CMPE-255 Project - Detection of Phishing/Suspicious Emails

Shreya Goyal
shreya.goyal@sjsu.edu

Rishitha Bandi
rishitha.bandi@sjsu.edu

Shashank Raghuvanshi
shashank.raghuvanshi@sjsu.edu

Praveen Kumar
praveen.kumar@sjsu.edu

***Abstract*—This paper focuses on proposing a model for phishing email detection using a set of features from the different email parts. Features are extracted from the EML file using the python program in the preprocessing phase. 15 features in total are extracted from the EML file. These features will be assigned weight according to the importance of each feature. Different classifiers will be used to classify emails based on the features extracted from the email. The goal is to find the best suitable classifier for phishing email detection with the greatest accuracy. The model discussed here has successfully achieved 93.65% accuracy with the Random forest algorithm. Other classification algorithms are also analyzed before achieving this accuracy.**

***Index Terms*—Phishing, Spam, Data Mining, Random Forest**

I. Introduction

Phishing is an online attack in which phishers used to send an email that seems to be legitimate but it is used to steal personal identity information and financial account credentials. Phishers want to engage users in the email and email may have inviting URLs, attachments, inline images, etc. A phishing email might have any or all of these defined things. Over time there has been a considerable amount of increase in online transactions and with that frequency of phishing email have also increased in our inbox. Training programs organized can help in mitigating phishing emails by raising awareness among users. However, training programs have to be made continuous for knowledge retention otherwise it is for a few days only.

A phishing attack can happen at different levels like a Web page, Email, SMS, or instant voice call. Whether the phisher took any of the approaches, it ends up routing to a fraudulent website. According to the survey, the most widely used technique is via email. Phishers try to deceive customers by sending an email that contains a link for the web page. The email contains instructions to go to a fake webpage that seems to be legitimate but originally it ends up stealing personal information.

Many solutions have been built to detect phishing attacks at different levels such as website level, Email level, etc. Models of website level are used to check that the website to which the user has been redirected is legitimate or spoofed. Models of email level are used to check that the email which is in the user's inbox is legitimate or not. This strategy is of an earlier stage.

In this study, a model is built to detect phishing attacks at an early stage as this is a much better approach than detection at the website level as it might lead to slowing down of the website. As when the user clicks on a link and after that model checks whether it is legitimate or not.

II. Relevant Work

We found a lot of recent interesting research around the problem of detecting Phishing Emails, many of which vary in the way they approach the problem in notable ways compared to each other. Here we are listing a brief description of some of them.

1) [1] Focused mainly on content-based filtering model using a statistical classifier, and along with the usual set of features (that are common) came up with few new features to look out for classification. For extracting topic features a new "latent Class Topic Model(CLTOM)" is used, which finds word clusters which are more specific for the distinction of phishing emails from legitimate emails rather than being general purpose. A model based on the Markov chain is used for sequential analysis of text and external links. A method to detect *Hidden Salting* is also developed, which is essentially a trick used in phishing emails to cause emails to appear normal to human eye, but contains text which makes automated message processing hard.

   While this approach was able to reduce rates of phishing attempts to less than 1%. The author has not commented on the efficiency of the approach as multiple statistical models are being used for feature extraction, however the algorithm seems to be robust, and less dependent on manual selection of features.

2) One of the research [2] proposed the use of lexical analysis of URL in the content of emails for classification, building upon their earlier work [3]. The main focus of the research is to enhance the classification accuracy of existing anti-phishing filters, by supplementing it with their approach. The proposed lexical URL analysis also considered relative positions of tokens to increase precision, and each token is assigned a real value based on how likely it is to be in a phishing email.

3) [4] Focussed mainly on the preprocessing step and extracted 23 hybrid features based on both content and header of the emails. The feature extraction method reduced the noise in the data because of missing values and had reasonable default values for each missing value. The authors did a comparative analysis on a set of classification algorithms with different methods of feature extraction along with their own. The results of the study indicated that the Random Forest, J48 and PART algorithm performed well, with PART having the lowest false positive rate.

4) [5] Used n-gram analysis of Message-ID field of email header to extract useful features for classification. The authors argued that even if the Message-ID field is an optional attribute, almost 99% of legitimate emails have that field, which forces the creators of phishing emails to include that, to avoid suspicion. Spoofing Message-Id requires advanced technical knowledge [6] so most of the phishing emails don't do that, which makes Message-Id field a valuable attribute for classification. [4] Also used Message-ID field to compare the domain name with the sender's domain name as one of the features.

III. KDD Process

The data mining process for knowledge extraction and pattern discovery has the following steps in the detection of suspicious emails.

A. Data Collection

In this phase of the KDD process, the primary focus is to collect emails. Collection of phishing and legitimate emails is needed among which some percentage of the dataset needs to be labeled also for the training purpose. For our model, we are using a 'SpamAssassin' [7] dataset having three folders two for ham emails and one for spam emails.
Ham email count: 2822 (2570 + 252)
Spam email count: 1398

B. Data Integration and Cleaning

In this step, data from different sources are combined and put into separate folders for legitimate and phishing emails. Inconsistent and incomplete data will be discarded from the datasets. However, after data cleaning the number of records may lessen.

C. Data Transformation and selection

In this step, unstructured data is converted into a structured format so that data models can be created.
Converted each of these files into separate eml files using a free tool 'mbx2eml'. These eml files are processed to extract features into CSV files.

There are two folders one for spam emails and one for ham emails so two CSV files will be generated one for each spam and ham.

D. Data Mining

In this step, data mining algorithms need to be applied to the preprocessed data that we have as output from previous steps of KDD.

Firstly, we will check the importance of different features that are extracted from the dataset. Correlation matrix will be drawn for the features to check which features are highly correlated so those can be removed. After this exploratory analysis, different classification algorithms will be applied to create the best model with the highest accuracy.

E. Evaluation

In this step, the model is evaluated using test data. We have divided the dataset into 70:30 for training and test data. The model will be evaluated using a 30% dataset. For evaluating the model, Precision, Recall, F1 score, AUC (Area under the curve), and accuracy metrics are used.

In further detail, Suppose NS is the number of suspicious emails, NL is the number of legitimate emails that are detected correctly. Ns is the number of suspicious emails detected legitimate and Nl is the number of legitimate emails detected as suspicious. S is the total number of suspicious emails, L is the total number of legitimate emails.

1) True positive TP: the number of suspicious emails correctly detected as suspicious.
$$TP = \frac{NS}{S}$$
2) True Negative TN: the number of legitimate emails correctly detected as legitimate.
$$TN = \frac{NL}{L}$$
3) False positiveFP: the number of legitimate emails incorrectly detected as suspicious.
$$FP = \frac{Nl}{L}$$
4) False Negative FN: the number of suspicious emails incorrectly detected as legitimate.
$$FN = \frac{Ns}{S}$$
5) Precision
$$Precision = \frac{TP}{TP + FP}$$
6) Sensitivity
$$Sensitivity = \frac{TP}{TP + FN}$$
7) accuracy
$$accuracy = \frac{TP + FN}{TP + FP + TN + FN}$$

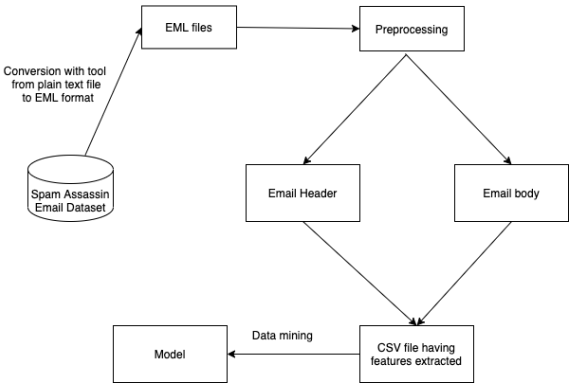
With the above metrics calculated the performance of the hypothesis is measured and the best suitable classifier is identified.

IV. Proposed Methodology for Phishing Email Detection

In this project, we started with the dataset having a text file for each email. Using an online mbx2eml tool, convert each text file to an eml file. Applied preprocessed techniques on these eml files to extract features. Eml files having unstructured data so converted it to a structured format and then extracted features. These features are treated as input to the model.

A. High Level Architecture

We are using the Spam Assassin email dataset [7]. First converted text email files into eml files. Then these files are preprocessed to extract key features from the email header, body, and URL embedded in the content of the email. These features will be written into a CSV file. The model will be created by applying different classification algorithms.



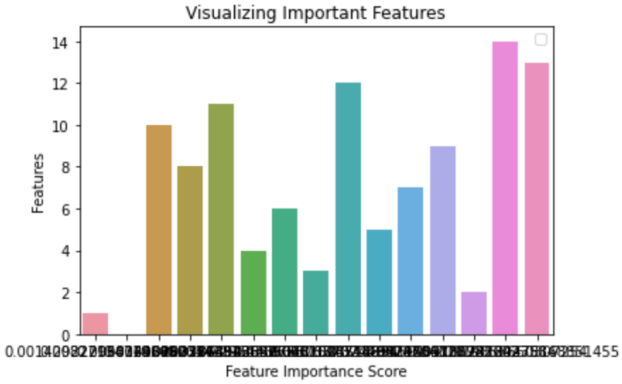
B. Feature Engineering

Below are the features that are extracted from each section.

- **Email Header:**
  1) Number of recipients from attributes need to be considered.
  2) Email content type from the header.
     a) Check if email is HTML type.
     b) Check if email has multipart.
  3) Check if there is a difference between sender and reply email address.
- **Email Body:**
  1) Check if the email body contains the URL. The features to be extracted from the URL section will be set 'False' if email doesn't contain any URL.
  2) Number of inline attachments.
  3) Number of attachments.
- **URL:**
  If the email body has URLs these features need to be extracted otherwise default values are stored.
  1) Check if the URL has @.
  2) Check if the IP address is used as a domain name in any of the URLs.
  3) If any of the URLs has // as this is used for redirection.
  4) If any of the URL has 'mailto' as this can be used to send information using email.
  5) If any URL domain has '-'
  6) Length of the URL. Calculated for each URL and returns maximum number.
  7) Check if the domain of the URL is getting resolved.
- **Subject:**
  1) Check the subject length
  2) Check if subject contains any non-ascii character
  3) Check if the subject contains any numbers.

15 features in total are extracted from eml files to create the model. URL features work like this.

- **Length of the URL :** If the length of the URL is more than 85 then it is very suspicious. These are used to hide the suspicious part.
- **URL with ip address:** If the URL is using an IP address as an alternative to the domain then the end user can have some doubt that personal information can be stolen.
- **URL has '-' :** This symbol is not used in legitimate emails. It is used to add suffixes and prefixes in the domain name.
- **URL has '@':** Everything prior to the @ sign is ignored in URL. Actual URL starts after the @ symbol. Used for spam emails.
- **URL has '//':** This sign is used for redirection in URL. Need to check the last occurrence as if it is greater than 7 then it is a phishing email. For URL starting with HTTP, it will be 6 and URL starting with HTTPS, it should be 7.



As above, the figure is showing the importance of the features and features are in the below order.

1) No of inline attachments
2) No of attachments
3) Length of subject
4) Subject having non-ascii character
5) Subject contains numbers
6) No of recipients
7) Sender and reply to are same
8) URL having IP address
9) Length of the URL
10) URL having '@'
11) URL having '//'
12) URL having '-' in domain name

13) 'Mailto' in URLs domain
14) Content type of the header
15) Length of the body

As we can observe, the Content-Type of the header has the highest importance and after that length of the body has the second-highest importance.

C. Comparison Of Classification Algorithms

Features have been extracted into the CSV file. Now, the following classification algorithms are applied to the features set:

1) **Logistic regression:** It is a binary classification algorithm used to classify and give scores between 0 to 1.
2) **SVM:** Support vector machine is based on the principle of creating a line in 2D or hyperplane in multi-dimension to separate data into classes.
3) **Naive Bayes:** Classifier based on Naive Bayes theorem. It works on the assumption that the presence of each feature in a class is independent of the presence of other features in class.
4) **KNN:** K nearest neighbour is a non parametric algorithm. It stores all available cases data and predicts new cases on the basis of similarity.
5) **Random Forest:** It works on the principle of decision trees which are created by selecting the random subset of decision trees. It selects the model by voting from different decision trees.

After going through the literature related to the classification of phishing and legitimate emails, the following algorithms were applied for classification. In other words, the most widely used algorithms for classifying ham and spam emails are these. Selected these top 5 algorithms and applied these onto our dataset, following metrics are observed.

TABLE I: Analysis of classification algorithms

Algorithms	***Accuracy***	***F1-Score***	***Precision***	***Recall***	***AUC***
Logistic Regression	80.84%	67.87%	86.53%	55.83%	84%
SVM	81.57%	68.07%	87.91%	55.53%	77%
Naive Bayes	78.40%	61.72%	82.69%	49.23%	82%
KNN	86.16%	79.22%	84.44%	74.61%	91%
Random Forest	92.26%	88.88%	92.72%	85.35%	97%

From the table, we can observe that Random forest is the best classification algorithm for phishing and legitimate emails. It is because of its nature. Random Forest (RF) algorithm creates different decision trees and creates vectors for each tree. RF is used to classify an input dataset using all the decision trees. In other words, each tree used to vote for the label. Label with the highest no of votes is selected as the result. As features are extracted from the email contents, not the hybrid features so Random Forest works best with this dataset.

TABLE II: Mean FPR and TPR for Classification Algorithms from ROC Curve

Algorithms	***FPR***	***TPR***
Logistic Regression	33.56%	73.62%
SVM	41.47%	69.94%
Naive Bayes	32.81%	69.41%
KNN	22.47%	66.11%
Random Forest	10.36%	80.30%

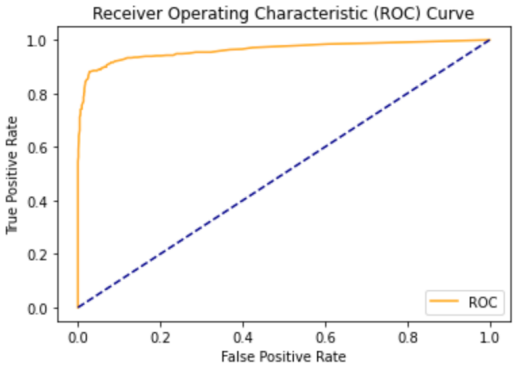


Fig. 1: ROC Plot for Random forest algorithm

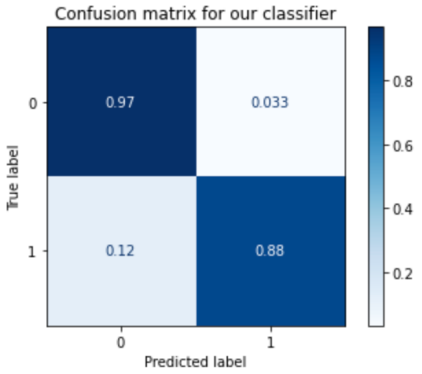


Fig. 2: Confusion matrix for Random forest algorithm

D. Selected Model- Random Forest

Curve plotting False positive rate on X-axis and True positive rate on Y-axis. Showing the model performance at different thresholds.
ROC curve for random forest and this is the model with the highest area under the curve.

Relying just on accuracy for deciding the model, as it does not specify where our model is giving error. It is a better approach to check where our model is lacking and how the model can be improved. Correct and incorrect predictions are summarized by the count of each class, from confusion matrix.
TPR: 0.97
FPR: 0.033
FNR: 0.12
TNR: 0.88

For this model, FNR is 0.12 which needs to be improved to increase the accuracy.

V. Rubric

- **Data Collection:** Using Spam Assassin [7] dataset as it is the only available approved dataset and has been used previously in other works of literature. Collected 2428 ham emails from easy_ham and hard_ham folder, 1276 spam emails from spam2 folder.
- **Variety:** The downloaded dataset has text files for each email. Each email has attachments, URLs, Html content, javascript, and Text content. In the preprocessing step, we processed all types of data mentioned in an email.
- **Visualization:** Feature analysis, Comparison of classification algorithms is done using figures that are attached here.
- **Significance to the real world:** In today's world when everything is online from shopping for clothes to buying a house. Online transactions are increasing day by day and with that phishing, activities are also increasing. Phishing can be done via emails, SMS, websites, voice calls, instant messages, etc. As pointed out by the Anti-Phishing Work Group's (APGW) report from the 4th quarter, emails are the common medium used for phishing. The importance of detecting phishing emails is only going to increase as more and more interactions among users are happening digitally.
- **Data reduction:** Extracted 20 features into the CSV but after drawing the correlation matrix and feature analysis graph, remove 5 features. Features having less importance were ignored like no. of inline attachments, no. of URL.
- **Veracity:**We analyzed a total of 3704 records having quality data for better model creation. This can be observed from the feature analysis graph showing the importance of each one.
- **Tools usage:**'mbx2eml' used to convert each text file into an eml file. We used Jupyter Notebook, Visual Studio Code (for local testing), and google collaboratory.
- **Volume:** 40 MB dataset is downloaded having 2428 ham emails and 1276 emails. However before settling on a dataset from spam assassin [7], we tried experimenting with Enron dataset [8], it had roughly about 5 lac emails in total, but all of the emails were only plain text type and were also specific to a single corporation, so there was not much variety in emails, even though it had large volume.
- **Version Control:** Google colab linked with Google drive. Version history is maintained in Google drive.
  https://colab.research.google.com/drive/1ckV7zTkhM8PxZXlCVVauv9FBaM_aH5sT?usp=sharing
  Github- Added the final colab notebook on github also. https://github.com/shrey1234/project_255

- **Lessons Learnt:**
  - We should never assume anything about the dataset without observing and validating our hypothesis based on our observations, whether it is true for all datasets.
  - Many times it is better to go with simpler approaches than the complicated ones.
- **Velocity:**Not applicable in this scenario as our dataset is not in real time.
- **Evaluation of performance:** Used Accuracy,ROC curve, Confusion matrix to evaluate the model.
- **Technical Difficulty:**
  - In the beginning, we tried to parse the dataset with Java, and most of the time we were getting so many exceptions, that we were only able to parse a small subset of the data. Then we decided to switch to python.
  - Different emails had different formatting for keys for the same type of data, so without exploring this problem and printing out a list of all keys we would have missed on a lot of values for the features, so we had to take that into account.
  - Initially, we thought of extracting some features from the text data of the emails, but there were different kinds of encodings in the dataset. Like there were printed quotable encodings used in some of the emails, which added some useless characters when we tried extracting text from Html.
  - We could not rely on mime type for the text, for example, many emails had Html in while the mime type was plain text.
  - Even after resolving the above problems while cleaning the text, we later came to realize that the emails were not all in English and were in multiple languages, so finally, we had to drop the idea of extracting text-based features because of this.
  - We were looking for a recent dataset which had variety, we experimented with the dataset of enron [8] but it had not much variety even though volume was large and from untroubledhttp://untroubled.org/spam/ which has a large number of spams, but there were no examples of legitimate emails, and many emails that we got were in Chinese for 2019, so we decided to drop the idea to use it as well.

References

[1] A. Bergholz, J. Beer, S. Glahn, M.-F. Moens, G. Paass, and S. Strobel, "New filtering approaches for phishing email," *Journal of Computer Security (JCS)*, vol. 18, 04 2009.

[2] M. Khonji, Y. Iraqi, and A. Jones, "Enhancing phishing e-mail classifiers: A lexical url analysis approach," *International Journal for Information Security Research*, vol. 2, pp. 236–245, 06 2012.

[3] ——, "Lexical url analysis for discriminating phishing and legitimate websites," in *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, ser. CEAS '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 109–115. [Online]. Available: https://doi.org/10.1145/2030376.2030389

[4] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2015, pp. 1–8.

[5] R. M. Verma and N. Rai, "Phish-idetector: Message-id based automatic phishing detection," in *Phish-IDetector: Message-Id Based Automatic Phishing Detection*, ser. ICETE 2015. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2015, p. 427–434. [Online]. Available: https://doi.org/10.5220/0005574304270434

[6] P. S., "Email 'message-ids' helpful for forensic analysis?" *Australian Digital Forensics Conference*, 2008.

[7] A. Beato. Spamassassin public corpus. [Online]. Available: https://www.kaggle.com/beatoa/spamassassin-public-corpus

[8] I. Androutsopoulos. Enron email dataset. [Online]. Available: http://www2.aueb.gr/users/ion/data/enron-spam/