Smishing Messages Detection by Using Machine Learning Ensemble Techniques

Amey Dubey*
CSBS, STME
NMIMS, Indore
Indore, India
ameydubeydin@gmail.com

Shreyas Arora
CSBS, STME
NMIMS, Indore
Indore, India
shreyasarora0123@gmail.com

Dhruv Jore
CSBS, STME
NMIMS,Indore
Indore, India
dhruvjore@gmail.com

Dr. Divya Gautam
CSBS, STME
NMIMS,Indore
Indore, India
divya.gautam@nmims.edu

Abstract- Phishing is a type of cybercrime where a target or targets are contacted via email, phone call, or text message by someone posing as a legitimate organisation in order to trick people into disclosing sensitive information like passwords, banking and credit card information, and personally identifiable information. Similar to phishing, smishing is a technique used to steal sensitive data from victims by sending SMS messages to their mobile devices. The necessity for efficient tools to detect and counteract smishing attacks is growing in importance as they become more common. As a result, these messages' detection and identification are crucial for information security and the protection of sensitive data. In this study, we suggest a technique for the automatic detection of smishing attacks that are based on machine learning which has an extensive dataset that contains both legitimate and smishing messages and has shown promising results in separating malicious messages from normal ones.

Keywords– smishing, machine learning

I. INTRODUCTION

For many years, the cyberattack known as phishing has posed a serious danger to information security. By tricking the victim into clicking on a link or downloading a file, the attacker hopes to get sensitive information, such as login credentials, credit card details, or other private data. Phishing attackers use alluring offers and convincing impersonations to lure victims into providing sensitive personal and business information in exchange for favours or money. Falsification attacks, also known as social engineering attacks, are used by attackers to lure victims into visiting websites and downloading malicious software in order to get sensitive data such as user names, passwords, and banking information. The first quarter of 2019 saw a total of 1,238,161 phishing assaults, according to the Anti-Phishing Working Group (APWG). Attackers frequently steal information using phishing by tricking their victims. The victims' negligence makes the assailants' work much easier. Most phishing assaults are carried out via SMS, emails, and websites.

SMS and phishing were combined to create the term "smishing." Smishing attacks, in contrast to traditional phishing attacks, target victims through SMS messages instead of using other media to steal personal information. In order to fool the victim into disclosing their sensitive information, attackers create these messages to include a variety of components, such as smartphone applications, website links, greeting messages, or cell phone numbers. A variety of techniques are currently being developed, and some are even in use.

These include building spam detection systems to identify spam messages that contain any URLs or other material that may be harmful. An example of a smishing message is shown in Fig. 1.

Fig. 1. An example of a spam text message.

After a little processing, these algorithms balance the words in a message using different categorization methods. There has been earlier research that employed various techniques for identifying smishing and spam communications by using feature extraction and then feeding the extracted features into a classification model. The performance of these models is then evaluated using massive datasets including both spam and regular communications.

This paper presents a machine learning-based methodology for identifying smishing messages in relation to past research. In the suggested paradigm, a message is preprocessed before transmission. Afterwards, distinct characteristics are retrieved. Eventually, the characteristics are input into several classification algorithms in order to

classify authentic and phishing communications. Using several success measures, the performance of the proposed technique is assessed on a large dataset comprising valid and smishing communications. The evaluation findings indicate that the suggested technique provides a promising performance in identifying smishing messages by recognising the best rule-based classification algorithm.

II. LITERATURE SURVEY

This paper[1] provides an overview of Smishing attacks and their impact on mobile devices. It discusses the different types of Smishing attacks, including SMS Spoofing, URL spoofing, and Malware-based Smishing attacks. The paper also reviews the existing detection and prevention techniques for Smishing attacks, including rule-based approaches and machine learning-based approaches.

This paper [2], the authors described cyber attacks. Phishers and malicious attackers are frequently using email services to send false kinds of messages. This results in gaining personal credentials such as credit card numbers, passwords and some confidential data.

This paper [3] proposes a machine learning-based approach for detecting Smishing attacks. The proposed approach uses a deep learning model to classify SMS messages as either legitimate or Smishing. The paper reports promising results, with the proposed approach achieving an accuracy of 97.2%.

This paper [4] presents a systematic review of machine learning-based techniques for Smishing detection. The paper reviews the existing literature on Smishing detection and categorizes the approaches into different types of machine learning algorithms, such as decision trees, support vector machines, and deep learning. The paper identifies the strengths and weaknesses of each approach and provides insights into future research directions.

This paper [5] proposes an ensemble learning approach for Smishing detection. The proposed approach combines multiple machine learning algorithms, including decision trees, random forests, and gradient boosting, to improve detection accuracy. The paper reports promising results, with the proposed approach achieving an accuracy of 98.9%.

III. APPROACH AND FINDINGS

This section provides the experimental dataset and the validation findings for our approach to the smishing and legitimate message dataset.

A. Dataset

For this paper, the dataset used is the Spam Collection Dataset [6] to train our model and classify spam and normal messages and determine the features that make the smishing messages different from the normal messages.

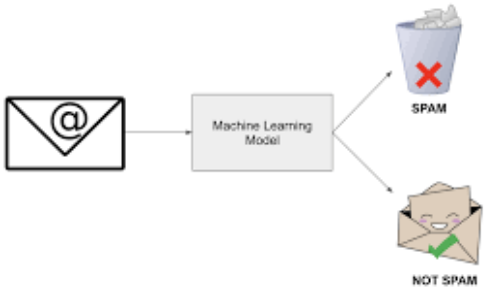

Fig. 3. Spam or not spam based on an ML Model

B. Preprocessing

Before proceeding with the data and using it to train our machine learning model, we perform preprocessing on our data to make the data viable to train a machine learning model for spam detection classification.

- Tokenization is performed on the data to consider every word as an individual token so that it can be processed individually.
- Stop words are removed from the sentences to make it easier to classify the sentences as spam. These words are usually conjunctions and prepositions which do not affect the topic that the sentence is discussing. Along with stop words, punctuations are removed as well since they play no role in deciding whether a message is spam or not.
- The words are usually converted to lowercase since this prevents the model from taking the lowercase and uppercase of the same word as different instances.
- Stemming is used to reduce the words to root form making it easier to process.

C. Attribute selection

A number of attributes are selected and chosen from the messages on which the classification will be based on:

1. URLs: it is observed that a large portion of smishing and spam messages contain malicious URLs which direct users to phishing websites.
2. E-mail addresses are another way that attackers use to lure in users in providing their personal information.
3. Phone numbers: The presence of phone numbers in messages are a sign that the message is spam message.
4. Money: Money is the most common topic in malicious messages since it is easier to attract people and convince them when there is money involved.
5. Common words: A good chunk of spam messages contain the same keywords which can be used to classify ham messages from smishing ones.

D. Feature Extraction

Post Preprocessing, the data is prepared for extracting the features from the dataset. Natural Language Toolkit or nltk [7], a python package is used to make the data balanced and easy to process.

New Columns containing the new processed sentences, number of sentences, characters and words in every message are added to the data. This is used to extract details about individual messages making it easier to process and classify different messages into spam and ham.

Fig. 3. and Fig. 4. show a plotting for the variation count between the number of characters that are used and the number of words that are used in the spam as well as the ham messages[8].

The plots show the count of the words, as well as the words that are used in the ham messages, are significantly more than when compared to ham messages.

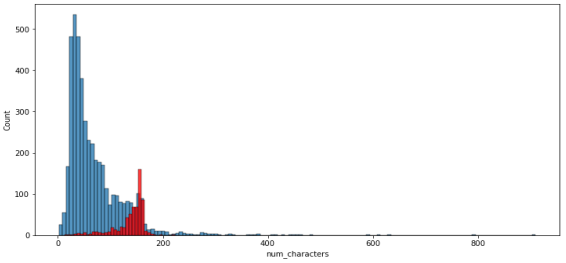

Fig. 3. Number of characters vs the count of messages. Red are spam messages and blue are ham messages.

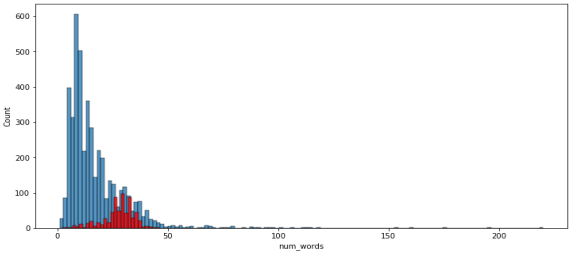

Fig. 4. Number of words vs the count of messages. Red are spam messages and blue are ham messages.

Extraction of the most repeated words which are being used in the spam and the ham messages, the Word cloud library is used paired with the collections library to find the count of all the words that were being used in the messages.

For further clarity in deciding what words are present in spam messages, Fig. 5. and Fig. 6. show a barplot of the 30 most common words and the frequency of their occurrences in messages that are spam and ham.

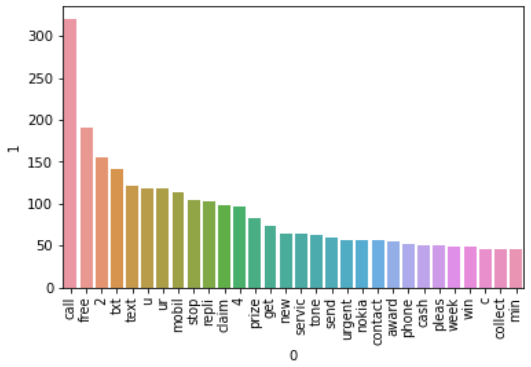

Fig. 5. Most common words in spam and the frequency of their occurrences.

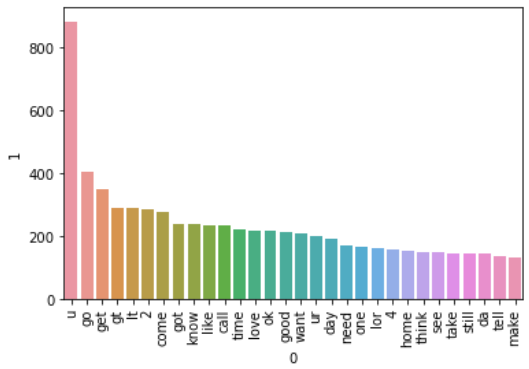

Fig. 6. Most common words in non-spam and the frequency of their occurrences.

E. Model Building

The features that are being extracted are then fed for numerical transformation as the algorithms take in numerical values. Therefore text vectorization[9] is performed for converting the textual values into an array. The chosen machine learning model will be based on the Naive Bayes Algorithm which involves and predicts on the basis of it. Gaussian Naive Bayes, Multinomial Naive Bayes and Bernoulli Naive Bayes as the classifier works on the principality of conditional probability to predict if a message is spam or if it is not- spam.The experimental ensembling work will further be used to improve the efficiency of the model and also compared with other algorithms for accurate results.

IV. EXPERIMENTAL WORK

In the proposed model, a variety of ensemble modeling techniques are used to compare the testing model achieved by the Naive Bayes as well as to predict an outcome to decide the final aggregated base of a model which is most suited according to the dataset.

1. With the help of Sklearn, various machine-learning algorithms are imported. Fig. 7. shows the classifier algorithms that were imported.

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

Fig. 7. Classifier-based algorithms imported

2. The models imported are trained with a function that has access to the training and testing data and later stored in a dictionary with their objects as keys and matrics scores as values in a sequential order without tweaking the hyperparameter tunings on a higher level.

3. Fig. 8. shows a graph with the accuracy and precision values plotted for the comparison of models.

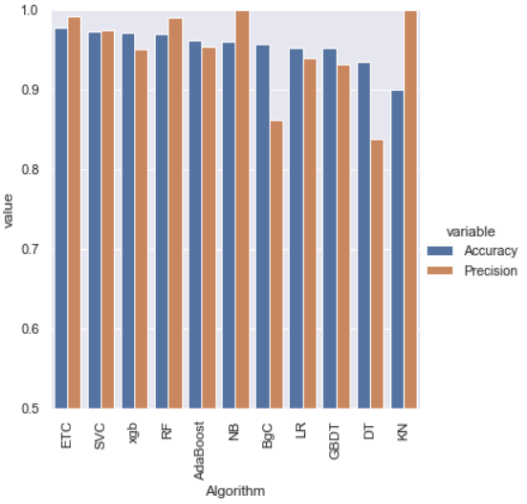

Fig. 8. Confusion Matrix values for Models

4. To further improve the Naive Bayes model, TF-IDF (term frequency-inverse document frequency) [10] based max features are vectorized by trials through experimentation with parameter features and scaling.

V. RESULTS AND DISCUSSIONS

In this section, the results from the experimental work are briefly discussed. The results show that the accuracy improved after applying the Max features to 3000, and the precision scores dropped significantly after scaling. Therefore Scaling will be flushed out and the Tfidf feature selection will be taken into consideration. Table. I. illustrate the scores for the matrics accuracy and precision before and after the max feature vectorization.

	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000
0	KN	0.900387	1.000000	0.905222	1.000000
1	NB	0.959381	1.000000	0.971954	1.000000
2	ETC	0.977756	0.991453	0.979691	0.975610
3	RF	0.970019	0.990826	0.975822	0.982906
4	SVC	0.972921	0.974138	0.974855	0.974576
5	AdaBoost	0.962282	0.954128	0.961315	0.945455
6	xgb	0.971954	0.950413	0.968085	0.933884
7	LR	0.951644	0.940000	0.956480	0.969697
8	GBDT	0.951644	0.931373	0.946809	0.927835
9	BgC	0.957447	0.861538	0.959381	0.869231
10	DT	0.935203	0.838095	0.931335	0.831683

Table. I. Matrics Scores for algorithms

Furthermore, Voting classifiers were compared with the Naive Bayes with the combined estimators of SVMs, MNBs, ETCs to check if a combination performs better than the chosen Naive Bayes algorithm. Fig. 9. Shows the scores for the classifier which is less than the results obtained through the Tfidf vectorization in Naive Bayes.

```
y_pred = voting.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

```
Accuracy 0.9816247582205029
Precision 0.9917355371900827
```

Fig. 9. Accuracy and Precision scores for Voting Classifier

VI. CONCLUSION

This paper proposes a machine learning-based methodology for identifying phishing messages. The proposed model consists mostly of processing, extractions and classifications. A sufficiently big dataset including genuine and smishing messages is utilized to test the proposed model's performance. The experimental work and Results reveal that the Multinomial Naive Bayes model based on conditional probability performs the best and has a high matrics score when compared to several other classifiers if trained with extra hyperparameter tuning and significant feature transformation. The experimental findings reveal that authentic and phishing communications are categorized with a high percentage of success. Due to the model's high efficiency after certain experimental techniques, the model may readily replace widely existing blacklist and whitelist techniques. Incrementing the accuracy of the model for expanding its performance capabilities and the study of smishing messages in various languages, distinct characteristics, and a variety of categorization techniques remain possible future developments.

REFERENCES

[1] Kharat, S., & Bhide, S. (2021). A review on Smishing Attacks: Detection and Prevention. International Journal of Emerging Trends in Engineering Research, 9(2), 213-218.

[2] Sunil B. Rathod, Tareek M. Pattewar "Content Based Spam Detection in Email using Bayesian Classifier", presented at the IEEE ICCSP 2015 conference.

[3] Singh, R., Singh, P., & Bhatnagar, S. (2020). Smishing Attack Detection Using Deep Learning. International Journal of Computer Science and Mobile Computing, 9(8), 20-26.

[4] Miah, S., Islam, M. S., & Kabir, M. H. (2021). Machine learning-based Smishing detection techniques: A systematic review. Journal of Information Security and Applications, 60, 102789.

[5] Wang, X., Wei, Z., Chen, Y., & Chen, Y. (2019). Smishing Detection with Ensemble Learning Algorithms. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (pp. 1232-1237). IEEE.

[6] "Kaggle Dataset- UCI Machine Learning". [Online]. Available at: [Accessed: January 22, 2023] https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[7] Li Junfei. (2019). Research on the Application of Natural Language Processing Toolkit in College English Teaching. Education Modernization 92, 136-137.

[8] Maram, Sai Charan Reddy, SMS Spam and Ham Detection Using Naïve Bayes Algorithm (August 21, 2021). Available at SSRN: https://ssrn.com/abstract=3908998 or http://dx.doi.org/10.2139/ssrn.3908998

[9] C. Liu, Y. Sheng, Z. Wei and Y. Yang, Research of text classification based on improved TF-IDF algorithm, in: 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), 2018, pp. 218–222. doi: 10.1109/IRCE.2018.8492945.