Performance Evaluation of Machine Learning Techniques for Identifying Forged and Phony Uniform Resource Locators (URLs)

N. A. Azeez*, A. A. Ajayi

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

ABSTRACT: Since the invention of Information and Communication Technology (ICT), there has been a great shift from the erstwhile traditional approach of handling information across the globe to the usage of this innovation. The application of this initiative cut across almost all areas of human endeavours. ICT is widely utilized in education and production sectors as well as in various financial institutions. It is of note that many people are using it genuinely to carry out their day to day activities while others are using it to perform nefarious activities at the detriment of other cyber users. According to several reports which are discussed in the introductory part of this work, millions of people have become victims of fake Uniform Resource Locators (URLs) sent to their mails by spammers. Financial institutions are not left out in the monumental loss recorded through this illicit act over the years. It is worth mentioning that, despite several approaches currently in place, none could confidently be confirmed to provide the best and reliable solution. According to several research findings reported in the literature, researchers have demonstrated how machine learning algorithms could be employed to verify and confirm compromised and fake URLs in the cyberspace. Inconsistencies have however been noticed in the researchers' findings and also their corresponding results are not dependable based on the values obtained and conclusions drawn from them. Against this backdrop, the authors carried out a comparative analysis of three learning algorithms (Naïve Bayes, Decision Tree and Logistics Regression Model) for verification of compromised, suspicious and fake URLs and determine which is the best of all based on the metrics (F-Measure, Precision and Recall) used for evaluation. 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.

KEYWORDS: Learning-algorithms, Forged-URL, Phoney-URL, performance-comparison.

[Received July 13, 2018; Revised January 23, 2019; Accepted February 10, 2019] Print ISSN: 0189-9546 | Online ISSN: 2437-2110

I. 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, however, being used by hackers to send malicious attack in form of pharming, phishing, e-mail spoofing and malware infection to users' computers and digital/electronic devices. 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 and ATM pin number. If however, such a customer assumes the message is from the authentic source he will be a victim (Choudhary & & Jain, 2017).

*Corresponding author: nazeez@unilag.edu.ng

doi: http://dx.doi.org/10.4314/njtd.v16i4.2

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 behaviours. 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 have been developed by web security (Nureni & Irwin, 2010) community and researchers to specifically identify these fake, vulnerable and compromised URL as well as 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 these challenges 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 Na¨ıve 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 (Nivedha et. al., 2017). This work is intended to meet the following objectives:

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

Having presented the introduction in Section 1, the rest of the paper is organized as follows: 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 findings, and finally, Section 6 provides the conclusion.

II. RELATED WORKS 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 customized classification software by PHP scripting language with MySQL database.

A. Naïve Bayesian (NB) Classifier

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 (Sahami, et al., 1998).

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$\vec{F} = (F_1, F_2, F_3, \dots, F_n)$ where each of the property, $F_1, F_2, F_3, \dots, F_n$ is independent. A Naive Bayes classifier can be represented as follows:

$$P(C = c|F_1, \dots, F_n) = \frac{P(C = c).P(F_1, \dots, F_n|\mathrm{C} = \mathrm{c})}{\sum_{k\in(spam,legitimate)} P(C = k).P(F_1, \dots, F_n|\mathrm{C} = \mathrm{k})} \quad (1)$$

The ''naive'' 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, $P(C = c|F_1, \dots, F_n)$ can be computed as:

$$(C = c|F_1, \dots, F_n) = \frac{P(C = c).\prod_{i=1}^{n} P(F_i|\mathrm{C} = \mathrm{c})}{\sum_{k\in(spam,legitimate)} P(C = k).\prod_{i=1}^{n} P(F_i|\mathrm{C} = \mathrm{k})} \quad (2)$$

Where $P(F_i|\mathrm{C})$ and $P(C)$ can be easily calculated from the training samples.

B. Decision Tree l Algorithm

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:

$$(x, Y) = (x_1, x_2, x_3 \dots \dots \dots \dots \dots, x_4, Y) \quad (3)$$

"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).

C. Linear Regression

Regression analysis is a technique for modelling the relationship between variables (Campbell & Campbell, 2008).

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

$$y = \beta 0 + \beta 1x \quad (4)$$

1. Not a perfect fit generally; account for difference between model prediction and the actual target value as a statistical error ε
   $y = \beta 0 + \beta 1x + \varepsilon$. This is a linear regression model
2. This error ε may be made up of the effects of other variables, measurement errors and so forth
3. x is called the independent variable (predictor or regressor) and y the dependent variable (response variable)
4. Simple linear regression involves only one regressor variable
5. if x is fixed, the random component ε shall determine the properties of y

Kolter & Maloof (2006) explained the machine learning and data mining approaches for classifying and detecting malicious URLs whenever they occur unexpectedly and uncontrollably. 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 a conclusion that 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. This is 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 unreliable (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 might seldom be 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 by firstly submitting random credentials before the real credentials in a login process of a website. A mechanism for analyzing 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.

Kan and Thi (2005), 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, modeled various sequential dependencies between different tokens. They concluded that the combination of feature extraction and URL segmentation enhanced the classification rate over other techniques. Similar research was carried out by Baykanet. al., though, they trained different binary classifiers for each point. They were able to improve on the result of previous f-measure.

Ma, et al., (2009) used the lexical and host-based features to detect malicious websites. Their approach could sift through numerous features and recognize the important URL metadata and components without demanding any domain expertise. They succeeded in evaluating up to 30,000 instances with good and promising results, specifically, a very high classification rate of 95% - 99% and a low false positive rate.

In (Ma et al., 2011) adopted online algorithms so as to handle many URLs whose feature evolve over a period of time. They developed a system to gather up-to-date URL features which was paired with a real-time feed of labeled URLs from a large mail provider. They reported a successful classification rate of 99% using confidence-weighted learning on a balanced dataset.

In the work of Yukun et. al. (2019), attempt was made to detect any phishing webpages using URL and HTML. This was achieved by presenting a stacking model for quick detection. With this approach, lightweight URL and HTML features were designed and later introduced HTML string without making reference to the third-party services. The stacking model was achieved by combining LightGBM, XGBoost and GBDT which allows various models to be complementary. This approach is believed o produce a better performance for detecting phishing webpage. To evaluate this approach, tow datasets - 50KPD and 50K-IPD were used. The approach achieves 98.60% on accuracy (Yukun et. al., 2019).

Having realized the fact that some of the existing approaches for preventing and detecting phishing are not reliable and efficient, Adebowale et. al., (2019) proposed an Adaptive Neuro-Fuzzy Inference System (ANFIS) based robust scheme using the combined features of text of legitimate and illegitimate websites, images, frames as well as associated artificial intelligence algorithms to develop an

integrated solution. 98.3% accuracy was achieved through this proposed solution (Adebowale et. al., in 2019).

The semantic-based attack structure is very complex to determine the status of any webpage whether legitimate or illegitimate. Having realized this complexity as well as inefficiency in some of the anti-phishing solutions, Sahingoz et. al. (2019) proposed a real-time anti-phishing solution which adopts seven different machine learning algorithms and the features of natural language processing (NLP). The system has seven distinguishing features from the existing solutions. The performance evaluation of the system was tested via a newly constructed dataset. The results obtained from the comparative and experimental analysis reveal that the excellent and performance accuracy rate of 97.98% was observed in Random Forest algorithm with only NLP based features (Sahingoz et. al., 2019)

III. 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 based on each URL. The software used is 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 was achieved. With PCA, the most effective and reliable transformation of the current attributes via a linear transformation technique is easily located. It equally assists to reduce the volume of dataset to be processed and stored. It assists to improve the visualization process. It provides much clear interpretation of the data under consideration (Sasan et. al., 2013).

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 tallies 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 identify those attributes with value of 1 as the most significant in categorizing an URL. 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.

IV. IMPLEMENTATION, RESULTS AND DISCUSSION

The system is a web-based application, which classifies a URL as malicious or legitimate based on certain predefined features or covariates. Based on the predefined criteria, if anyone of the features is found in the URL, the system classifies the URL as malicious else it is classified as legitimate and the malicious and legitimate are updated in the database.

Table 1: Naïve Bayes classification Algorithm.

REPORT GENERATED FOR NAIVE BAYES (ID 3) ALGORITHM

SN	URL _Name	55lenght	nohe ader	no_ip_address	nodots	age	at	percentage	verify	secure	account	confirm	signin	password	Probyes	Probno	Proboutcome
1	https://www.techmaish.com	1	0	1	1	1	1	1	1	1	1	1	1	1	0.004944	0.000642	0.004944
2	http://www.lnkd.in	1	0	1	1	0	1	1	1	1	1	1	1	1	0.004944	0.000321	0.004944
3	http://www.shrinkster.com	1	1	1	1	0	1	1	1	1	1	1	1	1	0.001648	0.000161	0.001648
4	http://yourls.org	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001648	0.000321	0.001648
5	http://scrnch.me	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001648	0.000321	0.001648
6	https://is.gd	1	0	1	1	1	1	1	1	1	1	1	1	1	0.004944	0.000642	0.004944
7	http://w3t.org	1	0	1	1	1	1	1	1	1	1	1	1	1	0.004944	0.000642	0.004944
8	https://www.po.st	1	0	1	1	1	1	1	1	1	1	1	1	1	0.004944	0.000642	0.004944
9	http://u.to	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001648	0.000321	0.001648
10	https://bitly.com	1	0	1	1	1	1	1	1	1	1	1	1	1	0.004944	0.000642	0.004944

Table 2: ID3 (Decision Tree) Algorithm Result.

REPORT GENERATED FOR DECISION TREE ID3 ALGORITHM																		
SN	URL **_Name**	**55lenght**	**nohead er**	**no_ip_ address**	**nodots**	**Age**	**at**	**percentage**	**verify**	**secure**	**account**	**confirm**	**signin**	**password**	**prob Highest**	**prob Lowest**	**prob Outcome**
1	https://www.techmaish.com	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
2	http://www.lnkd.in	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1
3	http://www.shrinkster.com	1	1	1	1	0	1	1	1	1	1	1	1	1	0.9183	0.66667	0.9183
4	http://yourls.org	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	http://scrnch.me	1	1	1	1	1	1	1	1	1	1	1	1	1	0.97095	0.95098	0.97095
6	https://is.gd	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0.9183	1
7	http://w3t.org	1	0	1	1	1	1	1	1	1	1	1	1	1	0.98523	0.85714	0.98523
8	https://www.po.st	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0.95121	1
9	http://u.to	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99108	0.98386	0.99108

Table 3: Logistic Regression Result.

REPORT GENERATED FOR BINARY LOGISTICS REGRESSION ALGORITHM																	
SN	**URL _Name**	**55lenght**	**Nohead er**	**no_ip_ address**	**nodots**	**age**	**at**	**percentage**	**verify**	**secure**	**account**	**confirm**	**signin**	**password**	**Prob Low**	**Prob High**	**Prob Result**
1	https://www.techmaish.com	1	0	1	1	1	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583
2	http://www.lnkd.in	1	0	1	1	0	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583
3	http://www.shrinkster.com	1	1	1	1	0	1	1	1	1	1	1	1	1	0.4433	0.5567	0.5567
4	http://yourls.org	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4433	0.5567	0.5567
5	http://scrnch.me	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4433	0.5567	0.5567
6	https://is.gd	1	0	1	1	1	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583
7	http://w3t.org	1	0	1	1	1	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583
8	https://www.po.st	1	0	1	1	1	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583
9	http://u.to	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4433	0.5567	0.5567
10	https://bitly.com	1	0	1	1	1	1	1	1	1	1	1	1	1	0.4417	0.5583	0.5583

The verification of fake URLs using supervised learning algorithms (Naïve Bayes classification algorithm, Decision Tree and Logistic Regression) is based on repetitive and redundancy values have been implemented. Experimentation was carried out to determine the algorithm that has the highest maximal level of effectiveness, accuracy and efficiency. The tables below provide analyze into the three supervised learning algorithms by showing their performance evaluation.

A. Results

1's and 0's are representations of results of each covariate or parameter from all the thirteen covariates or parameters that determined whether the status of the current URL is legitimate or not, yes or no, on or off, occurred or not-occurred.

Table 4: Compared Run 1 Results

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK										
		NAÏVE BAYES OUTCOME			**DECISION TREE OUTCOME**			**LOGISTICS REGRESSION OUTCOME**		
SN	**URL _Name**	**probyes**	**probno**	**proboutcome**	**dec Low**	**decHogh**	**dicOutcom**	**Log Low**	**Log High**	**Log Outcom**
1	https://www.techmaish.com	0.004944	0.000642	0.004944	0	0	0	0.4417	0.5583	0.5583
2	http://www.lnkd.in	0.004944	0.000321	0.004944	1	0	1	0.4417	0.5583	0.5583
3	http://www.shrinkster.com	0.001648	0.000161	0.001648	0.9183	0.66667	0.9183	0.4433	0.5567	0.5567
4	http://yourls.org	0.001648	0.000321	0.001648	1	1	1	0.4433	0.5567	0.5567
5	http://scrnch.me	0.001648	0.000321	0.001648	0.97095	0.95098	0.97095	0.4433	0.5567	0.5567
6	https://is.gd	0.004944	0.000642	0.004944	1	0.9183	1	0.4417	0.5583	0.5583
7	http://w3t.org	0.004944	0.000642	0.004944	0.98523	0.85714	0.98523	0.4417	0.5583	0.5583
8	https://www.po.st	0.004944	0.000642	0.004944	1	0.95121	1	0.4417	0.5583	0.5583
9	http://u.to	0.001648	0.000321	0.001648	0.99108	0.98386	0.99108	0.4433	0.5567	0.5567
10	https://bitly.com	0.004944	0.000642	0.004944	1	1	1	0.4417	0.5583	0.5583
11	http://cutt.us	0.001648	0.000161	0.001648	0.99403	0.97772	0.99403	0.4433	0.5567	0.5567
12	https://themeforest.net/category/site-templates	0.004944	0.000642	0.004944	1	0.97288	1	0.4417	0.5583	0.5583

Table 5: Compared Run 2 Results

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK										
		NAÏVE BAYES OUTCOME			**DECISION TREE OUTCOME**			**LOGISTICS REGRESSION OUTCOME**		
sn	**URL _Name**	**probyes**	**probno**	**proboutcome**	**dec Low**	**decHogh**	**dicOutcom**	**Log Low**	**Log High**	**Log Outcom**
51	https://git.io	0.004944	0.000642	0.004944	0.89743	0.86401	0.89743	0.4417	0.5583	0.5583
52	https://hec.su	0.004944	0.000642	0.004944	0.89049	0.85717	0.89049	0.4417	0.5583	0.5583
53	https://shorte.st	0.004944	0.000642	0.004944	0.88359	0.85037	0.88359	0.4417	0.5583	0.5583
54	http://www.ask.com	0.001648	0.000321	0.001648	0.87672	0.84361	0.87672	0.4433	0.5567	0.5567
55	https://tldrify.com	0.004944	0.000642	0.004944	0.86989	0.83691	0.86989	0.4417	0.5583	0.5583
56	https://tr.im	0.004944	0.000642	0.004944	0.86312	0.83026	0.86312	0.4417	0.5583	0.5583
57	http://www.ShortLinks.co.uk	0.004944	0.000642	0.004944	0.85641	0.82366	0.85641	0.4417	0.5583	0.5583
58	http://zypopwebtemplates.com/@@@jkjkfnjfni	0.001648	0.000321	0.001648	0.84975	0.81714	0.84975	0.4417	0.5583	0.5583
59	https://www.gooyaabitemplates.com/blogger-templates	0.004944	0.000642	0.004944	0.84316	0.81068	0.84316	0.4417	0.5583	0.5583
60	http://www.igbesa.com	0.001648	0.000321	0.001648	0.85995	0.82911	0.85995	0.4433	0.5567	0.5567
61	https://dcrazed.com	0.004944	0.000642	0.004944	0.85366	0.82293	0.85366	0.4417	0.5583	0.5583
62	http://www.webopedia.com	0.001648	0.000321	0.001648	0.86914	0.83989	0.86914	0.4433	0.5567	0.5567
63	http://www.cisco.com	0.001648	0.000321	0.001648	0.88322	0.85534	0.88322	0.4433	0.5567	0.5567
64	http://www.info.com	0.001648	0.000321	0.001648	0.89604	0.86942	0.89604	0.4433	0.5567	0.5567
65	https://www.walmart.com	0.004944	0.000642	0.004944	0.89049	0.86394	0.89049	0.4417	0.5583	0.5583
66	http://www.apple.com	0.001648	0.000161	0.001648	0.90239	0.877	0.90239	0.4433	0.5567	0.5567
67	http://www.berkshirehathaway.com	0.001648	0.000161	0.001648	0.91324	0.88291	0.91324	0.4433	0.5567	0.5567
68	http://www.mckesson.com	0.001648	0.000161	0.001648	0.92312	0.88598	0.92312	0.4433	0.5567	0.5567
69	http://www.unitedhealthgroup.com	0.001648	0.000161	0.001648	0.93211	0.88805	0.93211	0.4433	0.5567	0.5567
70	https://www.cvshealth.com	0.004944	0.000642	0.004944	0.92753	0.88137	0.92753	0.4417	0.5583	0.5583

Table 6: Compared Run 3 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK										
		NAÏVE BAYES OUTCOME			**DECISION TREE OUTCOME**			**LOGISTICS REGRESSION OUTCOME**		
SN	**URL_Name**	**Probyes**	**Probno**	**Proboutcome**	**Dec Low**	**Dechogh**	**Dicoutcom**	**Log Low**	**Log High**	**Log Outcom**
101	http://corporate.comcast.com	0.001648	0.000321	0.001648	0.98403	0.84544	0.98403	0.4433	0.5567	0.5567
102	http://www.target.com	0.004944	0.000321	0.004944	0.98218	0.84102	0.98218	0.4417	0.5583	0.5583
103	https://www.jnj.com	0.004944	0.000642	0.004944	0.98026	0.83662	0.98026	0.4417	0.5583	0.5583
104	https://www.metlife.com	0.004944	0.000642	0.004944	0.98286	0.84839	0.98286	0.4417	0.5583	0.5583
105	http://www.adm.com/en-US/Pages/default.aspx	0.001648	0.000321	0.001648	0.98523	0.84565	0.98523	0.4433	0.5567	0.5567
106	http://www.marathonpetroleum.com	0.001648	0.000161	0.001648	0.98738	0.84281	0.98738	0.4433	0.5567	0.5567
107	http://www.freddiemac.com	0.001648	0.000161	0.001648	0.98933	0.83989	0.98933	0.4433	0.5567	0.5567
108	http://www.utc.com/Pages/Home.aspx	0.001648	0.000321	0.001648	0.99108	0.8369	0.99108	0.4433	0.5567	0.5567
109	http://www.aetna.com	0.001648	0.000161	0.001648	0.99264	0.83385	0.99264	0.4433	0.5567	0.5567
110	http://www.lowes.com	0.004944	0.000321	0.004944	0.9914	0.82993	0.9914	0.4417	0.5583	0.5583
111	https://www.ups.com	0.004944	0.000642	0.004944	0.99008	0.82602	0.99008	0.4417	0.5583	0.5583
112	http://www.aig.com	0.001648	0.000161	0.001648	0.9917	0.82304	0.9917	0.4433	0.5567	0.5567
113	https://www.prudential.com	0.004944	0.000642	0.004944	0.99315	0.83376	0.99315	0.4417	0.5583	0.5583
114	https://www.humana.com	0.004944	0.000642	0.004944	0.99199	0.83005	0.99199	0.4417	0.5583	0.5583
115	http://www.disney.com	0.001648	0.000161	0.001648	0.99339	0.82701	0.99339	0.4433	0.5567	0.5567
116	http://www.pfizer.com	0.001648	0.000161	0.001648	0.99463	0.82391	0.99463	0.4433	0.5567	0.5567
117	http://www.dow.com	0.001648	0.000161	0.001648	0.99573	0.82078	0.99573	0.4433	0.5567	0.5567
118	http://www.sysco.com	0.001648	0.000161	0.001648	0.99668	0.81762	0.99668	0.4433	0.5567	0.5567
119	http://www.fedex.com	0.001648	0.000161	0.001648	0.9975	0.81443	0.9975	0.4433	0.5567	0.5567
120	http://www.caterpillar.com	0.004944	0.000321	0.004944	0.99679	0.81101	0.99679	0.4417	0.5583	0.5583

Table 7: Compared Run 4 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK										
		NAÏVE BAYES OUTCOME			**DECISION TREE OUTCOME**			**LOGISTICS REGRESSION OUTCOME**		
SN	**URL _Name**	**probyes**	**probno**	**proboutcome**	**dec Low**	**decHogh**	**dicOutcom**	**Log Low**	**Log High**	**Log Outcom**
151	http://www.nike.com	0.004944	0.000321	0.004944	0.99921	0.78728	0.99921	0.4417	0.5583	0.5583
152	http://www.3m.com	0.001648	0.000161	0.001648	0.99888	0.7842	0.99888	0.4433	0.5567	0.5567
153	http://www.exeloncorp.com	0.001648	0.000161	0.001648	0.99849	0.78113	0.99849	0.4433	0.5567	0.5567
154	https://www.21cf.com	0.004944	0.000642	0.004944	0.99805	0.78812	0.99805	0.4417	0.5583	0.5583
155	http://www.deere.com/en_US/regional_home.page	0.001648	0.000321	0.001648	0.99757	0.78503	0.99757	0.4433	0.5567	0.5567
156	http://tsocorp.com	0.001648	0.000161	0.001648	0.9981	0.79501	0.9981	0.4433	0.5567	0.5567
157	http://www.timewarner.com	0.001648	0.000161	0.001648	0.99763	0.79201	0.99763	0.4433	0.5567	0.5567
158	https://www.northwesternmutual.com	0.004944	0.000642	0.004944	0.99711	0.79847	0.99711	0.4417	0.5583	0.5583
159	https://www.northwesternmutual.com	0.004944	0.000642	0.004944	0.99654	0.80457	0.99654	0.4417	0.5583	0.5583
160	http://www.dupont.com	0.001648	0.000161	0.001648	0.99718	0.81382	0.99718	0.4433	0.5567	0.5567
161	http://www.avnet.com/en-us/Pages/default.aspx	0.001648	0.000321	0.001648	0.99663	0.81087	0.99663	0.4433	0.5567	0.5567
162	http://www.macysinc.com	0.001648	0.000161	0.001648	0.99604	0.80792	0.99604	0.4433	0.5567	0.5567
163	http://www.enterpriseproducts.com	0.001648	0.000161	0.001648	0.99671	0.81677	0.99671	0.4433	0.5567	0.5567
164	https://www.travelers.com	0.004944	0.000642	0.004944	0.99613	0.82233	0.99613	0.4417	0.5583	0.5583
165	https://www.pmi.com	0.004944	0.000642	0.004944	0.99552	0.82759	0.99552	0.4417	0.5583	0.5583

Table 8: Compared Run 5 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK										
		NAÏVE BAYES OUTCOME			**DECISION TREE OUTCOME**			**LOGISTICS REGRESSION OUTCOME**		
SN	URL _Name	**probyes**	**probno**	**proboutcome**	**dec Low**	**decHogh**	**dicOutcom**	**Log Low**	**Log High**	**Log Outcom**
201	http://www.cummins.com	0.004944	0.000321	0.004944	0.98047	0.82084	0.98047	0.4417	0.5583	0.5583
202	http://www.altria.com/Pages/default.aspx	0.001648	0.000321	0.001648	0.97947	0.81818	0.97947	0.4433	0.5567	0.5567
203	https://www.xerox.com	0.004944	0.000642	0.004944	0.98085	0.81739	0.98085	0.4417	0.5583	0.5583
204	http://www.kimberly-clark.com	0.001648	0.000161	0.001648	0.97987	0.81475	0.97987	0.4433	0.5567	0.5567
205	https://www.thehartford.com	0.004944	0.000642	0.004944	0.97887	0.81782	0.97887	0.4417	0.5583	0.5583
206	http://www.kraftheinzcompany.com	0.001648	0.000161	0.001648	0.97786	0.81519	0.97786	0.4433	0.5567	0.5567
207	http://www.lear.com	0.001648	0.000161	0.001648	0.97683	0.81258	0.97683	0.4433	0.5567	0.5567
208	http://www.jabil.com	0.001648	0.000161	0.001648	0.97828	0.82073	0.97828	0.4433	0.5567	0.5567
209	http://www.supervalu.com	0.001648	0.000161	0.001648	0.97967	0.82848	0.97967	0.4433	0.5567	0.5567
210	http://www.southerncompany.com	0.001648	0.000161	0.001648	0.9787	0.82596	0.9787	0.4433	0.5567	0.5567

Table 9: Compared Run 1 Results.

FULL COMPARISON REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK				
SN	**URL _Name**	**NAIVE BAYES OUTCOME**	**DECISTION TREE OUTCOME**	**LOGISTICS REGRESSION OUTCOME**
1	https://www.techmaish.com	0.00494384	0	0.5583
2	http://www.lnkd.in	0.00494384	1	0.5583
3	http://www.shrinkster.com	0.00164794	0.9183	0.5567
4	http://yourls.org	0.00164794	1	0.5567
5	http://scrnch.me	0.00164794	0.97095	0.5567
6	https://is.gd	0.00494384	1	0.5583
7	http://w3t.org	0.00494384	0.98523	0.5583
8	https://www.po.st	0.00494384	1	0.5583
9	http://u.to	0.00164794	0.99108	0.5567
10	https://bitly.com	0.00494384	1	0.5583
11	http://cutt.us	0.00164794	0.99403	0.5567
12	https://themeforest.net/category/site-templates	0.00494384	1	0.5583

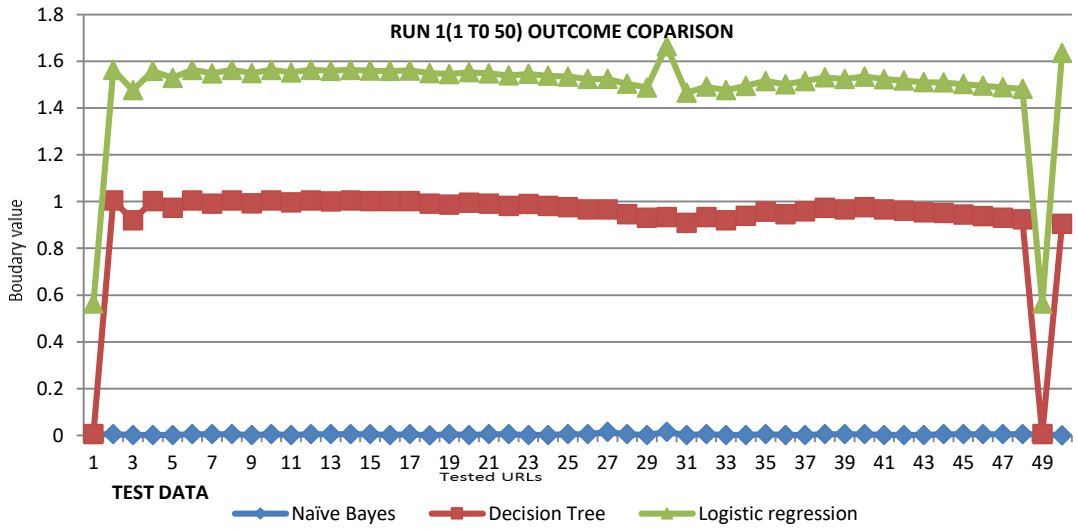

Figure 1 Graphical Comparison of the outcome.

Table 10: Compared Run 2 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK				
SN	**URL _Name**	**NAIVE BAYES OUTCOME**	**DECISTION TREE OUTCOME**	**LOGISTICS REGRESSION OUTCOME**
51	https://git.io	0.00494384	0.89743	0.5583
52	https://hec.su	0.00494384	0.89049	0.5583
53	https://shorte.st	0.00494384	0.88359	0.5583
54	http://www.ask.com	0.00164794	0.87672	0.5567
55	https://tldrify.com	0.00494384	0.86989	0.5583
56	https://tr.im	0.00494384	0.86312	0.5583
57	http://www.ShortLinks.co.uk	0.00494384	0.85641	0.5583
58	http://zypopwebtemplates.com/@@@jkjkfnjfni	0.00164794	0.84975	0.5583
59	https://www.gooyaabitemplates.com/blogger-templates	0.00494384	0.84316	0.5583
60	http://www.igbesa.com	0.00164794	0.85995	0.5567
61	https://dcrazed.com	0.00494384	0.85366	0.5583
62	http://www.webopedia.com	0.00164794	0.86914	0.5567
63	http://www.cisco.com	0.00164794	0.88322	0.5567
64	http://www.info.com	0.00164794	0.89604	0.5567

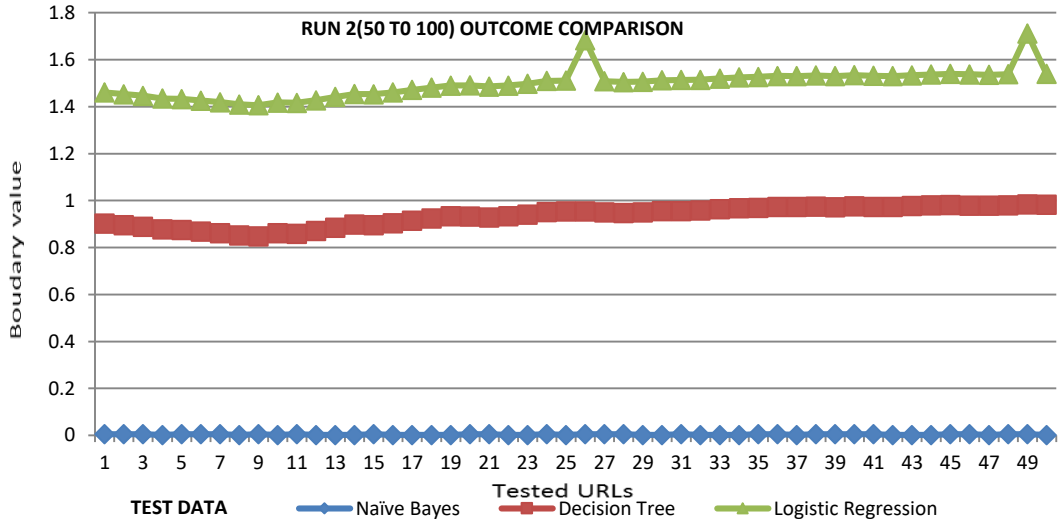

Figure 2: Graphical Comparison of the outcome.

Table 11: Compared Run 3 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK				
SN	**URL _Name**	**NAIVE BAYES OUTCOME**	**DECISTION TREE OUTCOME**	**LOGISTICS REGRESSION OUTCOME**
101	http://corporate.comcast.com	0.00164794	0.98403	0.5567
102	http://www.target.com	0.00494384	0.98218	0.5583
103	https://www.jnj.com	0.00494384	0.98026	0.5583
104	https://www.metlife.com	0.00494384	0.98286	0.5583
105	http://www.adm.com/en-US/Pages/default.aspx	0.00164794	0.98523	0.5567
106	http://www.marathonpetroleum.com	0.00164794	0.98738	0.5567
107	http://www.freddiemac.com	0.00164794	0.98933	0.5567
108	http://www.utc.com/Pages/Home.aspx	0.00164794	0.99108	0.5567
109	http://www.aetna.com	0.00164794	0.99264	0.5567
110	http://www.lowes.com	0.00494384	0.9914	0.5583
111	https://www.ups.com	0.00494384	0.99008	0.5583
112	http://www.aig.com	0.00164794	0.9917	0.5567
113	https://www.prudential.com	0.00494384	0.99315	0.5583
114	https://www.humana.com	0.00494384	0.99199	0.5583
115	http://www.disney.com	0.00164794	0.99339	0.5567

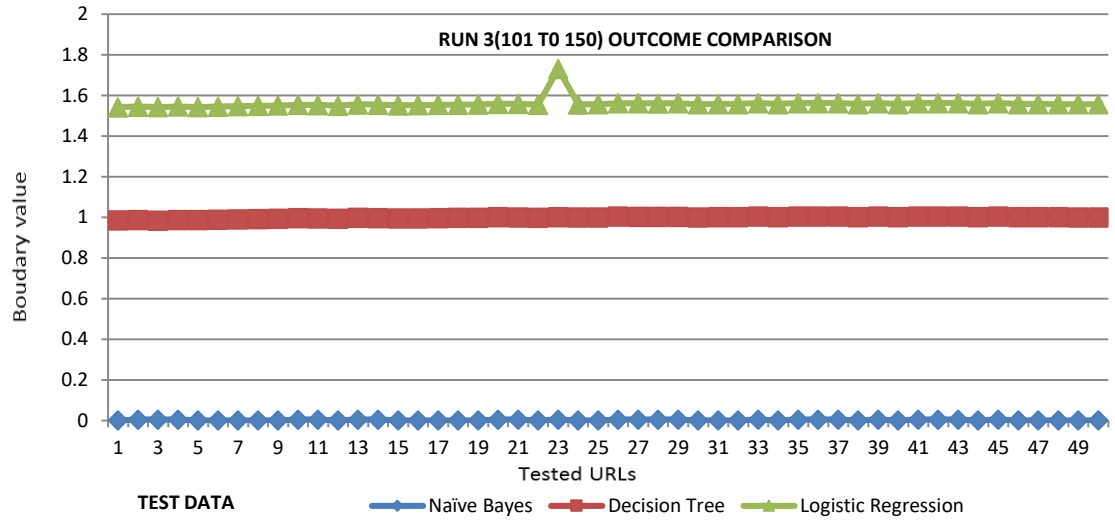

Figure 3: Graphical Comparison of the outcome.

Table 12: Compared Run 4 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK				
SN	**URL_Name**	**NAIVE BAYES OUTCOME**	**DECISTION TREE OUTCOME**	**LOGISTICS REGRESSION OUTCOME**
151	http://www.nike.com	0.00494384	0.99921	0.5583
152	http://www.3m.com	0.00164794	0.99888	0.5567
153	http://www.exeloncorp.com	0.00164794	0.99849	0.5567
154	https://www.21cf.com	0.00494384	0.99805	0.5583
155	http://www.deere.com/en_US/regional_home.page	0.00164794	0.99757	0.5567
156	http://tsocorp.com	0.00164794	0.9981	0.5567
157	http://www.timewarner.com	0.00164794	0.99763	0.5567
158	https://www.northwesternmutual.com	0.00494384	0.99711	0.5583
159	https://www.northwesternmutual.com	0.00494384	0.99654	0.5583
160	http://www.dupont.com	0.00164794	0.99718	0.5567
161	http://www.avnet.com/en-us/Pages/default.aspx	0.00164794	0.99663	0.5567
162	http://www.macysinc.com	0.00164794	0.99604	0.5567
163	http://www.enterpriseproducts.com	0.00164794	0.99671	0.5567

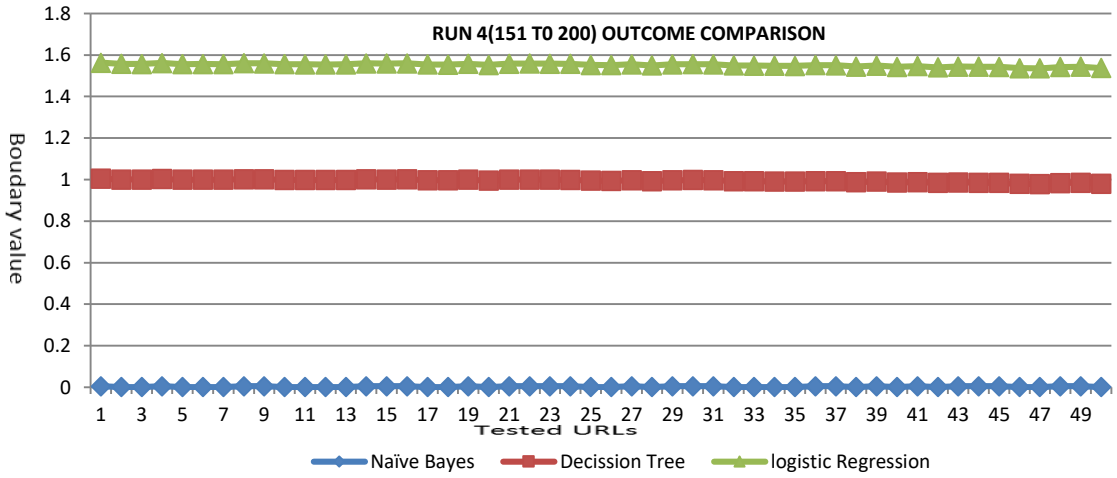

Figure 4: Graphical Comparison of the outcome.

Table 13: Compared Run 5 Results.

FULL REPORT GENERATED FOR ALL THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK				
SN	**URL _Name**	**NAIVE BAYES OUTCOME**	**DECISTION TREE OUTCOME**	**LOGISTICS REGRESSION OUTCOME**
201	http://www.cummins.com	0.0049438	0.98047	0.5583
202	http://www.altria.com/Pages/default.aspx	0.0016479	0.97947	0.5567
203	https://www.xerox.com	0.0049438	0.98085	0.5583
204	http://www.kimberly-clark.com	0.0016479	0.97987	0.5567
205	https://www.thehartford.com	0.0049438	0.97887	0.5583
206	http://www.kraftheinzcompany.com	0.0016479	0.97786	0.5567
207	http://www.lear.com	0.0016479	0.97683	0.5567
208	http://www.jabil.com	0.0016479	0.97828	0.5567
209	http://www.supervalu.com	0.0016479	0.97967	0.5567
210	http://www.southerncompany.com	0.0016479	0.9787	0.5567
211	http://www.nexteraenergy.com	0.0016479	0.98006	0.5567
212	http://www.thermofisher.com/ng/en/home.html	0.0016479	0.9791	0.5567
213	https://www.pnc.com	0.0049438	0.97812	0.5583
214	http://www.nucor.com	0.0016479	0.97713	0.5567
215	http://www.nucor.com	0.0016479	0.97853	0.5567

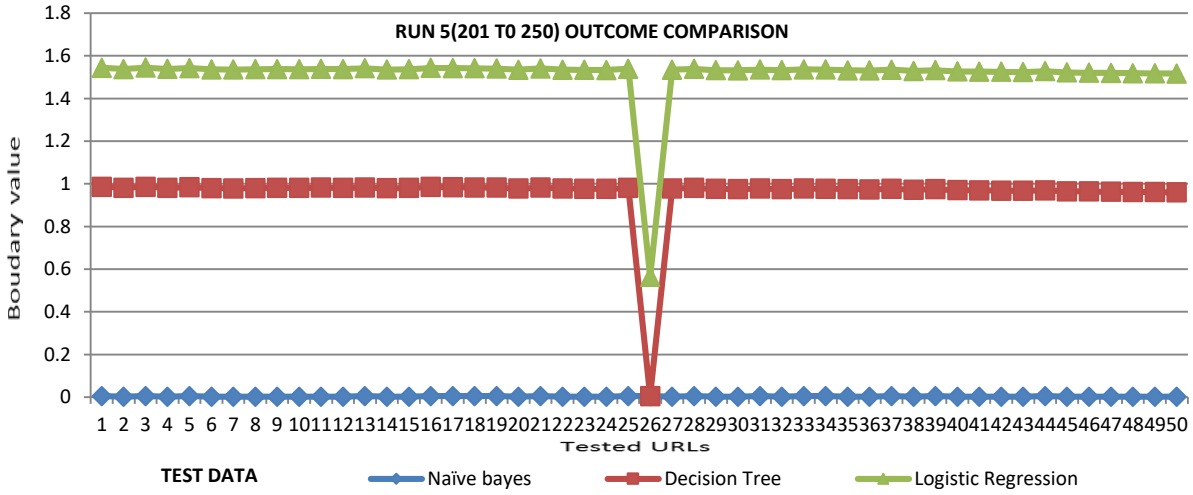

Figure 5: Graphical Comparison of the outcome.

CONFUSION MATRIX FOR PROBABILITY OF < 0.6 as 0 and >=0.6 as 1

Table 14: Probability of 0.5 Comparison Results of Confusion Matrix.

THE CONFUSION MATRIX FOR THE THREE LEARINING ALGORITHMS OF THIS RESEARCH WORK USING BOUNDARY OF 0.6									
SN	**NAIVE BAYES ALGORITHM**			**DECISION TREE (ID 3) ALGORITHMN**			**BINARY LOGISTICS REGRESSION ALGORITHMN**		
SN	**RECALL**	**PRECISION**	**F-MEASURE**	**RECALL**	**PRECISION**	**F-MEASURE**	**RECALL**	**PRECISION**	**F-MEASURE**
0	0	0	0	0.31	0.9	0.461	0	0	0
30	0	0	0	0.241	1	0.388	0	0	0
60	0	0	0	0.633	1	0.775	0	0	0
90	0	0	0	0.667	1	0.8	1	0.05	0.095
120	0	0	0	0.733	1	0.846	1	0.045	0.086
150	0	0	0	0.7	1	0.824	0	0	0
180	0	0	0	0.8	1	0.889	0	0	0
210	0	0	0	0.759	1	0.863	0	0	0
240	0	0	0	0.833	1	0.909	0	0	0
270	0	0	0	0.773	1	0.872	0.5	0.059	0.106

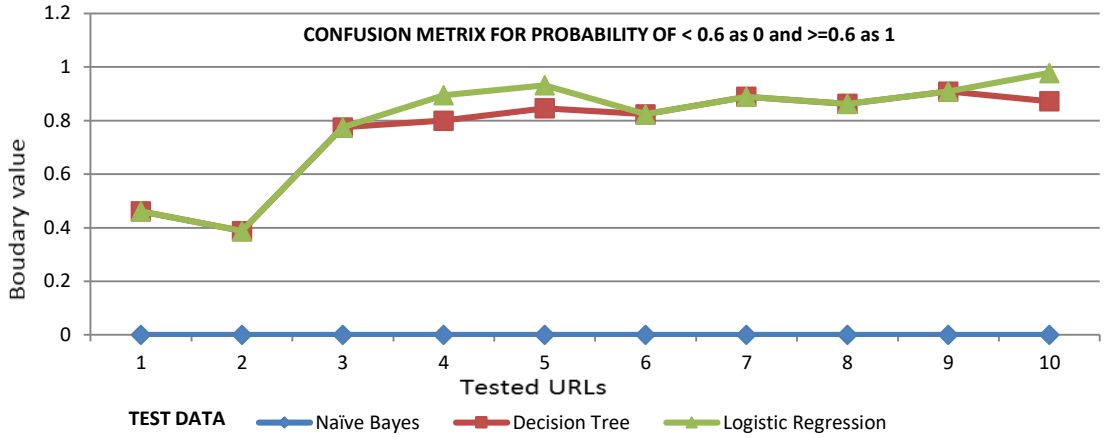

Figure 6: Graphical Comparison of results for probability of 0.5.

B. Results Discussion

The results presented in Table 4 represent the prediction result for the data verification at point 'YES' or 'NO' or point 'HIGH' or 'LOW' and also the point of the outcome by the three learning algorithms using the first set of URLs which start from URL One (1) to Fifty (50). The compared results shown in Table 5 are the prediction for the data verification at point 'YES' or 'NO' or point 'HIGH' or 'LOW'. Also, the point of the outcome by the three learning algorithms using the second run set of URLs which start from URL Fifty(50) to One Hundred(100).

The compared results shown in the Table 6 are the prediction result for the data verification at point 'YES' or 'NO' or point 'HIGH' or 'LOW'. This is the ***third*** run set of URLs which starts from One Hundred (100) to One Hundred and Fifty (150). The comparison results shown in the Table 8 are the predictions for the data verification at point 'YES' or 'NO' or point 'HIGH' or 'LOW'. This is the ***fifth*** run set of URLs which starts from Two Hundred (200) to Two Hundred and Fifty (250).

The result shown in Table 9 and its graphical representation in Figure 1 show the comparison outcome of the result from data one (1) to fifty (50). The result shown in Table 10 and its graphical representation in Figure 2 represent the outcome of the three learning algorithms obtained from tested fifty (50) to One Hundred (100) URLs.

Figure 1 shows a graphical representation of the comparative assessment of the algorithms when fifty URLs were tested. Figure 2 shows a graphical representation of assessment results of the three learning algorithms when the number of URL was increased from fifty (50) to One Hundred (100).

Figure 3 shows a graphical representation of assessment results of the three learning algorithms when the number of URLs was increased from One Hundred and One (101) to One Fifty (150). Table 11 and its graphical representation in Figure 3 show the outcome of the three learning algorithms obtained from the tested URLs. Table 12 and its graphical representation in Figure 4 show the outcome of the three learning algorithms resulted from One Fifty-One (151) to Two Hundred (200) URLs. Table 13 and its graphical representation in Figure 5 show the comparative assessment outcome of the three learning algorithms obtained when URL was increased from Two Hundred and One (201) to Two Fifty (250).

The result shown in Table 14 and its graphical representation in Figure 6 show the confusion matrix for the probability of result < 0.6 as 0 and result >= 0.6 as 1 for the three learning algorithms. The objective is to know an algorithm with the best level accuracy.

V. CONCLUSION AND FUTURE WORK

The evasion of anti-spam filtering techniques is made possible by hackers through embedment 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 14 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 and detection.

ACKNOWLEDGEMENT

The authors wish to acknowledge the efforts of anonymous referees for their valuable comments and helpful suggestions in shaping this paper into a publishable condition.

REFERENCES

Abu-Nimeh, S.; D. Nappa; X. Wang and S. Nair. (2007). A comparison of machine learning techniques for phishing detection, in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit. ACM, 60–69.

Adebowale, M.A.; K.T. Lwin; E. Sánchez and M.A. Hossain. (2019). Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text. Expert Systems with Applications 115:300–313.

Alnajim, A. and Munro, M. (2009). An Anti-Phishing Approach that Uses Training Intervention for Phishing Websites Detection. 2009 Sixth International Conference on Information Technology: New Generations. Las Vegas, NV, USA, USA: IEEE.

APWG. (2016). Phishing Activity Trends Report. USA: www.apwg.org.

Azeez, N. A. and Ademolu, O. (2016). CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification. 2016 International Conference Computational Science and Computational Intelligence (CSCI). Las Vegas, NV, USA: IEEE, 959-965.

Azeez, N. A. and Iliyas, H. D. (2016). Implementation of a 4-tier cloud-based architecture for collaborative health care delivery. Nigerian Journal of Technological Development, 13(1):17-25.

Azeez, N. A. and Venter, I. M. (2013). Towards ensuring scalability, interoperability and efficient access control in a multi-domain grid-based environment. SAIEE Africa Research Journal, 104(2): 54-68.

Blum, A.; B. Wardman; T. Solorio and G. Warner. (2010). Lexical feature based phishing URL detection using online learning, in Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security. ACM, 54–60.

Bo, S.; M. Akiyama; Y. Takeshi and M. Hatada. (2106). Automating URL blacklist generation with similarity search approach, IEICE TRANSACTIONS on Information and Systems, 99(4): 873–882.

Campbell, D. and Campbell, S. (2008). StatLab-IntroRegressionFa08. Retrieved May 04, 2017, from http://statlab.stat.yale.edu: http://statlab.stat.yale.edu/workshops/IntroRegression/StatLab-IntroRegressionFa08.pdf

Cao, J.; Q. Li; Y. Ji; Y. He and D. Guo. (2014). Detection of forwarding based malicious URLs in online social networks, International Journal of Parallel Programming, 1–18.

Chiba, D.; T. Yagi; M. Akiyama; T. Shibahara; T. Yada; T. Mori and S. Goto. (2016). Domainprofiler: Discovering domain names abused in future, in Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on. IEEE, 491–502.

Choudhary, N. and Jain, A. K. (2017). Towards Filtering of SMS Spam Mesages Using Machine Learning Based Technique. Advanced Informatics for Computing Research, First International Conference, ICAICR 2017. Jalandhar, India: Springer Nature Singapore, 18-30.

Chu, W.; B. B. Zhu; F. Xue; X. Guan and Z. Cai. (2013). Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs, in Communications (ICC), 2013 IEEE International Conference on. IEEE, 1990–1994.

Crawford, M.; T. M. Khoshgoftaar; A. N. Prusa; N. Richter and H. Al Najada. (2015). Survey of review spam detection using machine learning techniques, Journal of Big Data, 2(1): 23.

Doan, A.; R. Ramakrishnan and A. Y. Halevy. (2011). Crowdsourcing systems on the world-wide web, Communications of the ACM, 54(4):86–96.

Eshete, B.; A. Villafiorita and K. Weldemariam. (2013). Binspect: Holistic analysis and detection of malicious web pages, in Security and Privacy in Communication Networks. Springer, 149–166.

Fan, R.-E.; K.-W. Chang; C.-J. Hsieh; X.-R. Wang and C.-J. Lin. (2008). Liblinear: A library for large linear classification, The Journal of Machine Learning Research, 9: 1871–1874.

Felegyhazi, M.; C. Kreibich and V. Paxson. (2010). On the potential of proactive domain blacklisting. LEET, 10:6.

Heartfield, R. and Loukas, G. (2015). A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks, ACM Computing Surveys (CSUR), 48(3): 37.

Hou, Y.-T.; Y. Chang; T. Chen; C.-S. Laih and C.-M. Chen. (2010). Malicious web content detection by machine learning, Expert Systems with Applications, 37(1):55–60.

Hu, J.; H. Gao; Z. Li and Y. Chen. (2011). Poster: Cud: crowdsourcing for URL spam detection, in Proceedings of the 18th ACM conference on Computer and communications security. ACM, 785–788.

Joshi, Y.; S. Saklikar; D. Das and S. Saha. (2008). PhishGuard: A browser plug-in for protection from phishing. 2008, 2nd International Conference on Internet Multimedia Services Architecture and Applications. Bangalore, India: 1 - 6.

Kim, B.; I. Lm; C. T. and H. C. Jung. (2011). Suspicious malicious web site detection with strength analysis of a javascript obfuscation, International Journal of Advanced Science and Technology, 26: 19–32.

Kim, B.-I.; C.-T. Lm and H.-C. Jung. (2011). Suspicious malicious web site detection with strength analysis of a javascript obfuscation, International Journal of Advanced Science and Technology, 26:19–32.

Kirda, E. and Kruegel, C. (2005). Protecting users against phishing attacks with AntiPhish. 29th Annual

International Computer Software and Applications Conference (COMPSAC'05. Edinburgh, UK: IEEE, 1-8.

Kolbitsch, C.; B. Livshits; B. Zorn and C. Seifert. (2012). Rozzle: De-cloaking internet malware, in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 443–457.

Kolter, J. Z. and Maloof, A. M. (2006). Learning to detect and classify malicious executables in the wild. Journal of Machine Learning Research, 7: 2721-2744.

Lee, S. and Kim, J. (2013). WarningBird: A Near Real-Time Detection System for Suspicious URLs in Twitter Stream. IEEE Transactions on Dependable and Secure Computing, 10(3): 183-195 ·.

Ma, J.; L. K. Saul; S. Savage and G. M. Voelker. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1245–1254.

Ma, J.; L. K. Saul; S. Savage and G. M. Voelker. (2009). Identifying suspicious URLs: an application of large-scale online learning, in Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 681–688.

Ma, J.; L. Saul; S. Savage and G. Voelker. (2011). Learning to Detect Malicious URLs. ACM Transactions on Intelligent Systems and Technology, 2(3): 1-24.

Masud, M. M.; L. Khan and B. Thuraisingham. (2007). A Hybrid Model to Detect Malicious Executables. IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings, Glasgow, UK: 1443-1448.

McDonald, R.; K. Hall and G. Mann. (2010). Distributed training strategies for the structured perceptron, in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 456–464.

Nivedha, S.; S. Gokulan; C. Karthik and R. Gopinath. (2017). Improving Phishing URL Detection Using Fuzzy Association Mining. International Journal of Engineering and Science (IJES), 21-31.

Pao, H.-K.; Y. L. Chou and Y. J. Lee. (2012). Malicious URL detection based on kolmogorov complexity estimation, in Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01. IEEE Computer Society, 380–387.

Pao, H.-K.; Y.-L. Chou and Y.-J. Lee. (2012). Malicious URL detection based on kolmogorov complexity estimation, in Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01. IEEE Computer Society, 380–387.

Polat, H. and Du, W. (2003). Privacy-preserving collaborative filtering using randomized perturbation techniques. Third IEEE International Conference on Data Mining Melbourne, FL, USA, USA: IEEE, (1-15).

Qassrawi M. T. and Zhang, H. (2011). Detecting malicious web servers with honeyclients, Journal of Networks, 6(1):145–152.

Rokach, L. and Maimon, O. (2008). Data Mining with Decision Trees: Theroy and Applications. River Edge, NJ, USA: World Scientific Publishing Co., Inc.

Sahingoz, O.K.; E. Buber; O., Demir and B. Diri. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications 117: 345–357.

Sasan, K.; M. A. Shahidan, A. M. Azizah, Z. Mazdak and H. Alireza. (2013). An Overview of Principal Component Analysis. Journal of Signal and Information Processing, 4: 173-175

Seifert, C.; C. Welch and P. Komisarczuk. (2008). Identification of malicious web pages with static heuristics, in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, 91–96.

Seifert, C.; I. Welch and P. Komisarczuk. (2007). HoneyC - The Low-Interaction Client Honeypot. Proceedings of the 2007 NZCSRCS Hamilton, New Zealand: Waikato University: http://citeseerx.ist.psu.edu, 1–8.

Seifert, C.; I. Welch and P. Komisarczuk. (2008). Identification of malicious web pages with static heuristics, in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, 91–96.

Singh J. and Nene, M. J. (2013). A survey on machine learning techniques for intrusion detection systems, International Journal of Advanced Research in Computer and Communication Engineering, 2(11):4349–4355.

Sorio, E.; A. Bartoli and E. Medvet. (2013). Detection of hidden fraudulent URLs within trusted sites using lexical features, in Availability, Reliability and Security (ARES), 2013 Eighth International Conference on. IEEE, 242–247.

Sun, B.; M. Akiyama; T. Yagi; M. Hatada and T. Mori. (2015). Autoblg: Automatic URL blacklist generator using search space expansion and filters, in 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE, 625–631.

Surana, N.; P. Singh; U. Warade and N. Sabe. (2015). Detection and Prevention of Phishing Attacks in Web. International Journal of Scientific Engineering and Technology Research, 1595-1598.

Tao, Y. (2014). Suspicious URL and device detection by log mining, Ph.D. dissertation, Applied Sciences: School of Computing Science.

Thomas, K.; C. Grier; J. Ma; V. Paxson and D. Song. (2011). Design and evaluation of a real-time URL spam filtering service, in Security and Privacy (SP), 2011 IEEE Symposium on. IEEE, 447–462.

Vanhoenshoven, F.; G. Napoles; R. Falcon; K. Vanhoof and M. Koppen. (2016). Detecting Malicious URLs using Machine Learning Techniques. 2016 IEEE Symposium Series on Computational Intelligence (SSCI). Athens, Greece: 1-8.

Wang, J.; S. C. Hoi; P. Zhao; J. Zhuang and Z. Liu. (2013). Large scale online kernel classification," in Proceedings of Twenty-Third international joint conference on Artificial Intelligence, 1750–1756.

Wardman, B.; G. Shukla and G. Warner. (2009). Identifying vulnerable websites by analysis of common strings in phishing URLs, in eCrime Researchers Summit, eCRIME'09: 1–13.

Xiong, C.; P. Li; P. Zhang; Q. Liu and J. Tan. (2015). Mird: Trigrambased malicious URL detection implanted with random domain name recognition, in Applications and Techniques in Information Security. Springer, 303–314.

Yukun. L.; Y. Zhenguo; C. Xu; Y. Huaping and L. Wenyin. (2019). A stacking model using URL and HTML features for phishing webpage detection. Future Generation Computer Systems 94:27–39

Zhang, G.; T. W. Liu; W. Chow and W. Liu. (2011). Textual and visual content-based anti-phishing: a bayesian approach, IEEE Transactions on Neural Networks, 22(10):1532–1546.

Zhang, W.; Y.-X. Ding; Y. Tang and B. Zhao. (2011). Malicious web page detection based on on-line learning algorithm, in Machine Learning and Cybernetics (ICMLC), 2011 International Conference on, IEEE, 4:1914–1919.