JCRES Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6 –Issue 2,August 2023
Paper:61

MALICIOUS WEBSITE (URL) DETECTION USING MACHINE LEARNING TECHNIQUES

Ms. B. Suganya[1]
Assistant Professor
Department of Computer Science and Engineering
Dr. Mahalingam College of Engineering andTechnology
Pollachi, Tamilnadu, India.
suganyab@drmcet.ac.in

Silambarasan K[2]
UG Scholar Department of Computer Science and Engineering
Dr. Mahalingam College of Engineering andTechnology
Pollachi, Tamilnadu, India.
19bcs057@mcet.in

Siva P[3]
UG Scholar Department of Computer Science and Engineering
Dr. Mahalingam College of Engineering andTechnology
Pollachi, Tamilnadu, India.
19bcs059@mcet.in

Martin Joy J[4]
UG Scholar Department of Computer Science and Engineering
Dr. Mahalingam College of Engineering andTechnology
Pollachi, Tamilnadu, India.
19bcs067@mcet.in

Abstract—**Many unlawful activities, consisting of unsolicited mail-advertised e-commerce, economic fraud, and the spread of malware, are aided by the sector huge internet. The reality that unknowing visitors go to their web sites is the regular element all through those schemes, in spite of the fact that the precise objectives at the back of them may additionally range. email, on-line seek results, or connections from other websites can all encourage those visits. yet, the person need to usually carry out some form of movement, such clicking on a particular standard resource Locator (URL). Blacklisting offerings have been installed via the online security community with the intention to discover those dangerous web sites. Many harmful websites are always no longer blacklisted both because they had been never or incorrectly examined, or because they had been too current. on this paper, we examine the performance of many popular classifiers, inclusive of logistic regression, assist Vector Machines, choice bushes, Random wooded area, and k-Nearest Neighbors, in the detection of risky URLs as a binary classification hassle. additionally, we used a publicly to be had dataset made of 32 attributes. Numerical simulations have established that maximum classification algorithms are able to obtain first rate prediction costs without using both specialised feature selection methods to get the quality accuracy .**

Keywords—Machine learning, Decision Trees, Random Forest, Logistic Regression, K-Nearest Neighbors (KNN) Classifier, Support Vector Classifier (SVC)

I. INTRODUCTION

Our lives are now completely dependent on the internet, providing a wide range of services and resources. Yet, there has been an increase in cybercrime due to the increasing usage of the internet Cyberattacks through malicious websites are one of the most common types. Malicious websites are built with the intention of stealing users' sensitive data or infecting their devices with malware. Machine learning algorithms have become very popular in recent years for detecting dangerous websites. Machine learning algorithms have the ability to scan vast volumes of data and spot trends that point to criminal activities. The objective of this research is to create a machine learning model for identifying harmful websites. The objective of this project is to reliably identify harmful websites using machine learning algorithms and safeguard users from online threats. Data collection for the project will concern the domain name, IP address, content, and links of websites. The data will be pre-processed to highlight key features and eliminate ambiguity. Among other machine learning techniques, we will evaluate neural networks, support vector machines, decision trees, and random forests. Evaluation criteria including accuracy, will be used to compare the effectiveness of various algorithms. The final model will implement the algorithm that performs the best overall. Python and appropriate machine learning libraries like numpy Scikitlearn, pymongo ,pickle will be used to carry out the project. Users will be able to access the final model as a web-based service to check the security of websites. Ultimately, by giving consumers a dependable tool for identifying fraudulent websites, the project will contribute to improving internet security.

II. LITERATURE SURVEY

[1] MALICIOUS URL DETECTION USING SUPERVISED MACHINE LEARNING TECHNIQUES

Vara Vundavall et al [1] had proposed the method of blacklisting, as Malicious websites are a number one method of net crimes. Attackers can also attempt to get non-public records via malicious URL. those malicious URLS can reason untrustworthy sports, for instance, burglary of private and mystery statistics, ransomware set up on the customer devices that brings approximately large misfortunes constantly all around. Many Mitis clients continued to broadcast unauthorised URLS with the emergence of social media platforms that allowed interaction. Many of those URLS were later identified as self- and enterprise-advertising. However, several of those amazing supplier locators could pose a risk to green clients. The competitors' considerable security risks would be faced by the gullible clients who use harmful URLs. To ensure that the client should refrain from visiting harmful websites, the URL must be tested. There are many methods that have been suggested to distinguish rogue URLs. one of the predominant traits is - a factor should allow the benign URLs which can be stated by the consumer, prevent the malicious URLs earlier than accomplishing to users and alert customers. as opposed to relying on syntactical homes of the URLs, a gadget need to bear in mind homes of URL. traditional procedures, for instance, Black-list and Heuristic category, can apprehend those URLs and perceive them earlier than reaching to the client.

[2] USING PASSIVE DNS TO DETECT MALICIOUS DOMAIN NAME

Zhouyu Bao, et al [2] has proposed a domain call detection approach d based totally on word vector in aggregate with the language community surroundings . Considering how popular the internet is, The variety of bad domain names is substantial, and the severity and extent of the risks they create are expanding. Traditional popularity systems and opposite engineering techniques cannot be used in real-time to detect harmful area calls, and the process of doing so is cumbersome and difficult. This work uses passive DNS as the analytical records and machine learning to build a malicious domain name category detection version, which will compensate for the shortcomings and maintain accuracy. in line with the access traits and man or woman traits of area call, They created a comprehensive framework for function evaluation and suggested a multi-dimensional DGA domain call detection technique. Finally, they implemented prototype systems for detecting malicious domain names and got the desired results.

[3] EFFECTIVE ANALYSIS, CHARACTERIZATION, AND DETECTION OF MALICIOUS WEB PAGES

Birhanu Eshete [3] had cautioned the strategies to discover malicious internet pages were reactively powerful at unique training of assaults like pressure-by using-downloads. however, the superiority and The sophistication of attacks carried out by malicious websites is still concerning. Best-grained shooting and characterisation of assault payloads, growth of internet website artefacts and exhibitions, and scalability of detection systems in a rapidly changing threat landscape are the major challenging scenarios on this hassle domain. To do this, they put out a comprehensive approach that effectively scans and identifies harmful web sites by combining static assessment, dynamic evaluation, machine learning, and evolutionary searching and optimisation. They accomplish that through: introducing novel functions to seize satisfactory-grained picture of malicious internet pages, the comprehensive characterisation of dangerous web pages, as well as the awareness of evolutionary methods to excellent-song learning-based detection models relevant to the evolution of attack payloads.

[4] A WEBSITE DEFACEMENT DETECTION METHOD BASED ON MACHINE LEARNING TECHNIQUE

Xuan Dau Hoang [4] has proposed a technique for internet site defacement tracking and detection, including DOM tree assessment, checksum comparison, diff comparison, and sophisticated algorithms. However, some of them merely use static web pages, and the others demand a lot of processing power. A method based on system optimisation for detecting website defacement. Our approach uses device learning approaches to build classifiers (detection profiles) that divide pages into regular and attacked elegance categories. Our method is effective for both static and dynamic web sites since the detection profile can be learned from training data. According to experimental findings, our method achieves excellent detection accuracy of over 93% and a low false effective rate of significantly less than 1%. Additionally, the method is suitable for online deployment because it no longer calls for enormous computational resources.

[5] WEBSITE DEFACEMENTS DETECTION BASED ON SUPPORT VECTOR MACHINE CLASSIFICATION METHOD

Siyan Wu, et al [5] has advised some gadget studying algorithms which produces the excessive accuracy results . The loss of reputation, financial loss, or data leakage caused by website defacements can cause the website owner to suffer greatly Website defacements increased year over year due to the complexity and diversity of many online application structures, notably due to a lack of basic safety upgrades. By extracting website features and trojan features encoded in websites, they may determine whether the website has been altered. they used 3 styles of category gaining knowledge of algorithms which consist of The classification trials used Gradient Boosting Decision Trees (GBDT), Random Forests (RF), and the Support Vector Machine (SVM) thirteen, and the results show that the Support Vector Machine classifier performed better than other classifiers. It can identify website defacements with a 95%–96% accuracy across the board.

III. CLASSIFICATION TECHNIQUE

3.1 Logistic Regression

Logistic regression is widely recognized as one of the most in-demand approaches for machine learning when it comes to the class of "supervised learning". It leverages a group of independent variables to project the prospects of a categorical dependent variable. As categorical data typically possess fewer possible values, they allow for an easier analysis. Furthermore, this category can take on meanings aloud or written down using a textual form. Logistic regression approaches the task by projecting outcomes through assorted dependent categorizations, with results programmed into categories exclusive or discrete instead of precise values between 0 and 1. Probabilistic imprints shuttle between these two extremes rather than expressing exactness. Select one outcome over another boils down simply to 'Yes' or 'No', 'True' or 'False", or even 'One' and 'Zero'.

3.2 Support Vector Machine

To solve classification and regression issues, support vector machines (SVMs), one of the most well-liked supervised learning techniques. However, most of its uses are for machine learning classification. The SVM algorithm attempts to identify optimal decision boundaries or lines that allow subspace classification in n-dimensional space.

3.3 Random Forest

An effective supervised learning method is the well-known machine learning algorithm Random Forest. ML classification and regression problems can be Handled with it. It builds on the idea of ensemble learning, which mixes different classifiers to address difficult problems and improve model performance.

Rather than relying solely on decision trees, Random Forest determines outcomes based on which predictions are the most supported. Increasing the number of trees in the forest improves accuracy and prevents overfitting.

3.4 Decision Trees

A decision tree is a tree-like model that is built by recursively splitting the dataset based on the most in a description of how formative features . Each can be the dataset into two or more subsets, with the goal of maximizing the separation between malicious and benign websites. Decision trees are easy to interpret and can handle both categorical and numerical features.

IV. PROPOSED SYSTEM

In this paper, a system is developed for detecting the malicious website using the decision tree algorithm is proposed. The algorithm makes use of the dataset of website information, including features such as URL length, domain age, content type, page rank, etc. the decision tree algorithm to build a model that can classify websites as malicious or benign based on their features. The algorithm splits the dataset recursively into subsets using the most significant feature at each level, based on an impurity measure like Gini or entropy. We evaluate the performance of the model using metrics such as accuracy. Then we deploy the model to detect malicious websites in real-time by extracting their features and feeding them into the trained decision tree model. The model will output a prediction of whether the website is malicious or benign, based on the decision tree rules and Finally we Deploy the model to detect malicious websites in real-time by extracting their features and feeding them into the trained decision tree model. The model will output a prediction of whether the website is malicious or benign, based on the decision tree rules.

4.1 Data Collection

In data collection, the dataset is obtained from a website called kaggle.com which contains data of block list and legit URL of various websites. The dataset is organized so that the greater part of its residences and statistics is looked after out and put away independently in compact CSV record.

4.2 Data Preprocessing

In this stage, the collected data set is preprocessed by importing the libraries and reading the data set. In the data set consists of null values which will result in predicting wrong result for that we are handling the null values so that the data is much meaning for the prediction process in final the null handled data set is spitted into the test data and training data for the further training of the model.

4.3 Building the Model

In this stage , A machine learning model for detecting malicious websites can be created using a variety of classification algorithms. Some of the algorithms implemented in this project are listed below:

Decision Trees: A decision tree is a tree-like model that is built by recursively splitting the dataset based on the most in a description of how formative features . Each can be the dataset into two or more subsets, with the goal of maximizing the separation between malicious and benign websites. Decision trees are easy to interpret and may handle both specific and numerical features.

Random Forest: A remarkable method for enhancing forecasting accuracy is the deployment of a random forest, which involves integrating an ensemble of decision trees. Each tree in this ensemble is constructed employing specific characteristics selected at random from a data subset. By pooling together the predictions made by these independent trees, it becomes possible to obtain highly accurate forecasts. Furthermore, such models are adapted and resilient enough to handle high-dimensional data without encountering overfitting challenges.

Logistic Regression: One oft-utilized method for classification, referred to as logistic regression, involves the use of a logistic function in order to make predictions regarding the probability of a binary outcome. It works by estimating the parameters of a logistic function using a set of input features and a labeled dataset, and then using this function to make predictions on new data.

K-Nearest Neighbors (KNN) Classifier: The KNN algorithm employs non-parametric classification techniques to determine the k-nearest neighbours of a given data point. These identified neighbours' labels are then employed to forecast the label of new data points that may require classification. The value of k determines the number of neighbors to consider

Support Vector Classifier (SVC): The Support Vector Classifier is a popular classification algorithm that works by finding the optimal hyperplane that separates the different classes in the data. It seeks to maximize the distance between each class's closest data points and the hyperplane. Using the kernel method, it can handle non-linearly separable data.

4.4 Building the Web Page

In this stage we build a web page for the malicious website detection project using Flask, through the following process:

Setting Flask environment: Install Flask and any other necessary libraries, and create a new Flask application.

Defining routes: Decide on the different pages you want your web application to have, and define routes for each page. For example, you might have a homepage, a page for displaying the results of the malicious website detection, and a page for uploading a URL to check.

Creating templates: Use HTML, CSS, and JavaScript to create the templates for web pages. And using a template engine like Jinja to generate dynamic content, such as displaying the results of the detection.

Defining the functions that will handle the logic for each of your routes. For example a function that checks a URL for malicious content and returns the result.

Using the Flask development server to test the application

Deploying the application: When you're ready to make your application available to the public, deploy it to a web server or cloud service provider. Make sure to configure any necessary security settings to protect against potential attacks.

Overall, building a web page for the malicious website detection project using Flask involves creating HTML templates, defining routes and views, integrating the detection algorithm, and testing and deploying the application.

4.5 Integrating the Model to Web Page

Deploying the trained model to a webpage involves integrating the machine learning model into the web application to enable real-time malicious website detection. Here are the key steps involved in deploying the trained model to a webpage:

Model Serialization: The first step in deploying the trained model is to serialize it into a file format that can be easily loaded into the web application. This is typically done using libraries such as Pickle or Joblib in Python.

Model Integration: The serialized model is then integrated into the web application code. This involves importing the necessary libraries and functions and loading the model into memory.

Testing and Debugging: Once the web application is complete, it should be thoroughly tested and debugged to ensure that it works correctly and provides accurate predictions.

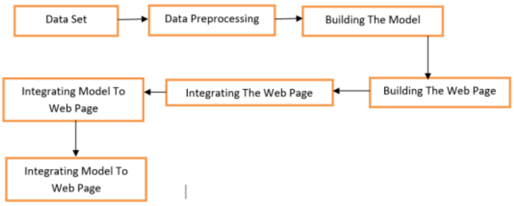
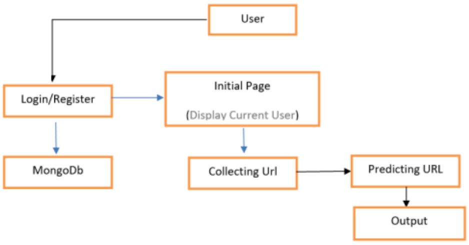


Figure 1: Block Diagram



V. RESULT

5.1. Performance Metrics

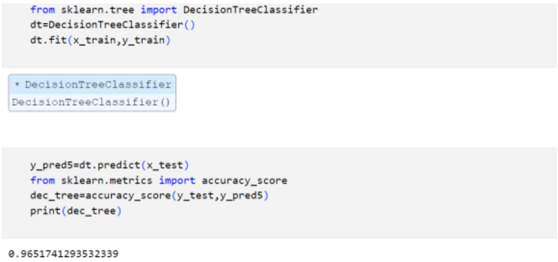
Accuracy

Accuracy is the most common method used to validate machine learning models to assess classification problems. One measure of model performance across all classes is accuracy. It helps if all classes are equally important.The computation is based on the total number of guesses divided by the number of right predictions. Its relative simplicity is a factor in its popularity. Easy to understand and easy to practise. In simple cases, accuracy is a useful metric for evaluating model performance. Using the below formula dataset accuracy is calculated.

$$\text{Accuracy} = ((TP + TN) / (TP + TN + FP + FN))$$

5.2. Performance Evaluation

In this model we have used classification algorithms like Decision Tree Classifier, Logistic Regression, Support Vector Classifier, Random Forest, and KNeighbors Classifier. Fig.5a shows the accuracy of the Decision Tree Classifier and it's the most popular Machine Learning techniques. Fig.5b shows the accuracy of the Logistic Regression, it's the most well-liked supervised learning techniques. Fig.5c represents the accuracy of the KNeighbor algorithm. Fig.5d represents the accuracy of Support Vector Machine. Fig.5e represents the accuracy of Random Forest algorithm. From this it is concluded as Decision Tree predicts the highest accuracy.



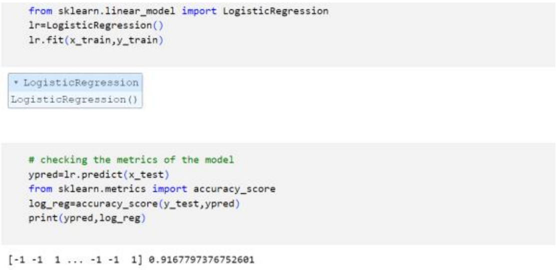
```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_pred5=dt.predict(x_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred5)
print(dec_tree)
```

```
0.9651741293532339
```

Fig.5. a. Decision Tree Classifier



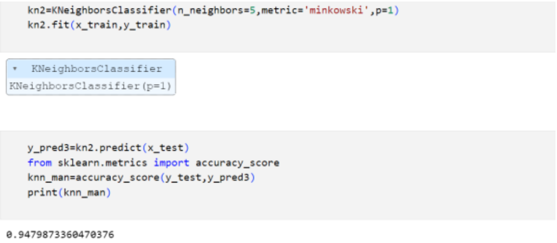
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
# checking the metrics of the model
ypred=lr.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score(y_test,ypred)
print(ypred,log_reg)
```

```
[-1 -1  1 ... -1 -1  1] 0.9167797376752601
```

Fig.5. b. Logistic Regression



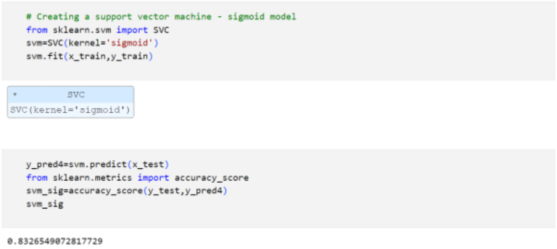
```
kn2=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=1)
kn2.fit(x_train,y_train)
```

```
▾  KNeighborsClassifier
KNeighborsClassifier(p=1)
```

```
y_pred3=kn2.predict(x_test)
from sklearn.metrics import accuracy_score
knn_man=accuracy_score(y_test,y_pred3)
print(knn_man)
```

```
0.9479873360470376
```

Fig.5. c. KNeighbors Classifier



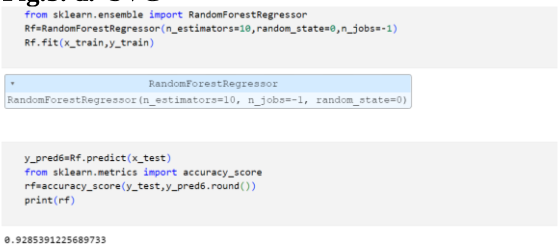
```
# Creating a support vector machine - sigmoid model
from sklearn.svm import SVC
svm=SVC(kernel='sigmoid')
svm.fit(x_train,y_train)
```

```
▾          SVC
SVC(kernel='sigmoid')
```

```
y_pred4=svm.predict(x_test)
from sklearn.metrics import accuracy_score
svm_sig=accuracy_score(y_test,y_pred4)
svm_sig
```

```
0.8326549072817729
```

Fig.5. d. SVC



```
from sklearn.ensemble import RandomForestRegressor
Rf=RandomForestRegressor(n_estimators=10,random_state=0,n_jobs=-1)
Rf.fit(x_train,y_train)
```

```
▾                    RandomForestRegressor
RandomForestRegressor(n_estimators=10, n_jobs=-1, random_state=0)
```

```
y_pred6=Rf.predict(x_test)
from sklearn.metrics import accuracy_score
rf=accuracy_score(y_test,y_pred6.round())
print(rf)
```

```
0.9285391225689733
```

Fig.5. e. Random Forest

VI. CONCLUSION

This paper provides a model that has been developed using a stored dataset with close to 1750 URLs and employing a variety of machine learning algorithms, from Support Vector Machines to Logistics Regression. Following training and testing iterations, it is discovered that Decision Tree generates the maximum accuracy. A decision tree generates results with 96% accuracy.

Future study will be a web browser plug-in that can identify Malicious websites and shield consumers in real time will be created based on an effective algorithm

REFERENCES

[1] Vara Vundavall, Farhat Barsha, Mohammad Masum, Hossain Shahriar, Hisham Haddad," Malicious URL Detection Using Supervised Machine Learning Techniques " International Conference on Security of Information and NetworksNovember 2020 Article No.: 21 https://doi.org/10.1145/3433174.3433592

[2] Zhouyu Bao, Wenbo Wang, Yuqing Lan," Using Passive DNS to Detect Malicious Domain Name ",ICVISP 2019: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing 2019, https://doi.org/10.1145/3387168.3387236

[3] Birhanu Eshete," Effective analysis, characterization, and detection of malicious web pages ",WWW '13 Companion: Proceedings of the 22nd International Conference on World Wide Web May 2013

[4] Xuan Dau Hoang," A Website Defacement Detection Method Based on Machine Learning Technique “, Proceedings of the 9th International Symposium on Information and Communication Technology December 2018 https://doi.org/10.1145/3287921.3287975

[5] Siyan Wu, Xiaojun Tong, Wei Wang, Guodong Xin, Bailing Wang, Qi Zhou," Website Defacements Detection Based on Support Vector Machine Classification Method ",Proceedings of the 9th International Symposium on Information and Communication Technology December 2018 https://doi.org/10.1145/3287921.3287975

[6] Nguyen Bac Trinh, The Duy Phan, Van-Hau Pham," Leveraging Deep Learning Image Classifiers for Visual Similarity-based Phishing Website Detection ",SoICT '22: Proceedings of the 11th International Symposium on Information and Communication Technology December 2022 https://doi.org/10.1145/3568562.3568629

[7] Md. Abu Ashraf Siddiq, Mohammad Arifuzzaman, M. S. Islam," Phishing Website Detection using Deep Learning ",ICCA '22: Proceedings of the 2nd International Conference on Computing Advancements March 2022 https://doi.org/10.1145/3542954.3542967

[8] Ch Rupa, Gautam Srivastava, Sweta Bhattacharya, Praveen Reddy, Thippa Reddy Gadekallu ," A Machine Learning Driven Threat Intelligence System for Malicious URL Detection ", Proceedings of the 16th International Conference on Availability, Reliability and Security August 2021 https://doi.org/10.1145/3465481.3470029

[9] Song Tan, Runyuan Sun, Zhifeng Liang," Detection of Malicious Web Requests Using Neural Networks with Multi Granularity Features ",ICBDT '22: Proceedings of the 5th International Conference on Big Data Technologies September 2022 https://doi.org/10.1145/3565291.3565304

[10] Marc Ohm, Felix Boes, Christian Bungartz, Michael Meier," On the Feasibility of Supervised Machine Learning for the Detection of Malicious Software Packages ",ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security August 2022 https://doi.org/10.1145/3538969.3544415

[11] Yi-Chung Tseng, Wei-An Chen," Hunting Malicious Windows Commands with Multi Machine Learning Technologies ",CMLT 2021: 2021 6th International Conference on Machine Learning TechnologiesApril 2021 https://doi.org/10.1145/3468891.3468893

[12] Meghna Dhalaria, Ekta Gandotra," Detecting Android Malicious Applications using Dynamic Malware Analysis and Machine Learning ",IC3-2022: Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing August 2022 https://doi.org/10.1145/3549206.3549271

[13] Yury Zhauniarovich, Issa Khalil, Ting Yu, Marc Dacier," A Survey on Malicious Domains Detection through DNS Data Analysis ",ACM Computing SurveysVolume https://doi.org/10.1145/3191329

[14] Mohammed Abutaha, Mohammad Ababneh, Khaled Mahmoud, Sherenaz Al-Haj Baddar," URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis" .