

Fake Website Detection Using Machine Learning & ANN

BY

MAHBUBA SULTANA

ID: 181-15-10927

AND

SHAMIMA AFROSE SUPTY

ID: 181-15-10540

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Most. Hasna Hena

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

Ms. Afsara Tasneem Misha

Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2022

APPROVAL

This Project titled “**Fake Website Detector Using Machine Learning and ANN**”, submitted by **Mahbuba Himu** and **Shamima Afrose Supty** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on ***04-01-2022***.

BOARD OF EXAMINERS



Chairman

Dr. Sheak Rashed Haider Noori
Associate Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



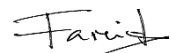
Internal Examiner

Abdus Sattar
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Saiful Islam
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



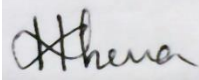
External Examiner

Dr. Dewan Md. Farid
Professor
Department of Computer Science and Engineering
United International University

DECLARATION

We hereby declare that; this project has been done by us under the supervision of **Most. Hasna Hena, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree.

Supervised by:



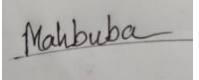
Most. Hasna Hena
Assistant professor
Department of CSE
Daffodil International University

Co-Supervised by:

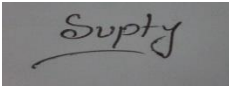


Ms. Afsara Tasneem Misha
Designation
Department of CSE
Daffodil International University

Submitted by:



Mahbuba Himu
ID: 181-15-10927
Department of CSE
Daffodil International University



Shamima Afrose Supty
ID: 181-15-10540
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Most. Hasna Hena, Assistant Professor Department of CSE** Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Machine learning algorithm*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Touhid Bhuiyan, Professor, and Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Day after day the number of internet users increases, phishing has grown increasingly breakneck. Phishing attacks pose a serious threat to people's daily lives and the online environment. For example, the attacker poses as a trustworthy source in order to get sensitive information or the victim's digital identity, such as a credit card number or certificate or other valuable information. For this reason, people lose their identity after falling into the trap of these raiders. As the name implies, phishing or faking sites are false copies of actual web sites. When a person's identification card gets stolen, they are cheating. To create the website for this paper debate publishing, we will be relying on a machine learning algorithm, Neural Network Classifier MLPC (Multilayer perceptron Classifier) and have differentiated the percentage of accuracy between them. We have used five machine learning algorithms: Naive Bayes algorithm, K-nearest neighbors (KNN), SVM, Decision tree, Random forest algorithm. Most accurate and well directed perspective of this approach may be found in our dataset that it's a scam or fake website. Among them, the Random Forest algorithm provided **97.9 %** accuracy.

TABLE OF CONTENTS

CONTENTS	PAGE NO
Approval	i
Declarations	ii
Acknowledgement	iii
Abstract	iv
List of figures	vii
List of Tables	viii
Key Abbreviations	viii
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1: Introduction	1
1.2: Motivation	1-2
1.3: Rationale of the Study	2
1.4: Research Questions	2
1.5: Expected Output	3
CHAPTER 2: BACKGROUND	4-10
2.1: Terminologies	4
2.2: Related Works	4-7
2.3: Comparative Analysis and Summary	7-9
2.4: Scope of the Problem	9
2.5: Challenges We Faced	10
CHAPTER 3: RESEARCH METHODOLOGY	11-26
3.1: Research Subject and Instrumentation	11

3.2: Data Collection Procedure	11
3.3: Data Preprocessing Steps	12-15
3.4: Statistical Analysis	16-17
3.5: Proposed Methodology	17-24
3.6: Implementation Requirements	24-26
CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	27-29
4.1: Experimental Setup	27
4.2: Experimental Results & Analysis	27-28
4.3: Discussion	28-29
CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	30
5.1: Impact on Society	30
5.2: Sustainability Plan	30
CHAPTER 6: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH	31-32
6.1: Summary of the Study	31
6.2: Discussion	31
6.3: Conclusion	31-32
6.4: Future Work	32
REFERENCES	33-34
PLAGIARISM REPORT	

LIST OF FIGURES

FIGURE	PAGE NO
Figure 3.3: Data Preprocessing Steps	12
Figure 3.3.2: Vectorization of string URLs	13
Figure 3.3.3: Sentence to word tokenization's code	14
Figure 3.3.4: Train and Test split	15
Figure 3.3.5: X Train data records	15
Figure 3.4.1: Total Null records for Noisy Dataset	16
Figure 3.4.2: Ratio of True and False URLs	16
Figure 3.4.3: Head of Dataset	17
Figure 3.5.1: Machine Learning Base URL Detection Techniques	17
Figure 3.5.2: Confusion Matrix of Naïve Bayes	18
Figure 3.5.3: Accuracy Score of Naïve Bayes	19
Figure 3.5.4: Confusion Matrix for Random Forest	19
Figure 3.5.5: Accuracy Score for Random Forest	20
Figure 3.5.6: Confusion Matrix for Decision Tree	20
Figure 3.5.7: Accuracy Score for Decision Tree	21
Figure 3.5.8: Confusion Matrix for KNN	21
Figure 3.5.9: Accuracy Score for KNN	22
Figure 3.5.10: SVM classification and Matrix	22

Figure 3.5.11: Artificial Neural Network Classifier and Matrix	23
Figure 3.6.1: Attackers Techniques to get Information	25
Figure 3.6.2: Flow Diagram of Fake Website Detection	25
Figure 4.2.1: Visualization Algorithm Accuracy	28

LIST OF TABLES

TABLES	PAGE NO
TABLE 2.3.1: ACCURACY FOR 3331 DATA (FALSE VALUE)	8
TABLE 2.3.2: ACCURACY FOR 3331 DATA (TRUE VALUE)	8-9
TABLE 2.3.3: ACCURACY FOR 3331 DATA (NEURAL NETWORKING)	9
TABLE 4.2.1: ALGORITHMS COMPARISONS	27

LIST OF ABBREVIATIONS

Key	Full Meaning
KNN	K-nearest neighbors
ANN	Artificial Neural Network
RF	Random Forest
NB	Naïve Bayes
SVM	Support Vector Machine
DT	Decision Tree
MLPC	Multilayer Perceptron Classifier

CHAPTER 1

INTRODUCTION

1.1: Introduction

Phishing is a form of cyber fraud that means criminals used various URLs and pages or website in the server program of real pages to place malware code on the site [14]. The eventual goal is nearly always the same: to get you to hand over your personal or financial data. This type of website could be a freestanding site, pop ups, or unlawful overlays on mainstream sites through click fraud. Regardless of how they are presented, these sites are designed to entice and mislead users. While some schemes are more complicated than others, the majority of them can be broken down into these three parts. Many communication methods, such as social media, email, and text messaging, can be used by a fraudulent website to entice internet users. Search engine optimization (SEO) techniques can sometimes be used to skew search results, resulting in harmful sites appearing towards the top. Users are more responsive to these schemes when they appear as an appealing offer or a terrifying alert message. The majority of scam websites rely on psychological tricks to function. We used a technique known as supervised machine learning in our study. We gathered information from different true or false site on various social media sites such as Facebook, Google. We had worked with 3331 data. We preprocessed our dataset and used five different machine learning classifiers, with the greatest accuracy of provided **97.9** percent coming from Random Forest Algorithm. Neural networks based used in our research. Because of its high accuracy, neural networks have gained popularity in data science in recent years.

1.2: Motivation

People nowadays rely on a variety of websites on the internet. In this era, we can easily access to a website or web page, and sometimes we don't know which the fake and which the real, and also, we can see the same web link but they are different, but we cannot define them and we face some problems, likes information's hacks. The fraudster released in order to alert visitors about dangerous websites. Some evil people create fake websites, like Facebook, Instagram and so on. They create a fake website and creates interface exactly

like Facebook, Twitter, Instagram etc. And some uneducated peoples, sometimes also educated peoples, unwarily uses their fake website and gives username and login password to that fake website owners. As a result, their private information is get accessed by attackers. In our study, our main aims to prevent or distinguish between real and a fake website. We hope it will help people to make a difference between a real and a fake website.

1.3: Rationality of the Study

There a lot more ways that can find out the difference between fake and real website but there no guarantee or there no such kind of algorithms are available that gives information about a website completely. They're having some algorithms named Decision Tree, Random Forest and support vector algorithms that are normally used to detect fake websites. In our study, our main goal is to detect the fake websites or URL by comparing false negative that means true and false positive that's mean negative rate of each algorithm and structures. Attacks can be carried out by people such as cybercriminals, pirates, or non-malicious (white-capped) attackers and hackers [1].

1.4: Research Questions

We had a tough time completing the task since we obtained data manually by searching Google and using Kaggle datasets. As a result, it took a long time. The main problem was then to achieve the desired accuracy. We had a lot of questions regarding it all. They are as follows:

1. How can we get Data?
2. What is the best way to sort the data?
3. How can we prepare our data for analysis?
4. How can we determine which method is suitable for our dataset?
5. What is the best way to get the highest level of accuracy?

1.5: Output to be expected

Our major goal was to figure out how to determine authentic or bogus websites for safety based on numerous facts. We were able to discover it using multiple machine learning approaches and algorithms, and its accuracy was 97.9 percent. We were able to recognize the type of website using textual data by applying supervised based classification.

First, we collected data in different ways then did data preprocessing. Following that, we have used machine learning algorithms such as Naive Bayes algorithm, K-nearest neighbors (KNN), SVM, Decision tree, Random Forest algorithm. Our accuracy climbed to 97.9 percent. We have created a comparison between machine learning and neural networks to find which gives the best results in our research. People will be able to readily determine if a website is real or false using this method. This will safeguard you against the inconvenience and misuse of bogus websites.

CHAPTER 2

BACKGROUND

2.1: Terminologies

This study describes related work, simple comparison and summary, problem prospects and difficulties. We dealt with research publications relevant to our research effort under the related work area. We spoke about how they gathered datasets, how they processed data, what algorithms they utilized, and what their accuracy rates were. We attempted to discover appropriate categories and methodologies for our work through comparative study and summary. We looked into a variety of methods to improve accuracy. We tagged our dataset in the most effective way possible while simultaneously working to reduce complexities. We researched some of these related websites from Google for summarization. We acquired data in various methods and then preprocessed it. After that, we employed machine learning methods and Neural Network Classifiers. Our main objective was to figure out how to identify genuine or fraudulent websites for safety based on a variety of factors. In the Overview of the Seeking in - depth understanding, we highlighted the issues we encountered when preprocessing, cleaning, and applying classifiers to our dataset. In the challenges section, we highlighted the issues we encountered in collecting our dataset and presenting it in a machine-readable format.

2.2: Related work

As the number of internet users has increased, phishing has grown more harmful. This paper will discuss the machine learning and deep learning algorithms and apply all these algorithms on our dataset and the best algorithm having the best precision and accuracy is selected for the phishing website detection. Anaconda environment is used to implement the work and, in this work, the dataset is taken from “Kaggle” website. The phishing website detection model has been tested and trained using many classifiers and ensemble algorithms to analyze and compare the model’s result for best accuracy. In this work Decision Tree Algorithm provides the high accuracy percentage. The methodology they discovered is a powerful technique to detect the phished websites and can provide more

effective defenses for phishing attacks of the future. [1] This Paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, Random Forest and Support vector machine algorithms are used to detect phishing websites. They have implemented python program to extract features from URL. Three machine learning classification model Decision Tree, Random Forest and Support vector machine has been selected to detect phishing websites. They achieved 97.14% detection accuracy using random forest algorithm with lowest false positive rate. [2]

In this paper they have developed a system that uses machine learning techniques to classify websites based on their URL. The data set they used from The University of California, Irvine Machine Learning Repository has nine attributes and contains 1,353 samples. They implemented four classifiers using MATLAB scripts, which are the decision tree, Naïve Bayes' classifier, Support Vector Machine (SVM), and the Neural Network. From these the Decision Tree Algorithm gives a high accuracy % in this job. [3]

Eint Sandi Aung†a) Hayato YAMANA†b) used Machine Learning Algorithms and neural network-based methods to analyze Malicious URLs and Detection Methods. They first identified phishing detection viewpoints depending on whether or not database access was required. They called them database-oriented and heuristic-oriented approaches, respectively. In the following portion of the research, they looked at current approaches and classified them as machine learning or neural network-based. [4]

Naresh Kumar D (2020), Nemala Sai Rama Hemanth (2020) Premnath S (2020) Nishanth Kumar V (2020), Uma S (2020) used machine learning algorithms in their research. In their research the random forest algorithm performs better with attack detection accuracy of 91.4%. The future work of the proposed system is to evaluate these machine learning classifiers with larger dataset. [5]

Ningxia Zhang, Yongqing Yuan build a model to Detect Phishing Using Neural Network. They utilize about 8762 emails, 4560 of which are phishing emails and the rest are spam. We use a feedforward neural network to identify phishing attempts by integrating some basic email structure and external link properties. They also compare neural networks' performance to that of other machine learning approaches. According to their findings, NNs have the highest recall while still maintaining a precision of >95 percent, implying

that they are good at identifying phishing emails while misclassifying just a tiny percentage of ham emails.[6]

Ammara Zamir(2019) used with a variety of machine learning techniques, including random forest [RF], neural network [NN], bagging, support vector machine, Nave Bayes, and k-nearest neighbor in their research. They compare several supervised learning algorithms on a variety of feature sets, including random forest (RF), SVM, bagging, kNN, neural network (NN), and j48. In terms of classification performance, Stacking1 (RF + NN + Bagging) surpassed all other classifiers with 97.4 percent accuracy in detecting phishing websites. The study is based on the data set of phishing websites. With 11,055 web visits, the data set comprises 32 pre-processed characteristics. The suggested approach might be integrated with different feature extraction models in the future to verify its applicability in a real-time environment.[7]

B. Geyik, K. Erensoy and E. Kocyigit (2021) used Machine learning technology to detect and prevent this type of intrusion. They gathered the websites used in phishing attempts into a dataset, then utilized this information to generate findings using four categorization methods. In their study, the Random Forest Classifier produced the greatest results (83 percent accuracy).[8]

S. S. Birunda and R. K. Devi (2021), here the suggested approach is used by Machine Learning (ML) classifiers in order to evaluate their performance in detecting false news. The testing findings show that the suggested framework with the Gradient Boosting algorithm has an efficacy of around 99.5 percent to the highest level. Using the TF-IDF approach, the top actual and false characteristics were retrieved from news articles. As a future research area, the presented system might be extended to more accurate XGBoost and deep learning algorithms. [9]

I. Kareem and S. M. Awan (2019) looked at two feature extraction techniques: Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) (TF-IDF). There are seven distinct supervised Machine Learning (ML) classification methods employed. In their study, the best performance classifier is K Nearest Neighbors (KNN), which has a 70% accuracy rate, while logistic regression has a 69% accuracy rate. This might be a future study area if other vectorization techniques like word2vec and deep learning models like LSTM are used for classification. [10]

P. Singh, Y. P. S. Maravi and S. Sharma (2015) used two algorithms called Adaline and Backpropion in conjunction with the support vector machine to improve detection and classification. They obtained phishing and legal URLs from a variety of sources, including phishtank [20] and Alexa [21]. It is demonstrated that the Adaline network, in conjunction with SVM, produced better results with a 99.1 percent accuracy.[11]

A. Basit, M. Zafar, A. R. Javed and Z. Jalil(2020) chose three machine learning classifiers to utilize in an ensemble technique with Random Forest Classifier (RFC): Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), and Decision Tree (C4.5). The experimental findings show that the KNN and RFC ensemble identifies phishing assaults with 97.33 percent accuracy. They utilized a dataset with 11055 occurrences and 30 characteristics from the UCI machine learning library. This study may be expanded to test performance on different datasets, and a pre-trained plug-in can be created to identify phishing assaults on websites in real time using web browsers.[12]

M. Akhavan and S. M. Hossein Hasheminejad (2021) here for web phishing data, an unsupervised feature selection approach named LAPPSO is presented. Results with experimental findings obtained by applying LAPPSO to two well-known phishing datasets, our method obtains an average F-measure of 96 percent while drastically lowering the amount of features. [13]

2.3: Comparative analysis and summary

This section will differentiate our work from all other types of research. In [1] research the dataset is taken from “Kaggle” website, they reviewed machine learning and deep learning algorithms in this research and applied all of them to their dataset before selecting the optimal algorithm with the best precision and accuracy for phishing website identification. On the other hand in [2] Phishing websites are detected using Decision Tree, Random Forest, and Support Vector Machine algorithms. Using the random forest approach, they were able to obtain 97.14 percent detection accuracy with the lowest false positive rate. Whereas in our research we have collected data manually by searching in internet platform such as google, Facebook, and from different necessary sites, and we also collected some data from kaggle. We have used machine learning algorithms and neural networking classification to get the best output and showed the comparison between each

algorithm. We preprocessed our dataset and used five different machine learning classifiers, with the greatest accuracy of provided **97.9** percent coming from Random Forest Algorithm.

In [3] they've created a system that classifies websites based on their URLs using machine learning techniques. The University of California, Irvine Machine Learning Repository data set they utilized comprises nine characteristics and 1,353 samples. On the contrary, fiction to get the best output showed the comparison between each algorithm. We preprocessed our dataset and used five different machine learning classifiers and 3331 data. The Decision Tree Algorithm delivers a high accuracy percent in this work based on this. Whereas in our research Random Forest Algorithm delivers a high accuracy percent.

In [4] they analyzed Malicious URLs and Detection Methodologies using Machine Learning Algorithms and neural network-based methods. Whereas, to acquire the best results, we employed machine learning algorithms and neural networking classification, and we compared each approach.

Our quality of accuracy is given below:

TABLE 2.3.1: ACCURACY FOR 3331 DATA (FALSE VALUE)

Classifier	Precision	Recall	F1-score	Accuracy
Naive Bayes	0.97	0.98	0.97	97.15%
Random Forest	0.99	0.97	0.98	97.9%
KNN	0.98	0.97	0.98	97.3%
SVM	0.97	0.98	0.97	96.85%
Decision tree	0.99	0.97	0.98	97.45%

TABLE 2.3.2: ACCURACY FOR 3331 DATA (TRUE VALUE)

Classifier	precision	Recall	F1-score	Accuracy
Naive Bayes	0.97	0.97	0.97	97.15%

Random Forest	0.97	0.99	0.98	97.90%
KNN	0.97	0.97	0.97	97.30%
SVM	0.97	0.96	0.96	96.85%
Decision tree	0.96	0.99	0.97	97.45%

TABLE 2.3.3: ACCURACY FOR 3331 DATA (NEURAL NETWORKING)

Classification Report	precision	Recall	F1-score	Accuracy
0.0	0.79	1.00	0.88	85.46
1.0	0.99	0.67	0.80	85.46

2.4: Scope of the problem

We ran across a few of issues, the breadth of which is listed below:

- 1) We were concerned about data gathering, so we resolved the problem by collecting real-life data from multiple social media platforms and from Kaggle.
- 2) In the data preprocessing portion, we've had a lot of problems. Because our data was web-based, extracting it in a standard format was extremely difficult. However, we employed various preprocessing approaches such as Stop Word removal, Null value elimination, punctuation removal, and white space removal to assist us reduce the dataset.
- 3) We were concerned about data collection, so we resolved the problem by gathering real-life data from a variety of social media platforms.
- 4) Although we used Vectorization to evaluate the polarity of each sentence and determine the primary accuracy level, it only provided us with a small number of results. The challenge is then solved by employing several supervised learning classifiers that deliver the highest level of accuracy.

2.5: Challenges We Faced

1. The algorithms we are using in our dataset didn't provide us 100% accuracy. As result we are failed to get the complete accuracy rate by these algorithms.
2. The proposed system enables the internet users to have a safe browsing. So, those who haven't connect to internet won't make any transaction.

CHAPTER 3

RESEARCH METHODOLOGY

3.1: Research Subject and Instrumentation

Our research topic is “Model comparison of URL based fake website detectors (Based on Machine Learning Algorithm and Neural Network Classifier)” .Before selecting our topic we have spent a lot of time to find our interested field. Finally we had found our interest on Machines Learning (ML) and Neural Network MLPC (Multilayer perceptron Classifier) .Almost everyday we visited several kinds of websites for ours daily work or reasons .Sometimes we visited fake sites deceives with us. So we think that we analysis the problem and for detect that’s kind of fake URL we choose this topics for our research. For designed the model We've used Google Colab notebook which is basically use for Python development .there are many notebook like,Jupyter Notebook, but we think that colab is better for our research.We also used various Python advance libraries such as Numpy, Pandas, Matplotlib, Seaborn,regular expression package, scikit-learn, word cloud etc. to develop our project .

3.2: Data collection procedure / Data set utilization

We have collected data from several sources of websites which we search in google and also, we take some data from Kaggle dataset and then we have manually arranged the data in the excel sheet. We couldn't find appropriate resources to collect data at first. We decided to collect data from several sources since we wanted real-life user data. Then we search in google for fake and real websites list and we get some data here. Then, because our acquired data was in a complex online format, we had to adapt it into a more universal one. The data was then translated to numerical representation for use in the classifier and algorithm.

3.3: Data Preprocessing Steps

In our study, we preprocessed our dataset with different preprocessing steps for read our excel dataset and then applied on it. The applied preprocessing steps are given below-

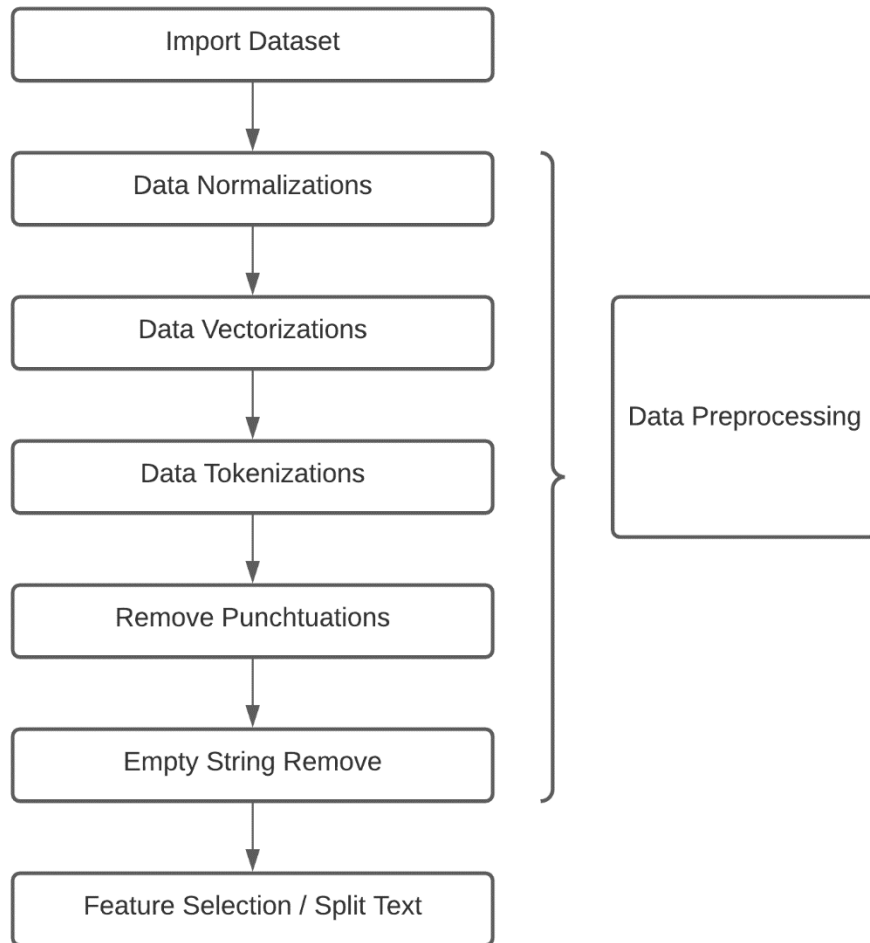


Figure 3.3: Data preprocessing Steps

3.3.1: Data Normalizations: After importing our dataset for the purpose of preprocessing we normalized our data to find our null values or noise data and also to remove empty record. Such that our dataset gets less noisy and it don't make any impact on our study. As a result, we will be able to find more accuracy from our given dataset.

3.3.2: Data Vectorizations: In Machine learning approaches vectorizations is a technique by which we are able to make our code fast and secure and it's the best way to optimize our algorithms when we are implementing from URLs. Vectorization is a phase in feature extraction in Machine Learning. The main goal of vectorizations by converting text to numerical vectors, the goal is to extract some identifiable features from the text for the model to learn on.

```
[21] from sklearn.feature_extraction.text import CountVectorizer
      from sklearn.feature_extraction.text import TfidfVectorizer

[31] vectorizer = TfidfVectorizer(tokenizer=makeTokens)

x= vectorizer.fit_transform(url)
print(x)
```

(0, 3027)	0.4854806794170542
(0, 4928)	0.36206848595374536
(0, 2959)	0.35948985078688567
(0, 2961)	0.3803833155668706
(0, 4925)	0.35948985078688567
(0, 2071)	0.1944108993578555
(0, 4015)	0.29550490634921117
(0, 3725)	0.3174929079101628
(0, 0)	0.06431994211658151
(1, 4967)	0.2746998824440028
(1, 4377)	0.5210034280591465
(1, 2533)	0.3672119010998621
(1, 60)	0.17440111305610956
(1, 3026)	0.4154961196825353
(1, 5006)	0.5210034280591465
(1, 2071)	0.19858695813377444
(1, 0)	0.06570157174552596

Figure 3.3.2: Vectorization of string URLs

3.3.3: Data tokenization's: Tokenization is the process of breaking a large piece of text into smaller tokens. Tokens can be words, characters, or subwords in this case. As a result, tokenization can be categorized into three parts: word, character, and sub word (characters) tokenization. In our dataset we applied python split function to convert sentence to word

tokenization's and finally we got the token for the further task and the code for tokenization are shown in below picture.

```
def makeTokens(f):  
    tkns_BySlash = str(f.encode('utf-8')).split('/')  
    total_Tokens = []  
    for i in tkns_BySlash:  
        tokens = str(i).split('-')  
        tkns_ByDot = []  
        for j in range(0, len(tokens)):  
            temp_Tokens = str(tokens[j]).split('.')  
            tkns_ByDot = tkns_ByDot + temp_Tokens  
        total_Tokens = total_Tokens + tokens + tkns_ByDot  
    total_Tokens = list(set(total_Tokens))  
    if 'com' in total_Tokens:  
        total_Tokens.remove('com')  
    return total_Tokens
```

Figure 3.3.3: Sentence to word Tokenization's code

3.3.4: Feature Selection

When building a predictive model, feature selection is the process of decreasing the number of input variables.

```
from sklearn.model_selection import train_test_split

[ ] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=1)

[ ] x_train

<2664x5186 sparse matrix of type '<class 'numpy.float64''>'
with 27991 stored elements in Compressed Sparse Row format>

[ ] y_train

1378  0.0
82    0.0
152   1.0
775   1.0
791   0.0
...
2764  0.0
906   0.0
1097  0.0
236   0.0
1062  0.0
Name: Class, Length: 2664, dtype: float64
```

Figure 3.3.4: Train and Test split

The number of input variables should be minimized to decrease the computational cost of modeling and, in some cases, to increase the model's performance. In our model, we use feature selection procedure just for the purpose of splitting the dataset into train and test set.

```
print(x_train)

(0, 1883)  0.3551357000375515
(0, 1476)  0.414308748750875
(0, 485)   0.40580864375004855
(0, 486)   0.4275547423481068
(0, 1139)  0.3742302058131614
(0, 1528)  0.3677761814755428
(0, 2340)  0.2593446602132475
(0, 2070)  0.08221297148911215
(0, 0)     0.07360020485586792
(1, 3866)  0.6823441001577624
(1, 3867)  0.6823441001577624
(1, 2070)  0.09611694119405248
(1, 60)    0.22840842141498102
(1, 0)     0.08604757173797745
(2, 3188)  0.47810560006188874
(2, 3187)  0.36367037492368237
(2, 3451)  0.4069556683366422
(2, 4473)  0.2941667732076091
(2, 4970)  0.2941667732076091
```

Figure 3.3.5: X train data records

3.4: Statistical analysis

The dataset having only a single null record and for that we are simply remove the null record using “df.dropna ()” command.

```
[ ] df.isnull().sum()

URL_List    1
Class       1
dtype: int64
```

Figure 3.4.1: Total Null records for noisy dataset

We used the supervised learning method in this research. The working technique is outlined in detail in the next section.

Dataset Collection and Properties: Our dataset has a total of 3331 urls, with 1851 categorized as False URLs and 1480 tagged as True URLs.

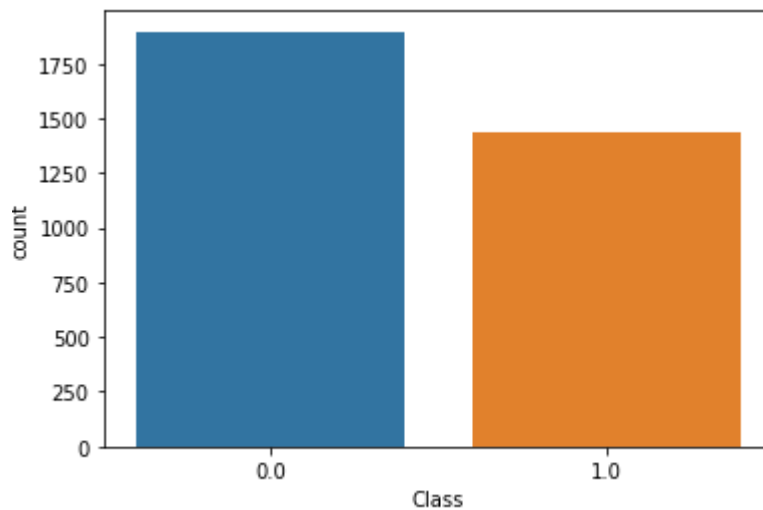


Figure 3.4.2: Ratio of False and True URL

Our data set consists of 3331 URLs where 1200 URL collected from google search and 2231 URLs were collected from Kaggle datasets.

	URL_List	Class
0	https://en.m.wikipedia.org/wiki/Facebook	1.0
1	https://www.sanagustinturismo.co/Facebook/	0.0
2	https://www.bdbudgetbeauty.com/	1.0
3	https://www.kinder.com/us/en/kinder-joy	1.0
4	https://www.facebook.pcirot.com/login.php	0.0
5	https://www.grandsultanresort.com/	1.0
6	https://deadlyplayerx.binhoster.com/Facebook/s...	0.0
7	https://www.seapearlcoxsbazar.com/	1.0
8	https://www.pranfoods.net/	1.0
9	https://dailymotion.com/	1.0


```
df.columns
```

Figure 3.4.3: Head of dataset

3.5: Proposed Methodology/Applied mechanism

We developed a model for detecting fake websites based on URL using machine learning approaches. We chose the supervised learning strategy from three primary fields of machine learning because it is compatible with developing and directing dynamic processes and also our datasets are supervised datasets.

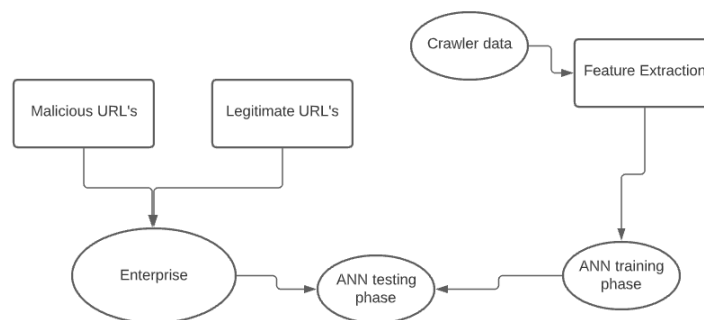


Figure 3.5.1: Machine learning based URL detection technique.

Here we also used Neural Networking Classifier algorithm MLPC (Multilayer perceptron Classifier) which is better for unsupervised data. But for our datasets its also give better

accuracy. For the implementation in our datasets we used 5 well known machine learning Classifier algorithms Naive Bayes algorithm, K-nearest neighbors (KNN), SVM, Decision tree, , Random forest algorithm. Based on our success in our model, each classifier has a brief discussion below.

Naive Bayes Algorithm

The following are some facts concerning the Naive Bayes Algorithm:

It is a machine learning approach for sorting that is based on the Bayesian probability theorem. Text classification is the most popular use, which demands massive training data sets. The Bayes theorem was applied, which is defined as follows:

$$P(h|d) = \frac{P(d|h).P(h)}{P(d)}$$

$P(h|d)$ is the probability of hypothesis h given the data d . The phrase for this is conditional probability. $P(d|h)$ represents the likelihood that data d is accurate if hypothesis h is valid. $P(h)$ is the probability that hypothesis h is right. This is known as the h prior probability. Here the data's probability is $P(d)$.

NB classifiers are the collection of different algorithms, It is not a single algorithm. In NB classifiers all the algorithms share their common principle. It is based on Bayes Theorem. For our dataset Multinomial Naïve Bayes performed very efficiently its confusion matrix was

```
confusion Matrix for Multinomial Naive Bayes:|
[[364   9]
 [ 10 284]]
```

Figure 3.5.2: Confusion Matrix of Naïve Bayes

With 97.15 percent accuracy, our classification report correctly predicted the output. The F1-score, precision, and recall are all extremely close to our output.

```

Classification Report:

              Precision    recall    f1-score   support

 0.0           0.97         0.98         0.97         373|
 1.0           0.97         0.97         0.97         294

 accuracy                    0.97         667
 macro avg           0.97         0.97         0.97         667
 weighted avg        0.97         0.97         0.97         667

```

Figure 3.5.3: Accuracy score of Naïve Bayes

Random Forest Algorithm

One of the most deserving algorithms for our job is random forest. For our data set, the Random Forest Algorithm generated an accuracy of 97.5 percent.

Random forest is a user-friendly, adaptable machine learning approach that produces outstanding results in most circumstances even when no hyper-parameters are altered. It is also one of the most commonly utilized algorithms due to its simplicity and adaptability (it can be used for both classification and regression tasks). In this article, we'll look at how the random forest technique works and how it differs from previous approaches.

Random forest is also a method of supervised learning. It creates a "forest" out of a group of decision trees trained by the "bagging" approach. The core principle of the bagging technique is that combining many learning models improves the end outcome.

RF is primarily used to solve classification issues. It's a learning algorithm that's supervised. It Creates a decision tree based on a data sample, and then votes on the best answer based on the predictions from each of them. It's the highest accuracy on our dataset was 97.9%, and its confusion matrix is as follows:

```

Confusion Matrix for Random Forest Classifier:
[[363  10]
 [  4 290]]
Score: 97.9

```

Figure 3.5.4: Confusion Matrix for Random Forest Algorithm

With 97.9 percent accuracy, our classification report correctly predicted the output. The F1-score, precision, and recall are all extremely close to our output.

```

Classification Report:|
                        Precision  recall  f1-score  support
0.0                   0.99      0.97      0.98      373
1.0                   0.97      0.99      0.98      294

accuracy              0.98              667
macro avg             0.98      0.98      0.98      667
weighted avg         0.98      0.98      0.98      667

```

Figure 3.5.5: Accuracy score of Random Forest Algorithm

Decision Tree

Decision tree-based algorithms are part of the supervised learning algorithm family. It has two applications: regression and classification. In the decision tree, the technique of the tree diagram at the top is used for prediction. There is a root node that is divided in the prevailing input feature, then divided again, and so on. These procedures will be repeated until the very last node gets the weights, at which point the input will be classified based on these weights.

DT is the most well-known and widely used algorithm for classification and prediction. It's a supervised classification system. It is commonly used for classification problems, but it is also utilized for regression difficulties in some circumstances. In comparison to other classifiers, it also has a great accuracy of 97.45%. For our data set, the DT algorithm's confusion matrix is

```

Confusion Matrix for Decision Tree:
[[360  13]
 [  4 290]]
Score: 0.9745

```

Figure 3.5.6: Confusion Matrix for Decision Tree

With 97.45 percent accuracy, our classification report correctly predicted the output. The F1-score, precision, and recall are all extremely close to our output.

```

Classification Report:

```

	precision	recall	f1-score	support
0.0	0.99	0.97	0.98	373
1.0	0.96	0.99	0.97	294
accuracy			0.97	667
macro avg	0.97	0.98	0.97	667
weighted avg	0.97	0.97	0.97	667

Figure 3.5.7: Accuracy score of Decision Tree

K-nearest neighbors (KNN) Algorithm

The KNN algorithm is used to train machine learning approach that may be used to tackle issues involving classification and regression prediction. Despite this, it is commonly used in industry to handle classification and prediction challenges.

KNN is a supervised learning algorithm since it lacks a distinct training phase but rather trains and classifies using all available data.

Because it makes no decisions based on data, KNN is a non-parametric learning approach. The KNN algorithm is a supervised learning method. It's incredibly basic and straightforward to put into practice. It is commonly used to tackle problems involving classification and regression. It performed admirably on our dataset, with a 97.3 percent accuracy rate. On our data set, the confusion matrix for KNN was:

```

Confusion Matrix for K Neighbors Classifier:
[[363  10]
 [  8 286]]
Score: 97.3
Classification Report:

```

Figure 3.5.8: Confusion matrix of KNN

With 97.3 percent accuracy, our classification report correctly predicted the output. The

F1-score, precision, and recall are all extremely close to our output.

Classification Report:				
	precision	recall	f1-score	support
0.0	0.98	0.97	0.98	373
1.0	0.97	0.97	0.97	294
accuracy			0.97	667
macro avg	0.97	0.97	0.97	667
weighted avg	0.97	0.97	0.97	667

Figure 3.5.9: Accuracy score of KNN

Support Vector Machine (SVM)

SVM algorithm is a supervised machine learning algorithm that is normally used for classification and SVM algorithm effective detect fake website

```
[ ] from sklearn.svm import SVC
svm = SVC(random_state=101)
svm.fit(x_train,y_train)
predsvm = svm.predict(x_test)
print("Confusion Matrix for Support Vector Machines:")
print(confusion_matrix(y_test,predsvm))
print("Score:",round(accuracy_score(y_test,predsvm)*100,2))
print("Classification Report:",classification_report(y_test,predsvm))
```

```
Confusion Matrix for Support Vector Machines:
[[364  9]
 [ 12 282]]
Score: 96.85
Classification Report:
              precision    recall  f1-score   support

   0.0         0.97       0.98       0.97         373
   1.0         0.97       0.96       0.96         294

 accuracy          0.97         667
 macro avg         0.97         667
 weighted avg      0.97         667
```

Figure 3.5.10: SVM classification and matrix

Artificial Neural Network

Biological neural networks develop the structure of the human brain, and the term "Artificial Neural Network" is derived from them. Artificial neural networks, such as the human brain, have neurons that are interconnected to one another in various layers of the networks. Nodes are the term for these neurons. In the field of artificial intelligence, an Artificial Neural Network attempts to mimic the network of neurons that make up a human brain so that computers can understand things and make decisions in a human-like manner. Computers are going to act like interconnected brain cells in order to create an artificial neural network. Our study uses Neural Network classifier that's named Multilayer Perceptron Classifier (MLPC).

```
[ ] from sklearn.metrics import f1_score, classification_report, confusion_matrix

from sklearn.utils import shuffle
Xtrain, ytrain = shuffle(x_train,y_train)

print("\n Neural Network")
from sklearn.neural_network import MLPClassifier
nnclf = MLPClassifier(solver='sgd', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=0)
nnclf = nnclf.fit(x_train,y_train)
nnclfpreds_test = nnclf.predict(x_test)

print("Confusion Matrix for Neural Network Classifier:")
print(confusion_matrix(y_test, nnclfpreds_test))

print("Score:",round(accuracy_score(y_test, nnclfpreds_test)*100,2))

print("Classification Report:")
print(classification_report(y_test, nnclfpreds_test))
```

Neural Network
Confusion Matrix for Neural Network Classifier:
[[372 1]
 [96 198]]
Score: 85.46
Classification Report:

	precision	recall	f1-score	support
0.0	0.79	1.00	0.88	373
1.0	0.99	0.67	0.80	294
accuracy			0.85	667
macro avg	0.89	0.84	0.84	667
weighted avg	0.88	0.85	0.85	667

Figure 3.5.11: Neural Network Classifier and matrix

It's a field in computer science that focuses into how simple models of biological brains may be used to solve difficult computational problems like predictive modeling in machine learning. The goal is to develop strong algorithms and data structures which can be used to model difficult problems, or to create realistic brain models. The power of neural networks comes from their ability to learn how to best relate the representation in your training data to the output variable you want to predict. Neural networks, in this sense, learn a mapping. They could learn any mapping function mathematically and have been proven to be a universal approximation algorithm.

In our model we found the accuracy for neural network algorithm are 85.46%.

3.6: Implementation requirements

The system that provides security to lose sensitive information to attackers using new and effective technology like Machine Learning and Artificial Neural Network algorithms needs some requirements.

Software Requirements

- * Python 3.10
- * Python Packages like number, scikit learn, matplotlib
- * Browser (Chrome)
- * Code editor (like Collab)

Hardware Requirements

- * Windows 10 installed computer(preferred)
- * 64 operating system (for scikit learn)

Design of Fraud & Flow chart

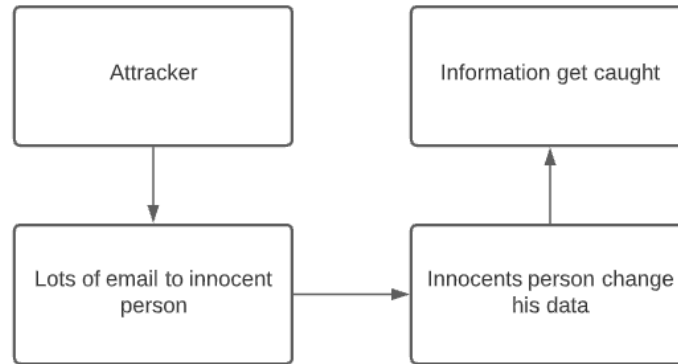


Figure 3.6.1: Attacker's technique to get information [4]

The basic idea of our propose work is the hybrid solution which uses all approaches that is black list, white list and visual similarity. In our proposed system, it has the algorithms are as follows

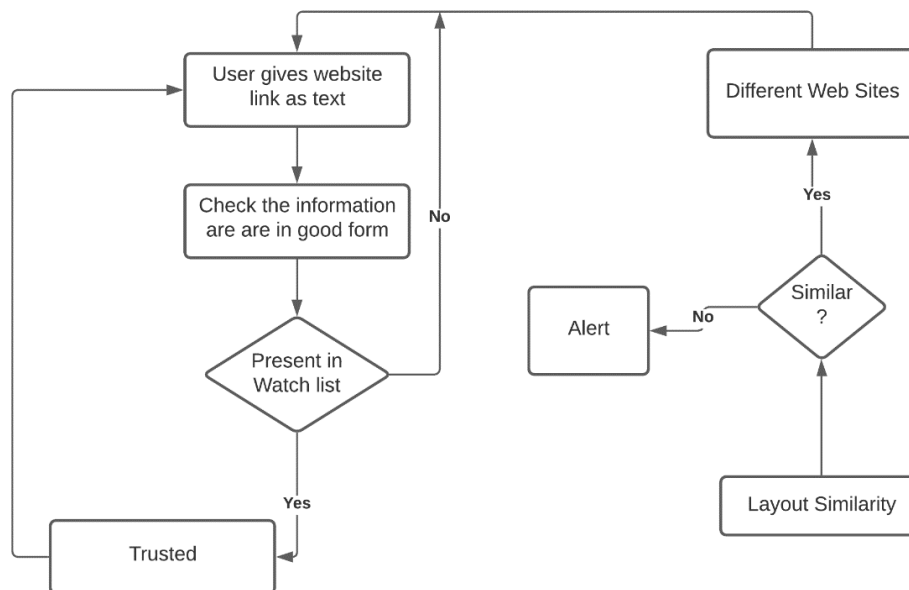


Figure 3.6.2: Flow diagram of Fake website detection.

1. Compare domain of each URL with the white list of trusted domain and also the black list domain. The data require for both the lists would be extracted dynamically [15].

2. The machine learning algorithms such as Random Forest, Decision Tree, logistic regression will be applied to the collected data and score is generate.
3. Our proposed solution provides normally three levels of security that is block and can prove to be more effective and more accurate than others.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1: Experimental setup

To begin testing, we had to use Google Colab, Google Research's Collaboratory, or "Colab" for short, is a product. Colab is a web-based Python editor that allows anyone to write and run arbitrary Python code. It's notably useful for machine learning, data analysis, and education. Then we used Numpy, Pandas, Matplotlib, Seaborn, NLTK, Cufflinks, scikit-learn, OS, Warnings, Strings, Wordcloud, and other Python libraries. We have to import several packages in order to import the above libraries. When we encountered an import issue, we had to reinstall the requirements. This is how we set up our system.

4.2: Experimental Results & Analysis

TABLE 4.2.1: ALGORITHMS COMPARISONS

Algorithm Name	Accuracy
Naïve Bayes	97.15
Random Forest	97.6
KNN	97.3
SVM	94.3
Decision Tree	97.45
Neural Network	84.46

Above table exploits the accuracy for different algorithms after successful implementation to our dataset. Some more previous has been published on this topic but our study gets higher accuracy and safest than other. The above exploitations can be visualized graphically that are shown below-

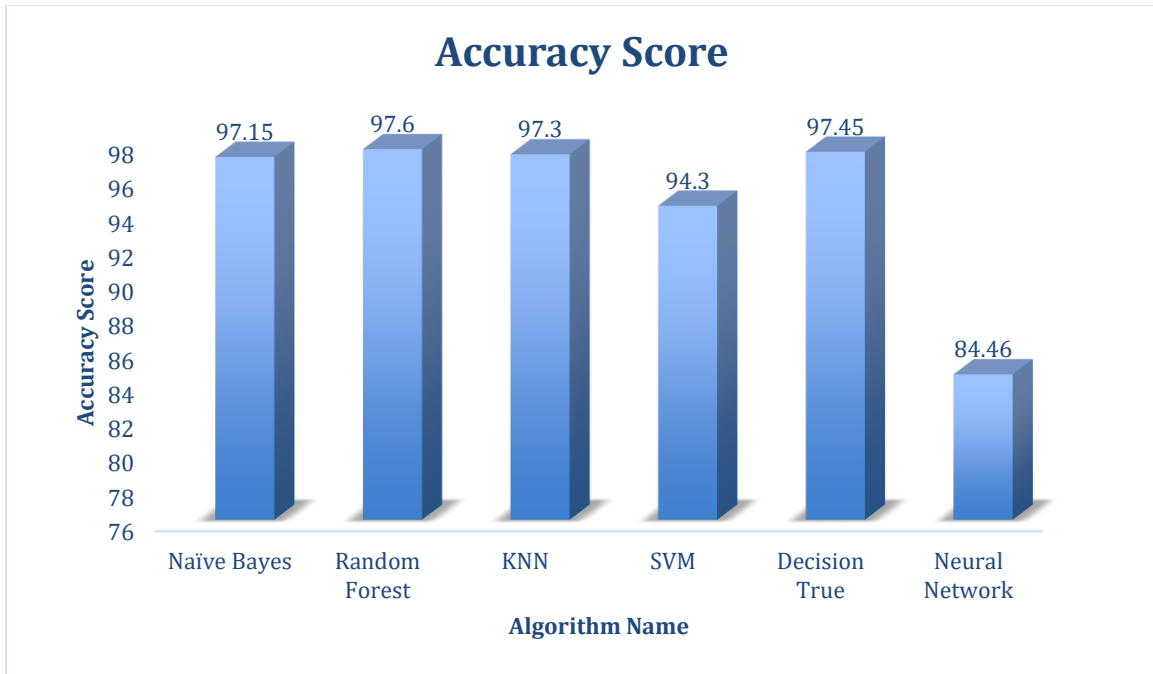


Figure 4.2.1: Visualize Algorithms accuracy

They're having lot more ways that can find out the difference between fraud and real website. To make the difference we applied some machine learning, Artificial Neural Network algorithms. The above figure exploits visually their comparisons among themselves. Random Forest, KNN, Decision tree and Naïve bayes algorithms are got high accuracy among them Random Forest classifier machine learning algorithms got the highest accuracy for our dataset.

4.3: Result Discussion

After applying the Machines learning algorithm and ANN we get a great accuracy for our dataset, And we measures those accuracy depends on some performance metrics. We describe those in bellow:

Precision: Precision refers to the number of correct documents returned by a machine learning model. It was the percentage of classifiers that were categorized as positive that were truly positive. That is, precision refers to the number of positive class predictions that are correct.

$$Precision = \frac{TP}{TP + FP}$$

Recall: The amount of positive class predictions made from all positive cases in the dataset is measured by recall.

$$Recall = \frac{TP}{TP + FN}$$

Here,

TP stands for true positive value.

TN stands for genuine negative.

FP stands for False Positive Value

FN stands for False Negative Value

F1-score: Harmonic mean of precision and recall, often known as F1-score.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Accuracy: Accuracy is the number of correct predictions divided by the total number of forecasts.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Confusion Matrix: A confusion matrix is a technique for summarizing a classification algorithm's performance.

TP	FP
FN	TN

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT & SUSTAINABILITY

5.1: Impact on Society & Environment

As we said before, our main aim of this conceptual study is to remove or reduce fraud by fake websites from our society and also make awareness among people such that they are able to distinguish between fake and real websites. Because millions of people are not aware about fraud website hacks. As a result, some innocent people lose or give most sensitive information to the attackers by logging into their portal or API without any knowledge and that are caught by attackers. The impact on society for fake website detection is that it will make people aware to distinguish between fake and real websites and it will reduce crimes to the society.

5.2: Sustainability Plan

The sustainable plan of our study is that we will easily detect a fake or real website and for that reason it will be steady if we develop a web application for that kind of fake website detection. Though the project we develop in our study is not completely sustainable or dynamic but if we make a web or android based application for these projects then it will be more sustainable to the users. Because users need to easily enter to the system and get the result with low cost and high accuracy.

CHAPTER 6

SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

6.1: Summary of the study

Fake websites are becoming much more prominent, generating billions of dollars in false income through unwary Internet users. Users would have a great deal of difficulty manually recognizing these websites as phony due to their design and appearance. Something we found websites that are full of malware but its appearance are looks like exactly same and we cannot make difference between real and fake website. Some innocent peoples logins or enter attackers system. As a result, users' sensitive information are got caught by attackers without any knowledge of innocent peoples. For that reason, we develop a model that will help people to distinguish between real and fake website. And finally, we found high accuracy for our model.

6.2: Discussion

This is the section in which an overall appraisal of the results of the work will be presented. It is here that one will have the opportunity to demonstrate the understanding of the work and to give a critical account of what has been achieved. This is a very important section of the report in terms of the assessment of work. Compare domain of each URL with the white-list of trusted domains and also the black-list of illegitimate domains. The data required for both the lists would be extracted dynamically by web scraping and stored on the server [2].

6.3: Conclusion

Fake websites are becoming much more prominent, generating billions of dollars in false income through unwary Internet users. Users would have a great deal of difficulty manually recognizing these websites as phony due to their design and appearance. Something we found websites that are full of malware but its appearance are looks like exactly same and we can not make difference between real and fake website. Some innocent peoples logins or enter attackers system. As a result, users' sensitive information

are got caught by attackers without any knowledge of innocent peoples. For that reason, we develop a model that will help people to distinguish between real and fake website. And finally, we found high accuracy for our model.

6.4: Future work

We will try to improve our project on dynamic field that will make easier for users. Though the result we got in our study that is satisfactory by comparing with other research study. But it has some lacking on providing 100% accuracy. Though, our study uses less noisy dataset as a result we got highest accuracy but in future we will try to implement our project dynamically that will work for all datasets.

REFERENCES

- [1] Selvakumari, M., Sowjanya, M., Das, S., & Padmavathi, S., "Phishing website detection using machine learning and deep learning techniques," *Journal of Physics: Conference Series*, vol.1916(1), pp. 012169, (2021).
- [2] Patil, Vaibhav; Thakkar, Pritesh; Shah, Chirag; Bhat, Tushar; Godse, S. P. (2018). [IEEE 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Pune, India (2018.8.16-2018.8.18)] 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Detection and Prevention of Phishing Websites Using Machine Learning Approach. , (), 1–5. doi:10.1109/ICCUBEA.2018.8697412
- [3] Kulkarni, A., & L., L., "Phishing Websites Detection using Machine Learning," *International Journal of Advanced Computer Science and Applications*, vol. 10(7), 2019.
- [4] Vilas, Mahajan Mayuri; Ghansham, Kakade Prachi; Jaypralash, Sawant Purva; Shila, Pawar (2019). [IEEE 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT) - Mysuru, India (2019.12.13-2019.12.14)] 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT) - Detection of Phishing Website Using Machine Learning Approach. , (), 384–389. doi:10.1109/ICEECCOT46775.2019.9114695
- [5] Naresh Kumar D., "Detection of Phishing Websites using an Efficient Machine Learning Framework," *International Journal of Engineering Research And*, vol. 9, 2020.
- [6]
- [7] Zamir, A., Khan, H., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A. and Hamdani, M., "Phishing web site detection using diverse machine learning algorithms," *The Electronic Library*, vol. 38, pp. 65-80,2020.
- [8] Geyik, K. Erensoy and E. Kocyigit, "Detection of Phishing Websites from URLs by using Classification Techniques on WEKA," *6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 120-125, 2021.
- [9] S. S. Birunda and R. K. Devi, "A Novel Score-Based Multi-Source Fake News Detection using Gradient Boosting Algorithm," *International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 406-414, 2021.
- [10] I.Kareem and S. M. Awan, "Pakistani Media Fake News Classification using Machine Learning Classifiers," *International Conference on Innovative Computing (ICIC)*, pp. 1-6, 2019.

- [11] P. Singh, Y. P. S. Maravi and S. Sharma, "Phishing websites detection through supervised learning networks," International Conference on Computing and Communications Technologies (ICCCT), 2015, pp. 61-65, 2015.
- [12] A. Basit, M. Zafar, A. R. Javed and Z. Jalil, "A Novel Ensemble Machine Learning Method to Detect Phishing Attack," IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-5, 2020.
- [13] M. Akhavan and S. M. Hossein Hasheminejad, "An Unsupervised Feature Selection for Web Phishing Data using an Evolutionary Approach," 7th International Conference on Web Research (ICWR), 2021, pp. 41-47, 2021.
- [14] Bai, Weighing (2020). [IEEE 2020 International Conference on Computing and Data Science (CDS) - Stanford, CA (2020.8.1-2020.8.2)] 2020 International Conference on Computing and Data Science (CDS) - Phishing Website Detection Based on Machine Learning Algorithm. , (), 293–298. doi:10.1109/CDS49703.2020.00064
- [15] Patil, Vaibhav; Thakkar, Pritesh; Shah, Chirag; Bhat, Tushar; Godse, S. P. (2018). [IEEE 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Pune, India (2018.8.16-2018.8.18)] 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Detection and Prevention of Phishing Websites Using Machine Learning Approach. , (), 1–5. doi:10.1109/ICCUBEA.2018.8697412
- [16] Korkmaz, Mehmet; Sahingoz, Ozgur Koray; Diri, Banu (2020). [IEEE 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) - Kharagpur, India (2020.7.1-2020.7.3)] 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) - Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. , (), 1–7. doi:10.1109/ICCCNT49239.2020.9225561

PLAGIARISM REPORT

Turnitin Originality Report

Processed on: 04-Dec-2021 20:03 +06
ID: 1720392907
Word Count: 6077
Submitted: 1

mahbuba By Most. Hena

Similarity Index	Similarity by Source
26%	Internet Sources: 15%
	Publications: 11%
	Student Papers: 12%

2% match (publications) Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCCBEA), 2018
2% match (Internet from 08-May-2010) http://www.staffs.ac.uk/assets/Project%20Guide2009-10%20%20MEng%20level%204%20september%202009_tcm44-25665.pdf
2% match (Internet from 28-Jun-2020) https://www.ijcaonline.org/archives/volume181/number23/mahajan-2018-ijca-918026.pdf
1% match (student papers from 02-Dec-2021) Submitted to University of Hertfordshire on 2021-12-02
1% match (student papers from 03-Dec-2021) Submitted to University of Hertfordshire on 2021-12-03
1% match (student papers from 06-Oct-2021) Submitted to Ghana Technology University College on 2021-10-06
1% match (Internet from 01-Oct-2021) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5263/161-15-7251%20%20%2820_%29.pdf?isAllowed=&sequence=1
1% match (publications) M Selvakumar, M Sovjanva, Sneha Das, S Padmavathi, "Phishing website detection using machine learning and deep learning techniques", Journal of Physics: Conference Series, 2021
1% match (student papers from 10-Dec-2017) Submitted to Indian Institute of Management, Bangalore on 2017-12-10
1% match (Internet from 14-Jul-2021) https://www.ijert.org/detection-of-phishing-websites-using-an-efficient-machine-learning-framework
1% match (Internet from 16-Oct-2021) https://iopscience.iop.org/article/10.1088/1742-6596/1916/1/012169
1% match (student papers from 15-Apr-2021) Submitted to Higher Education Commission Pakistan on 2021-04-15