African Journal of Science, Technology, Innovation and Development, 2022
Vol. 14, No. 6, 1618–1626, https://doi.org/10.1080/20421338.2021.1977087
© 2021 The Authors

Routledge
Taylor & Francis Group

Machine learning approach for identifying suspicious uniform resource locators (URLs) on Reddit social network

Nureni Ayofe Azeez[1], Ahmed Oladapo Lawal[1], Sanjay Misra [2]* and Jonathan Oluranti[3]

[1]*Department of Computer Sciences, Faculty of Science, University of Lagos, Lagos, Nigeria*
[2]*Department of Computer Science and Communication, Ostfold University College, Halden, Norway*
[3]*Center of ICT/ICE Research, CUCRID, Covenant University, Ota, Ogun, Nigeria*
**Corresponding author email: ssopam@gmail.com*

The applications and advantages of the Internet for real-time information sharing can never be over-emphasized. These great benefits are too numerous to mention but they are being seriously hampered and made vulnerable due to phishing that is ravaging cyberspace. This development is, undoubtedly, frustrating the efforts of the Global Cyber Alliance – an agency with a singular purpose of reducing cyber risk. Consequently, various researchers have attempted to proffer solutions to phishing. These solutions are considered inefficient and unreliable as evident in the conflicting claims by the authors. Against this backdrop, this work has attempted to find the best approach to solving the challenge of identifying suspicious uniform resource locators (URLs) on Reddit social networks. In an effort to handle this challenge, attempts have been made to address two major problems. The first is how can the suspicious URLs be identified on Reddit social networks with machine learning techniques? And the second is how can internet users be safeguarded from unreliable and fake URLs on the Reddit social network? This work adopted six machine learning algorithms – AdaBoost, Gradient Boost, Random Forest, Linear SVM, Decision Tree, and Naïve Bayes Classifier – for training using features obtained from Reddit social network and for additional processing. A total sum of 532,403 posts were analyzed. At the end of the analysis, only 87,083 posts were considered suitable for training the models. After the experimentation, the best performing algorithm was AdaBoost with an accuracy level of 95.5% and a precision of 97.57%.

Keywords: Internet, machine learning algorithms, phishing, Reddit, uniform resource locators

Introduction

Phishing remains a threat to the cyber world to date. It is referred to as a dishonest way employed to obtain vital, classified, and sensitive information from a system, usually by pretending to be an authentic entity in a given system (Ramzan 2010; Abdelhamid, Thabtah, and Abdeljaber 2017). The information of interest to the phisher may include login details, credit card details, etc. A common method of performing such activity is by posting the URLs of such malicious websites on social networking platforms (Basnet and Sung 2011; Clement 2019a). Reddit, pronounced /ˈrɛdɪt/ is one of those online social networks; it is home to thousands of communities, millions of threads, and endless conversations (Clement 2019b). These threads' topics can vary from breaking news, TV fan theories, or sports. According to Arielle (2018), as of the end of the year 2018, Reddit had an online population of about 330 million monthly active users, making it a good platform for targeting people for phishing attacks. According to Mediakix (2017), it is the 5th most famous website in the United States of America. Among the industries affected, social network accounts for 3.1% and some of the industries can be traced to the information cybercriminals obtain from social network attacks.

According to Mediakix (2017), over 168 billion pages are explored on Reddit each year, coupled with the fact that over 50% of Reddit posts contains URLs, an avenue for phishing attacks has been created for cybercriminals as it will be hard for a user to be able to differentiate between legitimate and phishing URLs on such a website.

Research carried by Sheng et al. (2010) suggests that females are more vulnerable to phishing attacks than males, with ages ranging between 18 and 25. Given that the percentage of women on Reddit in the United States is 8%, that is approximately 26.4 million users, and the percentage of users within the age group of 18–29 is 22% for both males and females, that is approximately 72.6 million users. And that is the data that we were able to obtain from the Internet. The figures above indicate a large number of potential victims of phishing attacks.

Phishing attacks generally leverage on the vulnerabilities of human users, so for this reason, some additional support systems are needed to protect the systems and users. To this end, two main groups of protection mechanisms exist, namely, the user awareness approach and software detection (Sahingoz et al. 2019). The software-based protection or detection method is preferred since an attacker can target even professional users using new techniques. The software-based system can also serve as a decision support system for the user. Machine learning is a popular software-based technique used to detect malicious websites (Silaa, Jazri, and Muyingi 2021).

Other methods include Heuristics, Blacklist/Whitelist, Visual Reality/ content evaluation, and hybrid approaches (Islam and Chowdhury 2016; Jain and Gupta 2018a; Sahingoz et al. 2019). This idea is that detection of a phishing attack can be viewed as a simple classification problem. A learning-based detection system requires training data that contain many features related to phishing and non-phishing classes (Kamal and Manna 2018; Preethi and Velmayil 2016).

African Journal of Science, Technology, Innovation and Development is co-published by NISC Pty (Ltd) and Informa Limited (trading as Taylor & Francis Group)

This work attempts to adopt a machine learning approach for identifying suspicious URLs on Reddit social networks. To carry out this work, six machine learning algorithms were considered. The algorithms considered are: AdaBoost, Gradient Boost, Random Forest, Linear SVM, Decision Tree, and Naïve Bayes Classifier.

The rest of the paper is arranged as follows. Related works about phishing are discussed in the next section, while the section thereafter focuses on the methodology for detecting phishing attacks based on URLs in which the techniques employed as well as the performance evaluation metrics are described. The section that follows presents the results obtained from the numerous experiments carried out in the study, while the last section concludes the study with some recommendations on the future direction the work may be extended (Katuri and Gorantla 2021). The paper is structured as per guidelines provided by Misra (2020).

Related works

Several studies exist in the literature that have employed machine learning approaches for the phishing detection system. Some of the studies include the work done by Jeeva and Rajsingh 2016, Babagoli, Aghababa, and Solouk 2018, Buber, Diri, and Sahingoz 2017, Feng et al. 2018, Smadi, Aslam, and Zhang 2018, Jain and Gupta 2018b, and Peng, Harris, and Sawa 2018. Mohammad, Thabtah, and McCluskey (2012) carried out a study based on seventeen (17) strategies for recognizing phishing attacks. None of the adopted machine learning algorithms have never been adopted concurrently on the Reddit social network.

Jeeva and Rajsingh (2016) applied apriori and predictive apriori algorithms to generate rules from some of the URL features they defined. The method achieved fast detection with rules. However, the dataset used was limited to 200 legitimate URLs and 1200 phishing URLs. Babagoli, Aghababa, and Solouk (2018) also constructed a meta-heuristic-based nonlinear regression algorithm. Two feature selection methods were employed, namely decision and wrapper. The approach involved third-party services with about 20 features and a limited dataset of 11,055 total data of both phishing and legitimate webpages (Chiew et al. 2020).

Buber, Diri, and Sahingoz (2017) used natural language processing (NLP) to create some features which were then used along with three machine learning algorithms to classify some URLs. The approach achieved an improvement in the performance of about 7% over the previous study. Their study had a limited dataset of about 3717 malicious URLs and 3640 legitimate URLs.

Feng et al. (2018) also constructed a classification system based on Neural Network and Monte Carlo algorithms. The system did not depend on third-party services and was based on real-time detection. The approach also achieved an improvement in the accuracy and stability of detection. However, the approach required the download of the entire page as well as the use of third-party services.

Smadi, Aslam, and Zhang (2018) merged Reinforcement learning and Neural Network to develop a classification system for phishing email detection. The approach employed was able to detect phishing emails before the users found out. Also, the approach did not depend on third-party services and was based on real-time detection. However, the approach had a limited dataset for which 50% of the 9118 data were phishing.

Jain and Gupta (2018b) applied machine learning techniques to achieve client-side detection of phishing webpages. The approach did not depend on third-party services, but the entire URL page had to be downloaded to access the source code. Also, only a total of 19 features were available to classify the URLs.

Peng, Harris, and Sawa (2018) developed a phishing email detection system by combining NLP techniques and machine learning. NLP was used to detect the accuracy and suitability of each sentence. The approach was limited by the fact that the text of the emails had to be analyzed first. The dataset used was also limited, comprising of 5009 phishing emails and 5000 legitimate emails.

Rao and Pais (2018) employed (PCA) principal component analysis and Random Forest Classifier using image analysis and new heuristic features. The approach had language independence and high detection accuracy. However, only a limited dataset comprising 2119 phishing data and 1407 legitimate data was used.

Niakanlahiji, Chu, and Al-Shaer (2018) used PhishMon, a feature-rich machine learning model in detecting phishing URLs. The PhishMon makes use of a set of 15 features that are efficiently generated without the involvement of any third-party application like WHOIS servers or search engines. The advantage of this framework lies in its ability to capture various features of legitimate websites and their supporting web infrastructures. The simulation of these features requires a significant amount of time and effort. The dataset used contained 17,500 distinct benign webpages and 4800 distinct phishing. It obtained an accuracy of 95.4% with a false positive rate of 1.3%.

Muppavarapu, Rajendran, and Vasudevan (2018) used a combination of two approaches namely Resource Description Framework (RDF) models and ensemble learning methods for the purpose of classifying websites. A total of 20 different features were used with a dataset comprising of 1256 phishing webpages and 800 legitimate sites. It achieved a true positive rate of 98.8%, with a false positive rate of 1.5%. The accuracy obtained was 98.68%. The strength of the model lies in the fact that it has almost zero false negatives.

Vanhoenshoven et al. (2016) used binary classification for the detection of fraudulent URLs. The performance comparison of a number of classifiers such as Random Forest, Decision Trees, K-Nearest Neighbors, Multi-Layer Perceptron, Naïve Bayes, and Support Vector Machines, was carried out. The study employed a public dataset containing 2.4 million URLs and about 3.2 million features. The results from numerical simulations indicate a number of classification methods achieved an acceptable level of prediction even when feature selection had not been applied. In all, Random Forest and Multi-Layer Perceptron had the highest accuracy values.

Al-Janabi (2018) used multiple supervised machine learning classification models for the detection of suspicious URLs on online social networks. The social network used was Twitter. The machine learning algorithms used include Gradient Boosting Trees, Random Forest, XGBoost, and Extra Trees. A set of features were applied to detect Twitter posts that contain suspicious URLs. Some of the features applied included webpage content, Twitter metadata, domain WHOIS record, and URL lexical and redirection data. It was discovered in order to avoid over-fitting while optimizing the performance of the model, it was necessary to control the complexity parameters of the Random Forest cCassifier. This idea was discovered after the analysis of the hyper-parameters of tree-based models and the significance of parameters used such as the depth of trees, the minimum size of leaf nodes on classification performance, and a number of trees were examined. Their major shared advantage was in the fact that they both were statistically better than the highest singular model. An online suspicious URL detection system 'SuspectRate' was built based on the research.

Sananse and Sarode (2015) demonstrated the use of two machine learning algorithms namely: Random Forest and Content-based algorithm in tackling the problem of phishing URL detection. They collated their phishing URL dataset from PhishTank (A community-based phish confirmation system on the Internet). Non-phishing URLs were collected from various credible sources. They grouped them into training and testing categories for training the models. For feature extraction, they extracted 24 lexical features, 48 WHOIS features, Alexa Rank, PhishTank-based, and PageRank features were extracted. The model was trained using 500 URLs and tested using 100 URLs. They applied web mining heuristics on the Random Forest algorithm which yielded a precision of more than 90% with False Negative Result and False Positive Result rates of less than 1%. The Precision obtained in the case of a Content-based algorithm was less than 65%. They concluded that for future works, there was still a need to work on a selection of more efficient features for the Content-based algorithm to increase the Precision and decrease the with False Negative Result and False Positive Result rates. The webpage content-based features can be integrated to make the system more robust.

Ali (2017) compared K-Nearest Neighbors (KNN), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Back-Propagation Neural Networks (BPNN), Radial basis function network (RBFN), and C4.5 algorithms for the detection of phishing URLs. The dataset used comprised 4898 illegitimate/phishing URLs and 6157 legitimate URLs used for the training and testing of the machine learning models. The source of the dataset used the UCI Machine Learning Repository. Five-fold cross-validation in conjunction with wrapper-based feature selection was used to evaluate the performances of the classifiers for the detecting of phishing URLs. The performance of the machine learning algorithms was carried after applying Information Gain and Principal Component Analysis. The results indicate that KNN, BPNN, and RF achieved the best Correct Classification Rate (CCR) while NB and RBFN had the lowest CCR for detecting phishing websites. Based on the statistical analysis, it was evident that the improvement of the CCRs of the machine learning classifiers was achieved by applying the wrapper-based features selection with KNN, C4.5, NB, and RF having the least CCR improvement. The analysis also demonstrated that the wrapper-based features selection method had a higher effect on machine learning models than PCA and IG methods.

In the work of Zhao et al., in 2021, they considered the prediction of cyberattacks as an essential approach for achieving efficient and stable cyberspace security. Doing this, they believed, will demystify any form of attack and proactively handle oncoming cyber threats. To achieve this, they proposed a framework named HinAp to predict cyberattack preference using attributed heterogeneous attention networks and transductive learning. They finally constructed an automated model for predicting cyberattack preferences. Experimental results based on real-world data prove that HinAp has an accuracy of 89.12%.

Table 1 provides a summary of all the reviewed articles.

Having noticed some of the challenges in a couple of the existing literature, the authors, therefore, deemed it necessary to carry out this work with the sole aim of addressing the research questions hereunder. To handle this challenge, two main research questions were raised: how can the suspicious URLs be identified on Reddit social networks by using machine learning techniques? And how can the internet users be safeguarded from unreliable URLs on Reddit social networks? This paper uses the machine learning technique to identify fake and suspicious URLs on Reddit social network.

Methodology

The methodology adopted comprises four phases, namely, a collection of data, extraction of features, training of the model, and evaluation of performance. These are described in the sections that follow.

Data collection and labelling

This phase describes how the data was collected and classified the posts into safe and malicious posts. The data used in this project is obtained from a public real-time stream of Reddit posts using a python library called 'praw'. Requests were made during a course of two months at different time intervals to achieve some level of randomization. Each result fetched was stored in a csv file. A total of 532,403 posts were obtained during the data collection.

Figure 1 presents the data collection process of the training dataset. The architecture is very simply to understand.

The machine learning models to be built for this project required a supervised machine learning approach, which means that the dataset used in training the models should be classified into legitimate URLs and phishing URLs.

In order to achieve this, the URLs in the dataset had to be tested for their validity on VirusTotal.com. Based on the status of each URL, the posts are stored in two different tables, one for posts with legitimate URLs and the other for posts with malicious URLs. The Microsoft

Table 1: Summary of the reviewed articles.

Author	Year	Approach	Strength	Weakness
Jeeva and Rajsingh	2016	Applied apriori and predictive apriori generation of algorithms	It detects Phishing and legitimate webpages	20 features and a limited dataset
Buber et al.	2017	Natural Language Processing (NLP) with three machine learning algorithms	7% performance improvement over the previous study	limited dataset
Feng et al.	2018	Neural Network and Monte Carlo algorithms	Improved accuracy and stability	The use of third-party services.
Smadi et al.	2018	Neural Network and Reinforcement learning	Independent of third party services	limited dataset
Jain and Gupta	2018	Machine-learning techniques	The approach did not depend on third-party services	A few features were considered
Peng et al.	2018	NLP techniques and machine learning	Relatively effective for detection	Limited amount of dataset
Rao and Pais	2018	Principal component analysis (PCA) and Random Forest Classifier	Language independence and high detection accuracy	Limited amount of dataset
Niakanlahiji et al.	2018	PhishMon – a feature-rich machine learning model	Ability to capture various features of legitimate websites	A few features were considered
Muppavarapu et al.	2018	Resource Description Framework (RDF) models and ensemble learning	It has almost zero false negatives	No real-life experimentation
Vanhoenshoven et al.	2016	Random Forest, Decision Trees, K-Nearest Neighbors, Multi-Layer Perceptron, Naïve Bayes and Support Vector Machines	It achieved an acceptable level of prediction	Lack of provision of public code repository link
Al-Janabi	2018	Gradient boosting trees, random forest, XGBoost, and extra trees	Statistically better than the highest singular model	A few features were considered
Sananse and Sarode	2015	Random Forest and Content-based algorithm	A precision of more than 90%	Limited and inefficient features
Ali	2017	Random Forest (RF), Naïve Bayes (NB), Back-Propagation Neural Networks (BPNN), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Radial basis function network (RBFN), and C4.5 algorithms	A relatively better performance	There is no novelty in the work

SQL Server database was used to store the datasets. For further scrutiny, the URLs in the table containing legitimate posts were checked to confirm their availability on Reddit. These posts were then moved to the table containing malicious posts. This process was done multiple times during the process of collecting data due to the fact that the malicious posts are not always detected early. At the end of the procedure, a total of 532,403 samples were obtained for both classes. After classification, there were 81,820 samples of legitimate posts and 5263 samples of malicious posts. To tackle the issue of having imbalanced classes of data, a 4:1 ratio was used for both classes as that was a universally agreed ratio for class balancing. To achieve the 4:1 ratio, the dataset of legitimate posts was under-sampled. A high-speed 4 g internet-enabled device and an Intel Core i5 8GB RAM were used to collate the samples in the datasets. The link to the code repository is inserted in Footnote 1.[1]

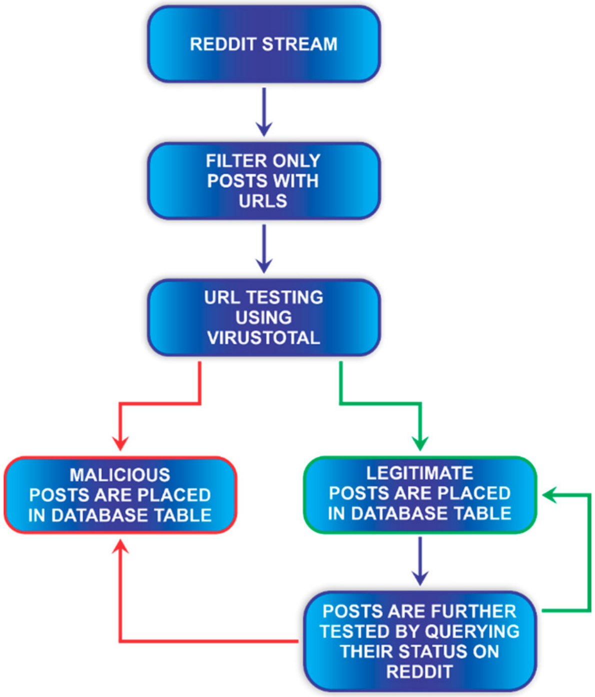
REDDIT STREAM
FILTER ONLY POSTS WITH URLS
URL TESTING USING VIRUSTOTAL
MALICIOUS POSTS ARE PLACED IN DATABASE TABLE
LEGITIMATE POSTS ARE PLACED IN DATABASE TABLE
POSTS ARE FURTHER TESTED BY QUERYING THEIR STATUS ON REDDIT

Figure 1: Data collection process of the training dataset.

In order to provide a professional way of sharing code repository and assist in reproducibility, GitHub was adopted.

Extraction of feature and engineering

In machine learning, features refer to the characteristics of a particular entity that can be used to differentiate it from another entity. Feature extraction in machine learning is the process of obtaining those characteristics from the entity. These features can be obtained from sources. For the purpose of this project, the feature sources are from the Reddit API stream and features that can be extracted through extra processing processes. A total of 12 features are being used. The features obtained from Reddit require no additional processing, unlike the features in Table 2. The proposed feature sources are given in Table 3.

Model training/machine learning

The problem of phishing URL detection is a type of classification problem and, therefore, the three machine learning algorithms that are being used in building the models are suitable for classification problems. A classification problem deals with the task of differentiating one entity from another. It always has its final result in the form of a discrete value as opposed to a regression problem. The algorithms used are AdaBoost, Gradient Boost, Random Forest, Linear SVM, Decision Tree, and Naïve Bayes Classifier.

Random Forest

This is a classification and regression machine learning algorithm. When used for classification purposes, it operates by building multiple decision trees and selecting the model class as a result. It is one of the most accurate machine learning algorithms available as it can even function with high accuracy in situations where a large chunk of data is missing. It performs with high efficiency on large databases. The Random Forest algorithm is based on the entropy values with respect to samples and classes. The major drawback of the Random Forest is its complex nature, which requires more computational resources due to many decision trees compared to other algorithms (Azeez and Vyver 2019).

Gradient Boost

This is a classification and regression machine learning algorithm. The prediction models produced by this method are in the form of an ensemble of decision trees. The models are generated in a stage-wise manner before they are generalized. The idea of gradient boosting is to build an optimization algorithm that makes use of an appropriate cost function.

Table 2: Reddit feature source.

Feature	Feature source
Signs of User name	User's Information
Length of User name	User's Information
Age of Account	User's Information
Digits of User name (number)	User's Information
Post Title	Post Content

Table 3: Preprocessed feature source.

Feature	Feature source
Age of Domain	Domain WHOIS Info
Whether it is a secured https?	User Info
Length of the Link	Post Content
Link letters (number)	URL Information
Number of dots in link	URL Information
Number of link signs	URL Information
Number of digits in link	URL Information

Likewise, with other algorithms, gradient boosting merges weak 'learners' with a single but strong learner in an iterative manner. It is easiest to explain in the least-squares regression setting, where the objective is to 'teach' a model F to foresee values of the form $\widehat{y} = F(x)$ by ensuring that the mean squared error is reduced $\frac{1}{n}\sum_i (\widehat{y}_i - y_i)^2$ where i indexes over some training set of n size of actual values of the output variable y:

- $\widehat{y}_i$ = the predicted value $F(x)$
- y_i = the observed value
- n = the number of samples in y

Assuming a gradient boosting algorithm has an M stage. At each stage m $(1 \le m \le M)$ of gradient boosting, if some deficient model F_m (for low m, the model might return $\widehat{y}_i = \bar{y}$, where RHS is the mean of y)

AdaBoost

This stands for adaptive boosting. This algorithm can work in conjunction with other machine learning algorithms for performance improvement. AdaBoost is said to be adaptive since it allows learning algorithms that are weak to be adjusted to favour those instances that were previously misclassified. AdaBoost is also sensitive to noise and outliers in data. It is less prone to overfitting compared to other learning algorithms. A strong learner is produced since the performance of the weak learning algorithms will eventually converge. AdaBoost allows the capture of many of the nonlinear relationships that occur during training, translating into better prediction accuracy on the problem of interest.

A boosted classifier is a classifier in the form:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \tag{1}$$

where each f_t is a weak learner that takes an object x as input and returns a value indicating the class of the object.

Each weak learner produces an output hypothesis,-$f(x_i)$, for each sample in the training set. At each iteration-t a weak learner is selected and assigned a coefficient α_t such that the sum training error E_t of the resulting t-stage boost classifier is minimized.

$$\sum E_t = \sum_i E[F_{t-1}(x_i) + \alpha_i h(x_i)] \tag{2}$$

Here $F_{t-1}(x)$ is the boosted classifier that has been built up to the previous stage of training, $E(F)$ is some error function and $f_t(x) = \alpha_t h(x)$ is the weak learner that is being considered for addition to the final classifier.

Decision Tree Classifier

This is a machine learning technique that is being used for solving application-related challenges, specifically for problem classification. It further provides a nonparametric approach for carrying out partitioning of datasets.

This algorithm gets as input, say a table X and allows the partition to be recursively performed on the table into different tables. With partitioning, this algorithm improves a clean score of the column that is labelled in each of the partitions. The purity score is a technique developed on the proportion of the classes of individual where there is a variety of class labels. As the proportion of one of the classes increases, the purer and more reliable the collection is. It has the capability of converting very large complex datasets into a comprehensive and graphical display information.

With this algorithm, the entropy value for the data under consideration can be determined after which the information gain for an attribute in a dataset can be calculated. In addition, an excellent performance on a very large datasets is guaranteed (Azeez et al. 2021).

The entropy of a given information source x, $H(x)$ is defined as follows:

$$H(x) = \sum_{xeX} p(x) \log p\ (x) \tag{3}$$

where $p(x)$ is the probability of occurrence of x (Azeez et al. 2021).

Linear Support Vector Classifier

The classifier assists in fitting the data as the best fit hyperplane that divides data into various categories. After getting the hyperplane, the classifier is supplied with some attributes to visualize the class that have already been predicted (Azeez et al., 2019a).

Naïve Bayes Classifier

This classifier is mainly used to carry out prediction of the possibility that a particular event will happen with the aid of evidence that is available in the data. The adoption of a multinomial Naïve Bayes algorithm classifier was because of its suitability and efficiency for features that explain discrete frequency counts which is comparable to the various features of the data that are available in the dataset obtained.

Given a class of variables or hypothesis (y) and a dependent feature or evidence ($x1$, ..., xn)

Therefore,

$$P(y|x1, x2, x3 \ldots .x\text{n}) = \frac{P(y)P(x1, x, x3, \ \ldots\ldots xn|y)}{\text{P}(x1, x, x3, \ \ldots\ldots .x\text{n})} \tag{4}$$

where: $P(y)$ are labels $P(x)$ and are comments

$P(y|x1, x2, x3 \ldots .xn)$ is the likelihood of the hypothesis (labels) given the observed evidence.

$P(x1, x2, x3 \ldots .xn|y)$ is the likelihood of the evidence, given that the hypothesis is true.

$P(y)$ is the likelihood of the hypothesis before observing the evidence.

$P(x1, x2, x3 \ldots\ldots xn)$ is the likelihood of the hypothesis considering the new evidence under all possible hypotheses.

Random Forest

Random Forest is a very dynamic, convenient to use algorithm that produces a reliable, result without the usage of hyper-parameter turning. Due to its diversity and simplicity, it is considered as one of the widely used algorithms. It can be applied for both regression and classification tasks. The 'forest' it develops is considered an ensemble of the entire decision trees majorly trained with the bagging approach. There is an increase in the overall result with bagging approach because of the combination of various learning models.

With this algorithm, over-fitting trees into the model is not allowed, specifically when there more trees.

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}} \tag{5}$$

where RF *fi* sub (i) represents the importance of feature i determined from all trees in the random forest model. Fi sub (i,j) is the significance of feature where node of i and j. i and j are the nodes (Azeez et al. 2019b).

Performance evaluation

The models used are evaluated using the following properties:

True Positives (TP) – these are entities that are presumed to be positive and are indeed positive

False Positives (FP) – these are entities that are presumed to be positive and are actually negative

True Negative (TN) – these are entities that are presumed to be negative and are indeed negative

False Negative (FN) – these are entities that are presumed to be negative and are actually positive (Azeez et al. 2021).

Accuracy:

$$\frac{(\text{TP}) + (\text{TN})}{(\text{TP}) + (\text{FP}) + (\text{TN}) + (\text{FN})} \tag{6}$$

Precision – This the fraction of correct positive or negative predictions out of the total predicted positive or negative instances. The denominator could be positive or negative.

$$\frac{(\text{TN})}{(\text{TN}) + (\text{FN})} \tag{7}$$

Recall – This is the fraction of positive predictions out of the total actual positive entities. In the formula, (TP) + (FN) is the actual sum of positive entities in the dataset. It is also known as the True Positive Rate.

$$\frac{(\text{TP})}{(\text{TP}) + (\text{FN})} \tag{8}$$

Specificity – This is the fraction of negative predictions out of the total actual negative entities. It is the opposite of Recall.

$$\frac{(\text{TN})}{(\text{TN}) + (\text{FP})} \tag{9}$$

False Positive Rate (FPR) – This is the fraction of all negative predictions that still yield positive test outcomes.

$$\frac{(\mathrm{FP})}{(\mathrm{TN}) + (\mathrm{FP})} \tag{10}$$

To compute F-score, the following equation was used:

$$F - score = 2\frac{Precision\ \times Recall}{Precision + Recall} \tag{11}$$

The Mathews Correlation Coefficient (MCC) is calculated as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)\ (TP + FN)(TN + FP)\ TN + FN)}} \tag{12}$$

Results

The proposed methodology is built upon a set of 12 features obtained by processing the contents of the posts in the dataset comprising of 81,820 ham posts and 5263 phish posts.

At the end of the implementation, the following explanation summarizes the feedbacks obtained when Random Forest was considered. A 90% accuracy was obtained. This implies that Random Forest can identify suspicious URLs on Reddit social network with 90% accuracy. This result is further provided with a Precision of 0.95, Recall of 0.93, F-Score of 0.94 and a False Positive Rate of 0.35. The corresponding values obtained for each of the performance metrics used – True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) are 40,213, 2083, 3821 and 2809, respectively. The summary of this result is provided in Table 4. The results obtained on other algorithms are completely different.

There is a significant improvement in the result obtained when the Gradient Boot Cassifier was considered. The accuracy is 91% while Precision, Recall, F-Score, and False Positive Rate are 0.97, 0.93, 095, and 0.33, respectively. This classifier is better in terms of performance when compared with a Random Forest Classifier. With the dataset used, the corresponding values for each of the performance metrics used are presented in Table 5.

AdaBoost is undoubtedly the best of all the six classifiers used in this work. It provides an accuracy of 95%. Table 6 provides Precision, Recall, F-Score, and False Positive Rate of 0.98, 0.97, 0.97, and 0.17, respectively. The corresponding values of the metrics are also presented in Table 6.

Table 7 provides a summary of the results obtained for all six algorithms.

Figure 2 shows a graphical representation of a comparison of the six models with the metrics used.

Conclusion and future works

The act of phishing has been severely exploited by cybercriminals over the years since the advent of online social network platforms. It has cost some users of social

Table 4: Results from Random Forest Classifier.

Random Forest	Precision	Recall	F-Score	False Positive Rate
Accuracy = 90 %	0.95	0.93	0.94	0.35
	Performance Metric		Value	
	True Positive (TP)		40,213	
	False Positive (FP)		2083	
	True Negative (TN)		3821	
	False Negative (FN)		2809	

Table 5: Results from Gradient Boot Classifier.

Gradient Boost	Precision	Recall	F-Score	False Positive Rate
Accuracy = 91%	0.97	0.93	0.95	0.33
	Performance Metric		Value	
	True Positive (TP)		39,022	
	False Positive (FP)		1290	
	True Negative (TN)		2572	
	False Negative (FN)		2782	

Table 6: Results from AdaBoost.

AdaBoost	Precision	Recall	F-Score	False Positive Rate
Accuracy = 95%	0.98	0.97	0.97	0.17
	Performance Metric		Value	
	True Positive (TP)		47,217	
	False Positive (FP)		1172	
	True Negative (TN)		5673	
	False Negative (FN)		1318	

Table 7: Comparison of AdaBoost, Gradient Boost, Random Forest, Linear SVM, Decision Tree, and Naïve Bayes Classifier models.

	False Positive Rate	F-Score	Precision	Recall	Accuracy	Spec	AUC	MCC
AdaBoost	0.17122	0.974312	0.97578	0.972844	0.955038	0.988	0.989	0.977
Gradient Boost	0.33402	0.950725	0.96800	0.933451	0.910831	0.923	0.945	0.967
Random Forest	0.33001	0.94020	0.95030	0.93002	0.90030	0.924	0.955	0.956
Linear SVM	0.45001	0.93005	0.95400	0.95401	0.94300	0.930	0.965	0.946
Decision Tree	0.32002	0.94060	0.95504	0.95703	0.94700	0.934	0.966	0.936
Naïve Bayes Classifier	0.37010	0.96010	0.95806	0.95504	0.94200	0.926	0.967	0.962

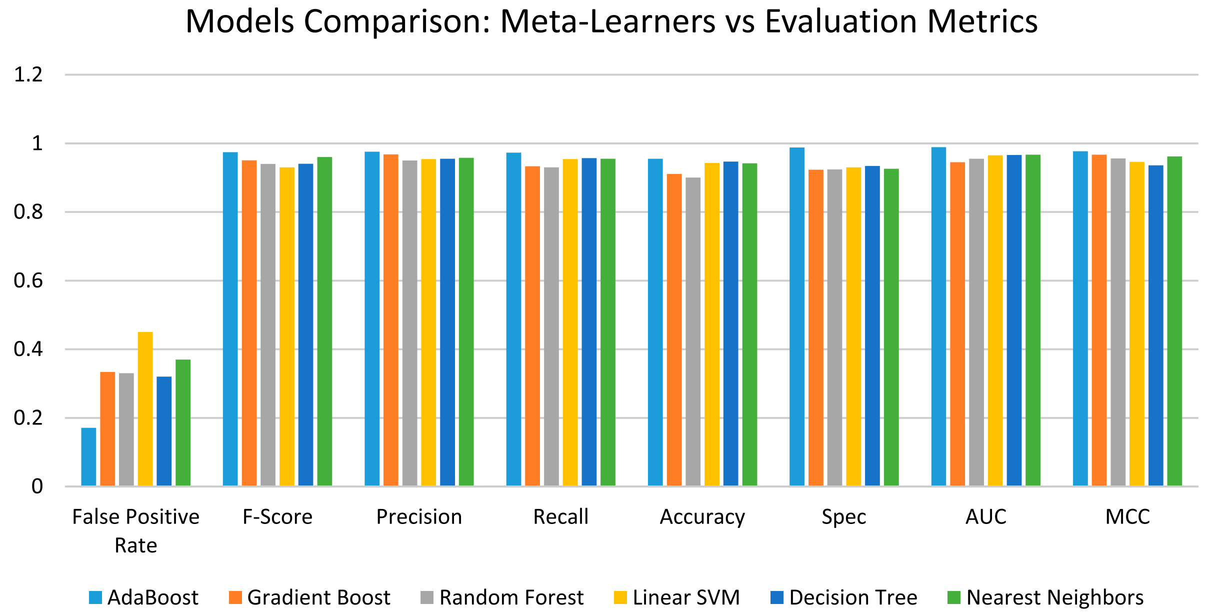

Figure 2: Graphical representation of a comparison of six models with the metrics.

networks a lot of money and resources. Individuals and even organizations are no longer safe when on such platforms. Reddit is a community-based social network platform; it is composed of multiple threads and sub-threads which have their own sub-threads and so on. It can be complex to detect phishing activities without the help of machines. That is where machine learning comes in – to use machines to detect such malicious activities. Given that supervised learning was the methodology used, the models were tested with ham and phish URLs and their performance metrics were taken and recorded. The following advanced work is currently being proposed for future work: automatically detect phishing activities from Reddit on a real-time basis. An attempt is being made to use Deep Learning with the ensemble for the detection of suspicious URLs on the Reddit social network.

Limitations and constraints

While Reddit provides a good stream of live posts, it was difficult to obtain this data in other languages aside from Python. A better API for testing URL validity would also come in handy in speeding up the validation process.

Disclosure statement

No potential conflict of interest was reported by the authors.

Note

1. https://github.com/soldierlytomcat/RedditCrawler

ORCID iD

Sanjay Misra http://orcid.org/0000-0002-3556-9331

References

Abdelhamid, N., F. Thabtah, and H. Abdel-jaber. 2017. “Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features.” 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), 2017: 72–77. doi:10.1109/ISI.2017.8004877

Ali, W. 2017. “Phishing Website Detection Based on Supervised Machine Learning with Wrapper Features Selection.” *(IJACSA) International Journal of Advanced Computer Science and Applications* 8 (9): 72–78.

Al-Janabi, M. F. 2018. *Detection of Suspicious URLs in Online Social Networks Using Supervised Machine Learning Algorithms*. Sydney: Keele University.

Arielle, P. 2018. “The Inside Story of Reddit’s Redesign.” *Wired*, February 4. https://www.wired.com/story/reddit-redesign/.

Azeez, N. A., T. J. Ayemobola, S. Misra, R. Maskeliūnas, and R. Damaševičius. 2019a. “Network Intrusion Detection with a Hashing Based Apriori Algorithm Using Hadoop MapReduce.” *Computers* 8 (4): 86.

Azeez, N. A., O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius. 2021. “Windows PE Malware Detection Using Ensemble Learning.” *Informatics* 2021 (8): 10. doi:10.3390/informatics8010010.

Azeez, N. A., B. B. Salaudeen, S. Misra, R. Damasevicius, and R. Maskeliunas. 2019b. “Identifying Phishing Attacks in Communication Networks using URL Consistency Features.” *International Journal of Electronic Security and Digital Forensics* (InderScience). https://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijesdf.

Azeez, N. A., and C. V. Vyver. 2019. “Verifying Literal and Conceptual Consistency for Anti-Phishing.” IST-Africa 2019 In *Proceedings of Conference Nairobi, Kenya, 08-10 May 2019*, edited by Paul Cunningham and Miriam Cunningham. IIMC International Information Management Corporation.

Babagoli, M., M. P. Aghababa, and V. Solouk. 2018. “Heuristic Nonlinear Regression Strategy for Detecting Phishing Websites.” *Soft Computing* 3: 1–13.

Basnet, R., and A. Sung. 2011. “Learning to Detect Phishing Webpages.” *Journal of Internet Services and Information Security* 4 (3): 21–39.

Buber, E., B. Diri, and O. K. Sahingoz. 2017. “NLP Based Phishing Attack Detection from URLs.” In *Intelligent Systems Design and Applications*, edited by A. Abraham, P. K. Muhuri, A. K. Muda, and N. Gandhi, 608–618. Cham: Springer International PUBLISHING.

Chiew, K. L., C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong. 2020. “A new Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System.” *Information Sciences* 484 (2019): 153–166.

Clement, J. 2019a. “Online Industries Most Targeted by Phishing Attacks as of 4th Quarter 2018.” Accessed from Statista https://www.statista.com/statistics/266161/websites-most-affected-by-phishing/.

Clement, J. 2019b. “Percentage of U.S. Adults Who Use Reddit as of February 2019, by Age Group.” Accessed from Statista https://www.statista.com/statistics/261766/share-of-us-internet-users-who-use-reddit-by-age-group/.

Feng, F., Q. Zhou, Z. Shen, X. Yang, L. Han, and J. Wang. 2018. “The Application of a Novel Neural Network in the Detection of Phishing Websites.” *Journal of Ambient Intelligence and Humanized Computing* 9: 2009–2010.

Islam, M., and N. K. Chowdhury. 2016. “Phishing Websites Detection Using Machine Learning Based Classification Techniques.” International Conference on Advances in Informatics and Communication Technologies (63).

Jain, A. K., and B. B. Gupta. 2018a. “A Machine Learning Based Approach for Phishing Detection Using Hyperlinks Information.” *Journal of Ambient Intelligence and Humanized Computing* 10: 2015–2028.

Jain, A. K., and B. B. Gupta. 2018b. “Towards Detection of Phishing Websites on Client–Side Using Machine Learning Based Approach.” *Telecommunication Systems* 68 (4): 687–700.

Jeeva, S. C., and E. B. Rajsingh. 2016. “Intelligent Phishing URL Detection Using Association Rule Mining.” *Human-Centric Computing and Information Sciences* 6 (10). doi:10.1186/s13673-016-0064-3.

Kamal, G., and M. Manna. 2018. “Detection of Phishing Websites Using Naïve Bayes Algorithms.” *International Journal of Recent Research and Review* XI (4): 34–38.

Katuri, R., and S. Gorantla. 2021. “Math Function-Based Controller Combined with PI and PID Applied to Ultracapacitor Based Solar-Powered Electric Vehicle.” *African Journal of Science, Technology, Innovation and Development* 13 (4): 509–526. doi:10.1080/20421338.2020.1857542.

Mediakix. 2017. “The Top 8 Reddit Statistics on Users, Demographics & More.” *Mediakix*. https://mediakix.com/blog/reddit-statistics-users-demographics/.

Misra, S. 2020. “A Step-by-Step Guide for Choosing Project Topics and Writing Research Papers in ICT Related Disciplines.” In *International Conference on Information and Communication Technology and Applications*, edited by S. Misra, and B. Muhammad-Bello, 727–744. Cham: Springer.

Mohammad, R.M., F. Thabtah, and L. McCluskey. 2012. “An Assessment of Features Related to Phishing Websites Using an Automated Technique.” *Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions*, 492–497. London, UK, 10–12 December 2012.

Muppavarapu, V., A. Rajendran, and S. Vasudevan. 2018. “Phishing Detection Using RDF and Random Forests.” *The International Arab Journal of Information Technology* 15 (55): 817–824.

Niakanlahiji, A., B.-T. Chu, and E. Al-Shaer. 2018. PhishMon: A Machine Learning Framework for Detecting Phishing Webpages. 978-1-5386-7848-0/18, 6.

Peng, T., I. Harris, and Y. Sawa. 2018. “Detecting Phishing Attacks Using Natural Language Processing and Machine Learning.” In *IEEE 12th International Conference on Semantic Computing (ICSC)*, edited by D. Bulterman, A. Kitazawa, D. Ostrowski, and P. Sheu, 300–301. Laguna Hills, CA: IEEE Computer Society.

Preethi, V., and G. Velmayil. 2016. “Automated Phishing Website Detection Using URL Features and Machine Learning Technique.” *International Journal of Engineering and Techniques* 2 (5): 107–115.

Ramzan, Z. 2010. “Phishing Attacks and Countermeasures (Vols. ISBN 978-3-642-04117-4).” In *Handbook of Notes Information and Communication Security*, edited by Mark Stamp, and Peter Stavroulakis, 433–448. Berlin, Heidelberg: Springer.

Rao, R. S., and A. R. Pais. 2018. “Detection of Phishing Websites Using an Efficient Feature-Based Machine Learning Framework.” *Neural Computing and Applications* 31: 3851–3873.

Sahingoz, K., E. Buber, O. Demir, and B. Diri. 2019. “Machine Learning Based Phishing Detection from URLs.” *Expert Systems with Applications* 117: 345–357.

Sananse, B. E., and T. K. Sarode. 2015. “Phishing URL Detection: A Machine Learning and Web Mining-Based Approach.” *International Journal of Computer Applications* 123 (13): 46–50.

Sheng, S., M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs. 2010. “Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions.” In *CHI ‘10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, edited by E. Mynatt, G. Fitzpatrick, S. Hudson, K. Edwards, and T. Rodden, 373–382. Atlanta Georgia: Association for Computing Machinery.

Silaa, J., H. Jazri, and H. Muyingi. 2021. “A Study on the use of Mobile Computing Technologies for Improving the Mobility of Windhoek Residents.” *African Journal of Science, Technology, Innovation and Development* 13 (4): 479–493. doi:10.1080/20421338.2020.1838083.

Smadi, S., N. Aslam, and L. Zhang. 2018. “Detection of Online Phishing Email Using Dynamic Evolving Neural Network Based on Reinforcement Learning.” *Decision Support Systems* 107: 88–102.

Vanhoenshoven, F., G. Nápoles, R. Falcon, K. Vanhoof, and M. Köppen. 2016. “Detecting Malicious URLs Using Machine Learning Techniques.” In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 8. Athens, Greece: IEEE. doi:10.1109/SSCI.2016.7850079.

Zhao, Z., R. Ye, C. Zhou, D. Wang, and T. Shi. 2021. “Control-theory Based Security Control of Cyber-physical Power System Under Multiple Cyber-attacks Within Unified Model Framework.” *Cognitive Robotics* 1: 41–57. doi:10.1016/j.cogr.2021.05.001.