Development of a Novel Approach to Phishing Detection using Machine Learning

Article · June 2024

1 author:

Taofeek Olayinka Agboola
Stephen F. Austin State University
6 PUBLICATIONS **48** CITATIONS

SEE PROFILE

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Development of a Novel Approach to Phishing Detection Using Machine Learning

Agboola Olayinka Taofeek
Stephen F. Austin State University,
Nacogdoches, Texas, USA.

ABSTRACT
Protecting and preventing sensitive data from being used appropriately has become a challenging task. Even a small mistake in securing data can be exploited by phishing attacks to release private information such as passwords or financial information to a malicious actor. Phishing has now proven so successful; that it is the number one attack vector. Many approaches have been proposed to protect against this cyber-attack, from additional staff training, and enriched spam filters to large collaborative databases of known threats such as PhishTank and OpenPhish. However, they mostly rely upon a user falling victim to an attack and manually adding this new threat to the shared pool, which presents a constant disadvantage in the fight back against phishing. In this paper, we propose a novel approach to protect against phishing attacks using machine learning. Unlike previous work in this field, our approach uses an automated detection process and requires no further user interaction, which allows for a faster and more accurate detection process. The experiment results show that our approach has a high detection rate. Machine Learning is an effective method for detecting phishing. It also eliminates the disadvantages of the previous method. We thoroughly reviewed the literature and suggested a new method for detecting phishing websites using feature extraction and a machine learning algorithm. This research aims to use the dataset collected to train ML models and deep neural nets to anticipate phishing websites.

ARTICLE INFO
Article History
Received: July, 2023
Received in revised form: September, 2023
Accepted: February, 2024
Published online: June, 2024

KEYWORDS
Phishing, Machine Learning, Security, Spam, Binary Visualisation.

INTRODUCTION

The Internet has become an integral part of our daily activities; from communicating through social networking sites and emails to banking, studying and shopping, the Internet has touched every aspect of our life [1]. Phishing has become prominent due to users interfacing daily with government, financial institutions, corporations and digital network platforms which sometimes results in ransomware. Phishers use different vectors to exploit vulnerabilities and users such as messaging “spear phishing”, VOIP, spoofed link and counterfeit websites. It is very easy to create counterfeit websites, which looks like a genuine website in terms of layout and content. Even, the content of these websites would be identical to that of legitimate websites. The wildly important goal of creating a dummy website some includes collection of personal identifiable information like user login details, account information, date of birth and home address for credit card access authorisation, etc. Moreover, attackers ask security questions to answer to posing as a high-level security measure providing to users. When users respond to those questions, they get easily trapped into phishing attacks. Many researches have been going on to prevent phishing attacks by different communities around the world. Phishing attacks can be prevented by detecting the websites and creating awareness to users to identify the phishing websites. Machine learning algorithms have been

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

one of the powerful techniques in detecting phishing websites. In this study, various methods of detecting phishing websites have been discussed.

Phishing has been purported as one of the greatest attack vectors that is causing great harm to online services and data security [3, 4]. This cyber-security threat attempts to trick internet users into revealing their private information such as passwords or financial account credentials, usually for the purpose of theft [3, 5]. Social engineering is the core of all phishing attacks, whether targeted or random [5]. This mechanism leads the victim to perform certain actions, such as submitting personal data directly to a malicious actor or executing malicious software that indirectly submits the data to the malicious actor without the victim knowing [4, 5].

Social engineering can manifest in the form of an email (spoofed email) or a clone of a legitimate website, so that the victim will not be able to differentiate between phishing and legitimate webpages. In addition, the attacker can use key phrases to emphasize the sense of urgency for the victim, for example, "You MUST complete this account check NOW". This fear tactic leads the victim to click on a malicious link or fill out a form on the phishing site. According to the PhishLabs report [6], 255,065 unique phishing attacks were found worldwide in the third quarter of 2018, where 83.9 % of attacks targeted credentials for financial, email, cloud, payment, and SaaS services. PhishLabs affirm that this global impact of phishing attacks will continue to increase and therefore requires more efficient anti-phishing techniques to handle new and emerging phishing patterns [6].

In recent years, great effort has been directed to curb the effectiveness of phishing. A variety of approaches has been proposed including additional staff training, enriched spam filters, and large collaborative data bases of known threats such as PhishTank and OpenPhish. Whilst these methods have proved effective in raising the awareness of this common cyber threat, they only have the ability to handle known phishing patterns, when a user falls victim to an attack, they manually added this new threat to the shared pool and thus leave internet users prone to new phishing attacks. However, phishers are not static in their activities; they change their mode of operation frequently to stay undetected and bypass existing techniques. When paired with the fact that cyber security is known as a reactive industry, this presents a constant disadvantage in the fight back against changing phishing patterns.

Types of Phishing Attacks

1. Deceptive Phishing: This is the most frequent type of phishing assault, in which a Cyber criminal impersonates a well-known institution, domain, or organization to acquire sensitive personal information from the victim, such as login credentials, passwords, bank account information, credit card information, and so on. Because there is no personalization or customization for the people, this form of attack lacks sophistication.
2. Spear Phishing: Emails containing malicious URLs in this sort of phishing email contain a lot of personalization information about the potential victim. The recipient's name, company name, designation, friends, co-workers, and other social information may be included in the email.
3. Whale Phishing: To spear phish a "whale," here a top-level executive such as CEO, this sort of phishing targets corporate leaders such as CEOs and top-level management employees.
4. URL Phishing: To infect the target, the fraudster or cyber-criminal employs a URL link. People are sociable creatures who will eagerly click the link to accept friend invitations and may even be willing to disclose personal information such as email addresses. This is because the phishers are redirecting users to a false web server. Secure browser connections are also used by attackers to carry out their unlawful actions. Due to a lack of appropriate tools for combating phishing attacks,

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

firms are unable to train their staff in this area, resulting in an increase in phishing attacks. Companies are educating their staff with mock phishing assaults, updating all their systems with the latest security procedures, and encrypting important Information as broad countermeasures. Browsing without caution is one of the most common ways to become a victim of this phishing assault. The appearance of phishing websites is like that of authentic websites.

STATEMENT OF PROBLEM

Phishing attacks pose a significant threat to cybersecurity, leading to substantial financial losses and data breaches. Traditional detection methods struggle to keep pace with the evolving techniques used by attackers, resulting in inadequate protection for users. This project aims to develop a novel machine learning-based approach to enhance phishing detection accuracy and efficiency, addressing the limitations of existing solutions and improving overall security measures against phishing threats.

Aim and Objectives of the Study

The primary aim of this study is to develop a novel machine learning-based approach for effective phishing detection, enhancing cybersecurity measures by improving detection accuracy and response times.

1. Conduct a comprehensive review of existing phishing detection methods and machine learning techniques.
2. Design and implement various machine learning models tailored for phishing detection.
3. Evaluate the performance of the developed models using standard metrics such as accuracy, precision, recall, and F1-score.

Scope of the Study

1. Phishing Detection Focus: This study specifically targets the detection of phishing attempts through various online communication channels, such as emails, websites, and messages, using machine learning techniques.
2. Machine Learning Techniques: The research will explore a range of machine learning algorithms, including supervised, unsupervised, and ensemble methods, to identify the most effective approach for phishing detection.
3. Dataset Utilization: The study will involve the collection and preprocessing of datasets comprising phishing and legitimate examples from publicly available sources and possibly proprietary datasets, ensuring a comprehensive and representative sample for model training and testing.
4. Feature Engineering: The research will investigate feature extraction and selection techniques to identify the most relevant attributes that contribute to distinguishing phishing attempts from legitimate communications.
5. Model Development and Evaluation: Multiple machine learning models will be developed, trained, and evaluated based on performance metrics such as accuracy, precision, recall, and F1-score. The best-performing model will be selected for further development.
6. Comparative Analysis: The novel approach will be compared against existing phishing detection methods to highlight improvements and potential benefits.

Limitations of the Study

1. Data Quality and Diversity: The effectiveness of the phishing detection model heavily relies on the quality and diversity of the dataset. Limited access to varied and comprehensive datasets may impact the model's generalizability to different phishing scenarios.
2. Evolving Phishing Techniques: Phishing strategies are continuously evolving. The models developed during the study

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

may not be effective against new and sophisticated phishing techniques that emerge after the research period.

3. Feature Extraction Challenges: Identifying the most relevant features for phishing detection can be complex and may require domain-specific knowledge. Inadequate feature selection could reduce the model's performance.
4. Model Interpretability: Some advanced machine learning models, such as deep learning, may offer high accuracy but lack interpretability. This can be a limitation when explaining the model's decisions to stakeholders or debugging the system.
5. Resource Constraints: Developing, training, and testing machine learning models require significant computational resources. Limited access to such resources can affect the scale and speed of experimentation.
6. False Positives and Negatives: While aiming for high detection accuracy, the model may still produce false positives (legitimate communications flagged as phishing) and false negatives (phishing attempts not detected), impacting its reliability.
7. Real-World Applicability: The controlled environment of the study may not fully replicate real-world conditions. Therefore, the model's performance in actual deployment scenarios could vary.
8. Ethical and Privacy Concerns: Handling sensitive data for training the model poses ethical and privacy concerns. Ensuring compliance with data protection regulations is essential but may limit the types of data used.

LITERATURE REVIEW

Over time a wide variety of approaches have been proposed to counter the ever-persistent threat of phishing in both commercial and public domains. These approaches can be classified into two main categories; user training approaches and software classification approaches [7]. Training approaches aim at raising the ability of end-users to identify phishing attacks [7, 8], which could reduce their susceptibility to falling victim to phishing attacks [8]. While classification approaches are typically designed to classify phishing and legitimate web pages on behalf of the user in an attempt to tackle issues of the human error and ignorance [7].

Numerous scholars have conducted analyses on phishing URL statistics. Our method integrates key insights from previous studies. We survey prior research on phishing site detection using URL features, which informs our current approach. Phishing is identified as a significant threat, facilitating hackers in obtaining user credentials covertly. Users often fall prey to these scams due to a combination of financial need, lack of experience, and trust issues. Mehmet et al. propose a phishing detection method based on URL analysis, utilizing various machine learning algorithms and hierarchical structures. Garera et al. employ logistic regression to classify phishing URLs, incorporating red flag keywords and features based on Google's recommendations. Yong et al. introduce a novel approach focusing on accurate URL detection using a capsule-based neural network. Our system, leveraging machine learning techniques, analyzes URL properties extensively, demonstrating competitive performance against contemporary methods.

Shahrivari et al., Kumar, Alswailem et al., and others explore machine learning approaches for phishing detection, with random forest often yielding high accuracy. Frameworks like "Fresh-Phish" aid in building machine learning datasets for phishing websites, showcasing high accuracy in classification. Zhang suggests a model based on semantic mining of Chinese web pages, utilizing various classifiers to detect phishing effectively. Aydin presents a versatile framework for feature extraction using data from Phish Tank and Google. Feature selection techniques like CFS and consistency subset-based selection, coupled with algorithms like Nave Bayes and SMO, are evaluated for phishing detection, with SMO being favored for its performance.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

They were able to achieve high accuracy using a Support Vector Machine and data that had been pre-processed using NLP approaches. Amani Alswailem et al. tried different machine learning model for phishing detection but was able to achieve more accuracy in random forest. Hossein et al. created the ―Fresh-Phish‖ open-source framework. This system can be used to build machine-learning data for phishing websites. They used a smaller feature set and built the query in Python. They create a big, labelled dataset and test several machine-learning classifiers on it. Using machine-learning classifiers, this analysis yields very high accuracy. These studies look at how long it takes to train a model. X.

Zhang suggested a phishing detection model based on mining the semantic characteristics of word embedding, semantic feature, and multi-scale statistical features in Chinese web pages to detect phishing performance successfully. To obtain statistical aspects of web pages, eleven features were retrieved and divided into five classes. To obtain statistical aspects of web pages, eleven features were retrieved and divided into five classes. To learn and evaluate the model, AdaBoost, Bagging, Random Forest, and SMO are utilized. The legitimate URLs dataset came from DirectIndustry online guides, and the phishing data came from China's Anti-Phishing Alliance. With novel methodologies, M. Aydin approaches a framework for extracting characteristics that is versatile and straightforward.

Phish Tank provides data, and Google provides authentic URLs. C# programming and R programming were utilized to btain the text attributes. The dataset and thirdparty service providers yielded a total of 133 features. The feature selection approaches of CFS subset based and Consistency subset-based feature selection were employed and examined with the WEKA tool. The performance of the Nave Bayes and Sequential Minimal Optimization (SMO) algorithms was evaluated, and the author prefers SMO to NB for phishing detection.

METHODOLOGY

A phishing website is a social engineering technique that imitates legitimate webpages and uniform resource locators (URLs). The Uniform Resource Locator (URL) is the most common way for phishing assaults to occur. Phisher has complete control over the URL's sub-domains. The phisher can alter the URL because it contains file components and directories. Methodologies

This research used the linear-sequential model, often known as the waterfall model. Although the waterfall approach is considered conventional, it works best in instances where there are few requirements. The application was divided into smaller components that were built using frameworks and hand-written code.

Research framework:

The steps of this research in which some selected publications were read to determine the research gap and, as a result, the research challenge was defined. Feature selection, classification and phishing website detection were all given significant consideration. It's worth noting that most phishing detection researchers rely on datasets they've created. However, because the datasets utilized were not available online for those who use and check their results, it is difficult to assess and compare the performance of a model with other models. As a result, such results cannot be generalized.

Language

For the preparation of this dissertation, I used Python as the primary language. Python is a language that is heavily focused on machine learning. It includes several machine learning libraries that may be utilized straight from an import. Python is commonly used by developers all around the world to deal with machine learning because of its extensive library of machine learning libraries. Python has a strong community, and as a result, new features are added with each release.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

Data Collection

The phishing URLs were gathered using the open source tool Phish Tank. This site provides a set of phishing URLs in a variety of forms, including csv, json, and others, which are updated hourly. This dataset is used to train machine learning models with 5000 random phishing URLs.

Data Sanitization

Fill in missing numbers, smooth out creaking data, detect and delete outliers, and repair anomalies to clean up the data.

Data Pre-processing

Data pre-processing is a cleaning operation that converts unstructured raw data into a neat, well-structured dataset that may be used for further research. Data pre-processing is a cleaning operation that transforms unstructured raw data into well-structured and neat dataset which can be used for further research.

Extraction of Features

In the literature and commercial products, there are numerous algorithms and data formats for phishing URL detection. A phishing URL and its accompanying website have various characteristics that distinguish them from harmful URLs. For example, to mask the true domain name, an attacker can create a long and complicated domain name. Different types of features that are used in machine learning algorithms in the academic study detection process are used. The following is a list of features gathered from academic studies for phishing domain detection using machine learning approaches. Because of some constraints, it may not be logical to use some of the features in specific instances. Using Content-Based Features to construct a quick detection mechanism capable of analyzing a huge number of domains may not be feasible. PageBased Features are not very effective when analyzing registered domains. As a result, the features that the detection mechanism will use are determined by the detection mechanism's purpose. So, which features should be used in the detecting technique been carefully chosen.

Models and Training

The data was split into 8000 training samples and 2000 testing samples, before the ML model is trained. It is evident from the dataset that this is a supervised machine learning problem. Classification and regression are the two main types of supervised machine learning issues. Because the input URL is classed as legitimate or phishing, this data set has a classification problem. The following supervised machine learning models were examined for this project's dataset training:

1. Decision Tree
2. Multilayer Perceptron
3. Random Forest
4. Autoencoder Neural Network
5. XGBoost
6. Support Vector Machines Design Specification:

The project is having three features that been extracted from data. The features are Address Bar based, Domain based, and HTML and JavaScript based. In the below section will discuss in detail.

Address Based

Below are the categories been extracted from address based

1. Domain of the URL Where domain which is present in the URL been extracted
2. IP Address in the URL The presence of an IP address in the URL is checked. Instead of a domain name, URLs may contain an IP address. If an IP address is used instead of a domain name in a URL, we can be certain that the URL is being used to collect sensitive information.
3. "@" Symbol in URL The presence of the'@' symbol in the URL is checked. When the —@ll symbol is used in a URL, the browser ignores anything before the —@ll symbol, and the

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

genuine address is commonly found after the —@ll symbol.

4. Length of URL Calculates the URL's length. Phishers can disguise the suspicious element of a URL in the address bar by using a lengthy URL. If the length of the URL is larger than or equal to 54 characters, the URL is classed as phishing in this project.
5. Depth of URL Calculates the URL's depth. Based on the'/', this feature determines the number of subpages in the given address.
6. Redirection "//" in URL: The existence of"//" in the URL is checked. The presence of the character"//" in the URL route indicates that the user will be redirected to another website. The position of the"//" in the URL is calculated. We discovered that if the URL begins with —HTTP, ll the —//ll should be placed in the sixth position. If the URL uses —HTTPS, ll however, the —//ll should occur in the seventh place.
7. Http/Https in Domain name: The existence of "http/https" in the domain part of the URL is checked. To deceive users, phishers may append the —HTTPSll token to the domain section of a URL.
8. Using URL Shortening Services: URL shortening is a means of reducing the length of a URL while still directing to the desired webpage on the "World Wide Web." This is performed by using a —HTTP Redirectll on a short domain name that points to a webpage with a long URL.
9. Prefix or Suffix "-" in Domain Checking for the presence of a '-' in the URL's domain part. In genuine URLs, the dash symbol is rarely used. Phishers frequently append prefixes or suffixes to domain names, separated by (-), to give the impression that they are dealing with a legitimate website.

Domain Based

This category contains a lot of features that can be extracted. This category contains a lot of features that can be extracted. The following were considered for this project out of all of them. **DNS Record:** In the case of phishing websites, the WHOIS database either does not recognize the stated identity or there are no records for the host name .

Web Traffic: This function determines the number of visitors and the number of pages they visit to determine the popularity of the website. In the worst-case circumstances, legitimate websites placed among the top100,000, according to our data. Furthermore, it is categorized as "Phishing" if the domain has no traffic or is not recognized by the Alexa database.

Age of Domain: This information can be retrieved from the WHOIS database. Most phishing websites are only active for a short time. For this project, the minimum age of a legal domain is deemed to be 12 months. Age is simply the difference between the time of creation and the time of expiry.

End Period of Domain: This information can be gleaned from the WHOIS database. The remaining domain time is calculated for this feature by determining the difference between the expiry time and the current time. For this project, the valid domain's end time is regarded to be 6 months or fewer.

HTML and JavaScript based: Many elements that fall within this group can be extracted. The following were considered for this project out of all of them.

1. IFrame Redirection IFrame is an HTML tag that allows you to insert another webpage into the one you're now viewing. The —iframell tag can be used by phishers to make the frame invisible, i.e., without frame borders. Phishers employ the —frame borderll attribute in this case, which causes the browser to create a visual boundary.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

2. Status Bar Customization Phishers may utilize JavaScript to trick visitors into seeing a false URL in the status bar. To get this feature, we'll need to delve into the webpage source code, specifically the "on Mouseover" event, and see if it alters the status bar.
3. Disabling Right Click Phishers disable the right-click function with JavaScript, preventing users from viewing and saving the webpage source code. This functionality is handled in the same way as "Hiding the Link with on Mouseover." Nonetheless, we'll look for the event "event. button==2" in the webpage source code and see if the right click is disabled for this functionality.
4. Website Forwarding: The number of times a website has been redirected is a narrow line that separates phishing websites from authentic ones. We discovered that authentic websites were only routed once in our sample. Phishing websites with this functionality, on the other hand, have been redirected at least four times.
5. Implementation: We'll examine at the implementation component of our artefact in this area of the report, with a focus on the description of the developed solution. This is a task that requires supervised machine learning.

Dataset

We collected the datasets from the open-source platform called Phishing tank. The dataset that collected was in csv format. There are 18 columns in the dataset, and we transformed the dataset by applying data pre-processing technique. To see the features in the data we used few of the data frame methods for familiarizing. For visualization, and to see how the data is distributed and how features are related to one another, a few plots and graphs are given. The Domain column has no bearing on the training of a machine learning model. We now have 16 features and a target column. The recovered features of the legitimate and phishing URL datasets are simply concatenated in the feature extraction file, with no shuffling. We need to shuffle the data to balance out the distribution while breaking it into training and testing sets. This also eliminates the possibility of over fitting during model training.

Machine Learning Models

For phishing website identification, we used many machine learning methods. We used the classification and regression algorithms listed below.

Decision Tree Classifier

For classification and regression applications, decision trees are commonly used models. They basically learn a hierarchy of if/else questions that leads to a choice. Learning a decision tree is memorizing the sequence of if/else questions that leads to the correct answer in the shortest amount of time. The method runs through all potential tests to discover the one that is most informative about the target variable to build a tree.

Random Forest Classifier

Random forests are one of the most extensively used machine learning approaches for regression and classification. A random forest is just a collection of decision trees, each somewhat different from the others. The notion behind random forests is that while each tree may do a decent job of predicting, it will almost certainly overfit on some data. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability.

MLPs

Feed-forward neural networks, or simply neural networks, are another name for multilayer perceptron‘s. MLPs are expansions of linear models that conduct many steps of processing to arrive at a decision. They can be used for both classification and regression problems.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

XGBoost

These days, XGBoost is one of the most prominent machine learning algorithms. eXtreme Gradient Boosting is the abbreviation for XGBoost. Regardless of whether the goal at hand regression or classification is, XGBoost is a high-performance and high-speed implementation of gradient boosted decision trees.

Autoencoder

A neural network with the same number of input neurons as output neurons is known as an auto encoder. The input/output neurons will have fewer neurons than the hidden layers of the neural network. The auto-encoder must learn to encode the input to the fewer hidden neurons since there are less neurons. In an auto encoder, the predictors (x) and output (y) are identical.

SVM

SVM are supervised learning models with related learning algorithms used in machine learning to examine data for classification and regression analysis. An SVM training algorithm creates a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples that are individually designated as belonging to one of two categories.

Libraries Used Pandas:

It's a Python-based machine learning library. Pandas is a free and open-source programming language. Pandas is a programming language that is commonly used for dataset loading and data analytics. Pandas is used for machine learning in a variety of domains, including economics, finance, and others. It is extremely user-friendly and can display datasets in a tabular style for easier comprehension. Sklearn: Sklearn is one of the most essential Python libraries for machine learning. Sklearn includes several tools for statistical classification, modelling, regression, dimensionality reduction and clustering. Numpy: Numpy is a Python-based machine learning package. In Python, Numpy is used to deal with arrays. NumPy is used for all calculations using 1-d or 2-d arrays. Numpy also has routines for working with linear algebra and the Fourier transform. MAPTlotlib: MAPTlotlib is a library for data visualization. It's a Python open-source module for plotting graphs from model results. These diagrams can aid in comprehending the circumstance of the outcomes. For easier comprehension, several components of the results can be graphically formatted.

Evaluation In this section, we use different models of machine learning for evaluating the accuracy. It has been explained about the different models in below sections. Where in this project the models are examined, with accuracy as the primary metric. In final stage we have compared the model accuracy. In all circumstances the testing and training datasets are splinted into 20:80 ratio. Experiment 1/ Feature Distribution Here in below figure shows how the data is distributed and how features are related to one another, a few plots and graphs are given.

Experiment 2/ Decision Tree Classifier

The method runs through all potential tests to discover the one that is most informative about the target variable to build a tree. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 82.6% and 81%. Below is the execution of Decision tree classifier algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model.

Experiment 3/ Random Forest Classifier We can limit the amount of over fitting by averaging the outcomes of numerous trees that all operate well and over fit in diverse ways. To construct a random forest model, you must first determine the number of trees to construct. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability. Where we are predicting the accuracy of the model on the samples collected on both trained and test

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

samples. On this we found accuracy of test and training datasets are 83.4% and 81.4%.

Experiment 4/ MLP MLPs can be thought of as generalized linear models that go through numerous phases of processing before deciding. Below is the execution of MLP algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 86.3% and 85.9%.

Experiment 5/ XGBoost Below is the execution of XGBoost algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 86.4% and 86.6%.

Experiment 6/ Auto encoder The auto-encoder must learn to encode the input to the hidden neurons with fewer neurons. In an auto encoder, the predictors (x) and output (y) are identical. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train. X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 81.8% and 81.9%.

Experiment 7/ SVM An SVM training algorithm creates a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples that are individually designated as belonging to one of two categories. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 81.8% and 79.8%.

APPROACH USED

In this section, we will discuss our proposed phishing detection system. As shown in Figure 1, the proposed system consists of two stages, the learning stage, and the detection stage. In the first stage, the samples and the topological structure of the machine learning TensorFlow is built, while in the second stage the submitted URLs are tested against the samples in the database to perform classification. Our approach relies on visualizing scraped HTML files onto 2D images, which are then processed by the TensorFlow that analyses them against its training modules, to distinguish between legitimate and phishing websites.

URLs passed through the system are recorded in a database, thus, each URL submitted by the user is tested to check for duplicates (see Figure 1). This helps in increasing the system overall performance as it could avoid the binary image reproduction process, which is a time-consuming process. If the submitted URL does not exist in the database, the system would automatically scrape the HTML code from the corresponding websites and store it in a string format. The automation of scraping the web page protects users from having to visit the potential phishing page and removes the risk of droppers and browser exploits. In addition, it prevents the user from viewing potentially inappropriate content that may be found on unknown or hacked websites. Once the website source code is scraped and stored, the corresponding binary file is passed to the image creation module, where the image visualisation method Binvis is used to convert the binary files into 2D images. Then, created images are analysed using the neural network TensorFlow to perform classification.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

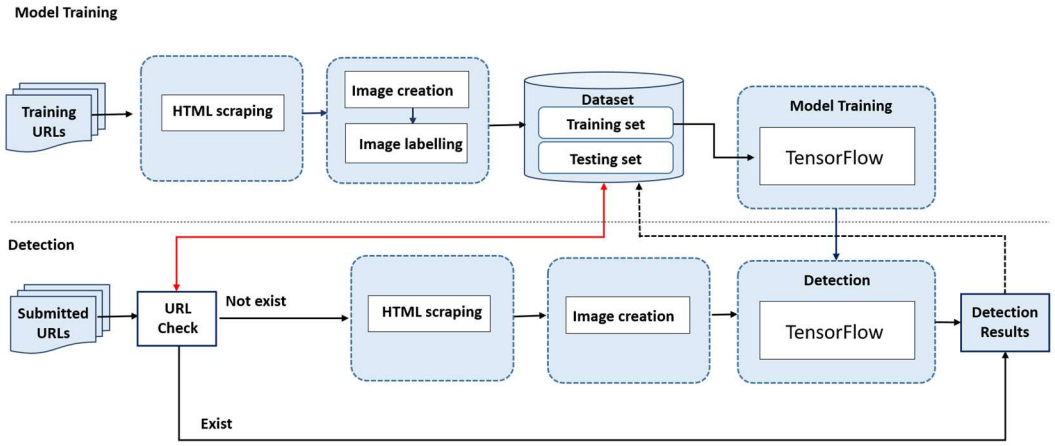

Fig. 1. Overview of the approach.

Binary visualisation

As aforementioned, once the system scrapes and stores a target website 's source code, it then transfers it into BinVis for the RGB image creation process. Binvis is a binary data visualization tool that converts the contents of a binary file to another domain that can be visually represented (typically a two-dimensional space) [25]. This tool takes individual char- acters from the created string in the previous step, translates them to a binary state and then converts them to RGB values (see Figure 2). Binvis divided the different ASCII characters into the following classes of colours;

1. Printable ASCII characters are assigned a blue colour.
2. Control characters are assigned a green colour.
3. Extended ASCII characters are assigned a red colour.
4. Null and (non-breaking) spaces are respectively represented by black (0x00) and white (0xFF) colours.

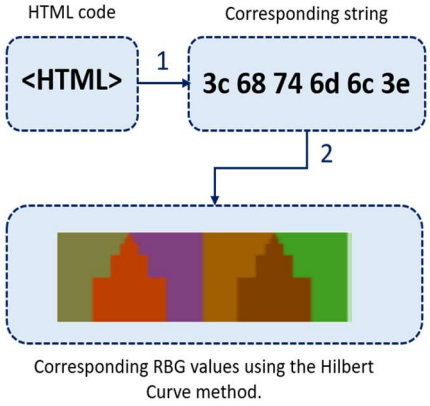

Fig. 2. Image Generation Process.

The final output of Binvis is an image with a pre-set size of 128 pixels. Figure 3 and Figure 4 show respectively Binvis images for a legitimate and a Phishing PayPal Login web page. The images were created using The Hilbert space-filling curve clustering algorithm [26], which overcomes other curves in preserving the locality between objects in multi-dimensional spaces [24, 26]. This helps to create much more appropriate RGB images for the machine learning classification process.

Positive results can be concluded from Figure 3 and Figure 4 as differences between a legitimate site and its phishing counterpart were clear and apparent. The legitimate site has a more detailed RGB value because it would be

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

constructed from additional characters sourced from licenses, hyperlinks, and detailed data entry forms. Whereas, the phishing counterpart would generally contain a single or no CSS reference, multiple images rather than forms and a single login form with no security scripts. This would create a smaller data input string when scrapped.

Image Recognition Classifier

In order to detect phishing pages, the machine-learning algorithm TensorFlow is used to analyse and classify the Binvis images against its in-depth training. TensorFlow is flexible and it has been used for deploying machine learning systems. into production across several areas of computer science, including image recognition, computer vision, robotics, information retrieval, natural language processing and geographic information extraction [27]. Its excellent image recognition ability makes it more appropriate for this application than other similar models. In fact, it could easily detect differences between the images, including differences that the human eye could not detect. TensorFlow takes as input the images produced in the previous steps to perform the classification. For speed of testing, the convolutional neural network MobileNet [28] is employed for the retraining element. This can greatly minimize the time and space for phishing websites classification.

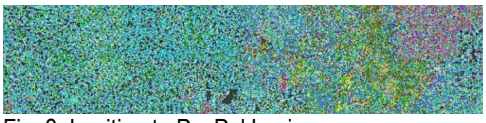

Fig. 3. Legitimate PayPal Login page.

Fig. 4. Phishing PayPal Login page.

IMPLEMENTATION AND RESULTS

In this section, we will discuss implementation details and results of the experiments carried out over our approach in order to demonstrate its effectiveness and reliability. Especially, accuracy metric (A) was used to analyse the results and evaluate the overall performance of our approach.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is the number of instances correctly classified as phishing, TN is the number of instances correctly classified as legitimate web pages, FP is the number of instances incorrectly classified as phishing, and FN is the number of instances incorrectly classified as legitimate web pages.

Precision (P), recall (R) and f1 value (F1) metrics were also used to evaluate the performance of the classifier, where;

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = \frac{2\ X\ P\ X\ R}{P + R} \quad (4)$$

Experiment Setup

For the initial web scraping, a python script was created by using the urllib library [29]. The script would scrape the target site source code and store it in a string format. The experiments are based on the MobileNet model on python with TensorFlow open source library. The TensorFlow framework is deployed in a virtual machine, running on Intel Core i5 CPU, 3.80 GHz, with 8 GB memory and the Ubuntu 14.04 64 bites OS. An NVIDIA GTX 1060 GPU with 6 GB memory is used as accelerator. In the training stage, the TensorFlow algorithm was trained by 250 images per category (Legitimate and phishing web pages) with a size of 128 pixels, for 4000 training steps. The learning rate was 0.005. Every image is used multiple times through training process. As shown in Table I, the phishing websites dataset contained a mixture of 25 samples from the Bank of America PHISH, PayPal Phish, ABSA Phish, DHL TRACKING Phish and Microsoft Login Phish.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(2), JUNE, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Table 1: Phishing websites samples

Category	Number of samples
Bank Of America PHISH	5
PayPal Phish	5
ABSA Phish	5
DHL TRACKING Phish	5
Microsoft Login Phish	5

EXPERIMENTAL RESULTS ANALYSIS

Several tests were carried out to determine the accuracy of the proposed classifier after the addition of more samples; five tests per trained target site were carried out to evaluate the success of the detection method. Figure 5 shows the results of the final test with the most training samples that were collected being used. It is apparent from the results that the classifier has achieved high accuracy for almost all categories, in particular, the ABSA and DHL URLs, where all submitted URLs were correctly labelled as expected. The classifier achieved lower accuracy with the PayPal URLs (85.71%), however, the precision was very high (100%).

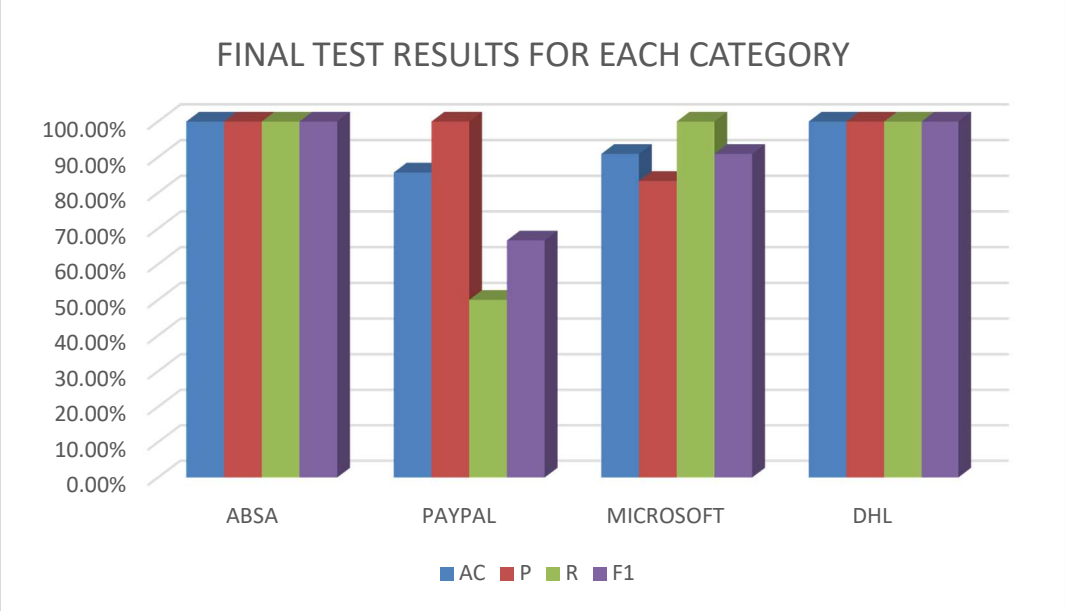

Figure 6: The overall results of the proposed approach

Figure 6 shows the overall results of the proposed approach, which achieved an overall detection accuracy of 94.16%, which is high and meets the required accuracy rate in practical use. The precision of the classification is also very high with a rate of 95.83%, which shows strong overall confidence in the pattern recognition process. This accuracy rate is interpreted as an acceptable and good result for phishing detection. The recall rate was lower than the precision rate (87.50%) because of the PayPal results that need further investigation in future works.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

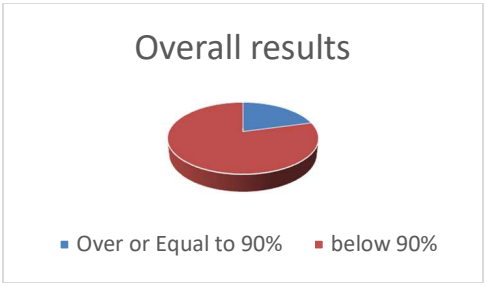

Figure. 6. Overall results for the final test

Result and Discussion

As a final step of evaluation, we have compared all the machine learning models. A data frame is constructed to compare the models' performance. The lists constructed to store the model's findings are the columns of this data frame. Below is the code snippet for comparing the models an accuracy result. The accuracy of the training and test datasets by the individual models.From our project we came to know that XGBoost ML model has the high accuracy compared to other model and the least accuracy is SVM. The XGBoost technique has 17the highest values in all the performance metrics used, indicating that it is the most robust of the complete algorithm, according to the experimental data. This could be due to the strategy used by the proposed model to avoid over fitting. Knowing that one of the most common problems with SVM, MLP, and Random forests is that they over fit for some datasets with poor classification objectives. The XGBOOST rows sub sampling, regularization term, shrinkage parameters, and are column sub sampling all approaches that XGBOOST uses to avoid over fitting. Auto encoder has the same issue in that it requires a lot of memory to store the structure and its execution is slow, but XGBOOST provides a lot of advantages over typical gradient boosting implementations. These are the main features of XGBoost to achieve more accuracy rate compared to other models.

CONCLUSION AND FUTURE WORK

Phishing has become a serious threat in online space, largely driven by evolving web, mobile, and social networking technologies. Due to the rapid spreading of new phishing websites and distributed phishing attacks, current phishing detection techniques need to be greatly enhanced to effec- tively combat emerging phishing attacks. In this paper, we have proposed a novel phishing detection method, leveraging multilevel artificial intelligence that uses a combination of neural network paired with a binary visualization. Using visual representation techniques allows to obtain an insight into the structural differences between legitimate and phishing web pages. From our initial experimental results, the method seems promising and being able to fast detection of phishing attacker with high accuracy. Moreover, the method learns from the misclassifications and improves its efficiency.

In the future, we plan to improve this work by the use of more samples for training and testing and utilising GPU for binary visualization and CNN classification, which will with no doubt enhance the predictive accuracy of the classifier. Furthermore, we intend to apply the proposed solution with more languages such as Russian, Greek and Chinese languages, and trained the system on 404-error HTML code.

A comparison of machine learning techniques for URL prediction is offered in this research. The major goal is to ensure security and prevent the user from gaining access to their sensitive data. It is possible to determine whether a website is legitimate or not using machine learning algorithms. With the comparison with other models in the research we found XGboost Classifier has a high accuracy by including 16 features. This project can be expanded upon by generating browser extensions and adding a graphical user interface. Using the current model, we can classify the Supplied URL as legitimate or phishing.

REFERENCES

[1] S. J. McMillan and M. Morrison, “Coming of age with the internet: A qualitative exploration of how the internet has become an integral part of young people’s lives,” New media & society, vol. 8, no. 1, pp. 73–95, 2006.

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

[2] K. Joshi. (2017) Mobile internet usage the 6 leading reasons that brought growth. [online] Tech Flix. Available: url = shorturl.at/ntRT4, [Accessed 29 May 2019].
[3] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list," EURASIP Journal on Information Security, vol. 2016, no. 1, p. 9, 2016.
[4] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," Journal of Applied Mathematics, vol. 2014, 2014.
[5] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," Journal of Information Security and applications, vol. 22, pp. 113–122, 2015.
[7] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.
[8] A. Alnajim and M. Munro, "An anti-phishing approach that uses training intervention for phishing websites detection," in 2009 Sixth International Conference on Information Technology: New Generations. IEEE, 2009, pp. 405–410.
[9] J. K. Keane, "Using the google safe browsing api from php," Mad Irish, Aug, vol. 7, 2009.
[10] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing at-tacks," in 2010 Proceedings IEEE INFOCOM. IEEE, 2010, pp. 1–5.
[11] J. Levine, "Dns blacklists and whitelists," Tech. Rep., 2010.
[12] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in Proceedings of the 4th ACM workshop on Digital identity management. ACM, 2008, pp. 51–60.
[13] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing black-lists," 2009.
[14] R. Alghamdi and K. Alfalqi, "A survey of topic modeling in text mining," Int. J. Adv. Comput. Sci. Appl.(IJACSA), vol. 6, no. 1, 2015.
[15] N. Toolbar, "Netcraft, ltd," 2009.
[16] Fortinet. Quarterly threat landscape report. [online] Fortinet. Available: url = shorturl.at/zRY29, [Accessed 1 Jun 2019].
[17] G. Xiang, B. A. Pendleton, and J. Hong, "Modeling content from human-verified blacklists for accurate zero- hour phish detection," CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCI- ENCE, Tech. Rep., 2009.
[18] EarthLimk. Earthlimk toolbar. [online] Fortinet. Avail- able: url = http://www.earthlink.net/, [Accessed 1 Jun 2019].
[19] A. K. Jain and B. B. Gupta, "Phishing detection: analysis of visual similarity based approaches," Security and Communication Networks, vol. 2017, 2017.
[20] S. Afroz and R. Greenstadt, "Phishzoo: Detecting phishing websites by looking at them," in 2011 IEEE Fifth In-ternational Conference on Semantic Computing. IEEE, 2011, pp. 368–375.
[21] P. A. Barraclough, M. A. Hossain, M. Tahir, G. Sexton, and N. Aslam, "Intelligent phishing detection and pro- tection scheme for online transactions," Expert Systems with Applications, vol. 40, no. 11, pp. 4697–4706, 2013.
[22] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "cantina+: A feature-rich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 2, p. 21, 2011.
[23] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in Proceedings of the 16th international

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.

conference on World Wide Web. ACM, 2007, pp. 649– 656.

[24] I. Baptista, S. Shiaeles, and N. Kolokotronis, "A novel malware detection system based on machine learning and binary visualization," arXiv preprint arXiv:1904.00859, 2019.

[25] Binvis. Binvis.io. [online] Binvis. Available: url = http://binvis.io/, [Accessed 1 Jun 2019].

Corresponding author: Abgoola Olayinka Taofeek
agboolato@jacks.sfasu.edu
Stephen F. Austin State University, Nacogdoches, Texas.