

**Pune Institute of Computer Technology
Dhankawadi, Pune**

**A SEMINAR REPORT
ON**

**DETECTION OF PHISHING WEBSITES USING MACHINE
LEARNING TECHNIQUES**

SUBMITTED BY

Name : Jash Shah

Roll No. : 31457

Class: TE-4

**Under the guidance of
Prof. R.A Kulkarni**



**DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2021-22**



DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

**“DETECTION OF PHISHING WEBSITES USING
MACHINE LEARNING TECHNIQUES”**

Submitted by
Jash Shah Roll No. : 31457

has satisfactorily completed a seminar report under the guidance of
Prof. R.A Kulkarni towards the partial fulfillment of third year
Computer Engineering Semester I, Academic Year 2021-22 of
Savitribai Phule Pune University.

Prof. R.A Kulkarni
Internal Guide

Dr. M.S.Takalikar
Head
Department of Computer Engineering

Place:Pune
Date: 10/11/2021

ACKNOWLEDGEMENT

It is my pleasure to present a report on "Detection of phishing websites using machine learning techniques". First of all, I would like to thank our Seminar Coordinator Prof. D.D. Kadam, Head of Department Dr. M.S.Takalikar and Principal Dr. R.Sreemathy for their encouragement and support.

I would also genuinely express my gratitude to my guide Prof. R.A Kulkarni, Department of Computer Engineering for her constant guidance and help. She has constantly supported me and has played a crucial role in the completion of this report. Her motivation and encouragement from beginning till end to make this seminar a success.

Last but not the least I would thank all the faculty, my parents and friends who have helped me.

Contents

1	INTRODUCTION	1
2	MOTIVATION	3
3	LITERATURE SURVEY	4
4	A SURVEY ON PAPERS	6
4.1	Detection of Phishing Websites by Using Machine Learning-Based URL Analysis	6
4.2	Phishing Web Page Detection Methods: URL and HTML Features Detection	6
4.3	Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection	6
4.4	Detecting Phishing Websites through deep reinforcement learning .	6
4.5	OFS- NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network	6
4.6	Proactive Phishing Sites Detection	7
5	PROBLEM DEFINITION AND SCOPE	8
5.1	Problem Definition	8
5.2	Scope	8
6	DIFFERENT ALGORITHMS USED FOR PHISHING WEBSITES DETECTION	9
6.1	KNN	9
6.2	SVM	9
6.3	Decision Tree	9
6.4	Random Forest	9
6.5	XGboost	9
7	METHODOLOGY	10
7.1	Workflow	10
7.2	Mathematical model	11
7.3	Algorithm	12
7.3.1	KNN	12
7.3.2	SVM	12
7.3.3	Decision Tree	12
7.3.4	Random Forest	13
7.3.5	XGBoost	13
8	DATASET DESCRIPTION	14
9	IMPLEMENTATION	15
10	RESULTS	16
11	CONCLUSION	17

References

18

List of Tables

1	Literature survey	4
2	Classification results for different methods	16

List of Figures

1	URL Structure	2
2	Overall itinerary planning framework: Workflow	10
3	Correlation of features in dataset	14
4	Implementation of different classifiers	15

Abstract

Phishing is the type of cyber-attack that is used to obtain sensitive user data including the login credentials and credit card details. It costs internet users billions of dollars per year. Attackers achieve the above goal through disguising illegal URLs as legitimate ones, attackers can induce users to visit the phishing URLs. Despite the fact that several ways for detecting phishing websites have been proposed, phishers have evolved their methods to beat these detection methods. Machine Learning is one of the most effective approaches for detecting these dangerous activities. This is due to the fact that most phishing attacks share some common characteristics that machine learning algorithms can detect. Advanced ML models based on feature selection methods and Neural networks have been quite more effective. I have compared the performance of different machine learning approaches for classification of URLs as legitimate or phishing URLs.

Keywords

Phishing, Phishers, Classification, Cyber-attack, Machine-learning, Neural Networks, Feature selection.

1 INTRODUCTION

Nowadays people do the majority of their work on digital platforms in their daily lives. In many ways, having a computer and access to the internet makes our work and personal lives easier. It assists us in timely completion of transactions and operations in the fields of finance, research, trade, entertainment, education, and engineering. With the advancement of mobile and wireless technologies, users who require access to a local network can now effortlessly connect to the Internet from anywhere and at any time. Despite the fact that this arrangement is extremely convenient, it has revealed significant information gaps. As a result, people in cyberspace must take precautions against potential cyber-attacks.

Cybercriminals, pirates, non-malicious (white-capped) attackers, and hacktivists are all examples of people who can carry out attacks. The goal is to gain access to the computer or the information it holds, as well as to acquire personal information in various methods. The most targeted industries are financial institutions, payment, e-commerce, Logistics, cryptocurrency, social media and web mails. These attacks have cost internet users billions of dollars per year.

The most common and dangerous of these attacks is phishing. Cybercriminals employ email or other social media communication channels in this type of attack. The attackers fool the victims into thinking the message came from a reputable source, such as a bank or an e-commerce site. As a result, they attempt to gain access to their personal data. Using this information, attackers get access to their victims' accounts. Thus, it results in both monetary and non-monetary losses.

Such attacks are generally carried out through malicious websites that steal all kinds of personal data that can be exploited. Traditional classification techniques such as blacklisting, regular expression, and signature matching are challenged in the identification of malicious URLs due to large data volumes, shifting patterns over time, and intricate relationships between features. Several dangerous websites, unsurprisingly, do not appear to have been blacklisted. A URL may be used to track down any website, much as any file on a computer can be found by specifying its filename. URLs are the addresses of resources on the WWW. For URL <https://www.google.com>, the protocol identifier is HTTPS. Hypertext Transfer Protocol Secure (HTTPS) is used to fetch hypertext documents. Other protocols include DomainName System (DNS), File Transfer Protocol (FTP), etc. For URL <https://www.google.com>, the resource name is www.google.com. The resource identifier is the address of a webpage on the internet.

The proposed work is based upon identifying the malicious URLs by feature extraction and further classification with help of various ML classifiers. And thus, examining the evaluation metrics for various machine learning classifiers.

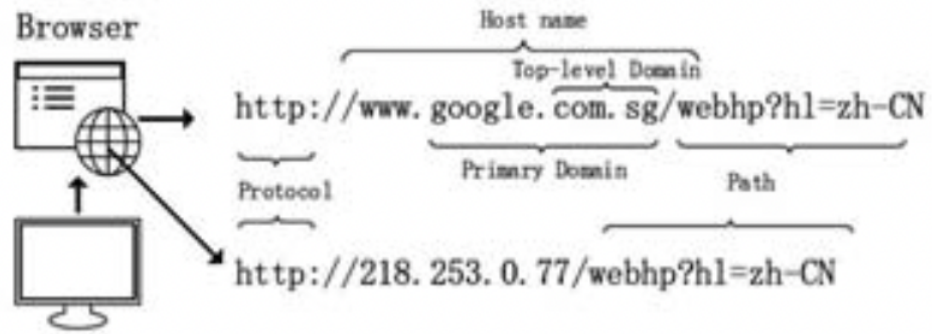


Figure 1: URL Structure

1

¹Shantanu, Janet, B., Joshua Arul Kumar, R. (2021). Malicious URL Detection: A Comparative Study. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS).

2 MOTIVATION

The sectors such as e-commerce and online banking have grown multi-folds in recent years. And a major boom in this sector was seen during the pandemic in the past two years when people couldn't go out to buy stuff and had to do it through online mode. People are now much inclined toward online purchasing and banking rather than the traditional methods. And with this rise in online business there is significant rise in the cyber attacks such as phishing in which phishers try to exploit users and try to gain sensitive information from them through phishing websites. Such attacks have cost internet users more than billions of dollars per year. The reasons why people are getting duped are they are not well educated about these frauds and attacks and secondly it sometimes becomes hard for a human to identify which websites are legitimate and which are phishing.

Thus, a machine learning approach which would extract important features and would be trained by advanced ML algorithms such as xgboost will help identify users which URLs are legitimate and which aren't , is a good approach to solve this problem.

3 LITERATURE SURVEY

The Following table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

Sr no.	Paper Title	Summary	Limitations
1	Detection of Phishing Websites by Using Machine Learning-Based URL Analysis	trained only with the features obtained from the URL. is expected to classify in a shorter time than other models	Accuracy and F-measure obtained is less.
2	Phishing Web Page Detection Methods: URL and HTML Features Detection	rules-based method with the aim of making the application more effective in terms of accuracy and faster detection ability	Features not selected optimally Lesser accuracy
3	Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection	Fuzzy Rough Set (FRS) to select the most effective features. Best accuracy achieved was 91.46%	Hard to implement. Lesser accuracy
4	Detecting Phishing Websites through Deep Reinforcement Learning	The proposed model can adapt to the dynamic behaviour of the phishing websites and thus learn the features associated with phishing website detection.	Not optimised for real-world implementation. Only lexical features were considered Parameters not tuned

5	OFS- NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network	FS-NN, an effective phishing website detection model based on the optimal feature and neural network	Hard to implement. Takes time to select features.
6	Proactive Phishing Sites Detection	authors proposed suspicious domain names generation and to predicts likely phishing web sites from the given legitimate brand domain name and scores and judges suspects by calculating various indexes to detect phishing websites	Method based on heuristic therefore not automatic.

4 A SURVEY ON PAPERS

4.1 Detection of Phishing Websites by Using Machine Learning-Based URL Analysis

The goal of this work is to develop a phishing detection system that analyses the URL of a webpage. In the literature review, it was discovered that obtaining effective characteristics from the URL improves the classification accuracy. Third-party service usage, site style, CSS, content, meta information, and other elements can all help to increase accuracy. These characteristics, on the other hand, will increase the time it takes to classify new websites that need to be categorised. The suggested model, which was trained solely on the features extracted from the URL, is projected to classify in less time than other models. In light of this, the study will focus solely on URL analysis. As a consequence, the classification results of collected features in various machine learning techniques are compared.

4.2 Phishing Web Page Detection Methods: URL and HTML Features Detection

In this study, the author presents a rules-based strategy with the goal of improving the application's accuracy and detection speed. This study also compared numerous machine learning algorithms to determine if they improved detection accuracy. The authors then compare and contrast some of these strategies to come up with a more effective strategy for detecting phishing web sites.

4.3 Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection

Mahdieh Zabihimayvan et al. employed the Fuzzy Rough Set (FRS) technique to pick the most effective features, claiming that the maximum F-measure gained by FRS feature selection is 95 percent when utilising Random Forest classification.

4.4 Detecting Phishing Websites through deep reinforcement learning

To model and detect malicious URLs, Moitrayee Chatterjee et al. devised a model based on deep reinforcement learning. The suggested model may learn the properties related with phishing website identification by adapting to the dynamic behaviour of phishing websites

4.5 OFS- NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network

The problem of overfitting in phishing classifiers was emphasised by

Erzhou Zhu et al., who proposed FS-NN, an effective phishing website detection model based on the ideal feature and neural network.

4.6 Proactive Phishing Sites Detection

Akihito Nakamura et al. evaluated phishing mitigation strategies such as blacklists, heuristics, visual similarity, and machine learning and determined that these techniques have limitations in dealing with zero-hour assaults and proactive phishing website identification. The authors advocated generating suspicious domain names and predicting possible phishing web sites from a legal brand domain name, as well as scoring and judging suspects using multiple indexes to detect phishing websites.

5 PROBLEM DEFINITION AND SCOPE

5.1 Problem Definition

Identification of Phishing websites (URLs) based on various factors such as IP address, URL length, Shortening Service,HTTPS token etc.

5.2 Scope

The features which are crucial and need to be considered for getting highly accurate ML model to classify legitimate and phishing websites are having ip address,URL length,shortening service,having @ symbol,SSL state,request url,favicon,dns record and some more. These features have a major impact on the optimization of results.

For these the data is first processed and all optimal features are extracted and then the data is trained with some ML algorithms and then the best fit model is gained.

6 DIFFERENT ALGORITHMS USED FOR PHISHING WEBSITES DETECTION

6.1 KNN

KNN is a ML technique for regression and classification problems. Feature similarity is used to predict the values for the new data points. The new datapoint is assigned its value depending on how closely it resembles the points in the training set.

6.2 SVM

SVM is a supervised machine learning (ML) technique that can be used to solve regression and classification issues. SVM's major goal is to establish a decision boundary that divides n-dimensional space into separate classes so that a new data point can be appropriately classified in the future.

6.3 Decision Tree

A tree-like structure is used to perform classification in the Decision Tree algorithm. In this algorithm the dataset is separated into small subsets which are used to generate Tree Nodes. These tree nodes can be Leaf Nodes or Decision nodes depending upon their function.

6.4 Random Forest

The random forest algorithm uses several different decision trees to anticipate the outcome. It's a collection of decision trees. To generate more precise answers, many decision trees are blended together. It aids in the elimination of decision tree flaws.

6.5 XGboost

XGBoost is a distributed gradient boosting toolkit that has been tuned for efficiency, flexibility, and portability. It uses the Gradient Boosting framework to create machine learning algorithms. XGBoost is a parallel tree boosting (also known as GBDT, GBM) algorithm that solves a variety of data science issues quickly and accurately.

7 METHODOLOGY

7.1 Workflow

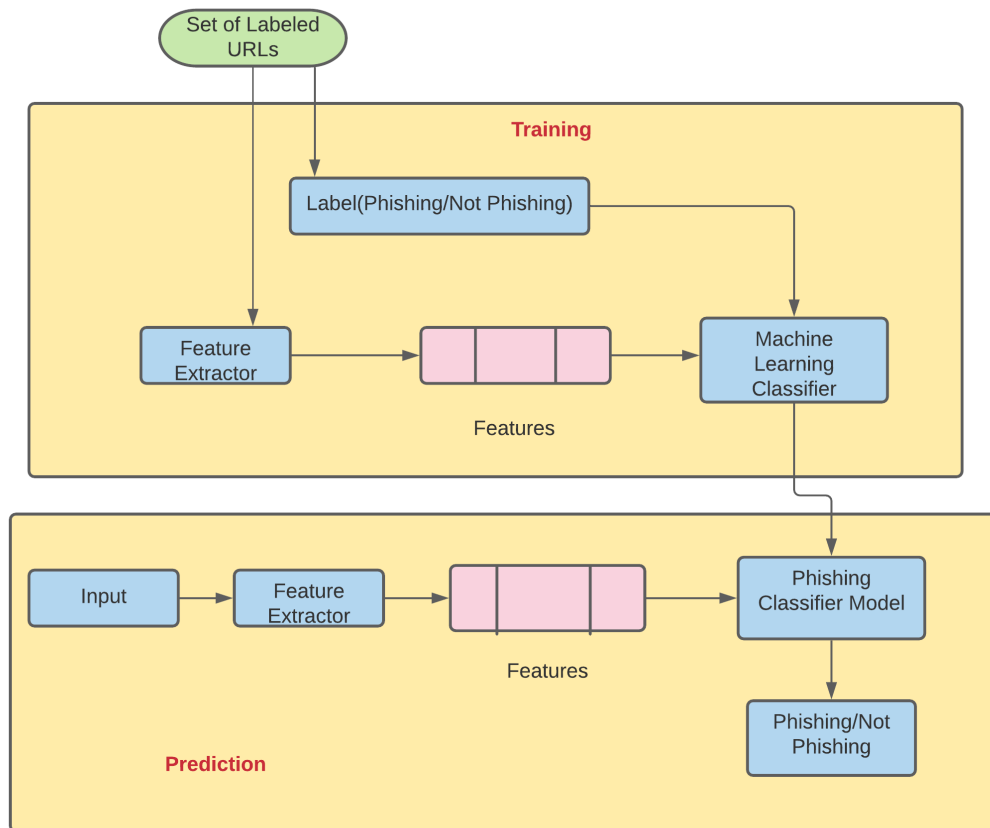


Figure 2: Overall itinerary planning framework: Workflow

7.2 Mathematical model

$N_{L \rightarrow L}$ = Number of legitimate websites classified as legitimate

$N_{L \rightarrow P}$ = Number of legitimate websites misclassified as phishing

$N_{P \rightarrow L}$ = Number of phishing websites misclassified as legitimate

$N_{P \rightarrow P}$ = Number of phishing websites classified as phishing

acc = Accuracy

r = Recall

p = Precision

F_1 = F1 Score

$$acc = \frac{N_{L \rightarrow L} + N_{P \rightarrow P}}{N_{L \rightarrow L} + N_{P \rightarrow L} + N_{P \rightarrow P}}$$

$$r = \frac{N_{P \rightarrow P}}{N_{P \rightarrow L} + N_{P \rightarrow P}}$$

$$p = \frac{N_{P \rightarrow P}}{N_{L \rightarrow P} + N_{P \rightarrow P}}$$

$$F_1 = \frac{2pr}{p + r}$$

7.3 Algorithm

7.3.1 KNN

```
procedure KNN
  Classify(X,Y,x)
  X:Traning data
  Y:class lables of X
  x: unknown sample
  for i=1 to m do
    Compute distance  $d(X_i,x)$ 
  end for
  Compute set I containing indices for the k smallest distances  $d(X_i,x)$ 
  return majority label for ( $Y_i$  where  $i \in I$ )
```

7.3.2 SVM

Require: **X** and **y** loaded with training labeled data, $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$ partially trained SVM

```
  C  $\Leftarrow$  some value (10 for example)
  repeat
    for all {  $x_i, y_i$  }, {  $x_j, y_j$  } do optimize  $\alpha_i$  and  $\alpha_j$ 
  end for
  until no change in  $\alpha$  or other constraint criteria met
Ensure: Retain only the support vectors  $i > 0$ 
```

7.3.3 Decision Tree

Input: S, where S=set of classified instances

Output: Decision Tree

Require: $S \neq \phi$, num_attributes>0

```
procedure BUILDTREE
  repeat
    maxGain  $\leftarrow$  0
    splitA  $\leftarrow$  null
    e $\leftarrow$  Entropy(Attributes)
    for all Attributes a in S do
      gain  $\leftarrow$  InformationGain(a,e)
      if gain > maxGain then
        maxGain  $\leftarrow$  gain
        splitA $\leftarrow$  a
      end if
    end for
    Partition(S,splitA)
  until all partitions processed
end procedure
```

7.3.4 Random Forest

To generate c classifiers:

```

for  $i=1$  to  $c$  do
    Randomly sample the training data  $D$  with replacement to produce  $D_i$ 
    Create a root node,  $N_i$  containing  $D_i$ 
    Call  $\text{BuildTree}(N_i)$ 
end for
BuildTree( $N$ )
if  $N$  contains instances of only one class then
    return
else
    Randomly select  $x\%$  of the possible splitting features in  $N$ 
    Select the feature  $F$  with the highest information gain to split on
    Create  $f$  child nodes of  $N, N_1, \dots, N_f$ , where  $F$  has  $f$  possible values
    ( $F_1, \dots, F_f$ )
    for  $i=1$  to  $f$  do
        Set the contents of  $N_i$  to  $D_i$  is all instances in  $N$  that match  $F_i$ 
        Call  $\text{BuildTree}(N_i)$ 
    end for
end if

```

7.3.5 XGBoost

M-Number of base learners

T-Number of leaves in tree

w^* -Optimised leaf weight

L-left branch

R-right branch

Data: Dataset and hyperparameters

Initialize $f_o(x)$;

for $k=1, 2, \dots, M$ **do**

Calculate $g_k = \frac{\partial L(y, f)}{\partial f}$

Calculate $h_k = \frac{\partial^2 L(y, f)}{\partial f^2}$

Determine the structure by choosing splits with maximised gain

$A = \frac{1}{2} [\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G^2}{H}]$

Determine the leaf weight $w^* = -\frac{G}{H}$

Determine the base learner $\hat{b}(x) = \sum_{j=1}^T wI$;

Add trees $f_k(x) = f_{k-1}(x) + \hat{b}(x)$;

end

Result: $f(x) = \sum_{k=0}^M f_k(x)$

8 DATASET DESCRIPTION

The dataset is taken from the PhishTank website. The dataset has around 11,000 sample websites URL, 90 percent of URLs are used in the training phase and 10 percent in testing phase. Each URL is marked either phishing or legitimate.

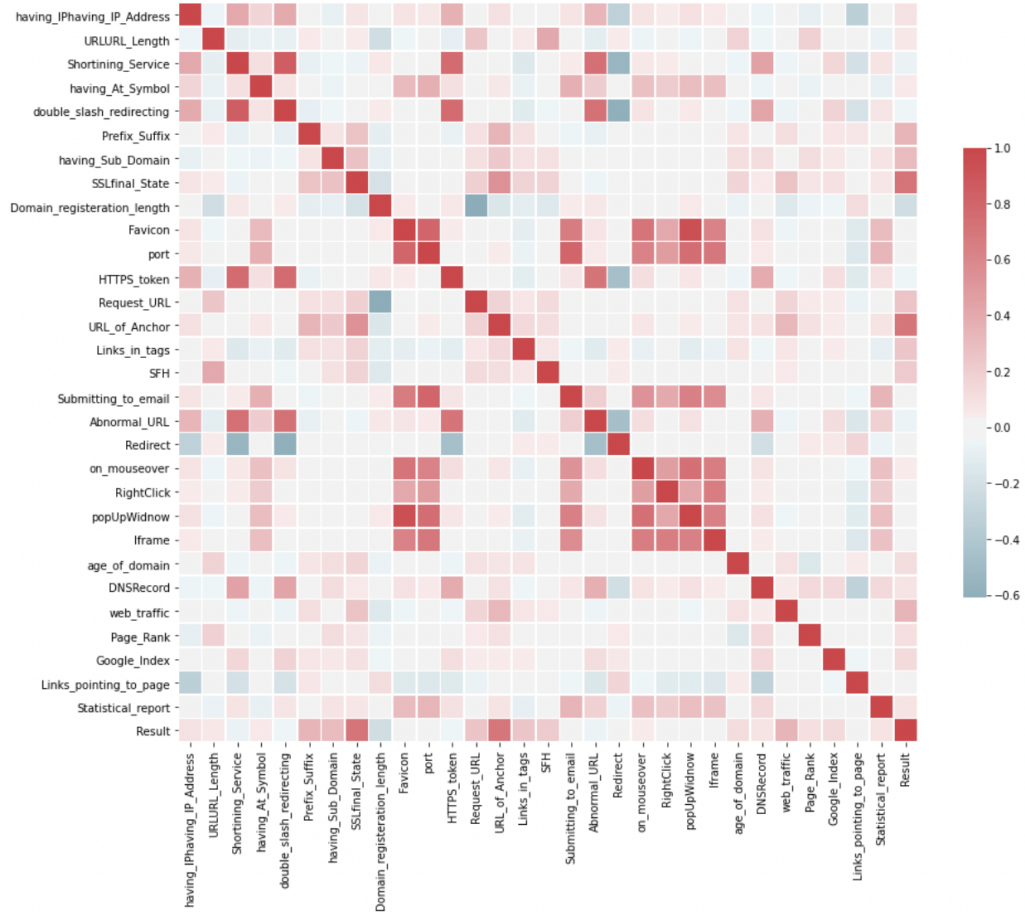


Figure 3: Correlation of features in dataset

9 IMPLEMENTATION

KNN

```
In [39]: KNeighbors_clf=KNeighborsClassifier(3)
cross_val_scores = cross_validate(KNeighbors_clf, X, y, cv=fold_count, scoring=scoring)
KNeighbors_clf_score = mean_score(cross_val_scores)
print(KNeighbors_clf_score)

{'fit_time': 0.11297237873077393, 'score_time': 0.3535628795623779, 'test_accuracy': 0.9527809643818579, 'test_recall': 0.9629
682715658324, 'test_precision': 0.9527832046787947, 'test_f1': 0.95782786394066}
```

SVM

```
In [42]: ###linear
linear_clf = svm.SVC(kernel='linear')
cross_val_scores = cross_validate(linear_clf, X, y, cv=fold_count, scoring=scoring)
linear_svc_clf_score = mean_score(cross_val_scores)
print(linear_svc_clf_score)
###poly
poly_clf = svm.SVC(kernel='poly')
cross_val_scores = cross_validate(poly_clf, X, y, cv=fold_count, scoring=scoring)
poly_svc_clf_score = mean_score(cross_val_scores)
print(poly_svc_clf_score)
###rbf
rbf_clf = svm.SVC(kernel='rbf')
cross_val_scores = cross_validate(rbf_clf, X, y, cv=fold_count, scoring=scoring)
rbf_svc_clf_score = mean_score(cross_val_scores)
print(rbf_svc_clf_score)
###sigmoid
sigmoid_clf = svm.SVC(kernel='sigmoid')
cross_val_scores = cross_validate(sigmoid_clf, X, y, cv=fold_count, scoring=scoring)
sigmoid_svc_clf_score = mean_score(cross_val_scores)
print(sigmoid_svc_clf_score)

{'fit_time': 1.6475388288497925, 'score_time': 0.05397987365722656, 'test_accuracy': 0.9277262647999803, 'test_recall': 0.9455
92070531095, 'test_precision': 0.9262681624188114, 'test_f1': 0.9357793221750376}
{'fit_time': 1.0482579469680786, 'score_time': 0.07420728206634522, 'test_accuracy': 0.9492547437670297, 'test_recall': 0.9688
163868651675, 'test_precision': 0.9417793952178888, 'test_f1': 0.9550833558973665}
{'fit_time': 1.3415402889251709, 'score_time': 0.1033292531967163, 'test_accuracy': 0.9521492803547904, 'test_recall': 0.96881
5331010453, 'test_precision': 0.9465803137802874, 'test_f1': 0.9575433423537307}
{'fit_time': 1.3446074724197388, 'score_time': 0.109696364402771, 'test_accuracy': 0.8274983021446165, 'test_recall': 0.846515
1515151514, 'test_precision': 0.8443119026086775, 'test_f1': 0.8453050000657976}
```

Decision Tree

```
In [47]: dtree_clf=DecisionTreeClassifier()
cross_val_scores = cross_validate(dtree_clf, X, y, cv=fold_count, scoring=scoring)
dtree_score = mean_score(cross_val_scores)
print(dtree_score)

{'fit_time': 0.02145226001739502, 'score_time': 0.0037374973297119142, 'test_accuracy': 0.9659887246037655, 'test_recall': 0.9
714145813536058, 'test_precision': 0.9676815772041456, 'test_f1': 0.9695318553822136}
```

Random Forest

```
In [32]: rforest_clf=RandomForestClassifier()
cross_val_scores = cross_validate(rforest_clf, X, y, cv=fold_count, scoring=scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)

{'fit_time': 0.4361262321472168, 'score_time': 0.021941876411437987, 'test_accuracy': 0.9726827751548527, 'test_recall': 0.981
4847956921128, 'test_precision': 0.9698521059240438, 'test_f1': 0.9756226033690053}
```

Gradient Boosting With XGBoost

```
In [34]: XGB_clf=XGBClassifier()
cross_val_scores = cross_validate(XGB_clf, X, y, cv=fold_count, scoring=scoring)
XGB_clf_score = mean_score(cross_val_scores)
print(XGB_clf_score)

{'fit_time': 0.5060725927352905, 'score_time': 0.0062372684478759766, 'test_accuracy': 0.9712358750705734, 'test_recall': 0.97
90470911202618, 'test_precision': 0.9696243951193955, 'test_f1': 0.9742930356745975}
```

Figure 4: Implementation of different classifiers

10 RESULTS

The following table represents the train time, test time, accuracy, recall, precision and f1 score for different ML classifiers.

Table 2: Classification results for different methods

Classifier	train time(s)	test time(s)	accuracy	recall	precision	F1 score
KNN	0.1129	0.3535	0.9527	0.9629	0.9527	0.9578
SVM_linear	1.6475	0.0539	0.9277	0.9455	0.9262	0.9357
Decision Tree	0.0214	0.0037	0.9659	0.9714	0.9676	0.9695
Random Forest	0.4361	0.0219	0.9726	0.9814	0.9698	0.9756
XGBoost	0.5060	0.0062	0.9832	0.9810	0.9872	0.9768

11 CONCLUSION

ML classifiers were implemented and tested on the phishing website dataset, which consisted of 6157 authentic websites and 4898 phishing websites, in this study. The examined classifiers were KNN , SVM , Decision Tree , Random forest , XGBoost. The results are shown in table 2. From the table it can be said that excellent results were gained in ensembling classifiers like as Random forest and XGBoost in terms of both computation time and accuracy










References

- [1] Korkmaz, Mehmet Sahingoz, Ozgur Diri, Banu. (2020). Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. 1-7. 10.1109/ICCCNT49239.2020.9225561.
- [2] Chapla, Happy Kotak, Riddhi Joiser, Mittal. (2019). A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier. 383-388. 10.1109/ICCES45898.2019.9002145.
- [3] H. Faris and S. Yazid, "Phishing Web Page Detection Methods: URL and HTML Features Detection," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 167-171, doi: 10.1109/IoTaIS50849.2021.9359694.
- [4] M. Chatterjee and A. -S. Namin, "Detecting Phishing Websites through Deep Reinforcement Learning," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, pp. 227-232, doi: 10.1109/COMPSAC.2019.10211.
- [5] Nakamura, Akihito Dobashi, Fuma. (2019). Proactive Phishing Sites Detection. 443-448. 10.1145/3350546.3352565.
- [6] Shantanu, Janet, B., Joshua Arul Kumar, R. (2021). Malicious URL Detection: A Comparative Study. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). doi:10.1109/icaais50930.2021.9396014
- [7] Kumar, J., Santhanavijayan, A., Janet, B., Rajendran, B., BS, B. (2020). Phishing Website Classification and Detection Using Machine Learning. 2020 International Conference on Computer Communication and Informatics (IC-CCI). doi:10.1109/iccci48352.2020.91

Document Information

Analyzed document	Seminar_Report_31457.pdf (D118751379)
Submitted	2021-11-16 06:53:00
Submitted by	Rekha Kulkarni
Submitter email	rakulkarni@pict.edu
Similarity	16%
Analysis address	rakulkarni.pict@analysis.orkund.com

Sources included in the report

W	URL: https://cupdf.com/document/adbms-seminar-report.html Fetched: 2021-11-16 06:58:00		2
W	URL: https://ouci.dntb.gov.ua/en/works/42LaypV9/ Fetched: 2021-11-16 06:58:00		4
W	URL: https://arxiv.org/pdf/2103.12739 Fetched: 2021-11-16 06:58:00		6
W	URL: https://www.researchgate.net/publication/333626385_OFS-NN_An_Effective_Phishing_Websites_Detection_Model_Based_on_Optimal_Feature_Selection_and_Neural_Network Fetched: 2021-11-16 06:58:00		1
W	URL: https://arxiv.org/pdf/2009.11116 Fetched: 2021-11-16 06:58:00		4
W	URL: https://www.semanticscholar.org/paper/OFS-NN%253A-An-Effective-Phishing-Websites-Detection-on-Zhu-Chen/7767c27d29651413b4fc0e1af63d313d82ffde49 Fetched: 2021-11-16 06:58:00		1
W	URL: https://www.researchgate.net/publication/303739321_PhishWHO_Phishing_webpage_detection_via_identity_keywords_extraction_and_target_domain_name_finder Fetched: 2021-11-16 06:58:00		1
W	URL: http://ieeexplore.ieee.org/document/8730309 Fetched: 2021-11-16 06:58:00		1
W	URL: https://www.ncbi.nlm.nih.gov/pmc/articles/pmc8504731/ Fetched: 2021-11-16 06:58:00		1