International Journal of Computer Engineering in Research Trends

Multidisciplinary, Open Access, Peer-Reviewed and fully refereed

Research Paper Volume 7,Issue 08, August 2020 Regular Edition E-ISSN: 2349-7084

Detection of Malicious URLs using Artificial Intelligence

Monisha.T[1], Sridevi.R[2], Tirumalini.K.R[3]

[1]Computer Science and Engineering, S.A. Engineering College, Anna University, Chennai, India
[2] Computer Science and Engineering, S.A. Engineering College, Anna University, Chennai, India
[3] Computer Science and Engineering, S.A. Engineering College, Anna University, Chennai, India

E-mail: monishathirumalini1909@gmail.com, rsridevirajkumar@gmail.com, malini4499@gmail.com

Available online at: http://www.ijcert.org

Received: 24/07/2020 Revised: 06/08/2020 Accepted: 10/08/2020 Published: 01/09/2020

Abstract:- Background/Objectives: The main objective of the project is to avoid various security threats and network attacks by detecting malicious Uniform Resource Locator(URL) based on the keyword text classification.

Methods/Statistical analysis: A semi-supervised technique, naive Bayes classification is proposed to locate malicious URL by text classification phenomena. The probabilities of the predicted and the exact values are calculated and it results with high probability. With more accuracy the malignant URL is predicted. A page rank algorithm is used to detect the blacklist which contains the URLs that are already noted as spam, malware or phishing URL.

Findings: With the persistent improvement of Web assaults, many web applications have been languishing from different types of security dangers and system assaults. The security identification of URLs has consistently been the focal point of Web security. One of the main sources of attacks is via malicious URLs; the attackers may send embedding executable codes or injects malicious codes through these URLs. Thus, it is important to improve the unwavering quality and security of web applications by precisely identifying malignant URLs. The utilization of profound figuring out how to group URLs to recognize Web guests' aims has significant hypothetical and scientific values for Web security investigate, giving new plans to canny security discovery.

Keywords: Link analysis, malignant code, malignant URL datasets, naive Bayes classification.

1. Introduction

The technology in this modern world has an enormous growth. Due to this fast emerging technology there may be so many security threats and attacks. Security threats may include intruding the privacy of others and attacks may be defined as revealing or stealing of information without permission. These threats and attacks occur majorly through URLs. The URLs can be embedded with malicious codes which may get injected into the device once they are accessed and get the information in an illegal way without any notice. These attacks can be prevented by introducing techniques through which these harmful URLs can be detected and blocked.

In this paper it is shown that these attacks can be prevented by applying certain techniques and algorithms. Two algorithms are used to detect the URLs for their originality. They are Naive Bayesian algorithm and page ranking algorithm. Naive Bayes algorithm is a popular

https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0802

algorithm used for text categorization which means identifying the documents or text belonging to the same category or belonging to different categories. It uses Bayes theorem in text classification. Page ranking is a link analysis algorithm which is used to measure the significance of the web pages. It works by adding the quality and the number of links that are referred to a page.

The URLs are fed to the model by the user and they are checked by matching the labels that are stored. It contains collection of labeled URLs where they are categorized as malicious or benign. Once the URL is found in the labels then it is predicted as malicious. These URLs are given to the model training to train those URLs with labels.

The remaining paper is organized as follows: Section 2 describes related work, Section 3 describes the Methodology, Section 4 describes the Results and Discussion, Section 5 describes the Conclusion and Future scope, Section 6 contains the References.

2. Related Work

Detection of Malicious URL is one of the biggest concerns in online media. Malicious URL may occur in any kind of online media and may cause serious issues in the network. There are more techniques in introducing malicious URL in the system.

Mohammed Al-Janabi et al.,[13] described about the supervised machine learning techniques to detect the malicious contents in the social media platform like Twitter. Justin Ma, Lawrence et al.,[14] analyzed the suspicious features of an URL using the host based properties of malicious websites.

Eric Lancaster et al.,[5] identified the embedded URL and content with wrong images, videos, and sounds. They seduced their work by determining how many wrong click that leads to malware injection to the system. They use five class features to classify the type of URL in the media. Suren Wu et al.,[4], Bo Feng et al.,[2] proposed a multistage and elastic detection framework which is based on the deep learning. It has a setup of detection system at the server and the mobile terminal. The messages are first detected by the mobile terminal and then transmitted to the server and it will detect messages elastically. Sina Weibo dataset is used in evaluating the detection framework. As a result of experiment shows the utilization rate of computing recourses. Surendra Sedhai et al.,[1] analyzed the spam detection technique which mainly focuses on blocking user who post spam tweets. Semi-supervised spam detection is proposed to detect spam at tweet level. There are two modules .They are spam detection and model update .Spam detection has four detectors.

They are blacklisted domain detector that detects the tweets that contains blacklisted URLs, near duplicate detector labels the near-duplicates, reliable ham detector labels the tweets that are posted by benign users and multi classifier-based detector labels all the remaining tweets. The information is updated in the batch module which is required by the detection module. It can learn the new patterns of new spam activities and it maintains better accuracy.

3. Methodology

The malicious URLs are detected using two major algorithms. One is Naïve Bayesian algorithm and the other is Page rank which is a link analysis algorithm. The URLs are taken as labels and the labels are trained to the system. This labelling helps in identifying the URLs originality. When the URLs are fed by the user they are sent to the feature extraction and after that they are directed to the prediction phase where they are classified as malicious or benign URLs. When the URLs enter they are checked for the labels in the collection that has labelled URLs. If they are present then they are transferred into the model training phase and they are predicted whether they are malicious or not malicious. Once they are detected the results are obtained.

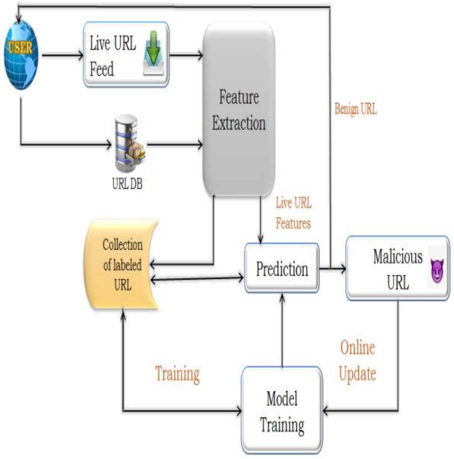

Figure1: Architecture of Malicious URL Detection System

3.1 Live URL Feed Phase

The URLs that are needed to be checked are fed to the model by the user. This module provides the information that is related to the User URL feed. Once the user feeds the URLs they are sent to the database through the internet. The URLs are checked whether they are available in the web or not.

3.2 Labelled URLs Collection

The collection of labelled benign URLs and malignant URL are available in this module. This module lists the malicious URLs into three types of listed URLs as spam, phishing and malware URL list. When the URLs are being checked for the availability of the labels this module allows to compare the user fed URLs with already existing labels. When the labels are matched then they are transferred to the model training phase.

3.3 Prediction with Model Training

This module describes the URL prediction and explains the how to predict the URL. Model training already have some model prediction. This model is used in prediction process. Once receive the URL link Model training check the malicious in that URL. Model training does not allow the detection of malicious URLs when they already exist in the collection. It directly predicts the URLs as malicious or benign.

3.4 Malicious URL Prediction Phase

Malicious URLs are generally used to scale various cyber-attacks including spam, malware and phishing. Detection of malicious URLs and identification of threat types are critical to identify these attacks. This method uses features like textual properties, link structures, webpage contents, Domain Name Service information (DNS), and network traffic.

4. Results and Discussion

4.1 Experimental Framework

The URLs that are malicious must be split into datasets. These datasets are labelled and are stored as labels in the collection. Datasets can be said to be as the set of information which are composed of several different elements but can be interpreted as a single source. The datasets that are stored allows the system to predict the new sets of data that enters the system which are fed by the user. Once these datasets are predicted they are gradually sent to the model training. The dataset is divided as training set and testing set. The three–fourth of the dataset is training set and the remaining dataset is testing set. Splitting of dataset consists of two portions of data. One portion is to promote a predictive model and the next portion is to analyze the models performance.

The datasets are classified with the help of naive Bayesian algorithm. Naïve Bayes is not a single algorithm rather it is a collection or group of several similar algorithms. All these algorithms follow the similar principle in which every pair of data or feature that is classified will be different from one another. The datasets can be divided into two major portions. They are feature matrix and response vectors. Feature matrix contains all the features which are arranged in a row and the response vector predicts the output for all the rows in the matrix.

Once the classification is completed the datasets are trained and stored. When a new dataset is fed by the user, the system first checks whether the labels are already available in the collection. If it is available it directly predicts the outcome. If the same datasets are not available then the datasets are sent to the model training and prediction phase to classify them. Once they are classified page rank algorithm helps in finding the effect of the identified URL. It collects data from several websites and produces the probability of the harmfulness of the identified URL. After this prediction the results are obtained and are displayed to the user.

	Verification Set	
Training Set	True Positive 7342	False Positive 307
	False Negative 121	True Negative 2230

Table1: Accuracy study of URL dataset.

Accuracy from the given dataset is found to be: 95.72%

4.2 Experiment Outcomes

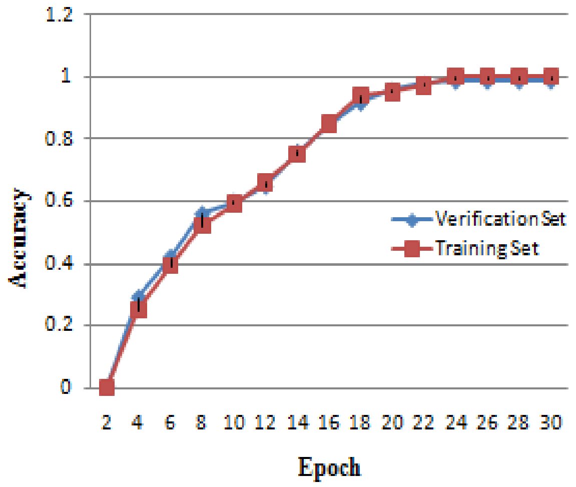

Figure2: Accuracy Curve of Naive Bayes Model on Training Set and Verification Set

The above figure2 showcase the accuracy rate between the training set and verification set over the period. The URL dataset contains both benign and malicious URLs which are classified using Naïve Bayes classifier corresponding to its features.

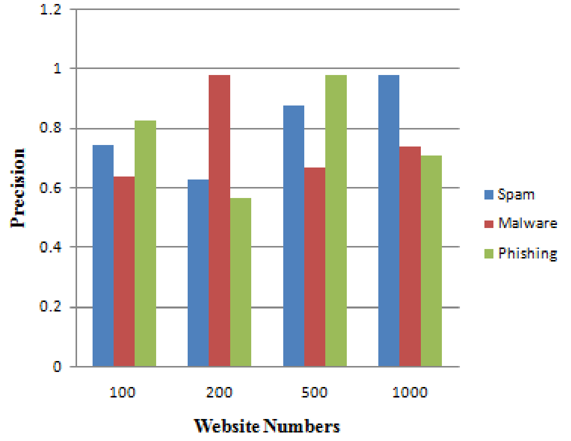

Figure3: Precision graph of websites that are reported as Spam, Malware or Phishing

This experimental result shows that the accuracy of the system is high using naïve Bayes classification and page rank algorithm when compared to the previous outcomes of various methods.

4.3 Comparison with Existing Systems

Malicious URL detection or spam detection has a major role in this modern world. These spam contents may affect the users systems and may cause severe damage. So it is very essential to prevent them from getting attacked. The previous works have given greater results in identifying malwares.

Some experiments have focused on a particular type of spam and have achieved good results in identified them. In a paper named Semi-supervised spam detection in Twitter stream by Surendra sedhai et al., [1] it is mainly focused on blocking users those who post spam tweets. This provides the results with good accuracy in the case of twitter attacks. Eric Lancaster et al., [5] identified the embedded URL and content with wrong images, videos, and sounds. This helped in identifying the URLs and made it easy to overcome the attacks.

Similarly, in this paper we have discussed a method to identify those malicious URLs which may cause damage to the users. There may be embedded codes inside those URLs which allows the spammers to embezzle the user contents. This can be avoided by detecting those malicious URLs. Naïve Bayes and Page rank detects the URLs in a more accurate way which helps in reducing the network and security attacks.

Label	URL (Unique values)
Benign (Good)	337222.54(82%)
Malicious (Bad)	74024.46(18%)
Total	**411247**

Table2: URL Dataset Distribution

4.4 Naive Bayes Algorithm

Naive Bayes Algorithm is based on Bayes theorem which has independence supposition that takes place between the features. It dwells in the group of probabilistic classifiers. Naive Bayes Algorithm is extremely scalable. It requires several parameters that are linear to the number of variables. Naive Bayes is also referred as independence Bayes and simple Bayes.

4.5 Feature Matrix

In machine learning the feature matrix is a term that consists of the columns with independent variables that needs to be refined. Feature matrix helps to differentiate between similar items. When machine learning, the feature selection is used to train faster. It is used to minimize the intricacy of a model. It is used increase the accuracy. Feature matrices are used to describe the meaning of certain word fields.

5. Conclusion and Future Scope

In this work, we have described how a URL detection system is capable to judge the URLs based upon the given data set. Specifically, we defined the feature set and an approach for classifying the given data set for malicious URL detection. When traditional approach falls short in detecting the new malicious URLs on its own, our proposed approach may be supplemented with the approach.

Some of the traditional methods are black listing and heuristic classification. Blacklisting means removing or blocking the particular URL that is malicious. Heuristic classification has a combination of several observations which can detect the intermediary or the final outcome. In these methods only the URLs which are already identified as malicious can be blocked or prohibited from accessing.

By implementing Naive Bayes Algorithm the features of the URLs can be identified even if the URL doesn't exist in the predefined URL set. In this way it provides better results than the other traditional methods.

It is anticipated to provide enhanced results. Here in this work, we proposed the feature set which is capable of classifying the URLs. The Future work is to fine tuning the machine learning algorithm that will produce the better result by making use of the given feature set. Adding to that the open query is how we will cope with the huge variety of URLs whose features set will evolve over time. Certain efforts have to be made in that direction so as to come up with the more robust feature set which can change with respect to the evolving changes.

References

[1] Surendra Sedhai; AixinSun&quot; Semi-Supervised Spam Detection in Twitter Stream&quot; IEEE Transactions on Computational Social Systems (Volume: 5, Issue: 1, March 2018)

[2] Bo Feng; Qiang Fu; Mianxiong Dong; Dong Guo; Qiang Li &quot;Multistage and Elastic Spam Detection in Mobile Social Networks through Deep Learning&quot;IEEE Network (Volume: 32 , Issue: 4 , July/August 2018)

[3] Jonghyuk Song; Sangho Lee; Jong Kim &quot; Inference Attack on Browsing History of Twitter Users Using Public Click Analytics and Twitter Metadata&quot; IEEE Transactions on Dependable and Secure Computing (Volume: 13, Issue: 3, May-June 1 2016)

[4] Longfei Wu; Xiaojiang Du; JieWu&quot; Effective Defense Schemes for Phishing Attacks on Mobile Computing Platforms&quot; IEEE Transactions on Vehicular Technology (Volume: 65, Issue: 8, Aug. 2016)

[5] Eric Lancaster ; TanmoyChakraborty ; V. S. Subrahmanian&quot;MALTP : Parallel Prediction of Malicious Tweets&quot;IEEE Transactions on Computational Social Systems(Volume: 5 , Issue: 4 , Dec. 2018)

[6] Hong Zhao; Zhaobin Chang; Weijie Wang; XiangyanZeng&quot; Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification&quot; IEEE Access (Volume: 7)

[7] Xuanzhe Liu ; Yun Ma ; Xinyang Wang ; Yunxin Liu ; Tao Xie ; Gang Huang&quot;SWAROVsky: Optimizing Resource Loading for Mobile Web Browsing&quot;IEEETransactions on Mobile Computing(Volume: 16 , Issue: 10 , Oct. 1 2017)

[8] DohoonKim&quot; Potential Risk Analysis Method for Malware Distribution Networks&quot; IEEE Access (Volume: 7)

[9] JoostBerkhout"Google's PageRank algorithm for ranking nodes in general networks"IEEE 2016 13th International Workshop on Discrete Event Systems (WODES)

[10] Zhou Hao; PuQiumei; Zhang Hong; ShaZhihao"An Improved PageRank Algorithm Based on Web Content" IEEE 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

[11] Yuguang Huang ; Lei Li"Naive Bayes classification algorithm based on small sample set" IEEE 2011 IEEE International Conference on Cloud Computing and Intelligence Systems

[12] HaiyiZhang; Di Li"Naïve Bayes Text Classifier" IEEE 2007 IEEE International Conference on Granular Computing (GRC 2007)

[13] Mohammed Al-Janabi: Ed de Quincey: Peter Andras: "Using supervised machine learning algorithms to detect suspicious URLs in online social networks" 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

[14] Justin Ma: Lawrence K. Saul: Stefan Savage: Geoffrey M. Voelker:"Identifying Suspicious URLs: An Application of Large-Scale Online Learning" 26th International Conference on Machine Learning, Montreal, Canada, 2009.

[15] Training Datasets: https://www.kaggle.com/teseract/urldataset

https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0802