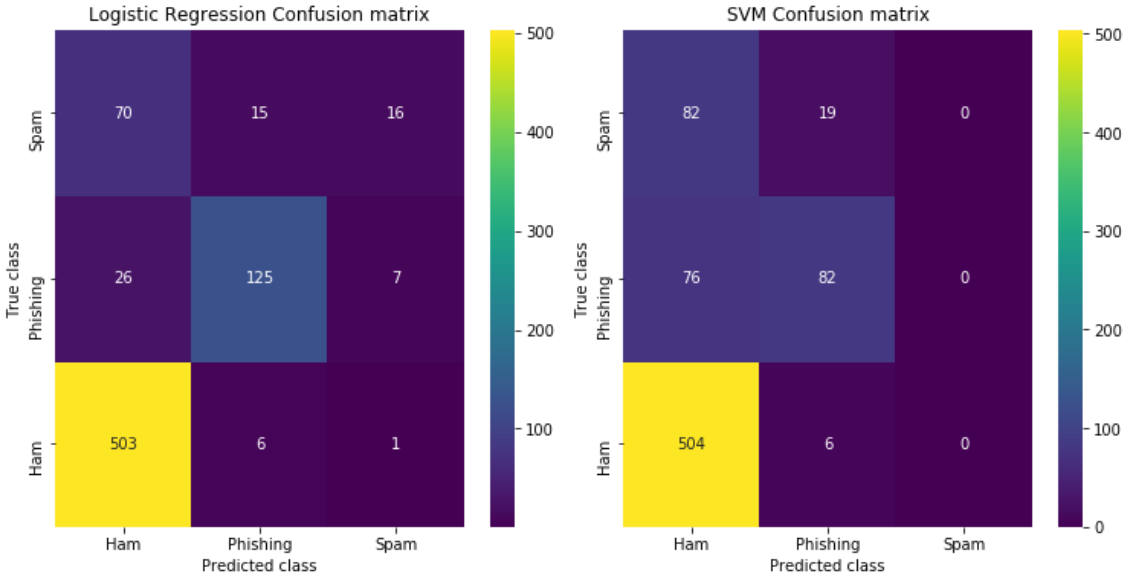
3844 rows × 41 columns

ebscript	...	url_noDomains	url_noExtLinks	url_noImgLinks	url_noIntLinks	url_noIpAddresses	url_noLinks	url_noPorts	url_nonModalHereLinks	url_ports	label
False	...	2	1	0	0	0	1	0	False	False	S
False	...	3	2	0	0	1	2	0	False	False	S
False	...	3	1	1	0	0	1	0	False	False	S
False	...	4	2	0	0	0	2	0	False	False	S
False	...	8	28	16	3	1	31	0	False	False	S
...	...	...	...	...	...	...	...	...	...	...	...
False	...	7	11	14	10	0	21	0	False	False	P
False	...	9	13	9	9	0	18	0	False	False	P
False	...	3	1	0	1	0	2	0	False	False	P
False	...	2	0	0	0	0	0	0	False	False	P
False	...	2	0	0	12	0	12	0	False	False	P

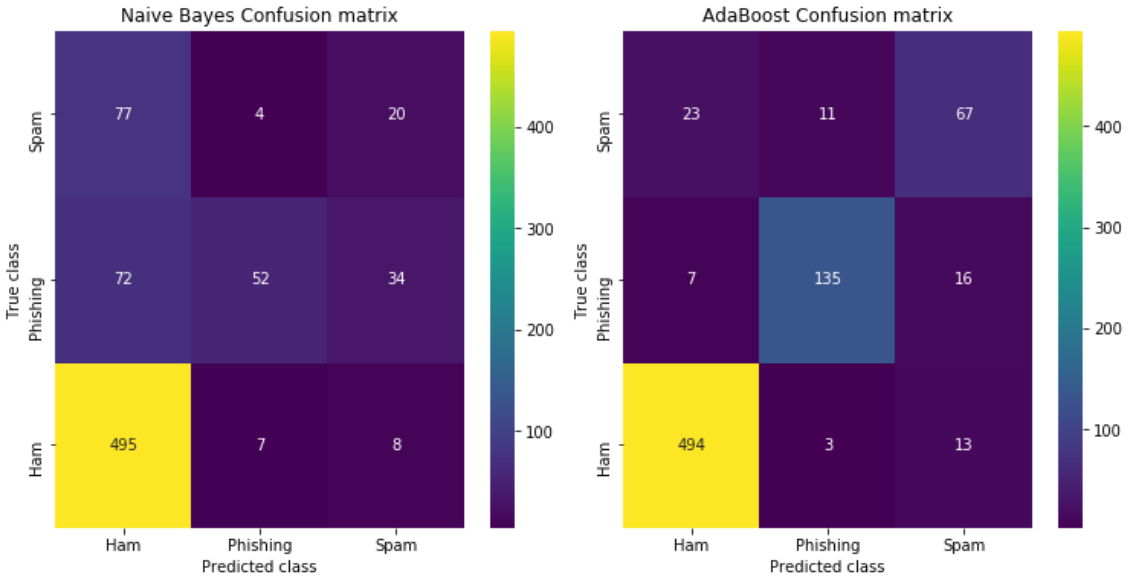
Dataset labels graph

ISSN- 2394-5125 VOL 10, ISSUE 03, 2023

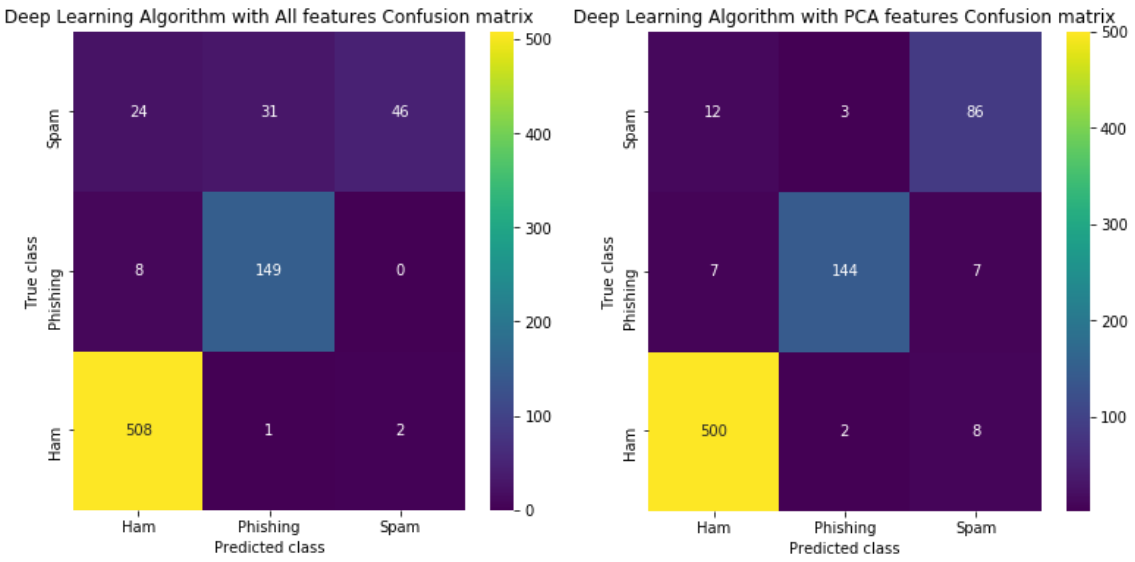



Logistic Regression Confusion matrix
True class
Spam
Phishing
Ham
70
15
16
26
125
7
503
6
1
Ham
Phishing
Spam
Predicted class
500
400
300
200
100

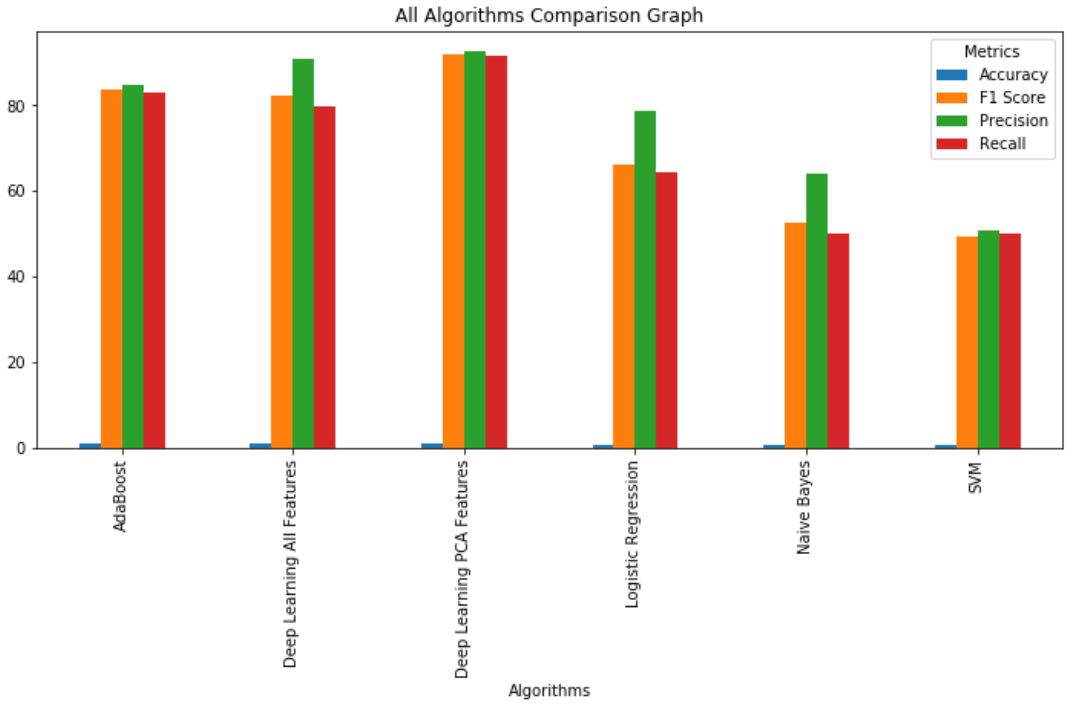
SVM Confusion matrix
True class
Spam
Phishing
Ham
82
19
0
76
82
0
504
6
0
Ham
Phishing
Spam
Predicted class
500
400
300
200
100
0


Naive Bayes Confusion matrix
True class
Spam
Phishing
Ham
77
4
20
72
52
34
495
7
8
Ham
Phishing
Spam
Predicted class
400
300
200
100

AdaBoost Confusion matrix
True class
Spam
Phishing
Ham
23
11
67
7
135
16
494
3
13
Ham
Phishing
Spam
Predicted class
400
300
200
100


Deep Learning Algorithm with All features Confusion matrix
True class
Spam
Phishing
Ham
24
31
46
8
149
0
508
1
2
Ham
Phishing
Spam
Predicted class
500
400
300
200
100
0

Deep Learning Algorithm with PCA features Confusion matrix
True class
Spam
Phishing
Ham
12
3
86
7
144
7
500
2
8
Ham
Phishing
Spam
Predicted class
500
400
300
200
100



6. CONCLUSION

This work has examined the performance of two kinds of random forest models. A real-life B2C dataset on credit card transactions is used in our experiment. Although random forest obtains good results on small set data, there are still some problems such as imbalanced data. Our future work will focus on solving these problems. The algorithm of random forest itself should be improved. For example, the voting mechanism assumes that each of base classifiers has equal weight, but some of them may be more important than others. Therefore, we also try to make some improvement for this algorithm.

REFERENCES

[1] S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 680-683, doi: 10.1109/Confluence47617.2020.9057851.

[2] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, "Credit Card Fraud Detection - Machine Learning methods," 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), 2019, pp. 1-5, doi: 10.1109/INFOTEH.2019.8717766.

[3] V. N. Dornadula, S Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms”, Procedia Computer Science, Volume 165, 2019, Pages 631-641, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.01.057.

[4] Itoo, F., Meenakshi & Singh, S. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. Int. j. inf. tecnol. 13, 1503–1511 (2021). https://doi.org/10.1007/s41870-020-00430-y.

[5] S. Dhankhad, E. Mohammed and B. Far, "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study," 2018 IEEE International

Conference on Information Reuse and Integration (IRI), 2018, pp. 122-125, doi: 10.1109/IRI.2018.00025.

[6] Ileberi, E., Sun, Y. & Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. J Big Data 9, 24 (2022). https://doi.org/10.1186/s40537-022-00573-8.

[7] More, Rashmi & Awati, Chetan & Shirgave, Suresh & Deshmukh, Rashmi & Patil, Sonam. (2021). Credit Card Fraud Detection Using Supervised Learning Approach. International Journal of Scientific & Technology Research. 9. 216-219.

[8] F. Carcillo, Y. L Borgne, O. Caelen, Y. Kessaci, F. Oblé, G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection", Information Sciences, Volume 557, 2021, Pages 317-331, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2019.05.042.

[9] S. Mittal and S. Tyagi, "Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 320-324, doi: 10.1109/CONFLUENCE.2019.8776925.