Comparative Analysis of the Performance of Selected Learning Algorithms for Verification of vulnerable and Compromised Uniform Resource Locators (URLs)

N.A. Azeez¹*, A.D. Ajayi, ¹, C.O. Yinka-Banjo¹

¹Department of Computer Sciences, Faculty of Science, University of Lagos, Nigeria

Received: 8th June, 2017. Accepted: 23rd October, 2017. Published: 5th May, 2018.

Abstract

The fact that cybercriminals have caused serious havoc and unprecedented financial loss through internet activities is well acknowledged by internet users across the globe. Different nefarious activities of the internet fraudsters have undoubtedly resulted in monumental loss of life and property of immeasurable values. From available literatures, many people have become victims of their handiworks by giving feedback to fake and phony Uniform Resource Locators (URLs) sent to their electronic mails. In the recent works by researchers in the area of cybersecurity, it has been established that machine learning approaches have been proposed to identify various compromised and fake URLs in order to safeguard internet users from becoming victim. Consequently, discrepancies noted in some the available results give room for doubt and reliability of the results obtained in their experimentations. In an attempt, however to protect internet users from experiencing further loss and to establish the performances of these algorithms, the authors carried out a comparative analysis of three learning algorithms (Naive Bayes, Decision Tree and Logistics Regression Model) for verification of compromised, phony and fake URLs and determined which is the best of all the three based on the metrics (F-Measure, Precision and Recall) used for evaluation. After the experimentation, it was finally observed that the decision tree provides optimal and efficient solution of all the tree algorithms with full and absolute F-Measure when 0.6 is considered as boundary. With optimal solution provided by the decision tree, internet users can be given reliable information and consequently be guarded against further attacks.

Keywords: Algorithms, Vulnerable URL, Compromised URL, performance

*Corresponding author: E-mail: nazeez@unilag.edu.ng
1 Introduction

In this new age, the fastest and the easiest way of sharing data, information and files is the World Wide Web (WWW). The www is used by hackers to send malicious attack in form of pharming, phishing, e-mail spoofing and malware infection to people. Phishing is a social engineering technique whereby electronic mail appears like a legitimate one coming from a renowned and reliable source. The fraudster will eventually use the information obtained in the process to commit atrocities on behalf the legitimate owner (Azeez & Venter, 2013).

According to the "Anti-Phishing Working Group (APWG) the rate of phishing activity report for the 4th quarter of 2016 indicates that the total number of unique phishing websites detected was 277,693 while the total number of unique phishing e-mail reports received by APWG from consumers was 211,032. It was estimated that 70% of Internet users have received phishing e-mails, out of which approximately 15%" has provided their personal information which was subsequently used for fraudulent activities (APWG, 2016).

A report released by Cloud mark in 2014 indicated how internet fraudsters used Twilio to broadcast over 385,000 spam messages through fake urls. What is more? A media report was published by National Fraud Intelligence Bureau (NFIB) about the latest scams which was analyzed in 2016 by action fraud (Choudhary & Jain, 2017). Spammers are directing their nefarious activities to bank customers and many financial institutions by sending fake urls to them requesting their bank details such as password, ATM pin number. If however, such a customer assumes the message is from the authentic source he will be a victim.

Aside from the above, in 2016, Symantec Internet Security presented a report that elaborates various global threats pose by sending fake and compromised urls to include corporate data breaches, various attacks on websites and browsers, corporate data breaches and other forms of fraudulent cyber behaviors. One of the approaches being used by cybercriminals is baiting the internet user to intentionally click on a compromised and fake url in order to achieve any of the objectives stated previously (Symantec, 2016).

Blacklisting services has been developed by web security (Nureni & Irwin, 2010) community and researchers to specifically identify these fake, vulnerable and compromised url and malicious websites (Azeez & Ademolu, 2016). These so called blacklists are developed by numerous approaches such as honeypots, manual reporting as well as web crawlers with
website analysis heuristics. It is unambiguous that blacklisting of url has been very helpful and effective to certain extent, it is however easy for cybercriminals to cajole and even deceive the system by modifying features of the url string. Unavoidably, some fake and compromised sites are not blacklisted because they are new.

Many researches in the past have handled this challenge from a Machine Learning point of view. They compiled a selected list of urls that have been categorized as either legitimate or malicious and thereafter characterized each of the urls through a set of specified attributes (Rokach & Maimon, 2008). Machine Learning algorithms are then applied to train and learn the boundary among various decision strata and classes. This work addresses the verification and detection of compromised and vulnerable urls as a classification problem and examines the performance of three popular classifiers, namely Naïve Bayes, Decision Trees and Linear Regression (Azeez & Iliyas, 2016). Finally, the results obtained were properly studied and compared using recall, precision and f-measure as metrics. This work is intended to meet the following objectives:

1. To identify the host-based features and lexical features of a malicious URL
2. To design a system to detect suspicious links in e-mails and notify users in order to protect them from falling for phishing attacks
3. To determine which of the algorithms is the best suitable for determining compromised and fake URLs by using standard metrics for measuring the performance of the learning algorithm (Naïve Bayes, Decision Tree and Logistics Regression Model) to benchmark the performance of the algorithms.

The rest of the paper is organized as follows: First of all, section 2 discusses the set of related work in the subject domain addressed in this paper. Section 3 presents the methodology adopted. Section 4 presents the three (3) algorithms considered and justification for their choice. Section 5 presents the experimental details including results of our experimentation. Finally, section 6 provides the conclusion.

2 Related Works

Kolter & Maloof (2006) explained the machine learning and data mining approaches for classifying and detecting malicious URLs anytime they appear in the wild. The authors were able to collate "1,971 benign and 1,651 malicious" executable and used n-grams of byte codes as a training example. The processing approach yielded over 255 million different n-grams. After considering the most useful and relevant grams for prediction including Naïve
Bayes, decision trees, support vector machines, and boosting, they arrived at conclusion that boosted decision trees performed best of all other approaches under the ROC curve of 0.996.

Embedding malicious URLs in e-mails is one of the most common web threats facing the Internet community today. Malicious URLs have been widely used to mount various cyber-attacks like spear phishing, pharming, phishing and malware. In an attempt to find solution to this challenge, Azeez and Ademolu (2016) explored how malicious links in e-mails can be detected from the lexical and host-based features of their URLs to protect users from identity theft attacks. This research uses Naïve Bayesian classifier as a probabilistic model to detect if a URL is malicious or legitimate. The Naïve Bayesian classifier is used to count up the occurrence of each feature in an email and calculate the cumulative score.

In an effort to solve the challenge being posed by phishing in the cyberspace, Kirda and Kruegel (2005) developed AntiPhish, a mechanism that aims at preventing Internet users against any form of phishing attack. The system tracks information considered sensitive and quickly provide warning against divulging such information to any website that is considered untrusted (Azeez, Iyamu & Venter, 2011).

Alnajim & Munro (2009) proposed anti-phishing approach for detecting phishing website tagged APTIPWD. This approach assists Internet users to differentiate between legitimate and phishing websites. It provides useful information to the end user to quickly recognize either a fake or genuine site. This approach is adjudged to be one of the best approaches for recognizing if a site is either of the two classifications. It is however difficult to implement and browser dependent.

An algorithm considered novel was proposed by Joshi, Saklikar, Das, & Saha, (2008). The objective of the work was to identify any forged website firstly submitting random credentials before the real credentials in a login process of a website. A mechanism for analysing feedbacks from the servers against the submitted credentials was also proposed. The aim of this was to determine through the credentials if a website is original or phished one. It is however observed that the technology is basically meant for a website that supports HTTP with both userid and passwords as credentials The approach seems reliable and efficient but it is stressful to implement and also browser dependent.

Kan and Thi (2005), they carried out classification of web pages without considering their content but by applying their URLs. The latter is considered faster as there is no delay when parsing the text and fetching the page content. The features used in their work modelled various sequential dependencies between different tokens. They concluded in their research that the combination of feature extraction and URL segmentation enhanced the classification rate over other techniques. Similar research was carried out by Baykan et. al., though, they trained different binary classifiers for each point. They were able to improve
3 Methodology

Having established that three learning algorithms were used, the need to further explain the approach and the source of the dataset used is important.

The dataset used was obtained from Irvine, California, United States (UCI) Machine Learning Repository. URLs from different mails were used to validate the models (Logistics Regression, Decision Tree and Naïve Bayes). The features extracted for classifications are thirteen (13) in number (see Appendix A) based on each URL. The software used is a customized classification software by PHP scripting language with MySQL database.

The source of the dataset used was from United States (UCI) Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/URL+Reputation while the size of the dataset is 496MB in matlab format.

With principal component analysis (PCA), the objective of reducing the dimensions of a d-dimensional dataset used by projecting it onto a (k)-dimensional subspace with the aim of increasing the computational efficiency while considering and retaining most of the information was achieved.

The size of the training dataset used is 82MB while testing dataset is 150MB. Feature selection remains a complex and intricate issue when dealing with a dataset of numerous entries with uncountable attributes. In the dataset, features of an URL are tagged and coded as a set of binary attributes with each tallying to one of the likely value. When distributing a categorical value across dual binary attributes, it was noted that none of the attributes has detailed information about the feature except its value is 1. There is need to detect and
The three (3) learning algorithms were implemented for the classification of the dataset extracted using MATLAB 2015 and the dataset was also exported as a file and downloaded to Mysql database with further training using customized PHP.

4 Algorithms Used For Comparison

Decision was reached on the three algorithms because of their popularity along with observable contradictory results obtained on them from previous researches (Vanhoenshoven, Napoles, Falcon, Vanhoof, & Koppen, 2016) (Choudhary & Jain, 2017). What is more, they can provide relatively good performance on the classification task in this work [21].

The dataset used was obtained from Irvine, California, United States (UCI) Machine Learning Repository. URLs from different mails were used to validate the models (Logistics Regression, Decision Tree and Naïve Bayes). The features extracted for classifications are thirteen (13) in number based on each URL. The software used is a customized classification software by PhP scripting language with MySQL database.

4.1 Naïve Bayesian (NB) Classifier

Naïve Bayesian (NB) classification algorithm makes use of Bayes’ theorem strong assumptions between the features as the case may be. NB is a popular and one of the most useful learning algorithms for classification of text along the word frequencies. It is commonly used in spam filtering (Alnajim & Munro, 2009). Given a dependent class variable C with a small number of outcomes or classes which is conditional on several feature variables, each URL in an email is represented by a feature vector $F = (F_1, F_2, F_3, \ldots, F_n)$ where each of the property, $F_1, F_2, F_3, \ldots, F_n$ is independent. A Naïve Bayes classifier can be represented as follows:

$$P(C = c | F_1, \ldots, F_n) = \frac{P(C = c) \cdot P(F_1, \ldots, F_n | C = c)}{\sum_{k \in \{spam, legitimate\}} P(C = k) \cdot P(F_1, \ldots, F_n | C = k)}$$

(1)

The naïve conditional independence assumes that each feature $F_i$ is conditionally independent of every other feature $F_j (j \neq i)$ given a class $C$. Hence, $P(C = c | F_1, \ldots, F_n)$ can be computed as:

$$(C = c | F_1, \ldots, F_n) = \frac{P(C = c) \cdot \prod_{i=1}^{n} P(F_i | C = c)}{\sum_{k \in \{spam, legitimate\}} P(C = k) \cdot \prod_{i=1}^{n} P(F_i | C = k)}$$

(2)

where $P(F_i | C)$ and $P(C)$ can be easily calculated from the training samples.
4.2 Decision Tree Model Algorithm

Decision trees are always used as a predictive model. It can be used to map an item in order to draw a conclusion about item’s expected value. Decision tree learning is majorly used in data mining to create a model for prediction based on several variables (Rokach & Maimon, 2008). Data comes in the form:

\[(x, Y) = (x_1, x_2, x_3, \ldots, x_m, Y)\]  \hspace{1cm} (3)

The dependent variable, \(Y\), is the target variable that we are trying to understand, classify or generalize. The vector \(x\) is composed of the input variables, \(x_1, x_2, x_3\) etc., that are used for that task at hand (Rokach & Maimon, 2008).

4.3 Linear Regression

Regression analysis is a technique for modeling the relationship between variables (Campbell & Campbell, 2008). In brief:

1. Assume two variables, \(x\) and \(y\). Model relationship as \(y \sim x\) (that is, \(y = f(x)\)) as a linear relationship

\[y = \beta_0 + \beta_1 x\]  \hspace{1cm} (4)

2. Not a perfect fit generally: account for difference between model prediction and the actual target value as a statistical error \(\varepsilon\), s.t. \(y = \beta_0 + \beta_1 x + \varepsilon\). This is a linear regression model.

3. This error \(\varepsilon\) may be made up of the effects of other variables, measurement errors and so forth

4. \(x\) is called the independent variable (predictor or regressor) and \(y\) the dependent variable (response variable)

5. Simple linear regression involves only one regressor variable

6. if \(x\) is fixed, the random component \(\varepsilon\) shall determine the properties of \(y\).

5 Algorithm Evaluation Based on Confusion Matrices

A confusion matrix is a tabular description of the performance of a classifier on a given dataset which consequently reveals the true values. It is very simple to comprehend and
analyze but seldom, similar terminology might be difficult and confusing. Table 1 provides a typical example of confusion matrix for a binary classifier (Markham, 2014).

Table 1: An Example of Confusion Matrix

<table>
<thead>
<tr>
<th>n = 165</th>
<th>Predicted: NO</th>
<th>Predicted: YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual: NO</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>Actual: YES</td>
<td>5</td>
<td>100</td>
</tr>
</tbody>
</table>

The following are the basic terms and definitions:

1. **True positives (TP)**: This occurs in a situation whereby there is a 'yes' prediction (legitimate) and it is legitimate.

2. **True negatives (TN)**: This occurs in a situation whereby there is a 'no' prediction, and it is a fake Url.

3. **False positives (FP)**: This occurs in a situation whereby there is a 'yes' prediction, but it is a fake Urls. (Also known as a "Type I error.")

4. **False negatives (FN)**: This occurs in a situation whereby there is a 'no' prediction, but it is a legitimate. (Also known as a "Type II error.")

**Precision** or **Positive Predictive Value** can simply be defined as the fraction of cases and occurrences that are relevant, that is: PRECISION= TP/TP+FP.

**Recall** (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. RECALL= TP/TP+FN (El-Buhaisi, 2013).

The confusion matrices for different probabilities follow next.

5.1 **Confusion Matrix for Probability of < 0.001 as 0 and >= 0.001 as 1**

In the result shown in Table 2 and its graphical representation in Figure 1 shows the confusion matrix for the probability of result < 0.001 as 0 and result >= 0.001 as 1 for the three learning algorithms to know their better accuracy.
5.2 Confusion Matrix for Probability of $< 0.3$ as 0 and $\geq 0.3$ as 1

In the result shown in Table 3 and its graphical representation in Figure 2 shows the confusion matrix for the probability of result $< 0.3$ as 0 and result $\geq 0.3$ as 1 for the three learning algorithms to know their better accuracy.
Table 3: Probability of 0.3 Comparison Results of Confusion Matrix.

<table>
<thead>
<tr>
<th>SN</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.31</td>
<td>0.9</td>
<td>0.461</td>
<td>0.333</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.241</td>
<td>1</td>
<td>0.388</td>
<td>0.233</td>
<td>1</td>
<td>0.378</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.633</td>
<td>1</td>
<td>0.775</td>
<td>0.633</td>
<td>1</td>
<td>0.775</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.667</td>
<td>1</td>
<td>0.8</td>
<td>0.667</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>1</td>
<td>0.824</td>
<td>0.7</td>
<td>1</td>
<td>0.824</td>
</tr>
<tr>
<td>180</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>1</td>
<td>0.889</td>
<td>0.8</td>
<td>1</td>
<td>0.889</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.759</td>
<td>1</td>
<td>0.863</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
</tr>
<tr>
<td>240</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.833</td>
<td>1</td>
<td>0.909</td>
<td>0.833</td>
<td>1</td>
<td>0.909</td>
</tr>
<tr>
<td>270</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.773</td>
<td>1</td>
<td>0.872</td>
<td>0.773</td>
<td>1</td>
<td>0.872</td>
</tr>
</tbody>
</table>

Figure 2: Graphical Comparison of results for probability of 0.3.

5.3 Confusion Matrix for Probability of < 0.5 as 0 and >= 0.5 as 1

In the result shown in Table 4 and its graphical representation in Figure 3 shows the confusion matrix for the probability of result < 0.5 as 0 and result >= 0.5 as 1 for the three learning algorithms to know their better accuracy.
Table 4: Probability of 0.5 Comparison Results of Confusion Matrix.

<table>
<thead>
<tr>
<th>SN</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.31</td>
<td>0.9</td>
<td>0.461</td>
<td>0.333</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.241</td>
<td>1</td>
<td>0.388</td>
<td>0.233</td>
<td>1</td>
<td>0.378</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.633</td>
<td>1</td>
<td>0.775</td>
<td>0.633</td>
<td>1</td>
<td>0.775</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.667</td>
<td>1</td>
<td>0.8</td>
<td>0.667</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>1</td>
<td>0.824</td>
<td>0.7</td>
<td>1</td>
<td>0.824</td>
</tr>
<tr>
<td>180</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>1</td>
<td>0.889</td>
<td>0.8</td>
<td>1</td>
<td>0.889</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.759</td>
<td>1</td>
<td>0.863</td>
<td>0.733</td>
<td>1</td>
<td>0.846</td>
</tr>
<tr>
<td>240</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.833</td>
<td>1</td>
<td>0.909</td>
<td>0.833</td>
<td>1</td>
<td>0.909</td>
</tr>
<tr>
<td>270</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.773</td>
<td>1</td>
<td>0.872</td>
<td>0.773</td>
<td>1</td>
<td>0.872</td>
</tr>
</tbody>
</table>

Figure 3: Graphical Comparison of results for probability of 0.5.

5.4 Confusion Matrix for Probability of < 0.6 as 0 and >= 0.6 as 1

In the result shown in Table 5 and its graphical representation in Figure 4 shows the confusion matrix for the probability of result < 0.6 as 0 and result >= 0.6 as 1 for the three learning algorithms to know their better accuracy.
6 Conclusion

The evasion of anti-spam filtering techniques is made possible by hackers through the embedding of fake URLs in the content of electronic mails. The malicious actions of hackers have undoubtedly caused several monumental economic damages to many financial institutions. The numerical values obtained after experimentation and the corresponding statistical interpretation imply that the differences among the methods used are very significant hence the need to identify the best. The status and ranking of each of the methods as depicted in Table 5 can be taken as the best, correct and important rating in terms of recall, precision and f-measure. Decision tree (ID3) algorithm appears to be the most appropriate classification
for the problem followed by Binary Logistics Regression algorithm. Based on the confusion metrics measurement, the result obtained shows that the Decision Tree (ID3) algorithm achieves the highest values for recall, precision and f-measure. It unarguably provides efficient and credible means of maximizing the detection of compromised and malicious URLs. Finally, for future work, authors are of the opinion that two or more supervised learning algorithms can be hybridized to form a single effective and more efficient algorithm for fake URLs verification.

References


