National College of Ireland

DETECTION OF PHISHING AND SPAM EMAILS USING ENSEMBLE TECHNIQUE

MSc Internship
CyberSecurity

Michael Oluwasegun Akinrele
Student ID: X18109489

School of Computing
National College of Ireland

Supervisor: Mr Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	**Michael Oluwasegun Akinrele**
Student ID:	**X18109489**
Programme:	**CyberSecurity** **Year: 2019**
Module:	**MSc Internship**
Supervisor:	**Mr Vikas Sahni**
Submission Due Date:	**12/12/2019**
Project Title:	**Detection of Phishing and Spam Emails Using Ensemble Technique**

Word Count: 5234 **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Phishing and Spam Emails Using Ensemble Technique

MICHAEL OLUWASEGUN AKINRELE
X18109489

Abstract

Most of the cyber breaches in the world today are done based on fraudulent activities. Phishers and Spammers come up with new and hybrid techniques all the time to circumvent the available software and techniques, which shows that all organizations are covered by unbroken threat. Among the approaches developed to stop email spam and phishing, filtering is a popular and important one. Common uses of email filters include organizing incoming emails and removal of spam, while phishing is detected by validating email body, URLs, etc. In this study, we proposed an ensemble approach for phishing and spam filter-based feature selection methods with the goal to lower the feature space dimensionality and increase the accuracy of spam and phishing review classification. We collected different public datasets and trained on Machine Learning (ML) based mRMR (Minimum Redundancy Maximum Relevance) models and Ensemble models. Experimental results with seven classifiers show an average of 83% accuracy which made the feature selector improves the performance of spam and phishing classifiers. And can legitimate future email cyber-attacks with a scope for future research and expansion.

1 Introduction

In recent years, phishing has grown tremendously and presents a critical challenge to world security and economy. Criminals attempt to persuade naive online users to reveal sensitive information, such as account numbers, passwords, social security or other personally identifiable information records. Spam refers to unsolicited bulk mail (junk email), which usually involves sending to a significant number of recipients, who never submitted a message with ads or even meaningless content. Spam is induced by supplying recipients with a payload containing advertisements for an item (probably useless, unlawfully or not existing), incentive for theft, endorsement of a cause or software malware to hijack the recipient's device. Since it is so cheap to send out emails, only a very small number – maybe one in ten thousand or less – of targeted recipients need to accept and reply to the cost load so that spam can be useful to their transmitters. Spam in law courts all over the world is a highly contested topic, especially with regards to the authorization to send messages to private or public email addresses. Spammers are continuously developing new ways to attack, apart from email messages, e.g. by using Instant Messenger, weblogs, SMS or spam filtering tools to bogus search engines.

Research Question

1. Question: Are there any open corpus for detecting spam, ham and phishing emails?

Description: As the existing literature suggest that there is very less online resources for detecting spam, ham and phishing emails. The aim of this question is to explore for open source data corpus for detecting spam, ham and phishing emails.

2. Question: Does combining all three Spam, Ham, Phishing datasets help in finding a future cyber-attack?

Description: Current techniques uses either spam filtering or phishing detection but there are very less research done. Proposes a new hypothesis to examine the behaviour when three of these metrics are clustered.

3. **Question: What features are helpful in detecting Spam, Ham, Phishing emails?**

Description: Feature selection is an important problem for pattern classification systems. This plays a vital role in detecting Spam, Ham, Phishing emails.

4. **Question: How does different machine learning classifier's perform for detecting spam, ham and phishing emails?**

Description: This is to test how different ML classifiers like mRMR and non-mRMR classifiers work.

2 Background and Related Work

2.1 Background

2.1.1 Aims of email spam

Spammers tend to lure innocent computer users to purchase legal or prohibited products and services. A newsgroup or mailing list had been flooded with irrelevant or inappropriate messages in the past, the spam has changed considerably in the present days because it has been oriented to the money side. The popular targets for spamming are:

- Products and services marketing and distribution.
- Collection of sensitive information like emails and passwords to bank accounts through online gambling, bank fraud, and assistance requests.
- Concepts and philosophies of ads.
- To send spam viruses:
  - The computer of the recipient is corrupted and transformed into robot PCs which create harmful botnets.
  - Theft personal information and crime.

2.1.2 Methods for Email Spamming

Spammers generally offer their services (sometimes illegal) to individuals or organizations seeking a "less costly" way of advertising their brands. Spammers offer the databases to advertisers or sell the whole service: set, message layout to avoid detection, and spam email delivery.
The spam response is sent to the email address lists, compiled in different ways:

- Using software in public spaces, websites or unsafe mail servers to look for email addresses
- Flooding or dictionary spamming
- *e-pending*-Valid search addresses and criteria for individuals

- *Usenet posting*
- Subscribing to e-mail lists, to view all the e-mail addresses accessible
- Malware access to user directories or confidential information
- Wiretapping network traffic
- Stealing databases

Phishing attacks on non-traditional sites, such as automotive associations, are also underway. Spear phishing is referred to as extremely attack on employees or members of a company, government agency, or organization. The targeted scams have a great deal of potential harm. As techniques such as secure encryption of emails are still not common and require high administrative burdens, we are focusing on metameasures based on phishing email content.

2.1.3 Types of Phishing Attacks

It is possible to distinguish between two different forms of phishing: malware-based phishing and deceptive phishing. Malicious software is transmitted by defective e-mails or by using the computer's security vulnerabilities and loaded on the user's machine for malware-based phishing. Afterward, the malware can capture user input and the phisher can receive confidential information. The other is deceptive phishing, where a phisher sends tricky emails from a reputable institution such as a bank. In general, the pishers urges the user to click on a link to a fraudulent website where the user is requested to disclose personal information, for example, passwords. The attacker exploits this information, e.g. by withdrawing money from the user account. A variety of techniques in phishing are common:

- Social engineering: The creation of plausible stories, situations, and techniques for the production and use of personalized information in a convincing context.
- Mimicry: Both the website and the email link are very closely related to the official e-mails and the official websites of the target group.
- Email spoofing: Phishers mask the sender's actual identity and give the client a fake sender address.
- URL hiding: Phishers try to make official, legal and obscure the actual link addresses of the URLs in e-mails and the linked website.
- Invisible content: In phishing emails or the website, phishers insert information that is invisible to the user and aims to fool automatic filters.
- Image content: Phishers only graphically project images containing the text of the message

Phishing is causing huge financial losses. The targeting organizations are hesitant to give accurate information on losses in order to prevent a bad press. Gartner published an online survey in 2017 with 4500 US individuals. Some 3.3% reported losing money in phishing scams in 2017. That's 3.6 million people in the United States. The average loss was $886, resulting in $3.2 billion in total losses. Such estimates will not

compensate for credibility erosion and decreased customer confidence[1]. Alarmingly, 11% of those involved say they do not use any security software, like anti-virus or anti-spyware.

2.1.4 Different types of spam

Spam is divided into three categories - nuisance; scams and phishing and malware.
Figure 1: Types of spam

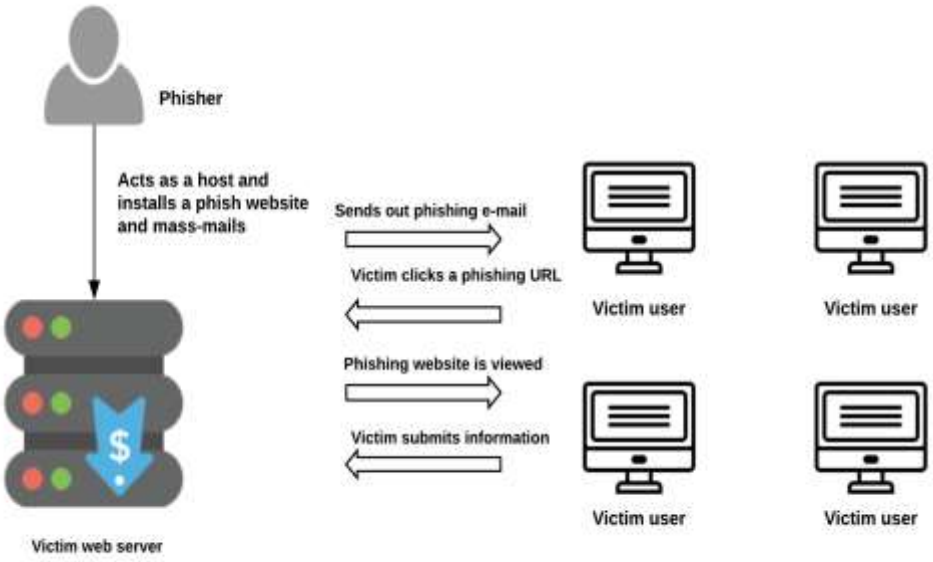

Figure 2: Life cycle of phishing email

[1] https://www.forbes.com/sites/leemathews/2017/05/05/

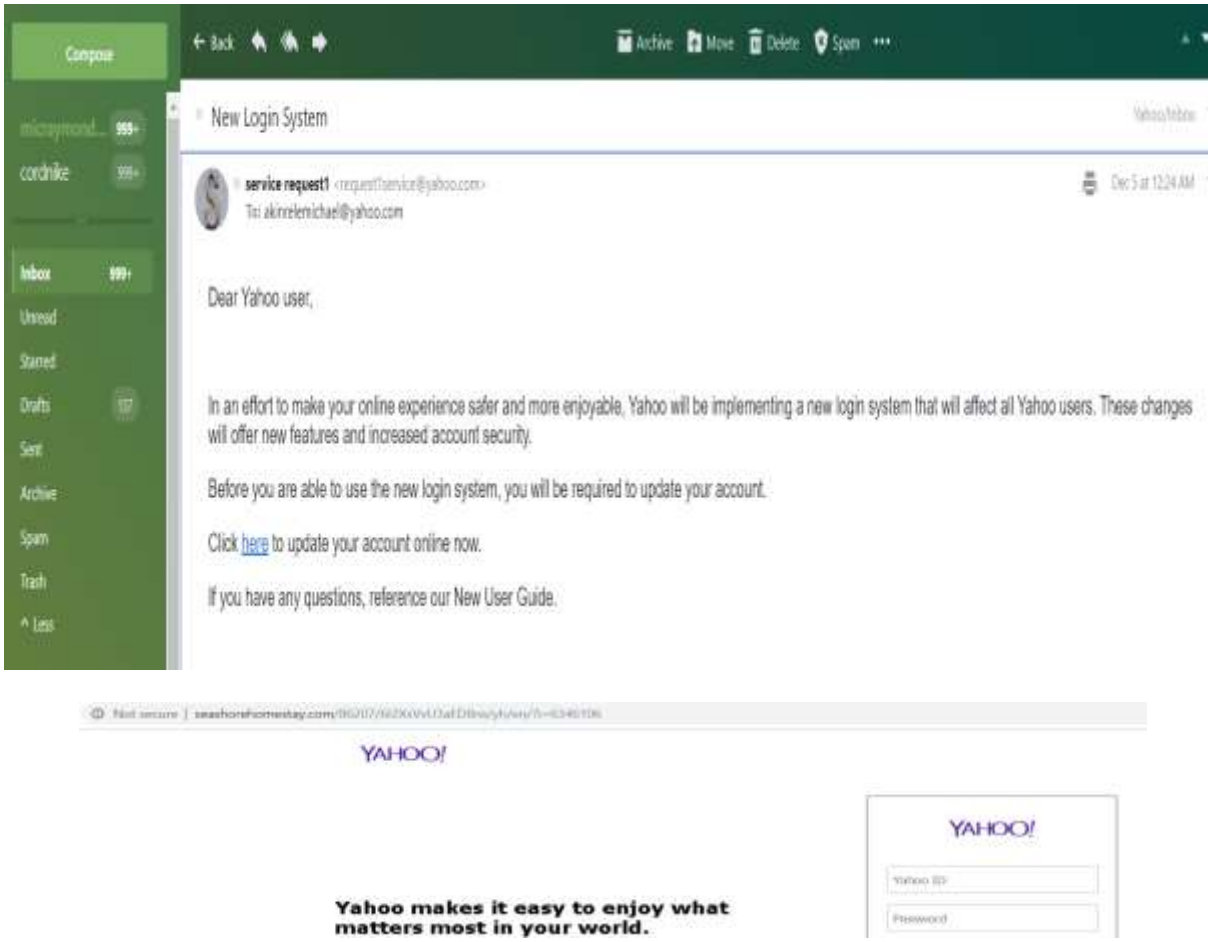

Figure 3: Example of phishing email using Z-Shadow

2.2 Related Work

Many researcher and computer scientist work effectively to address this dangerous and malicious act of emailing. Because spammers regularly find means of circumventing spam filters and spreading spam messages, researchers should remain as a visionary in this topic so that spam notifications can be minimized. This section discusses existing work on the identification and recognition of email phishing and spamming strategies and techniques addressed as follows.

2.2.1 Phishing detection

2.2.1.1 Toolbars

The initial attempts to detect phishing attacks took the form of browser toolbars, for example, the Spoofguard [1] and Netcraft[2] toolbars. Many toolbars, as mentioned in [2], are pleased to have 85% accuracy detecting websites for phishing. Apart from the accuracy, the toolbars have pros and cons over email filtering. The first drawback is toolbar is a reduction in contextual information in contrast to email filtering. The e-mail gives the explanation for the attack to the user. An email filter can see which words the user uses to act, which is currently unknown to a filter running in a browser other than the user's email client. An email-filter is also open to the headers, which include data not only about who sent the text but also about the way to access the recipient of the message. This background is not visible in the browser with certain implementations of the toolbar.

[2] Netcraft Ltd. Netcraft toolbar, 2006. http://toolbar.netcraft.com/.

The second downside for toolbars is the failure to protect the user from judgments. Interface toolbars usually request a dialog that is ignored and misinterpreted or, even worse, interface-space malware may intercept such alert dialogs. To avoid the risk that these warning messages are ignored and withheld from the user by filtering out phishing emails before they can be seen by users.

2.2.1.2 Email Filtering

While the filtering of phishing attacks at email level does have clear advantages, there are currently not many specific methods for targeting phishing emails, in contrast to general spam. The closest related previous attempt by [3] in order to determine whether the authors use the structural characteristics of an e-mail. The traits are primarily linguistic and include items like email terms, the vocabulary's assets, subject line composition and the inclusion of 18 keywords. Other examples include the filter built into Thunderbird 1.5. The filter, however, is super simple and only searches for one or three things, namely IP-based URLs, inappropriate URLs, and the HTML form component. This filter is also relatively straightforward. The built-in Thunderbird blocker just alerts the consumer and does not stop storage or time costs from occurring.

We try to fill this void in email phishing filters by our deployment and analysis. However, our approach is generalized over and above filtering e-mails, and we note how it can be used and what changes are required in contrast to e-mails in the context of web filtering.

2.2.1.3 Machine Learning

Phishing emails processes rely on classification approaches that can be handled in several ways, such as extraction features machine-learning and clustering [4]. [5] presents a framework for the identification and use of a machine learning feature set designed to show the user focused deception of electronic communication in its most general form. A phishing e-mail filter is a system for the classification of new messages. It can evaluate messages for different identification or via a learning-based filter that analyzes a set of defined training data, etc. [6]. In the study conducted by [7] envisaged approach for identification and filtering phishing e-mails using Stochastic Learning-Based Weak Estimators in real-life environment. This research is implemented based on Naïve Bayes classification for filtering phishing emails that are unpredictable in nature. Two datasets were used: 1200 legitimate, harmless emails and 600 actual phishing emails. They contrasted their findings from the SLWE method with the Maximum Likelihood Estimator (MLE) in order to evaluate the feasibility of their proposal. However, the results seems to be fine with 81.2 % accuracy but with an enormous number of features, impacting system performance and unrestricted data training can consume large amounts of space are failing from the proposed method.

In order to enhance phishing emails detection reliability, the researchers [8] suggested a lexical URL review methodology. The application of LUA (Lexical URL Analysis) to the methodology has shown to be an efficient way to enhance the classification quality with almost all tested subsets of features by testing empirically for publicly available phishing or legitimate e-mail sets. The idea of running sub-sets with two feature sets is to prevent potential features from increasing the classifier's time and space complexity.
This also prevents the deterioration in reliability of the classification system. To test their proposal, they used the publicly available positive and phishing datasets. It includes 4,150 harmless and 4,116 phishing messages. Their suggested lexical URL review methodology was successful in increasing the quality of identification by discussing their collected data.

The researchers [9] proposed that the phishing predictions should be focused on a neural network model. To extract phishing website functionality, they used the AntiPhishing Working group and PhishTank. In order to train and evaluate the model, they used the extracted functions. Researchers also noticed that phishing networks had only been down for 2.25 days. Nevertheless, they did not submit standardized tests, so the validity of their proposed design is difficult to assess.

2.2.2 Spamming detection

Several researchers also experimented for textual and image data analysis for spam emails. [10] uses an innovative approach to Naïve Bayes (NB) algorithms and Particle Swarm Optimization (PSO) based mathematical intelligence for the analysis of email spam. The Naïve Bayes algorithm is used here to learn and classify spam and no spam email content. PSO is characterized by stochastic distribution and swarm behavior and is considered to optimize global NB approach parameters. For the experiment, the Ling spam dataset is examined, and the accuracy of the data is evaluated. Though the results yielded a good accuracy, but this theory failed to show a proper integration with PSO and mostly depended on NB approach.

[11] suggests Logistic Regression (LR) and Decision Tree (DT) hybrid blend for email spam identification. Spam base datasets are used in this study to evaluate the method proposed. The findings of the study revealed that 91.67 % of the experimental method results were excellent and positive. Nonetheless, the experiment didn't provide the analysis on DT training set as DT has an over-sensitive limitation for the dataset training and noise information or example that can decrease the performance.

[12] proposes the HTML email abstraction system, enabling the near-duplicate spam phenomenon to be more efficient. Also adds an adaptive data protection system that provides a comprehensive structure for secrecy expectations based on information for a specified account. But, not sure how this system can tackle for enterprise policies where they are predefined by the employer and will override the secrecy policies and open a question for user privacy management.

A study by [13] highlighted an automated sorting and screening strategy to spam and genuine mail. The unified alternative to the traditional approach has improved the reliability of the real-world data collection by more than 1% from 96.46% to 97.3%. Though this allows internet users essentially to prevent spam but didn't include computational processing speed and time.

2.2.3 Recent discoveries

Authors [14] attempt to integrate spam filter using the measurement of details and email classification system based on the context in order to improve the reliability of spam detection by 90%. Junk filters are used to first delete all spam emails from the mailbox of the proposed solution. Thus, emails may be classified into several folders in the context-based email classification model. Research has shown that the LingerIG spam filters are extremely efficient to isolate spam from a group of standard functioning e-mails. Though this research gave potential results but failed to give a comparative study on different approaches and relied only on LingerIG spam filter.

The flaws in the above approach are rightly judged by authors [15] The proposed system trains the algorithm and classifies emails by training from a previously classified datasets and then applies this to the identification and classification of incoming email. However, security concerns were not considered while executing.

An algorithm for classifying emails was introduced by [16] using Artificial Neural Network. Regarding training purposes, the model uses the backpropagation algorithm. Model factors were taken from the 1501 page Mill Rd., Palo Alto and 94304 records of Jaap Suermondt Hewlett-Packard Labs. The template was checked with 85.31% of the final output. This study showed the potential of the artificial neural network for classification of emails.

[17] focused on the user profile classification created by ontology in spam filtering based on ontology. Therefore, the mails can be sorted by user's personal interest and a box that contains only the mails needed may be given. And adopted machine learning techniques for the experiment also states that the results are empirical and need to test on real-time of environment. However, the fact is that web3.0 is started making its footprint in enterprise-level, this approach may be helpful for future research.

[18] underlines the various current approaches for effective spam detection deployment in several software fields. A strategy of e-mail spam detection using the 0.5 membership limit was proposed using Fuzzy C Means. With other machine-learning methods associated with it, this method can be extended further for improving the performance and security.

[19] uses a fixed collection of engineered devices, with functions removed automatically. With end-to-end authentication, the solution is just as successful as the feature set with authenticated e-mails remains unchanged. Some experiment have been actualized to use algorithm or ML to classify emails that belongs to phishing and email spam categories. One of the most paramount approach for the progress of any algorithm is the set features used to show instances [20]. Among the few features that have been proposed over the years to represent phishing and spam email instances, no paper has revealed a full study of the possible features and an interpretation of their capability utility in this work.

3 Research Methodology

3.1 Possible characteristics

This section explains all the features of this project. The features included are those which are relevant to the e-mails, than external. Some researchers have used features from alternative sources, like spam assassins, database registry data or search engine results. For a variety of reasons, in the search for the best insightful feature collection, we decided to bypass all these external features and inspired from the research [21]. The criteria are:

- Mail is the only bit of data access which is assured for all participants in the identification of spam and phishing.
- Some electronic information periodically varies, e.g. DNS data or search results.
- Blacklist solutions allow people/organizations to make it impossible for a fully automated spam/phishing detection system to be implemented.

  In this analysis, 40 features of emails are identified. After a review of the literature in the field, these features are determined. In many cases, writers seem to pick apps randomly before understanding how much they can benefit from using specific features. The characteristics we recognized were approximately segmented into five different categories. The following categories are:

- **Features based on email body:** Several features are explicitly taken from the e-mail body text, containing details such as the content type of the e-mail.
- **Features based on URL:** The anchor tags in the HTML e-mails extract this feature.
- **Features based on Subject:** These features are taken derived from email subject line.
- **Features based on Script:** Such characteristics are due to the inclusion and lack of scripts in the message and the impact of patterns on the usability.
- **Features based on Sender:** Those attributes were stripped out from the e-mail address of the recipient.

The following are the Features based on email body:

- html content in email body *(body html)*: The HTML presence of the email corpus is a binary feature. [22] previously used these the html body feature.
- forms in email *body forms*: This is a numeric characteristic that shows HTML in the e-mail system. This binary function shows forms in HTML e-mail organizations. Historically, [22] used the body form feature.
- Number of words in email body *body noWords*: The maximum number of words on the email calculations this feature [23], body noWords were used.

- Number of characters in email body *body noCharacters*: The total number of characters in the email body is calculated. This was used in [23]

- Number of distinct words in email body *body noDistinctWords*: This feature was used by [23], it calculates the maximum number of distinct words in the email body.

- Richness of the email body*body richness*: The richness is defined as the proportion of words to character numbers in the text. Mathematically, this is expressed in equation 1. This feature is adapted from [20]

$$\text{body\_richness} = \frac{\text{body noWords}}{\text{body noCharacters}} \quad (1)$$

- Number of Function Words in email *body noFunctionWords*: The following function words included: account; access; banking; credit; click, inconvenience; information; limited; log; minutes; password; recently; risk. [24] also listed the words: social security and security; service; limited. The body noFunctionWords measures the total number of redundancies in the email body of these function words.

- Count of word suspension in email body *body suspension* This quantitative function shows the suspension of the expression in the message.

- Count of word verifyYourAccount *body verifyYourAccount*: This binary characteristic describes the inclusion of the sentence in the email that verifies your account. This is used in [20].

4 Design Specification

We see this problem as a classification task as we tackle spam filtering from the Machine Learning viewpoint of the system. This is to determine whether an e-mail is spam or ham or phishing, depending on its features. In this case, a variety of features were identified from section 2 and consolidated in the previous section 3 in the e-mail is the functionality.

A machine learning system usually works in two ways: training and testing

• Training

The machine learning system receives labeled data from a training dataset during training. The labeled training data in this project include a wide range of spam-labeled or ham-labeled or phishing-labeled e-mails. Throughout the training process, a machine learning classifier determines labels for potential e-mails by defining the links between an e-mail and its tag.

• Testing

The machine learning program is supplied with unlabeled information during testing. In this project, such details were spam/ham/phishing e-mails.

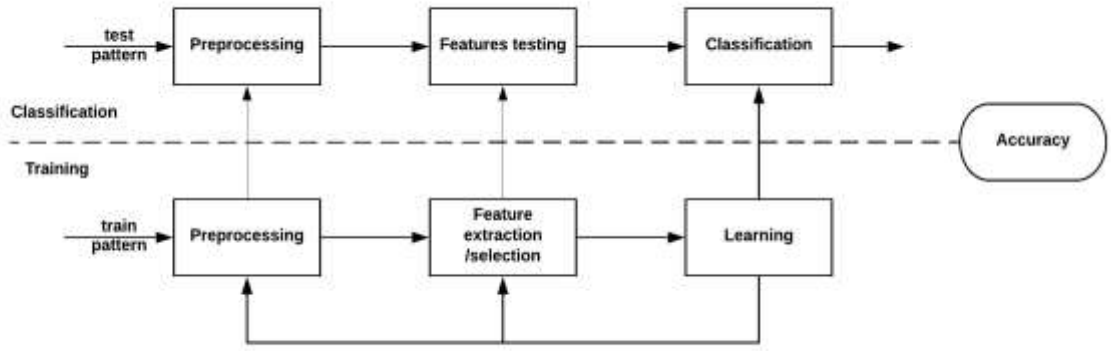

Figure 4: Design flow

Dataset	Number of emails
Spam	501
Ham	2551
Phishing 2017	303
Phishing 2018	490

Table 1: Dataset facts

5 Implementation

5.1 Datasets

We used combinations of three open datasets to detect spam, ham and phishing. One is the set of emails from the Ham (genuine) Spam Assassin Project[3]. The other is a spam email dataset from the same source. The third was phishing e-mails of 2017 and 2018 collected from open source by Jose[4]. Table 1 outlines the facts for these datasets, with the use of these datasets a cumulative dataset is formed which would be used for analysis and prediction of the illegitimate emails hitting the users mailbox. Several data pre-processing techniques were used for best fit and better accuracy.

5.2 Data preprocessing

The pre-processing of data can have very a strong influence on a Machine Learning algorithm especially when working with raw data. There are several techniques available to convert raw data into insightful data. This section gives an overview of the steps taken for data preprocessing.

Figure 5 shows the technical flow of the proposed approach to identify Spam/Ham/Phishing emails.

5.2.1 Prerequisites

In this project we used the open-source Anaconda Distribution which is simple to perform Python machine learning tasks.

- Programming language: Python 3.7
- Libraries: sci-kit learn, NumPy, Pandas

[3] https://spamassassin.apache.org/old/publiccorpus/
[4] https://monkey.org/ jose/phishing/

- Operating system: Windows 10

Python libraries: As depicted in technical flow figure 5, Pandas and NumPy were used to carry over the data preprocessing tasks due to their user-friendly approach and best fit results.

```
import numpy as np # used for handling numbers import pandas as pd # used for handling the dataset from sklearn.model selection import train test split
#used for splitting training and testing data
```

5.2.2 Process feature extraction

As analyzed in section 3.1 all the features are extracted from the raw data corpus and extracted into a structured csv file. This csv file holds the details of Ham, Spam, Phishing emails with a label indicating its category. In total there was 3845 rows were created from the data corpus.

5.2.3 Pandas Dataframe

Pandas dataframe[5] is helpful for data manipulations. Dataframes from Pandas is a two-dimensional, possibly interdependent tabular data structure with axes (rows and columns) labelled. The extracted csv file is fed on pandas dataframe for pre-processing and building machine learning classifier models.

```
data frame = pd.read csv('SpamHamPhishing.csv')
```

5.2.4 Handling of Missing Data

The first idea is to remove lines where certain data are missing. But that can be very risky, as this dataset includes important information. Removal of an observation would be quite risky. In this project, all numerical features such as *body richness, subj richness, url noLinks, url noExtLinks, url noDomains, body noCharacters, body html, url maxNoPeriods, body noWords* were verified for 'nan' values and replaced with a minimum value but fortunately this dataset didn't had any null values.

```
# handling the missing data and replace missing values with
#nan from numpy and replace with mean of a l l the other values imputer = SimpleImputer(missing values=np.nan,
strategy='mean') imputer = imputer.fit(X[:, 1:])
X[:, 1:] = imputer.transform(X[:, 1:])
```

5.2.5 k-Fold Cross-Validation

Any machine learning algorithm needs to be tested for accuracy. Cross-validation is the re-evaluation of machine learning models for a limited dataset. A procedure is called k which refers to the number of groups to be divided into in a provided data sample. As such, k-fold cross-validation is often referred to. When k is specified for a value, k in the model relation, for example, k=10 can be used to cross-validate 10 times.

[5] https://pandas.pydata.org/

```
kfold = model_selection.KFold(n_splits = 10)
```

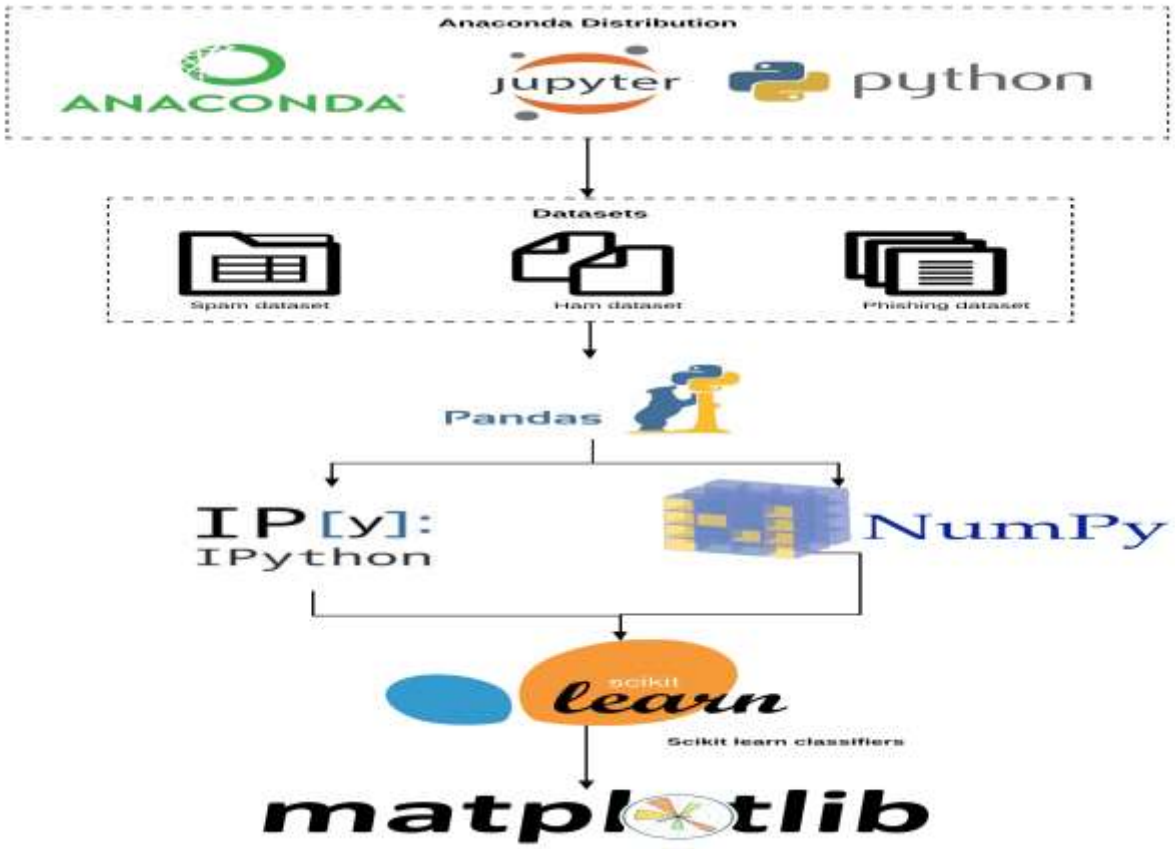

Figure 5: Technical flow

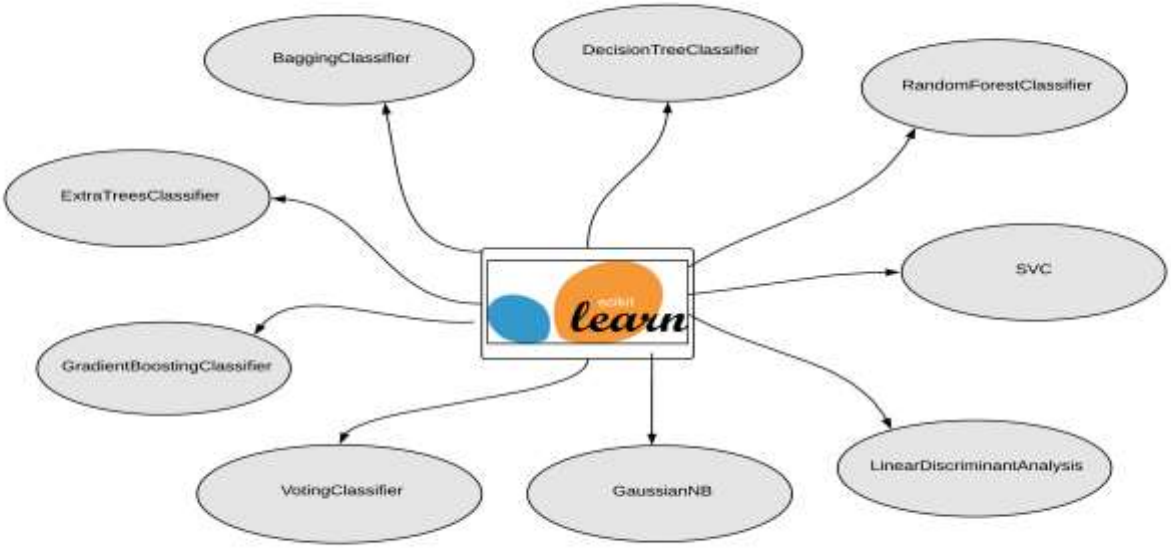

Figure 6: Applied Sci-kit learn classifiers

6 Evaluation

To determine the quality of the assumed features and approach we have trained machine learning classifiers[6] to extract the effectiveness of the proposed system. We focus on the discussion of mutual-information-based feature selection. The two random parameters of a and b describe some mutual information according to the functions $p(a)$, $p(b)$, and $p(a,b)$ of their probabilistic frequency.

$$I(a;b) = \iint p(a,b) \log \frac{p(a,b)}{p(a)p(b)} dadb \tag{2}$$

Throughout this scenario, it is easy to quantify mutual data, since the measurements of categorical variables can be calculated by counting all joint and marginal probability tables. If at least one of the variables a and b is constant, though, their $I(a;b)$ reciprocal information is difficult to measure, since it is always difficult to calculate the integral on the basis of a limited sample size in a continuous space. Considering N samples of a variable a, $\hat{p}(a)$ has the following density function:

$$\hat{p}(a) = \frac{1}{N} \sum_{j=1}^{N} \delta \left(a - a^{(j)}, h\right) \tag{3}$$

where $\delta(.)$ is the Parzen window function in which $x^{(i)}$ is the ith sample, and h is the window width. No certain classifiers are included in our mRMR selection method. We therefore assume that the features chosen by this system should work well in various classifying forms. We consider two common classifiers in order to test this: Naive Bayes (NB), Support Vector Machines (SVM).

Given a sample $s = \{x_1, x_2,...,x_m\}$ for 'm' features, the posterior probability that s belongs to class c_k is

$$p\left(c_k|s\right) \propto \prod_{i=1}^{m} p\left(x_i|c_k\right) \tag{4}$$

Table 2: mRMR Algorithms performance

Algorithm	**Accuracy**
Navie Bayes	72.79
SVM	67.81

where $p(x_i|c_k)$ is the conditional probability table. SVM is a more modern classification tool that uses kernels for the development of linear classification limits in higher spaces.

Applying k-fold as 10 splits for mRMR algorithms and calculating mean using score[7]function as derived in equation 5 yielded the accuracy as tabulated in 2 and compared in figure 7.

$$Mean\ \bar{x} = \frac{\Sigma x}{n} \tag{5}$$

where ^{P}X is sum of all data values and n is number of data items in sample

[6] https://scikit-learn.org/stable/modules/cross*validation.html*

[7] https://scikit-learn.org/stable/modules/cross*validation.html*

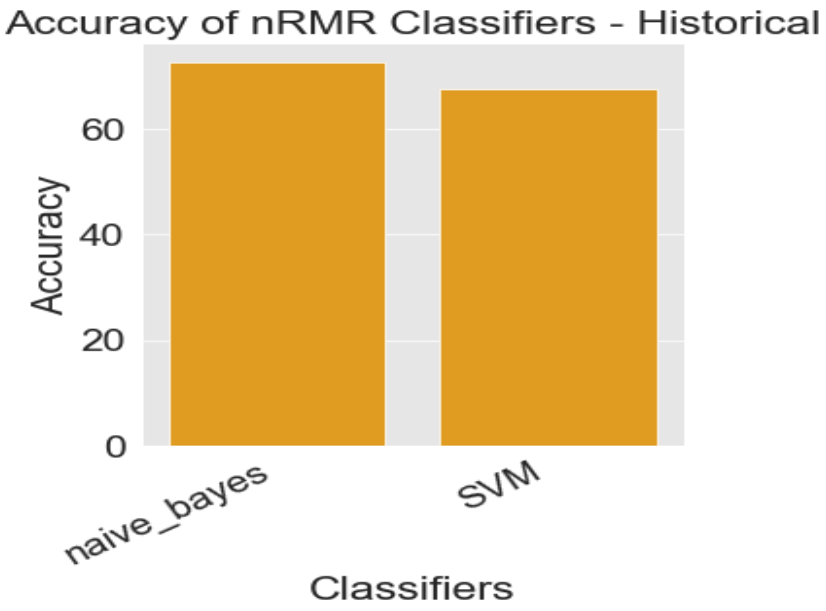

Figure 7: mRMR Algorithms performance

The nRMR approach results didn't show promising and it was decided to apply ensemble algorithms and test the quality of proposed system.

Table 3: Ensemble Algorithms performance without any filter

Algorithm	**Accuracy**	**Time build(s)**
Bagged Decision Tree	89.23	5.17
Random Forest	89.34	1.02
Extra Trees	90.53	0.99
Adaboost	83.90	2.93
Stochastic Gradient Boosting	82.55	3.10
Voting Ensemble	84.55	-

Ensemble learning: Ensemble learning contributes by merging various models to enhance machine learning outcomes. This strategy enables better predictive performance in comparison to a single model. Ensemble techniques consist of meta-algorithms merged into a single predictive model by integrating several machine-learning techniques to reduce variance (bagging), bias, or increase predictions.

$$f(x) = 1/M \sum_{m=1}^{M} f_m(x \qquad) \tag{6}$$

Bagging: Bagging implies the accumulation of bootstrap. The aggregation of several evaluations is one means of reducing the uncertainty of an estimation. Bagging uses bootstrap samples to obtain data subsets. It takes the vote for classification and the mean for regression to aggregate the outputs. In contrast with the k-NN Bagging ensemble, the decision tree bagging ensemble obtained greater precision. Each tree in the ensemble is constructed in random forests from a sample taken from the training set and substituted (i.e. a bootstrap sample).

Boosting: The core idea of boosting is to adapt a sequence of weak learners to weighted versions of the data – models that are only a little better than a random devaluation, for example, small decision trees. Examples

unclassified in earlier rounds are given greater weight. The predictions are then merged to deliver the final prediction through an outright majority (classification) vote.

Gradient Tree Boosting of the Gradient Tree is an improvement in conditional failure functions. And can be used for questions of regression and grouping. It sequentially builds the model

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (7)$$

Voting Classifier Voting Ensembles is used for finding the average of the predictions for any arbitrary models.

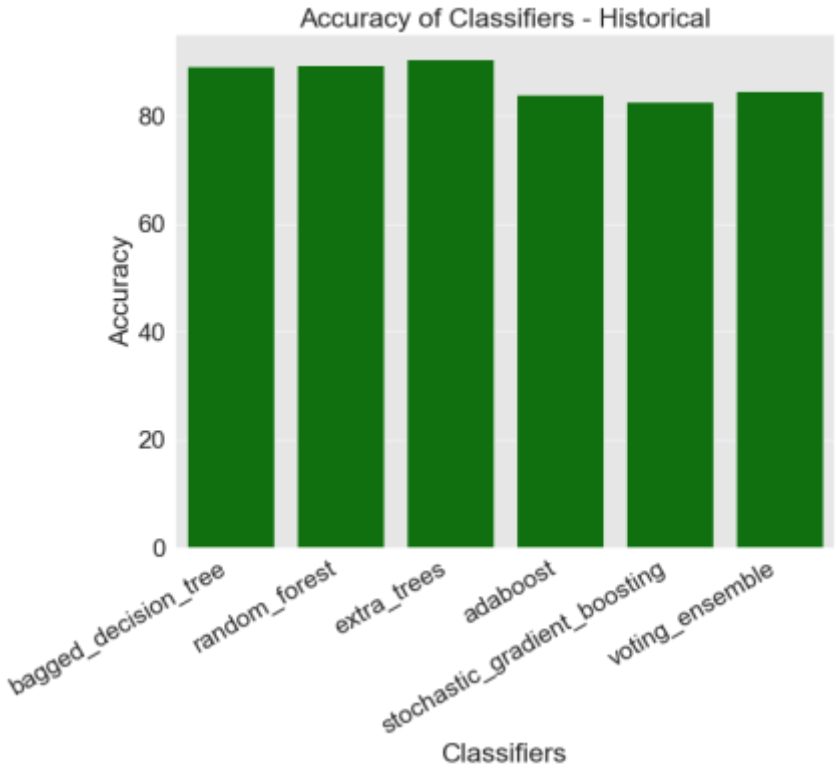

Figure 8: Ensemble Algorithms performance without any filter

Table 4: Ensemble Algorithms performance with low variance filter

Algorithm	Accuracy	Time build(s)
Bagged Decision Tree	85.82	1.78
Random Forest	86.45	0.97
Extra Trees	86.99	0.78
Adaboost	81.31	1.86

Stochastic Gradient Boosting	80.62	1.88
Voting Ensemble	84.13	-

Table 5: Ensemble Algorithms performance with low correlation filter

Algorithm	**Accuracy**	**Time build(s)**
Bagged Decision Tree	89.36	3.86
Random Forest	88.45	0.90
Extra Trees	90.61	0.72
Adaboost	83.90	2.28
Stochastic Gradient Boosting	82.19	2.52
Voting Ensemble	84.89	-

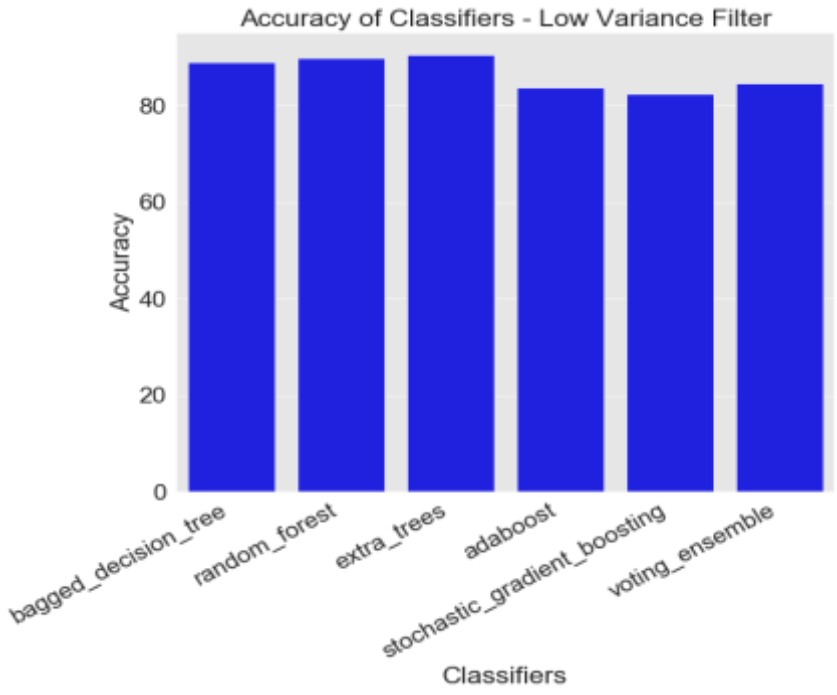

Figure 9: Ensemble Algorithms performance with low variance filter

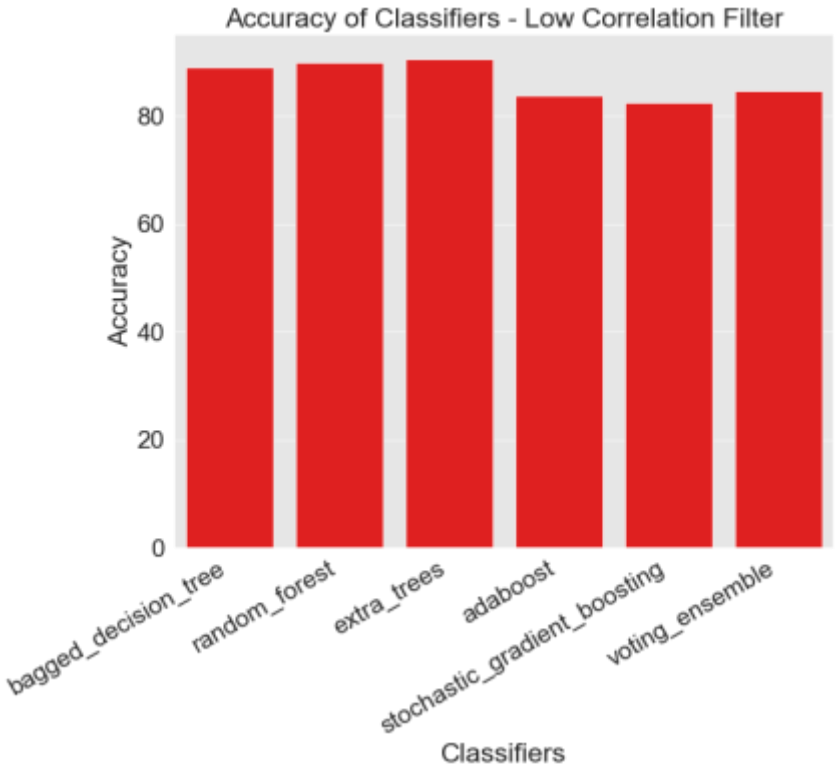

Figure 10: Ensemble Algorithms performance with low correlation filter

Table 6: Ensemble Algorithms performance with no importance filter

Algorithm	**Accuracy**	**Time build(s)**
Bagged Decision Tree	89.13	4.55
Random Forest	89.94	0.86
Extra Trees	90.61	0.68
Adaboost	83.90	2.32
Stochastic Gradient Boosting	82.55	2.66
Voting Ensemble	84.60	-

Figure 11: Ensemble Algorithms performance with no importance filter

Discussion

The experimental results from both mRMR and ensemble classifiers seems to be acceptable for detecting spam, ham and phishing emails. However, ensemble classifiers were more promising with the additional computational techniques. The overall average accuracy of 83% is good for the initial research whereas Bagged Decision Tree outperformed with the best accuracy of ~90%. This shows that the proposed approach would be a good fit for detecting future spam, ham and phishing emails with a scope for future research and expansion.

7 Conclusion and Future Work

It is crucial to indicate that phishing and email spam are detrimental, and its consequence can be faced for long period of time, which also can crumple the entire system. Few available tools are accessible to halt this, but the use of classification with combination of algorithms is one of the ultimate ways to uncover it. With the identified research question, we conclude that our model classify and analyze them as Ham, Phished and Spam email — successfully extracted features from the different public datasets, which split amongst three divisions: phishing, ham, and spam. We then properly ensembled and mRMR the information gain from the features. This approach further created ensembles classifiers and mRMR models using four groups of filters, those with the best accuracy and time builds were actualized. As predicted in each study, the classifier trained on the best filters exceeded all the others. Though mRMR algorithms showed slightly poor results when compared with ensemble methods. Future work can center on capturing real-time phishing and email spam datasets while also restraining algorithm to avoid words from the dictionary and recurrence to achieve a better accuracy.

References

[1] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, “Client-side defense against web-based identity theft,” 01 2004.

[2] R. Fatima, A. Yasin, L. Liu, and J. Wang, “How persuasive is a phishing email? a phishing game for phishing awareness,” *Journal of Computer Security*, no. Preprint, pp. 1–32, 2019.

[3] P. PRIYATHARSINI and C. Chandrasekar, “Classification techniques using spam filtering email,” *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, p. 402, 2018.

[4] I. R. A. Hamid and J. Abawajy, “Hybrid feature selection for phishing email detection,” in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2011, pp. 266–275.

[5] I. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 649–656.

[6] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, “A survey of phishing email filtering techniques,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 2070–2090, Fourth 2013.

[7] J. Zhan and L. Thomas, “Phishing detection using stochastic learning-based weak estimators,” in *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, April 2011, pp. 55–59.

[8] M. Khonji, Y. Iraqi, and A. Jones, “Enhancing phishing e-mail classifiers: A lexical url analysis approach,” *International Journal for Information Security Research (IJISR)*, vol. 2, no. 1/2, p. 40, 2012.

[9] A. Martin, N. Anutthamaa, M. Sathyavathy, M. M. S. Francois, D. V. P. Venkatesan *et al.*, “A framework for predicting phishing websites using neural networks,” *arXiv preprint arXiv:1109.1074*, 2011.

[10] K. Agarwal and T. Kumar, “Email spam detection using integrated approach of na¨ıve bayes and particle swarm optimization,” in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, June 2018, pp. 685–690.

[11] A. Wijaya and A. Bisri, “Hybrid decision tree and logistic regression classifier for email spam detection,” in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct 2016, pp. 1–4.

[12] P. Rajendran, M. Janaki, S. M. Hemalatha, and B. Durkananthini, “Adaptive privacy policy prediction for email spam filtering,” in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, Feb 2016, pp. 1–4.

[13] A. Iyengar, G. Kalpana, S. Kalyankumar, and S. GunaNandhini, “Integrated spam detection for multilingual emails,” in *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, Feb 2017, pp. 1–4.

[14] M. K. Chae, A. Alsadoon, P. W. C. Prasad, and A. Elchouemi, “Spam filtering email classification (sfecm) using gain and graph mining algorithm,” in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2017, pp. 1–7.

[15] A. A. Alurkar, S. B. Ranade, S. V. Joshi, S. S. Ranade, P. A. Sonewar, P. N. Mahalle, and A. V. Deshpande, “A proposed data science approach for email spam classification using machine learning techniques,” in *2017 Internet of Things Business Models, Users, and Networks*, Nov 2017, pp. 1–5.

[16] A. Alghoul, S. Al Ajrami, G. Al Jarousha, G. Harb, and S. S. Abu-Naser, “Email classification using artificial neural network,” 2018.

[17] N. M. Shajideen and V. Bindu, “Conventional and ontology-based spam filtering,” in *2018 International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR)*, July 2018, pp. 1–3.

[18] A. K. Singh, S. Bhushan, and S. Vij, “Filtering spam messages and mails using fuzzy c means algorithm,” in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, April 2019, pp. 1–5.

[19] T. Krause, R. Uetz, and T. Kretschmann, “Recognizing email spam from meta data only,” in *2019 IEEE Conference on Communications and Network Security (CNS)*, June 2019, pp. 178–186.

[20] Ammar Almomani, B. B. Gupta, Samer Atawneh, A. Meulenberg, Eman Almomani, "A Survey of Phishing Email Filtering Techniques", Communications Surveys & Tutorials IEEE, vol. 15, no. 4, pp. 2070-2090, 2013.

[21] F. Toolan and J. Carthy, “Feature selection for spam and phishing detection,” in *2010 eCrime Researchers Summit*, Oct 2010, pp. 1–12.

[22] I. Qabajeh and F. Thabtah, “An experimental study for assessing email classification attributes using feature selection methods,” in *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*. IEEE, 2014, pp. 125– 132.

[23] J. Clark, I. Koprinska, and J. Poon, “A neural network-based approach to automated e-mail classification,” in *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*. IEEE, 2003, pp. 702–705.

[24] M. M. Al-Daeef, N. Basir, and M. M. Saudi, “A method to measure the efficiency of phishing emails detection features,” in *2014 International Conference on Information Science & Applications (ICISA)*. IEEE, 2014, pp. 1–5.

[25] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, “Phishing email detection based on structural properties.”

Appendix

Table-A: Feature list taken from [21]

Attribute	Data type	Description
body_forms	Number	The presence of forms in HTML email bodies.
body_html	Number	The presence of HTML in the email body
body_noCharacters	Number	The total number of characters occurring in the email body
body_noDistinctWords	Number	The total number of distinct words occuring in the body of the email.
body_noFunctionWords	Number	The total number of occurrences of these function words in the email body such as access; bank; credit; etc
body_noWords	Number	The total number of words occuring in the email.
body_richness	Number	The richness is defined as the ratio of the number of words to the number of characters in the document.
body_suspension	Number	A binary feature with word *suspension* in the body of the email
body_verifyYourAccount	Number	The phrase *verify your account* in the body of the email.
script_javaScript	Number	The presence of javascript in the email body
script_nonModalJsLoads	Number	The presence of external javascript forms that come from domains other than the modal domain.
script_onClickEvents	Number	Counts the number of onClick events in the email
script_popups	Number	Binary feature if the email contains pop-up window code
script_scripts	Number	The presence of scripts in the email body.
script_statusChange	Number	A binary feature that is true if the script attempts to overwrite the status bar in the email client.
send_diffSenderreplyTo	Number	Is a difference between the sender's domain and the reply-to domain.
send_noCharacters	Number	The total number of characters in the sender field.
send_nonModalSenderDomain	Number	The sender's domain is different from the email's modal domain
send_noWords	Number	The total number of words in the send field.
subj_bank	Number	0.07449
subj_debit	Number	0.00269
subj_forward	Number	The email is forwarded from another account to the recipient.
subj_noCharacters	Number	The total number of characters in the email's subject line
subj_noWords	Number	The total number of words in the subject line of the email
subj_reply	Number	The email is a reply to a previous email from the sender.
subj_richness	Number	The richness is defined as the ratio of the number of words to the number of characters in the subject.
subj_verify	Number	The email's subject line contains the word *verify*.
url_atSymbol	Number	The presence of links that contain an @ symbol.
url_ipAddress	Number	The use of IP addresses rather than a qualified domain name.
url_linkText	Number	Text like click; here; login; or update in email body
url_maxNoPeriods	Number	The link with the highest number of periods
url_noDomains	Number	The total number of domains in all URLs in the email.
url_noExtLinks	Number	The number of links whose target is outside the email body.
url_noImgLinks	Number	The number of links where the user needs to click on an image in the email body
url_noIntLinks	Number	The number of links whose target is internal to the email body
url_noIpAddresses	Number	The number of links in an email that contain IP addresses rather than fully qualified domain names
url_noLinks	Number	The number of links in the email body
url_nonModalHereLinks	Number	Captures here links that link to a domain other than the modal domain.
url_noPorts	Number	The number of links in the email that contain port information in the address.
url_ports	Number	URL accesses ports other than 80.