National College of Ireland

Title

MSc Research Project
Detecting Spear-phishing Attacks using Machine Learning

Yamah Hanson Shonibare
Student ID: 21106941

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Yamah Hanson Shonibare		
Student ID:	21106941		
Programme:	MSc Cybersecurity	**Year:**	2022
Module:	Research Project		
Supervisor:	Jawad Salahuddin		
Submission Due Date:	15th December, 2022		
Project Title:	Detecting Spear-phishing Attacks using Machine Learning		
Word Count:	6900	**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: Yamah Hanson Shonibare

Date: 15th December, 2022

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):	

Detecting Spear-phishing Attacks using Machine Learning

Yamah Hanson Shonibare
Student ID: x21106941

Abstract

The threat landscape has become larger as a result of the growing number of internet users and its features, particularly in the usage of email communication, and as a result of this, attacks have increased resulting in the loss of money, reputation, data, and emotional well-being of individuals and organizations. These threat actors use phishing as one of their tactics, especially spear-phishing, which has evolved into one of their most successful attack vectors due to its high success rate. The use of social engineering, which takes advantage of the victim's emotional and psychological state to disguise hostile emails as legitimate ones, has allowed this attack method to grow so complex that it is challenging to identify them and protect victims. This study examines spear-phishing detection using traditional and automated techniques, and a novel model was developed to identify spear-phishing emails using Random Forest algorithms, and Ensemble learning on a trained and tested dataset of 3000 emails consisting of 1500 normal emails and 1500 spear-phishing emails. This research also classified the rate of accuracy between both algorithms, from which the Random Forest (RF) algorithm performed the best in detecting spear-phishing emails with a 96.33% accuracy rate.

1. Introduction

The use of email for data exchange is widely utilized by many individuals and organizations due to the strong demand for information interchange and the growing number of email users. By 2025, Statista predicts that there will be 4.6 billion email users, up from the current 4 billion daily users, and over 306 billion emails per day are sent and received from that total. Due to the frequent interchange of documents and links via email within and between these environments, emails have become a common attack vector for attacking businesses and organizations because they seem to be ideal means of delivering harmful payloads to a victim. This can be accomplished by using the spear-phishing approach, in which attackers use carefully designed emails to directly target certain employees of a business. For instance, an attacker might choose a relevant subject, use acceptable language, and impersonate a well-known sender to persuade the recipient that the email is genuine. Since each targeted attack is specifically tailored to the surroundings and behavior of the victim, they are more sophisticated than standard phishing or spam attacks as a result of this, there aren't many spear-phishing attacks that are the same, which makes it difficult to defend against.

Although users are becoming more aware of the risk they are exposed to, they are still dependent on the email client's suggestions to identify fake information. Several email applications, like Mozilla Thunderbird and Microsoft Outlook, only show the From and Reply-To fields by default, which provides only a limited amount of information for locating the sender. Even an experienced user may find it challenging to distinguish between well-designed attacks and valid content when dealing with emails from unknown senders because these and other fields can be falsified. Emails from unknown senders can be properly identified and treated individually. If the attacker perfectly adapts all fields, making the email's text and

headers appear entirely genuine, it becomes challenging. Inconsistent combinations of these fields can be quickly discovered and utilized to warn the user of a potential threat.

The sender of an email can be verified in this circumstance with the use of popular anti-spoofing tools like the Sender Policy Framework [SPF], DomainKeys Identified Mail [DKIM], and Domain Message Authentication Reporting & Conformance [DMARC]. Similar to this, methods for digitally signing emails like PGP and S/MIME allow the sender to be verified. Sadly, these methods are still not frequently used in daily life.

The aim of this report is to answer this question “How effective do Random Forrest Algorithm, and Ensemble Learning compare against one another in the detection of spear phishing attacks?"

This study's objective is to:

- Implement a machine learning algorithm that is able to detect spear phishing.
- Select the most effective classification algorithm between random Forest and Ensemble Learning for detecting spear phishing.
- Assess the outcomes of the implementation.

The need for alternative methods to protect users from highly targeted spear-phishing emails results from this. In this research, we present a method to check the subject and content of an email and whether it contains trigger words that could mean that the email is from an illegitimate user. Our approach is based on the discovery that a sender often leaves characteristics in the content and subject of an email that is often overlooked. These characteristics, which include specific header combinations, and encoding formats reflect user behavior, email client quirks, and delivery channels and show a significant difference between each sender. Based on this finding, we created a detection model that takes a user's mail as input and uses machine learning techniques to create profiles for every sender in the mailbox, These profiles give us a view of the header and content and help us recognize spoof emails as deviations from the learned profiles.

A set of 3000 emails from a total of more than 500,000 emails from Enron datasets was analyzed using the Random Forest Algorithm and Ensemble Learning models and showed that our models can analyze these emails in the dataset and identify spear-phishing emails with a detection accuracy of 96.33% when the random forest algorithm was used and 94.38% when ensemble learning was used with particular reference to the proposed feature extraction to remove irrelevant features from spear-phishing e-mail data that was developed.

This research work is organized as follows: Section 2 examined the Literature Review that has been conducted within similar research areas to get the perspective and opinions of other authors. Section 3 covers the Research Methodology which would provide a detailed analysis of the tools, techniques, and research work. Design specification and the proposed architecture of this research work were covered in Section 4 while Section 5 describes in detail the proposed Random Forest and Ensemble Learning algorithm. Section 6 examined the model's performance using the evaluation parameters that we selected while developing the model. The paper is concluded in Section 7 with findings from the research carried out and suggestions for future work.

2.0. Literature Review

2.1. Background on Spear-phishing Attacks

Using sophisticated social engineering techniques, spear-phishing is a focused type of phishing that often involves an email attack. It aims to persuade users to reveal important account, personal, or company information or to allow access to the computing infrastructure (Goel et al., 2017; Sjouwerman, 2015). Spear-phishing is challenging to spot because it employs a targeted strategy to persuade users to let their guard down and act on emails by invoking feelings of urgency. Spear-phishing attacks have become harder to spot as technological advancement has made remote communication more widely available which has made spear-phishing a conduit for other online crimes like ransomware and identity theft, which collectively result in billions of dollars in losses.

Majority of the time threat actors rely on spear-phishing since it is much more specifically targeted and allows for the theft of credentials, the use of ransomware, and other means of obtaining money. According to several reports, spear-phishing is quickly gaining ground on traditional phishing in terms of popularity such that a sophisticated spear-phishing attempt was encountered by 88% of enterprises and 64% of security professionals according to reports from Proofpoint. Account breaches, malware such as ransomware, and data theft were many of the objectives of these attacks. Spear-phishing reports have dramatically grown after 2020. 74% of American firms reportedly suffered a successful phishing attack, according to the Data Breach Investigation Report (DBIR) version 2021 of which the use of email is the most popular spear-phishing delivery method, which has accounted for 96% of all attempts. According to the Symantec Internet Security Report 2019, approximately two-thirds (65%) of all known groups that conduct targeted cyberattacks utilize spear-phishing emails as their primary attack vector, and 96% of targeted attacks, according to the research, are carried out to gather intelligence.

2.2. Traditional Phishing Mail Detection

Three basic categories make up the current traditional phishing detection technology namely blacklist and whitelist-based detection, heuristic rule-based detection, and machine learning-assisted detection. For URLs, IP addresses, or keywords that have been detected as phishing sites, the black-and-white list detection technique often creates a list of blacklists from which people may correctly recognize phishing websites.

Although this method is straightforward and practical, it has a short renewal time and is prone to leaking. For this flaw, (Prakash et al. 2010) presented a better technique dubbed PhishNet. They presented five heuristics to identify new phishing URLs by listing simple combinations of well-known phishing sites, and their approximation matching method splits a URL into several parts that are then checked against entries in the blacklist individually. Heuristic rule-based detection can fix the black-and-white list method's flaw by creating heuristic rules that take into account how similar phishing sites are to one another. However, this system struggles with a high rate of false alarms and challenging rule updates when dealing with massive data. A classification method is then provided for detection after machine learning-based detection treats the content of phishing sites and emails as text. Currently, it is widely utilized in the majority of literature to identify phishing emails and can successfully identify phishing mail by adding or removing the attributes collected from the message. Using a variety of classifiers for training and testing, (Fette et al. 2006) suggested a phishing detection

approach based on 10-dimensional characteristics, and subsequently, the number of characteristics was increased to 47 by (Khonji M et al. 2011).

2.3. Spear-Phishing Detection

The identification of phishing emails has been the subject of extensive study over the years, most of which used machine learning, deep learning, or natural language processing techniques to address the issue. In a study, to identify targeted spear-phishing emails, a unique model that combines stylometric information from emails and social features from the online social network was developed. (Dewan et al. 2014) used 10-fold cross-validation to test 27 features, including 18 stylometrics and nine social variables, to demonstrate their claim. The outcome of the experiment demonstrates that without utilizing social variables like LinkedIn, the model had a better accuracy rating of 98.28%. (Han 2016) concentrated on spear-phishing campaign identification and developed a semi-supervised learning algorithm based on affinity graphs for campaign attribution and detection. (Grant Ho 2017) demonstrated a method for identifying credential spear-phishing attempts in corporate settings; in over 370 million real-world emails, they only missed two spear-phishing emails with less than a 0.005% false positive rate.

(Duman et al. 2016) suggest an intriguing methodology called EmailProfiler for identifying spear-phishing using the sender metadata. This model combines two processes: the first involved evaluating incoming emails using recipient-trained profiles, and the second involved building profiles at the sender and making them accessible for querying at a reliable server. This technique gathered 222 features from the emails' body, header, and sending time in order to create the profiles. The outcome demonstrates that the method was examined with accuracy rates ranging from 67% to 100%.

Another strategy, IdentityMailer, was put up by (Stringhini et al. 2015) It involves creating a user profile for their email-sending habits. There were three different kinds of features used: interaction habits, composition habits, and writing habits. To determine if the sender is real or phony, the retrieved features from the emails are compared with a behavioral profile. If the sender is discovered to be a fraud, the verification is terminated; if it is a real sender, an identity check is then conducted using a more sophisticated technique or by responding to a security question.

(Han et al. 2016) provided an affinity graph-based semi-supervised learning strategy for identifying spear-phishing emails that includes email profiling features such as origin features, text features, attachment features, and recipient features. (Stembert et al. 2015) proposed a prototype strategy that identifies spear-phishing attacks utilizing a combination of alerts, blocking, informational messages, and reporting. Along with maintaining immunological memory cells (IMCs) to quickly identify subsequent attacks from the earliest known or suspected phishing sources, this model also updated the user's knowledge for this phase. This strategy is created and used in three mockups, including a reporting button, email filtering and warnings, and educational recommendations. This model has two different kinds of sensors: one that was user-produced and the other that was generated by an intrusion-detection system (IDS).

A novel model called Anti-Spear-phishing Content-based Authorship Identification (ASCAI) was put forth by (Khonji et al. 2011) to reduce spear-phishing attacks using the document authorship method. To identify the authorship of the senders, a profile of regular

users is created without relying on the sender's user IDs, and the write-print of the most recent message is calculated using Jaccard's similarity index which yielded an accuracy rating of 87%.

2.4. Machine Learning Algorithms

Random Forest (RF) Classifier uses numerous decision trees to produce predictions. It functions by applying a number of decision tree classifiers to various dataset subsamples. The best qualities were also randomly chosen for each tree in the forest before being combined to create each one. Decision trees are produced during the training phase and they are used for class prediction. They are obtained by taking into account the classes that received the most votes for each unique tree, with the class that receives the most votes being regarded as the output. (Shapire et al. 1998) and (Breiman 1996) introduced Boosting and Bagging classification trees RF techniques. Boosting entails assigning additional weight to points that earlier forecasters incorrectly predicted; frequently, a weighted vote is finally taken for the projection. Bagging eliminates this dependency by building each tree independently using a bootstrap sample of the data set. The forecast is ultimately established by a simple majority vote. It is also very user-friendly because it just has two parameters—the number of variables in each node's random subset and the number of trees in the forest—and is typically not overly sensitive to their values. It is a fast algorithm compared to other supervised learning algorithms since it requires less training time and is resistant to noise and outliers. For high-dimensional data categorization, provides great scalability and parallelism, supports big datasets, prevents over-fitting, and produces high accuracy.

Ensemble Learning makes predictions based on characteristics extracted by a variety of projections on data, an ensemble learning algorithm combines findings with different voting processes. This results in performances that are superior to those produced from any individual component algorithm alone. The goal of ensemble learning is to seamlessly include various machine learning algorithms into a unified framework, effectively utilizing the complementary knowledge of each algorithm to enhance the performance of the entire model. This perspective claims that ensemble learning can be used in conjunction with a number of machine learning models for a variety of tasks, such as common classification tasks, clustering tasks, and other similar activities. The earliest ensemble learning research may be found from the previous century. To improve the effectiveness of identification systems, Dasarathy and Sheela (1999) proposed using component classifiers learned from many categories to construct a composite classification system. The link between the weak and strong learning algorithms in the PCA learning model was the same problem that (Kearns, 1995) looked at. Following that, Schapire and Robert (1990) investigated if it was possible to combine several weak learning models into a high-precision model. Some works conducted analyses on the characteristics of features included in the original data from various perspectives at the feature level.

3.0. Research Methodology

In this study, we propose a novel approach for using machine learning algorithms to analyze a dataset of emails to automatically identify spear-phishing and to classify the most efficient between two models. Consequently, the method by which this study will be implemented will be the main topic of this section. Before the commencement of any project, a thorough outline of the architecture, technique, and procedures required must be created. This detailed outline is often referred to as the research methodology. To achieve the objectives of this research, the SEMMA research methodology was adopted detailing all the steps involved

in this research. The steps in this methodology are Sample, Explore, Modify, Model, and Assess, from which the acronym was derived. The SEMMA methodology is suitable for the project as it aims to compare the performance of the random forest classifier against the ensemble learning classifier in the prediction of spear-phishing emails. The figure below shows the flow of the SEMMA methodology adopted for this research. Following this is an explanation of each step in the methodology as they relate to this research.

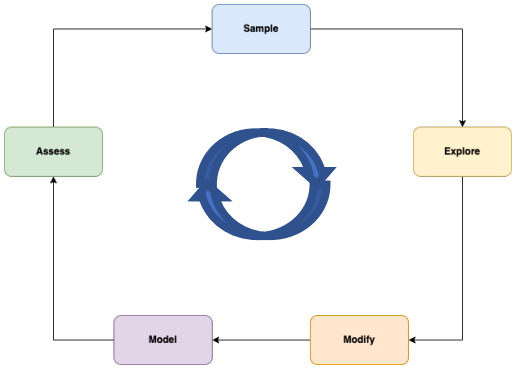

Figure 1: SEMMA Methodology

3.1. Data Collection & Dataset Description

The dataset utilized in this study was put together and created by the CALO (Cognitive Assistant that Learns and Organizes) Project which contains emails that the Federal Energy Regulatory Commission (FERC) initially made publicly accessible online as part of an investigation. It contains over 500,000 records of folder-organized data from 150 users, mostly from senior executives of the Enron Corporation. For this project, attachments are not included in the dataset, and certain communications were deleted as part of an effort at redaction at the request of the impacted employees. Invalid email addresses were corrected, whenever possible, to something like user@enron.com.

3.2. Exploration

A dataset can be better understood through exploration, which also makes it simpler to manage and utilize the data in the future. A researcher's analysis will be better if they are more knowledgeable about the data they are using (Idreos et al., 2015). The exploration phase of this research involves checking the datatype of each column, checking the number of null rows in each column as well as the number of normal emails against the number of spear-phishing emails present in the dataset.

3.3. Data Information

Prior to analyzing the data, it is important to understand the information contained in the dataset to ascertain its suitability for modelling. In addition, the data information also reveals the number of non-null rows in the dataset, from which we can infer the number of null rows.

3.4. Count of Null Values

Null values/ missing values are defined as values or data which are not present for some columns in a dataset. Handling null values is pertinent prior to modelling as machine learning algorithms are unable to support data with missing values. (Kaiser, 2014) suggests different ways of dealing with missing values in a dataset. For columns that have values measured on a continuous scale such as (height, weight, or price), an acceptable technique would be to input the mean/ average of the column into rows with missing values. The author also proposed that

columns with missing values could be dropped from the dataset if more than 60% of the rows are null. The latter technique was adopted as the columns with missing values were strings- which could not be easily deduced from other values in the column, and more than 60% of those columns had missing values. Below is an image showing the columns in the dataset and the number of missing values present in each column.

3.5. Data Distribution

Upon exploring the distribution of the dataset, it was discovered that the dataset was highly imbalanced. Less than 1% of the dataset was classified as spear-phishing emails – with only about 1700 emails. All other emails were classified as normal emails. Building a model on an imbalanced dataset could introduce bias to the model even though it may achieve high accuracy during training as explained by (Lemaître et al., 2017). The bias introduced by an imbalanced dataset is often towards the majority class. This means when the model is tested or introduced to real-world data, it tends to make predictions in favor of the label with a larger sample size.

Following this insight, it was pertinent to apply a data balancing technique to the dataset.

Random Oversampling and Undersampling are examples of data balancing techniques used for data with insufficient size for modelling (Yen and Lee, 2006). The random oversampler creates augmented data based on the sample with a smaller size, thus, increasing it to the same size as the larger sample. However, the disadvantage of oversampling is that there is a possibility of introducing bias and overfitting into the model. The undersampler reduces the number of the majority class to match the minority class. While more ideal than the oversampler, there is a chance of losing some information from the majority class once it is undersampled.

A similar approach to the undersampler was employed in this research. This technique involves randomly selecting the "n" number of random rows from both the majority class and the minority class to get a balanced dataset, where "n" is defined by the researcher. For this research n was chosen as 1500, meaning the program randomly selected 1500 normal emails and 1500 spearphishing emails, thus resulting in a balanced dataset.

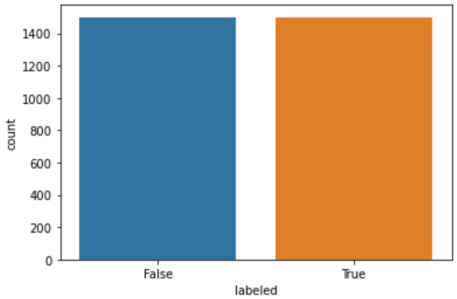

Figure 2: Balanced Dataset after Hybrid Undersampling

3.6. Modify

The data modification phase of the project involves the further preparation of the dataset for modelling. These steps typically involve transforming the original dataset into a machine-readable format (Bhagoji et al., 2018). The modification steps are explained below:

3.7. Data Encoding

Data encoding involves the modification of values into numerical values. Categorical variables are typically represented between 0 and n-1 where n is the number of categories

present in the column (Schuld et al., 2021). For example, the labelled column in the dataset has only 2 categories – normal and spear-phishing, consequently, the labelled column will be encoded to 0 and 1 for normal and spear-phishing respectively. All the columns in the dataset except the email subject and the email content were encoded to numerical values. The email subject and content were excluded from the encoding process as other modification steps would be applied to them.

	Message-ID	Date	From	Subject	X-From	X-Origin	X-FileName	content	user	labeled
0	1950	2354	363	Next NERC Reliability Meeting on Legislation -...	788	181	90	See NERC memo below that was just received. Th...	111	1
1	2584	2210	319	NaN	417	28	13	FYI. From today's Post. Davis seems to continu...	19	1
2	1314	618	619	Re: 7/00 Sithe Financial Liquidations	693	109	246	Please use the following two files as your det...	68	0
3	240	2301	746	Re: California Amendments DEFEATED!	817	85	271	Congrats Linda Robertson 07/19/2001 09:44 AM T...	56	1
4	1389	2237	816	FW: Donnie Vinson thanks you for your ALS dona...	875	207	114	-----Original Message----- From: Vinson, Donal...	138	0
...	...	...	...	...	...	...	...	...	...	...
2995	2488	2836	6	RE: Flordia	582	113	287	Ok, I will try to take a look at it this after...	72	0
2996	1477	85	746	Duchesne - Melissa called and cancelled.	817	85	271	Put FH on AAE retainer Bd Meeting on Sun in Bd...	56	1
2997	678	80	746	Andy Fastow	817	85	271	Response to Tarpey on Schnitzer/Tierney (303)5...	56	1
2998	2478	912	107	Re: (no subject)	214	60	129	When do you take the bike to the shop? I have ...	36	0
2999	2175	2786	193	Williams Energy News Live -- today's video new...	415	23	222	=09 =09 Dear Michael, I'm Washington Bureau Ch...	13	0

Figure 3: Dataset Post Encoding

3.8. Natural Language Processing

Natural Language Processing (NLP) allows machine learning algorithms and computers, in general, to understand and process human language. NLP is, therefore, crucial to machine learning and artificial intelligence as it allows us to train models in the basic human language (Nadkarni et al., 2011). The authors further explained that special characters and texts such as punctuation, emojis, and stopwords need to be removed from any model as part of the NLP. Some of these steps under NLP were adopted in this research.

3.9. Handling Punctuations

An in-built function in python provides all the punctuations in available in the English language. This made it easier to remove the punctuation from the email subject and content.

3.10. Repeating Characters

In addition to removing punctuation, this research also removed repeating characters from words. As explained by (Bird et al., 2009), people often include multiple characters in-between words, however, these machine learning algorithms require consistent data across the dataset to retrieve meaningful information for modelling. For example, words like “cool” and “coool” would be read differently by an algorithm even though human beings would read them as the same word. Removing repeating characters helps the algorithm tackle this challenge. Therefore, the repeated characters were removed from both the email subject and the email content to improve consistency.

3.11. Remove Stopwords

Stopwords are basically the most common words in any language. Each language has a series of stopwords that should be excluded from datasets before modelling as they do not add any value to the model being developed (Raulji and Saini, 2016). For this research, the English stopwords were removed from the dataset. There are a total of 179 stopwords in the English Language.

3.12. Tokenization

This is the process of splitting sentences or paragraphs into words that can be easily understood by the algorithm (Webster and Kit, 1992). (Manning et al., 2014), more recent research further details the importance of tokenization in natural language processing – explaining that tokenization, paired with vectorization is fundamental for preparing a text-based dataset for modelling.

```
0    [Se, NERC, memo, received, They, scheduled, ho...
1    [FYI, From, todays, Post, Davis, sems, continu...
2    [Please, use, folowing, two, files, detail, su...
3    [Congrats, Linda, Robertson, 0719201, 094, AM,...
4    [Original, Mesage, From, Vinson, Donald, Wayne...
Name: content, dtype: object
```

Figure 4: Snapshot of Applied Tokenization

Subsequently, this research applied vectorization as the next step in preparing the dataset for modelling.

3.13. Vectorization

Vectorization is a feature engineering technique that extracts distinct features from within a specific column, converts them to independent columns, and then represents each row with 1 if that row contains the extracted feature or 0 if the row does not contain the extracted feature. The vectorization techniques were applied after the tokenization was applied to the dataset. The technique applied extracted the top 500 tokenized features from the email subject and email content and converted them to independent columns as recommended by (Manning et al., 2014).

The resulting dataframe had a total of 1008 columns/ features and 3000 rows as shown below.

	Message-ID	Date	From	X-From	X-Origin	X-FileName	user	labeled	1	10	...	west	western	wholesale	wil	work	wptf	wsj	year	york	youre
0	1950	2354	363	788	181	90	111	1	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	2584	2210	319	417	28	13	19	1	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1314	618	619	693	109	246	68	0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	240	2301	746	817	85	271	56	1	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1389	2237	816	875	207	114	138	0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2995	2488	2836	6	582	113	287	72	0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2996	1477	85	746	817	85	271	56	1	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2997	678	80	746	817	85	271	56	1	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2998	2478	912	107	214	60	129	36	0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2999	2175	2786	193	415	23	222	13	0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

3000 rows × 1008 columns

Figure 4: Resulting Dataset After Modification

3.14. Modelling

3.14.1. Random Forest Algorithm

For a more precise forecast, Random Forest produces numerous decision trees that are then combined. The rationale behind the Random Forest model is that a collection of uncorrelated algorithms, such as the various decision trees, performs significantly better as a whole than it does separately (Machado et al., 2015).

Step 1: Samples from the training dataset are selected.

Step 2: A decision tree is generated for each training data selected.

Step 3: Predictions are made from each decision tree generated.

Step 4: Voting is then implemented in favor of the majority prediction.

Step 4: Finally, the result of the voting is taken as the final prediction result.

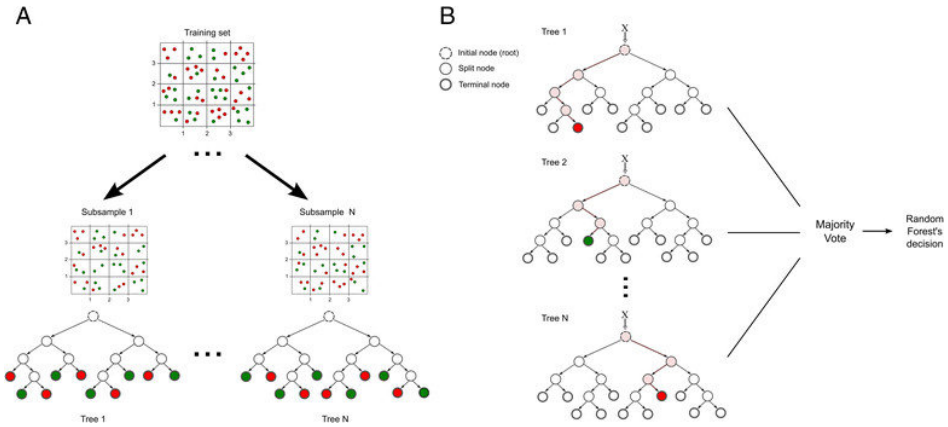

Figure 5: Random Forest Classifier (Machado et al., 2015)

3.14.2. Ensemble Learning Algorithm

Ensemble learning is a wide conceptual machine learning technique that seeks to enhance prediction outcomes by combining the forecasts from several algorithms. Even though there appear to be a limitless amount of ensembles you can design for the predictive analysis, there are just three strategies that dominate the area of ensemble learning. (Gomes et al., 2017).

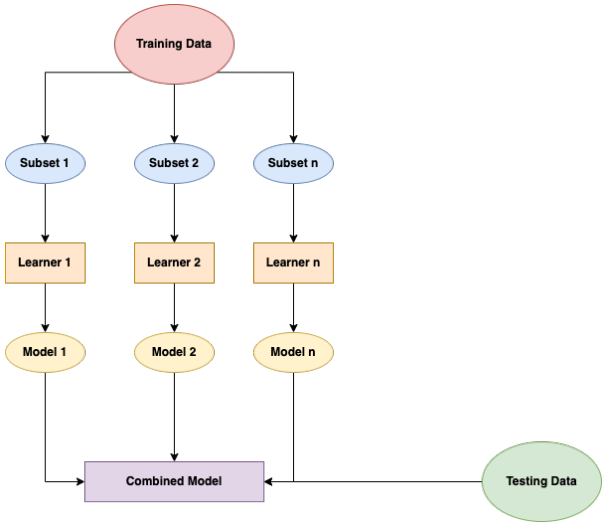

Figure 6: Ensemble Learning Algorithm

The random forest algorithm and the ensemble learning were implemented and compared in this research.

3.15. Model Evaluation

After training and testing the model, the next stage of the research involves assessing the performance of each model. While accuracy is an important assessment for every model, there are other metrics used to assess the predictive power of a model. Assessments such as the confusion matrix, precision, and recall form a vital part of model assessment depending on the goal of the research (Naghibi and Pourghasemi, 2015).

3.16. Confusion Matrix

The confusion matrix is ideal for classification with 2 possible outcomes, as is the case in this research. It shows the actual values against the predicted values in the form of a 2x2 matrix. The sections in the matrix represent the true negative, true positive, false negative, and false positive predictions. The true negative values are the number of negative predictions which were actual negative values, the true positive values are the number of positive

predictions which are actual positive values, the false negative values are the number of negative predictions which are actual positive values while the false positive values are the number of positive predictions which are actual negative values (Visa et al., 2011).

		Predicted Values	
		Negative	**Positive**
Actual Values	**Negative**	**True Negative (TN)**	**False Positive (FP)** Type 1 Error
	Positive	**False Negative (FN)** Type 2 Error	**True Positive (TP)**

Figure 7: Confusion Matrix

i. True Positive (TP) is considered to be True Positive (TP) when the predicted value turns out to be accurate.
ii. True Negative (TN) is when the anticipated value, which in this case is negative, actually occurs, the value is said to be True Negative.
iii. False Positive (FP) is one in which the predicted value, in this case, a positive one turns out to be false.
iv. False Negative (FN) is one in which the expected value, in this case, a negative one turns out to be true.

Precision is the number of true positive values divided by the total number of positive values. The formula is shown below:

$$\frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

Recall is the number of true positives divided by the total number of possible positive values. The formula is shown below:

$$\frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

F1 Score

Also known as F1, is the harmonic mean of the precision and recall calculated above. The formula for the f-score is

$$\frac{2 * precision * recall}{precision + recall}$$

4.0. Design Specification

The design specification provides a visual representation of the flow of the entire research project. Various stages of this research have been explained as part of the research methodology above. The project begins with downloading the mbox file from the authors, after which data processing commences. The processing steps include balancing the dataset and eliminating null values. In addition, the data is transformed by encoding the dataset and removing punctuations, repeated characters, and stopwords from the dataset.

All the columns except the subject and contents are then encoded to make them machine-readable. The subject and contents are then tokenized by splitting the sentences into words. The result from the tokenization is then vectorized. This process involves extracting the top 500 words from the subject and content and then converted to columns and representing them with 1 and 0 depending on if they are present in the row or not respectively. The result from the dataset is then combined into a single dataframe in preparation for modelling.The final

dataframe is split into 65% and 35% for training and testing respectively. (Kearns, 1995) suggests a split of at least 60:40 between the training and the testing dataset as the model would have sufficient data to learn how to make predictions.

The random forest algorithm and the ensemble learning algorithm are then used to create models used to make predictions. Finally, both models are evaluated using the assessment metrics defined above. As the research is aimed at comparing the performance of the random forest algorithm against the ensemble learning algorithm, the models are used to make predictions on normal emails and spearphishing emails.

4.1. Project Prerequisite

The design, specification, and configuration of the models that have been utilized for this research work were carried out using a PC with a 2.2GHz quad-core Intel Core i7 processor, 16GB of RAM, and a 1TB hard drive running macOS Monterey.

The software setup needed to run the project includes:

i. Anaconda IDE
ii. Jupyter Notebook
iii. Python
iv. Libraries – pandas, numpy, nltk, sklearn, seaborn, prettytable, and matplotlib.

A visual representation of the design specification is shown below:

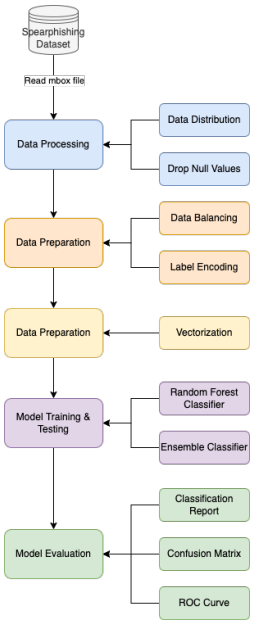

Figure 8: Proposed Spearphishing Detection Design Specification

5.0. Implementation

5.1. IDE and Packages

An IDE (Integrated Development Environment) is an application used to run programming languages. There are numerous IDEs such as pycharm, vscode, and jupyter notebook. The IDE of choice for this research is the jupyter notebook as it provides a web interface for writing, running, and visualizing the code. The programming language used in the entirety of this project is python because it allows for the manipulation of large datasets, as well as modelling and evaluation (Raschka, 2015). The following packages were used during this research.

MBox Files: The mailbox library was used to read the mbox file.
Data Processing: The pandas and NumPy libraries were used for data processing.
Natural Language Processing: the nltk (natural language toolkit) was used for tokenization and stopword removal while the feature extraction library in the sklearn package was used for vectorization.
Data Visualization: the seaborn and matplotlib libraries were used for visualizing the data.
Data Modelling & evaluation: various libraries in the sklearn package were used for splitting, training, testing, and evaluating the models.

5.2. Experiment 1: Random Forest Classifier

As explained in the research methodology, the random forest classifier uses a number of decision tree classifiers on a different subset of the training data to make a prediction. The following default parameters were used when defining the random forest classifier (Ahmad et al., 2018).

N_estimators = 100: the number of estimators can be described as the number of trees in the random forest classifier. This means 100 decision trees were used in producing predictions, thus producing more accurate predictions.

Max_depth = None: the maximum depth of each tree is equal to the number of nodes or branches in the tree. When selecting none, all nodes are explored until a prediction is made.

Random State = 123: the random state allows for consistency and reproducibility of the code. This means that every time the code is run, the same sample of the dataset will be used to train and test the model thus resulting in consistent evaluation.

5.3. Experiment 2: Ensemble Learning Classifier

Like the random forest algorithm, the ensemble learning classifier also uses a voting system to weigh predictions. There are 2 main voting techniques in ensemble learning; they are hard voting and soft voting.

Hard voting, also known as majority voting, means that each estimator makes its own predictions. Once all the predictions are made, the value predicted by the majority of estimators becomes the final prediction of the ensemble learning.

Soft voting, also known as a weighted average, on the other, finds the probability of occurrence for each label in the class, and the label with the higher probability is taken as the final prediction.

While both methods have their advantages, the hard voting technique has been shown to be more suitable for binary classifications as described by (Miller and Yan, 1999).

6.0. Evaluation

This section of this research work discusses the evaluation of the various experiments carried out on our two models by comparing different parameters of the confusion matrix such as Precision, Recall, F1-score, and Accuracy which estimates the efficacy and performance of the predictions from the developed models. Results are presented in graphs and tables

6.1. Experiment 1: Random Forest

The random forest model achieved an average of 95% across all evaluation metrics detailed in the assessment section above. The breakdown of the evaluation indicators and outcomes of the Random Forest model is shown below.

Models	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9533	0.9545	0.9527	0.9536

Table 1: Evaluation of Random Forest Model

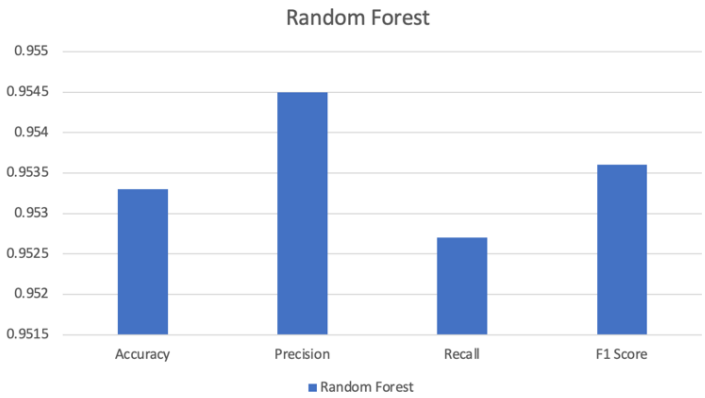

Figure 9: Accuracy, Precision, Recall of Random Forest Model

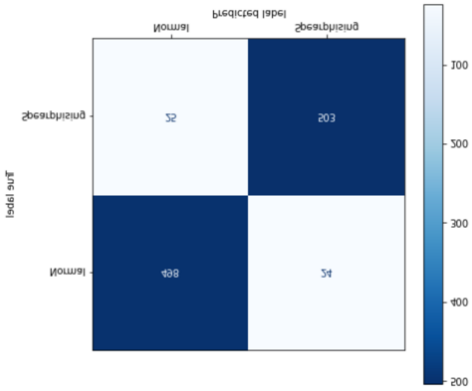

Figure 11: Confusion Matrix of Random Forest Model

6.2. Experiment 2: Ensemble Learning

Similarly, after training the ensemble learning model, it was evaluated using the same metrics as with the random forest model. A breakdown of the evaluation metrics for the ensemble learning model is shown below

Models	Accuracy	Precision	Recall	F1 Score
Ensemble Learning	0.9438	0.9681	0.9186	0.9427

Table 2: Evaluation of Ensemble Learning

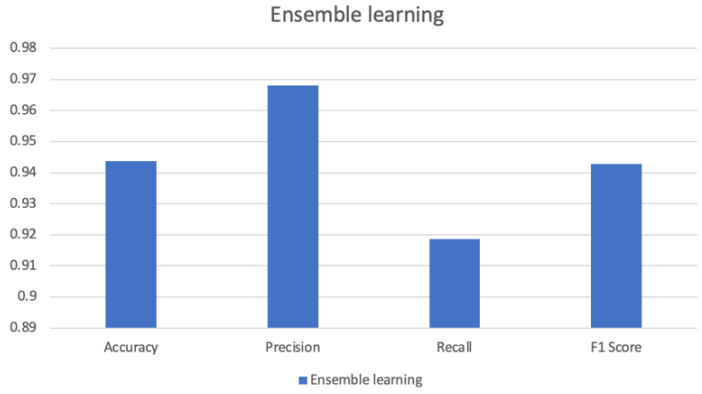

Figure 12: Accuracy, Precision, Recall of Ensemble Learning Model

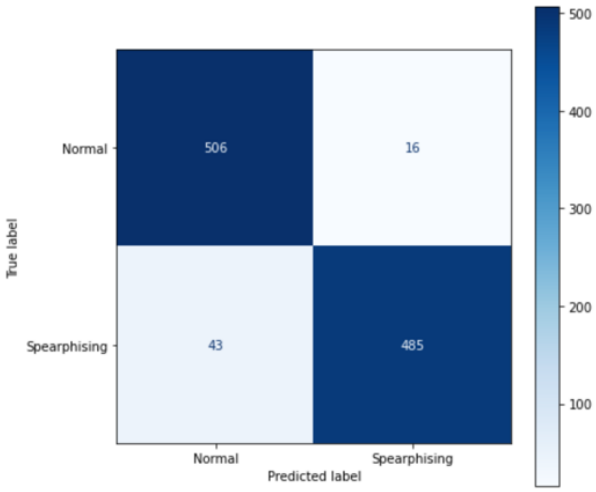

Figure 13: Confusion Matrix of Ensemble Learning Model

6.3. Model Comparison

After performing a comparative analysis between the two algorithms, we can conclude that the models had similar values across all models. the random forest classifier, which is an ensemble learning method outperforms as compared to the ensemble model.

6.4. Accuracy, Precision, Recall & F-Score

The comparison reveals that the random forest model outperformed the ensemble learning in terms of accuracy, precision, and f-score. However, the ensemble learning model achieved a higher precision of about 97% as opposed to 95.5% by the random forest model. As discussed in the assessment section above, the precision explains the ratio of positive predictions against the total positive values.

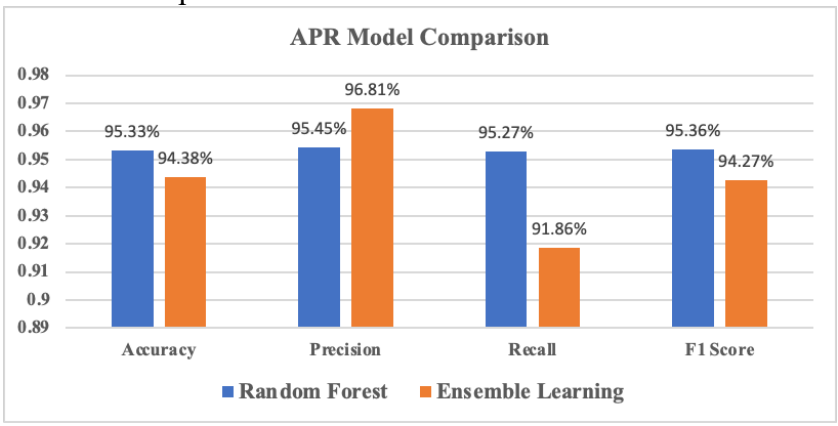

Figure 14: Accuracy, Precision, Recall Comparison

6.5. ROC Curve

The performance of a classification model at each classification threshold is depicted on a graph called a Receiver Operating Characteristic (ROC) curve. The True Positive Rate (TPR) and False Positive Rate (FPR) are two metrics that this curve plots (FPR). TPR and FPR trade-offs are depicted by the ROC curve. The performance of classifiers is better shown by curves that are closer to the top-left corner. This gives credibility to the models that were developed, the testing and training of the dataset, and the results achieved.

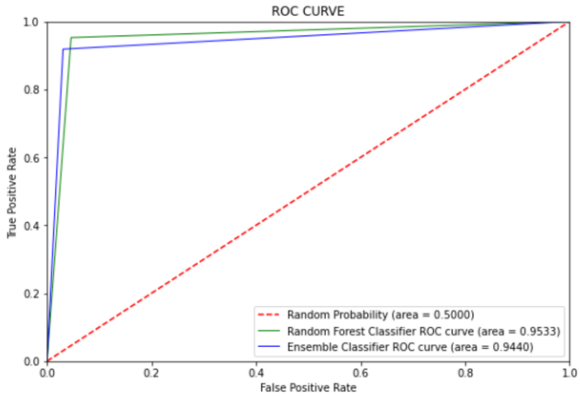

Figure 12: ROC Curve Comparison

7.0. Conclusion and Future Work

Nowadays, phishing attack methods are changing, and several hazardous phishing techniques have been developed that prey on users' and computer systems' vulnerabilities. However, several countermeasures have also been developed, including detection and prevention systems. This paper proposed a model that detects spear-phishing attacks. A model was built that was able to differentiate between spear-phishing emails from normal emails using feature extraction, which was used to train and test our models for classification. The final detection model built was able to answer our request question which says, how effectively do Random Forrest Classifier, and Ensemble Learning compare against one another in the detection of spear phishing attacks? In the future, the use of a much bigger dataset could be used as well as the use of more machine learning algorithms to ascertain the most accurate classification and most effective model to detect spear-phishing attacks. This research work can further be implemented in real-time in the detection of spear-phishing emails.

References

[1] Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A., 2018. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access 6, 33789–33795.

[2] Bhagoji, A.N., Cullina, D., Sitawarin, C., Mittal, P., 2018. Enhancing robustness of machine learning systems via data transformations. Presented at the 2018 52nd Annual Conference on Information Sciences and Systems (CISS), IEEE, pp. 1–5.

[3] Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.

[4] Dasarathy B V, Sheela B V. A composite classifier system design: concepts and methodology. Proceedings of the IEEE, 1979, 67(5): 708–713

[5] Dewan, P., Kashyap, A., & Kumaraguru, P. (2014, September). Analyzing social and stylometric features to identify spear phishing emails. In 2014 apwg symposium on electronic crime research (ecrime) (pp. 1-13). IEEE.

[6] Duman, S., Kalkan-Cakmakci, K., Egele, M., Robertson, W., & Kirda, E. (2016, June). Emailprofiler: Spearphishing filtering with header and stylometric features of emails. In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 408-416). IEEE.

[7] Enron Email Dataset [WWW Document], 2004. URL https://www.cs.cmu.edu/~enron/ (accessed 12.4.22).

[8] Fette, I., Sadeh, N., & Tomasic, A. (2007, May). Learning to detect phishing emails. In Proceedings of the 16th international conference on World Wide Web (pp. 649-656).

[9] Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A., 2017. A survey on ensemble learning for data stream classification. ACM Computing Surveys (CSUR) 50, 1–36.

[10] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. Detecting credential spearphishing in enterprise settings. In 26th {USENIX} Security Symposium ({USENIX} Security 17), pages 469–485, 2017.

[11] Han, Y., & Shen, Y. (2016, April). Accurate spear phishing campaign attribution and early detection. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 2079-2086).

[12] Han, YuFei, and Yun Shen. "Accurate Spear Phishing Campaign Attribution and Early Detection." *Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC '16*, 2016, 10.1145/2851613.2851801.

[13] Idreos, S., Papaemmanouil, O., Chaudhuri, S., 2015. Overview of data exploration techniques. Presented at the Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 277–281.

[14] Kaiser, J., 2014. Dealing with Missing Values in Data. JoSI 42–51. https://doi.org/10.20470/jsi.v5i1.178

[15] Kearns, M., 1995. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. Advances in neural information processing systems 8.

[16] Khonji, M., Iraqi, Y., & Jones, A. (2011, December). Mitigation of spear phishing attacks: A content-based authorship identification framework. In 2011 International Conference for Internet Technology and Secured Transactions (pp. 416-421). IEEE.
[17] Lemaître, G., Nogueira, F., Aridas, C.K., 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research 18, 559–563.
[18] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit. Presented at the Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60.
[19] Nadkarni, P.M., Ohno-Machado, L., Chapman, W.W., 2011. Natural language processing: an introduction. Journal of the American Medical Informatics Association 18, 544–551.
[20] Naghibi, S.A., Pourghasemi, H.R., 2015. A comparative assessment between three machine learning models and their performance comparison by bivariate and multivariate statistical methods in groundwater potential mapping. Water resources management 29, 5217–5236.
[21] Prakash P., Kumar M., Kompella R., and Gupta M., “Phishnet: Predictive Blacklisting to Detect Phishing Attacks,” in Proceedings IEEE INFOCOM, San Diego, pp. 1-5, 2010.
[22] Raschka, S., 2015. Python machine learning. Packt publishing ltd.
[23] Raulji, J.K., Saini, J.R., 2016. Stop-word removal algorithm and its implementation for Sanskrit language. International Journal of Computer Applications 150, 15–17.
[24] Statista, "Statista," Statista, [Online]. Available: http://www.statista.com/. [Accessed December 2022].
[25] Schapire, Robert E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197–227
[26] Symantec, "Symantec internet security threat report," Symantec, 2019. [Online]. Available: http://www.symantec.com/enterprise/threatreport/index.jsp.. [Accessed December 2022].
[27] Stringhini, G., & Thonnard, O. (2015, July). That ain’t you: Blocking spearphishing through behavioral modelling. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 78-97). Springer, Cham.
[28] Stembert, N., Padmos, A., Bargh, M. S., Choenni, S., & Jansen, F. (2015, September). A study of preventing email (spear) phishing by enabling human intelligence. In 2015 European intelligence and security informatics conference (pp. 113-120). IEEE.
[29] Visa, S., Ramsay, B., Ralescu, A.L., Van Der Knaap, E., 2011. Confusion Matrix-based Feature Selection. MAICS 710, 120–127.
[30] Webster, J.J., Kit, C., 1992. Tokenization as the initial phase in NLP. Presented at the COLING 1992 volume 4: The 14th international conference on computational linguistics.
[31] Yen, S.-J., Lee, Y.-S., 2006. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset, in: Intelligent Control and Automation. Springer, pp. 731–740.