Phishing URL Detection using Hybrid Ensemble Model

Anurag Pandey
3rd year B.Tech Computer Science,
Vellore Institute of Technology,
Vellore

Jay Chadawar
3rd year B.Tech Computer Science,
Vellore Institute of Technology,
Vellore

***Abstract*— Nowadays we hear the news of people losing their money by unknowingly performing transactions through a given link by an anonymous person. There are several ways of defraud people like email, SMS, calls, fake websites or even face to face. These types of attacks or defraud people are called phishing attacks. So, in this project we are focusing on one of the methods by which a phishing attack can be done, that is, by using a malicious URL or website. It is hard to identify whether a URL visited by anyone is legitimate or not because these URLs are written in such a way that it looks almost similar to a legitimate URL. These malicious URLs may be sent in private or in public and if there is no system used for blocking or removing these malicious URLs, soon the credentials of the user accessing the link will be transferred to the attacker. Aim of our project is to build a machine learning based model which helps in classifying whether a URL is safe to use or not. Objective of this project is to identify malicious URLs and to build an accurate machine learning model for identification of malicious and legitimate URLs.**

Keywords*— *Classification, phishing, URL, ensemble model

I. INTRODUCTION

In today's environment, phishing is still a major source of security issues and the majority of cyber-attacks. According to Cisco's 2021 Cybersecurity Threat Trends report, at least one person in 86 percent of firms clicked on a phishing link. According to the company's research, phishing accounts for over 90% of data breaches. Businesses in the United States lose $2 billion every year as a result of phishing attacks on their customers. The main goal of this project is to employ machine learning techniques to detect dangerous URLs and alert users to the possible risk of any phishing attempts that may be there. Phishing URLs can be classified as authentic or malicious using a variety of approaches. One way is to ban the URL and update it whenever a new dangerous URL is discovered.

Another is heuristic-based detection, which includes characteristics that have been observed in real-world phishing attacks and can detect zero-hour phishing attacks, but the characteristics are not guaranteed to be present in such attacks all of the time, and the false positive rate in detection is very high. The Deep Learning strategy is utilized, which has a 98 percent accuracy, however the disadvantage of this method is that it requires a very large dataset due to its complicated models. Convolutional neural networks were utilized to recognise characteristics through their hidden layers. Because our dataset is so vast, we'll have a lot of features to identify, which will aid in discovering new URLs. For the detection of phishing URLs, a hybrid technique was utilized, although the number of characteristics used was less than ten due to their tiny dataset. This method can have drawbacks when a new URL is presented that does not fit any of the criteria they are recognising. To categorize the URL as phishing or authentic, we will use a hybrid ensemble model that includes MLP, SVM, Decision tree, and Random Forest in our project.

The blacklist method has the disadvantage of not being able to detect zero-hour phishing assaults, which can be recognised using a heuristic approach. The main disadvantage of a heuristic-based strategy is that it takes a long time to implement. We'll be incorporating HTML and JavaScript-based capabilities to improve the model's ability to recognise phishing URLs.

II. STATE OF THE ART (LITERATURE SURVEY)

Arun Kulkarni1 and Leonard L. Brown proposed a method in which They have used decision tree, Naive Bayesian classifier, support vector machine (SVM), and neural network as there four classifiers. The classifiers were evaluated on a data set of 1,353 real-world URLs that could be classified as legitimate, suspect, or phishing sites, with 10 features retrieved for each.

Mr. Kondeti Prem Sai Swaroop1 and Ms. Konka Renuka Chowdary2 proposed a method in which the features were retrieved and then compiled using ML algorithms.

Nandhini.S and Dr.V.Vasanthi 2 (2017) have proposed a method in which they used five different data mining algorithms. To classify the web phishing data set, examine the findings, and select the most effective technique to classify the web page phishing data set, Naive Bayes, KNN, Random Forest, SVM, and j48 were employed.

Jaiswal and Vaishali Bhole have proposed a method in which they ahve used The Apriori and FP-Tree algorithms to compute the association rules in this experiment. These association criteria can also be used to detect phishing URLs.

Anindita Khade and Dr. Subhash K Shinde (2013) have proposed a method for identifying phishing websites with a layer structure, three different phishing types and six separate criteria have been defined. For classification, they used the RIPPER data mining technique.

III. PROPOSED METHODOLOGY

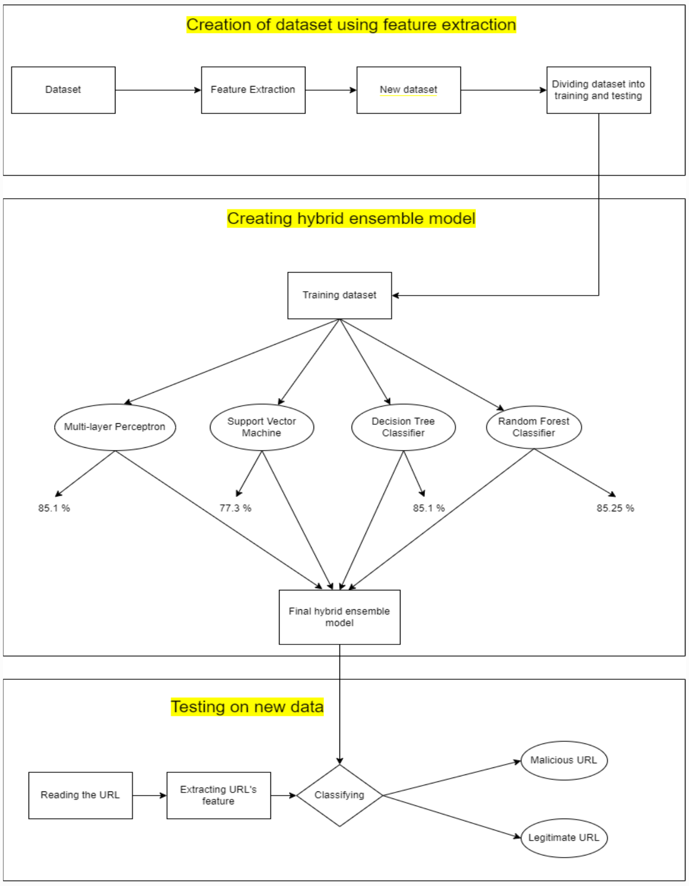


Figure 1: Proposed Architecture

We are using a hybrid ensemble model to improve the accuracy of phishing URL identification in this research. The terms "bagging" and "boosting" are used to describe two different types of ensemble learning. The bagging category includes the popular ensemble learning model random forest. Another famous ensemble learning model that falls into the boosting group is AdaBoost. The bagging models only use a small portion of the dataset, whereas the boosting models use the complete dataset.

Our model is a collection of weak learners who are brought together to demonstrate their combined strength because we will be employing diverse classifiers, resulting in a heterogeneous collection of models, also known as a hybrid ensemble model, the URL class is determined by a vote of the weak students. The accuracy can be improved by adding additional weak students.

1) Dataset: The dataset considered is a combination of legitimate and malicious URLs of size 20,000.

2) Extracting features: URLs in the dataset are passed to various features which return 0 or 1 depending on the conditions. The returned values are then stored in a csv in a tabular format.

3) Dividing the dataset into train and test: The dataset is divided into training and testing data in variable ratios.

4) Hybrid ensemble model: The classifiers are imported and applied on the dataset and the respective accuracies are calculated. In this work, we will define some numbers of models a variable number of times to generate weak learners. Then finally, the Max Voting Classifier method is used where the class which has been predicted mostly by the weak learners will be the final class prediction of the ensemble model.

IV. RESULTS

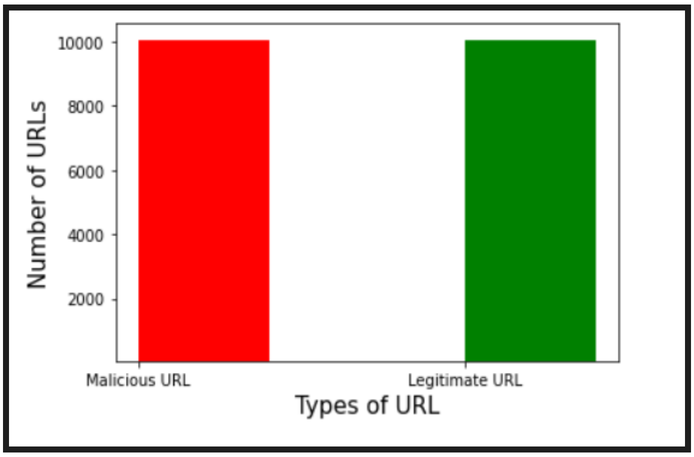
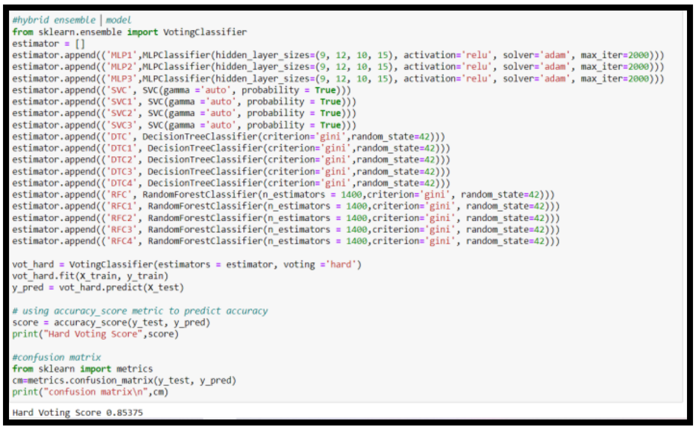


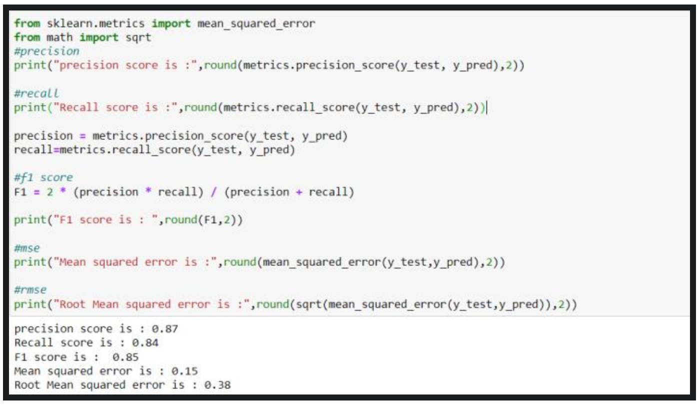
Figure 2: Legitimate v/s malicious URLs



```
#hybrid ensemble | model
from sklearn.ensemble import VotingClassifier
estimator = []
estimator.append(('MLP1',MLPClassifier(hidden_layer_sizes=(9, 12, 10, 15), activation='relu', solver='adam', max_iter=2000)))
estimator.append(('MLP2',MLPClassifier(hidden_layer_sizes=(9, 12, 10, 15), activation='relu', solver='adam', max_iter=2000)))
estimator.append(('MLP3',MLPClassifier(hidden_layer_sizes=(9, 12, 10, 15), activation='relu', solver='adam', max_iter=2000)))
estimator.append(('SVC', SVC(gamma ='auto', probability = True)))
estimator.append(('SVC1', SVC(gamma ='auto', probability = True)))
estimator.append(('SVC2', SVC(gamma ='auto', probability = True)))
estimator.append(('SVC3', SVC(gamma ='auto', probability = True)))
estimator.append(('DTC', DecisionTreeClassifier(criterion='gini',random_state=42)))
estimator.append(('DTC1', DecisionTreeClassifier(criterion='gini',random_state=42)))
estimator.append(('DTC2', DecisionTreeClassifier(criterion='gini',random_state=42)))
estimator.append(('DTC3', DecisionTreeClassifier(criterion='gini',random_state=42)))
estimator.append(('DTC4', DecisionTreeClassifier(criterion='gini',random_state=42)))
estimator.append(('RFC', RandomForestClassifier(n_estimators = 1400,criterion='gini', random_state=42)))
estimator.append(('RFC1', RandomForestClassifier(n_estimators = 1400,criterion='gini', random_state=42)))
estimator.append(('RFC2', RandomForestClassifier(n_estimators = 1400,criterion='gini', random_state=42)))
estimator.append(('RFC3', RandomForestClassifier(n_estimators = 1400,criterion='gini', random_state=42)))
estimator.append(('RFC4', RandomForestClassifier(n_estimators = 1400,criterion='gini', random_state=42)))

vot_hard = VotingClassifier(estimators = estimator, voting ='hard')
vot_hard.fit(X_train, y_train)
y_pred = vot_hard.predict(X_test)

# using accuracy_score metric to predict accuracy
score = accuracy_score(y_test, y_pred)
print("Hard Voting Score",score)

#confusion matrix
from sklearn import metrics
cm=metrics.confusion_matrix(y_test, y_pred)
print("confusion matrix\n",cm)
```

```
Hard Voting Score 0.85375
```

Figure 3: Accuracy Score of Model



```
from sklearn.metrics import mean_squared_error
from math import sqrt
#precision
print("precision score is :",round(metrics.precision_score(y_test, y_pred),2))

#recall
print("Recall score is :",round(metrics.recall_score(y_test, y_pred),2))

precision = metrics.precision_score(y_test, y_pred)
recall=metrics.recall_score(y_test, y_pred)

#f1 score
F1 = 2 * (precision * recall) / (precision + recall)

print("F1 score is : ",round(F1,2))

#mse
print("Mean squared error is :",round(mean_squared_error(y_test,y_pred),2))

#rmse
print("Root Mean squared error is :",round(sqrt(mean_squared_error(y_test,y_pred)),2))
```

```
precision score is : 0.87
Recall score is : 0.84
F1 score is :  0.85
Mean squared error is : 0.15
Root Mean squared error is : 0.38
```

Figure 4: Various metric scores

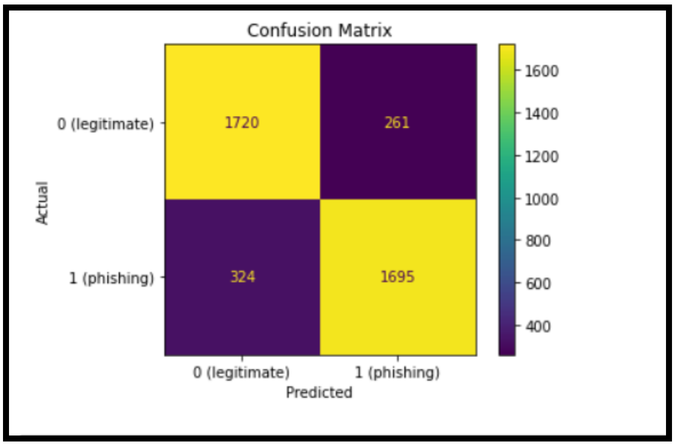


Figure 5: Confusion Matrix

V. ANALYSIS

Split ratio	Classifier	Accuracy score
80:20	Random forest	85.2
80:20	Decision tree	85.1
80:20	MLP	85.1
80:20	Svm	77.3
80:20	Hybrid ensemble model	85.37

Table 1: Comparison between several classifiers

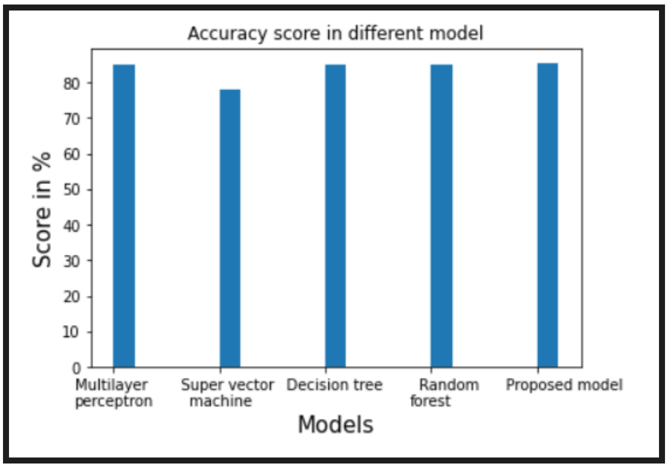


Figure 6: Comparison of accuracy scores.

Internally when we applied individual models to our dataset instead of the hybrid ensemble model, then it resulted in lesser accuracies like MLP and Decision Tree produced 85.1% accuracy whereas SVM produced 77.3% and random forest produced 85.25%. But when we used the hybrid model it went up to 85.37% which is better than all the models individually.

We have developed a hybrid ensemble model by combining MLP (3 weak learners), SVM(4 weak learners) ,decision tree(5 weak learners) and random forest(5 weak learners) and combination of these classifiers results in a hybrid model. We have achieved an accuracy of 85.37%.

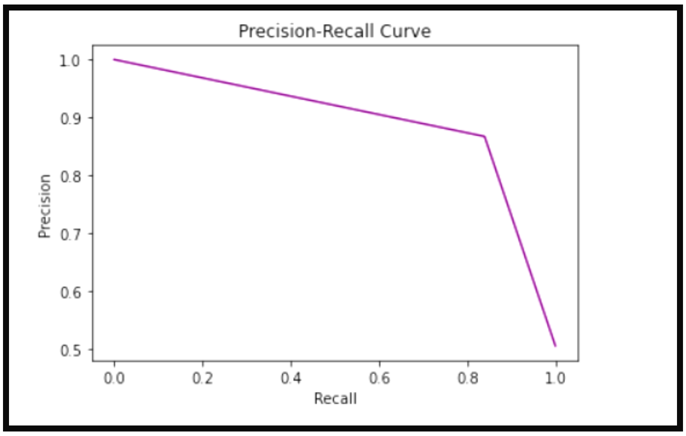


Figure 7: Precision - Recall Curve for hybrid ensemble model

Precision score is 86.65 % which tells us about the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions.
Recall score is 83.95 %. It is calculated as the number of true positives divided by the total number of true positives and false negatives. Model recall score represents the model's ability to correctly predict the positives out of actual positives.

VI . COMPARITIVE ANALYSIS

	Proposed model	Previously used model [14]	Previously used model [18]
Dataset size	20000	1353	1050
Accuracy score	85.375%	90.04%	82.6 %
Precision score	87 %	-	67.1 %
Number of features	19	8	29
Model Used	Hybrid ensemble model	Hybrid KNN-SVM	C4.5 data mining algorithm
Recall Score	84 %	89 %	94 %
Error rate	14.6%	10%	18%

Table 2: Comparative Analysis

In the previous paper [18] they achieved the accuracy of 82.6% and our model achieved the accuracy of 85.37%. Whereas in the other research paper [14] they achieved the accuracy of 90% using a hybrid KNN-SVM model. In [14] a hybrid model is created using KNN followed by an SVM classifier. The key benefit of utilizing a KNN classifier is that it has a lower computational complexity because it does not require the building of a feature space, and then the SVM method is used as a classification engine in the second stage of this hybrid model. In contrast, we constructed a hybrid ensemble model using several classifiers in our proposed methodology.

VII. CONCLUSION AND FUTURE WORK

The main significance of this work is that this model can be used as a web browser extension to determine whether the website we are currently visiting is malicious or legitimate. This could help users avoid any kind of malwares that may creep into their device. We have achieved an accuracy of 85.37%. Precision score is 86.65 %. Recall score is 83.95 %. We faced difficulty while creating the hybrid model as we had to decide about which weak learners had to be included in the hybrid architecture.

The current work can be compiled and deployed to a browser extension which will automatically detect if the site is malicious or safe to visit as we browse through the internet. Further, this model can be enhanced by the use of various deep learning techniques to increase the overall accuracy of the model.

REFERENCES

[1] Mr. Kondeti Prem Sai Swaroop1, Ms. Konka Renuka Chowdary2, Ms. S. Kavishri 3 Phishing Websites Detection using Machine Learning Techniques International Research Journal of Engineering and Technology (IRJET)

[2] Mahajan, Rishikesh & Siddavatam, Irfan. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications. 181. 45-47. 10.5120/ijca2018918026.

[3] Mahajan, Rishikesh & Siddavatam, Irfan. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications. 181. 45-47. 10.5120/ijca2018918026.

[4] Bhagyashree E. Sananse Tanuja K. Sarode, Phishing URL Detection: A Machine Learning and Web Mining-based Approach International Journal of Computer Applications (0975 – 8887)

[5] Suman Bhattacharyya1 , Chetan kumar Pal2 , Praveen kumar Pandey3 Detecting Phishing Websites, a Heuristic Approach International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

[6] B.A.S. Dilhara (2021) Phishing URL Detection: A novel hybrid Approach using Long Short-Term Memory and Gated Recurrent Units International Journal of Computer Applications (0975 – 8887)

[7] Tomas RASYMAS, Laurynas DOVYDAITIS.(2020). Detection of Phishing URLs by Using Deep Learning Approach and Multiple Features Combinations Baltic J. Modern Computing, Vol. 8 (2020), No. 3, 471-483

[8] Ray, K.S., Kusshwaha, R. (2021). Detection of Malicious URLs Using Deep Learning Approach. In: Chakraborty, M., Singh, M., Balas, V.E., Mukhopadhyay, I. (eds) The "Essence" of Network Security: An End-to-End Panorama. Lecture Notes in Networks and Systems, vol 163. Springer, Singapore.

[9] Luong Anh Tuan Nguyen, Huu Khuong Nguyen, and Ba Lam To.(2016).An Efficient Approach Based on Neuro-Fuzzy for Phishing Detection .Journal of Automation and Control Engineering Vol. 4.

[10] Ashritha Jain R,Chaithra Kulal ,Mrs. Mangala Kini,Deekshitha S .(2019).A Review Paper on Detection of Phishing Websites using Machine Learning.International Journal of Engineering Research & Technology (IJERT).

[11] Arun Kulkarni1 , Leonard L. Brown, III2.(2019).Phishing Websites Detection using Machine Learning.International Journal of Advanced Computer Science and Applications, Vol. 10.

[12] Mehanović, D., Kevrić, J. (2020). Phishing website detection using machine learning classifiers optimized by feature selection. Traitement du Signal, Vol. 37, No. 4, pp. 563-569. https://doi.org/10.18280/ts.370403

[13] S. Mercy Shalinie,Ming Hour Yang,Raja Meenakshi U. Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions, IET Network

[14] Taha, Altyeb. (2017). Phishing Websites Classification using Hybrid SVM and KNN Approach. International Journal of Advanced Computer Science and Applications. 8. 10.14569/IJACSA.2017.080611.

[15] Anindita Khade, Dr. Subhash K Shinde, 2013, Detection of Phishing Websites Using Data Mining Techniques, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 02, Issue 12 (December 2013),

[16] Nandhini.S 1 , Dr.V.Vasanthi 2.(2017).Extraction of Features and Classification on Phishing Websites using Web Mining Techniques.IJEDR.Volume 5.

[17] V. Suganya.(2016).A Review on Phishing Attacks and Various Anti Phishing Techniques.International Journal of Computer Applications (0975 – 8887) Volume 139

[18] Priya, Akansha and Er. Meenakshi. “Detection of phishing websites using C4.5 data mining algorithm.” 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (2017): 1468-1472.

[19] Jaiswal, Vaishali Bhole and Pramod Sekharan Nair. “A Study of Phishing URL Detection using Apriori and FP-Tree algorithm.” International Journal of Advanced Research in Computer and Communication Engineering 6 (2017): 460-467.

[20] Pankaj Kumar Kandi and Pankaj Agarkar (2020) Detection of Phishing Web Sites Based On Feature Classification and Extreme Learning Machine EasyChair Preprint 2425

[21] Palse, Vishal et al. “Real Time Phishing Detection on Generated URLs.” International Journal of Engineering Research and V6.06 (2017): n. pag. Web.

[22] Dirash A R1, Mehtab Mehdi2 Phishing URL Detection International Journal of Trend in Scientific Research and Development (IJTSRD) Volume 5 Issue 1, November-December 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

[23] Yadollahi, Mohammad Mehdi & Shoeleh, Farzaneh & Serkani, Elham & Madani, Afsaneh & Gharaee, Hossein. (2019). An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features. 281-286. 10.1109/ICWR.2019.8765265.

[24] Kausar, Firdous et al. “Hybrid Client Side Phishing Websites Detection Approach.” International Journal of Advanced Computer Science and Applications 5 (2014): n. pag.

[25] Aydin, Mustafa et al. “Using Attribute-Based Feature Selection Approaches and Machine Learning Algorithms for Detecting Fraudulent Website URLs.” 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) (2020): n. pag. Web.