A Proactive Approach to Predict Phishing Websites

MSc Research Project
Cybersecurity

Keerthi Prabhakar
Student ID: x19211023

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: KEERTHI PRABHAKAR

Student ID: x19211023

Programme: MSc in Cybersecurity **Year:** 2020-2021

Module: MSc Internship

Supervisor: Imran Khan

Submission Due Date: 16/12/2021

Project Title: A Proactive Approach to Predict Phishing Websites

Word Count: 7087 **Page Count**: 29

I hereby certify that the information contained in this (my submission) is the research information I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project. ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other authors' written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: 16/12/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

List of Figures

List of Tables

Abstract

Among Cyber-attacks, Phishing is the most prevailing attack that deceit users to provide sensitive data like credentials, bank and financial details which later wields to loss of funds. Although there are several sources like text messages, voice calls, typically, e-mails are the main source to target naive users. Prototypical users are misled by the attackers who create an exact duplicate of a legitimate website however making it malicious. As a result, it is paramount to efficiently detect and eliminate this attack. The fundamental focus of this experiment is to predict the URLs that are not genuine, and to make users informed of such social engineering attempts. This study mainly deals with the extraction of relevant features of URL to detect the unsafe links that are a copy of authentic web pages. An ensemble model is developed employing Deep learning (DL) approaches and Machine learning (ML) techniques to foresee the authenticity of the URL. This study has developed an optimal model that has achieved an accuracy of 99.1% for the Random Forest classifier.

CHAPTER 1: INTRODUCTION

1.1 Background of Study

Most organizations and online users are using third party mediums for various transactions and other purposes which has been made feasible by this premise. However, cyber-attackers often take advantage of this and post a source as trusted and gain confidential information of the users including credit card details, transactional details, login credentials, and other important data using social engineering techniques(Blum *et al.*, 2010). It usually happens whenever an intruder pretends as a credible service provider and lures a victim to access an e-mail, text-messages, harmful website, or an instant-chat. Due to lack of technological complexities, the end-users cannot identify the difference between an original webpage and a fake one which made them victims of cyber-attacks. In this chapter, the reason for the successful phishing attack and the background of the study would be discussed. Moreover, research aims, and objectives would be developed in this chapter for keeping the research in a veritable manner.

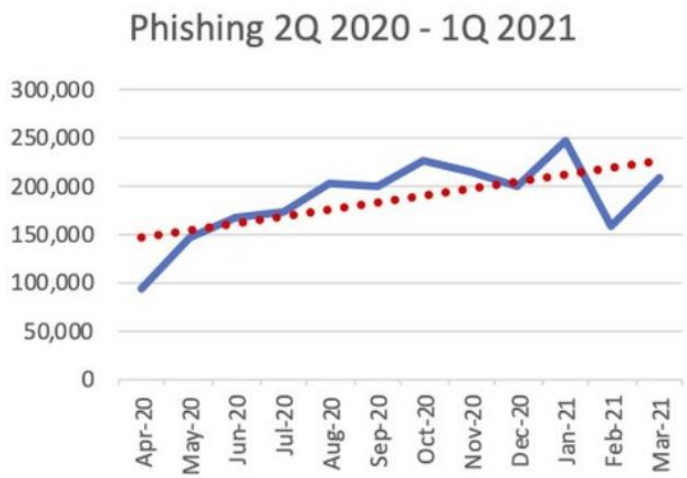


Figure 1: Increase of phishing sites

Internet-related criminality has advanced at a rapid rate due to technical developments. The fact that individuals are uninformed of certain high-tech abilities is a key reason for the rise in phishing attacks(*Phishing Report*, *Figure 1*). Many internet users are unaware of how web applications work on a technical level. Differentiating a fake website from a legitimate website, like "comprehending the structure" or "comprehending the importance" of URLs, is challenging for ordinary users(Bahnsen *et al.*, 2017). The security indicators that are normally available in web

browsers are not being used. The phishing attack utilises graphical deception to influence the users to access the phishing links. In this case, it has been a serious issue in the development of the websites as the cyber-attackers mimic the websites appropriately in making the users input their confidential data.

Figure 2:Enhancement of Phishing reports during Covid

From (*Phishing Reports*, Figure 2) , it can be inferred that phishing has been increased significantly at the end of the year 2019 during the pandemic. Some methods (Abroshan *et al.*, 2021) for thwarting phishing scams are widely implemented. The most effective technique to prevent phishing attacks is to educate members of the organization about the “significance and severity” of phishing attempts. Phishing protection relies heavily on safeguarding information(Balogun *et al.*, 2021). However, training staff for each possible phishing event is not practical. Professionals are inevitably deceived as cybercriminals use “social engineering” in luring end-users into becoming phishing victims. Previous attempts should be restricted from “domain URLs” and “known IPs”. Nonetheless, cyber attackers always find new IP addresses using the latest technologies for different attempts in cyber-attacks. Hence, proactively detecting phishing is the aim of this research which is discussed in this report.

1.2 Problem statement

Phishing poses as a trustworthy source and steals confidential user information. Although there are various sources like text messages, calls it is generally done by e-mail. The intent is to pilfer confidential information like credentials, credit-cards details or to install malicious on the victim's system(Alzuwaini and Yassin, 2021). Because of inadequate knowledge and cluelessness,

attackers may simply trick online people into clicking on illegal websites that appear legitimate, and this is the most pressing issue to fix. Further, the study deals with the victim's inability to distinguish authentic websites from phishing websites due to a lack of understanding of phishing website techniques that causes significant issues. Additionally, end-users lack of understanding while browsing fake sites on the internet may result in the loss of private information to cybercriminals. Access to a person's Social Security card allows for the collection of all papers about that person's nationality, i.e., the theft of his identity. Even hacked credit card details could be used to re-create a person's identity. Phishing attacks are amongst the most prevalent security issues that both businesses and individuals face when it comes to protecting their privacy. Businesses are especially a valuable target(Alkhalil *et al.*, 2021) and alot of web-users are being victims to such social-engineering assaults. Thus, difficulty in the implementation of an acceptable and successful strategy for anticipating phishing attempts via spoof websites.

1.3 Objective

The key motive of this work is to establish a model which envisages and recognizes phishing sites. End-users have been observed to be victims of cyber-attacks while surfing fake websites on the internet. Cyber-attackers acquire data and information from consumers by creating phoney websites like the actual ones and propagating ransomware threats. This study discovers the effectiveness and processes of URL feature analysis by employing deep learning and machine learning techniques in predicting Phishing URLs. Lack of technical knowledge has impacted the end-users significantly as they are tricked. Hence, the first objective would detect phishing web applications which facilitates the end-users to differentiate the original websites from fake phishing websites(Korkmaz, Sahingoz and DIri, 2020). It also would increase their perception of phishing attacks and other cyber-attacks. Having less knowledge on phishing and fake websites, the end-users often diverted to fake websites which resulted in phishing attacks. Hence, the third objective would help to understand the challenges in preventing phishing attacks. In this way, the primary aim of the study would be achieved, a platform that, proactively detect phishing attack which also contributes to spreading awareness among the internet users to differentiate phishing websites.

The primary question to consider in this study is, how efficient are ML and DL techniques in predicting malicious URL? This proposed question assists to achieve the objective and tangibly conduct the research. The research question helps to create an ensemble model which will predict a fake website. Therefore, assisting online users to be vigilant about the cyberattacks on web transactions and other purposes(Basit *et al.*, 2020). Hence the research questions help in attaining a sustainable model to predict phishing websites.

1.4 Implication of the study

Significance to research

This research covers most types of phishing techniques in terms of collecting data of internet users from the web. Moreover, it also poses two main approaches in identifying the phishing websites properly in making the end-users aware(Moreno-Fernández *et al.*, 2017). Hence, this research would be significant to the academic researcher who might continue to work on Phishing detection projects. Moreover, it also would be beneficial for the students in gaining appropriate knowledge about phishing processes. It would also be beneficial for students to be aware of the strategies that may be used to effectively thwart phishing attempts.

Significance to end-users

Due to the advent of technology, it has been observed that end-users who are uninformed of cyber-attacks are the victims of phishing assaults. (Mao *et al.*, 2017) This research would educate end-users in fully comprehending the process of social engineering techniques so that they are warned when accessing a phishing website. As an outcome, internet users can avoid becoming a victim of phishing. Furthermore, standard browsers would aid in the prevention of phishing assaults, as well as make people more conscious when surfing and entering private information.

Significance to cybersecurity organization

This research has introduced a unique approach for identifying phishing websites in preventing phishing attacks properly. In this case, URL feature analysis has been introduced which has facilitated various approaches and processes which would help to point out the phishing websites effectively. Hence, this research would be helpful to develop their typical processes of preventing

phishing attacks. Moreover, it would also help to implement the process in various ways so that the organization would have significant development in their security.

1.5 Structure

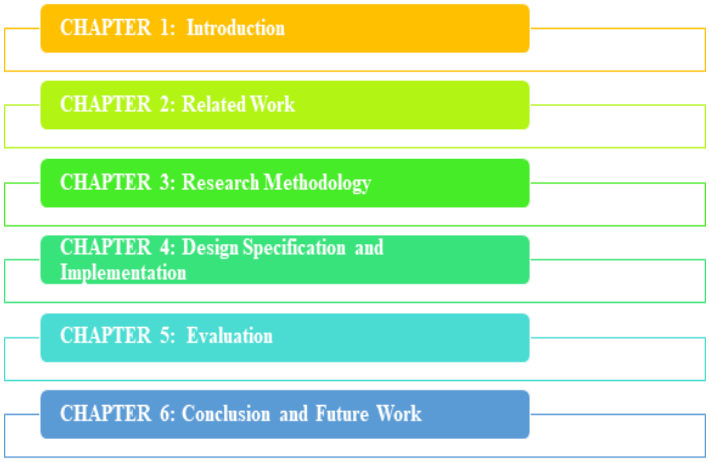


Figure 3: Research Structure

Chapter 1 deals with the basic background of the study where an overview of the phishing attack and the research objective.

Chapter 2 deals with the related work in the past which reflects previous research. In this chapter phishing and prevention methods are mainly approached so that the end-users can be made aware.

Chapter 3 deals with the research methodology which would help to analyse and collect data on the topic.

Chapter 4 deals with the techniques and architectures which would allow the implementation of prevention approaches. additionally, it discusses the implementation of the model.

Chapter 5 deals with the evaluation of the main findings and evaluation through comprehensive analysis. Describes the utilization of statistical tools in developing justified conclusions.

Chapter 6 deals with the conclusion on the topic depending on the findings and analysis and future scope also would be shown here for next research.

CHAPTER 2: Literature Review

Even though software firms release new anti-phishing tools that include blacklists, algorithms, visuals as well as machines attempting to learn techniques, these technologies cannot eliminate all

phishing assaults. As the web evolves, user safeguards must grow in lockstep to ensure people remain secure online.

2.1 Approaches

The COVID-19 pandemic has increased the use of technologies in all sectors, culminating in the physical relocation of activities such as organising business events, attending classes, purchasing, making payments, and so forth. This means that phishers will have more possibilities to carry out assaults that may harm the victim economically, mentally, and professionally(Gowtham and Krishnamurthi, 2014). Detecting phishing sites is difficult due to the need for URL encryption to abbreviate the URL; link requests are routed, altering links to appear trustworthy. This requires a departure from traditional development approaches moves towards advanced techniques. Ad hoc approaches have generally been applied to detect phishing attempts, content-based recommendations, the URL of the website, and so forth. There are three main types of detecting attacks.

- ***Content-Based Approach:*** Examines a page's content material utilising copyright, a null foot wide links, zero body Web pages links, and links with both the highest frequency domains. Simply using the TF-IDF algorithms, 97% of phishing websites may be identified, with only 6% of wrongful convictions. (Patil and Dhage, 2019)
- ***URL-Based Approach:*** Incorporates page rank alongside additional metrics generated from URLs based on past knowledge. This technique can detect up to 97% of phishing websites.
- ***Machine Learning Approach:*** Existing machine learning classifiers are trained on characteristics such as whether the Address contains @, whether it has quadruple slash redirection, the page authority of the URL, this same number of external sites contained on the website, and so on. This method might achieve a TPR of 92%.

2.2 Related works on Phishing

2.2.1. Research using Machine-Learning:

Concept drift is a substantial issue in dynamically evolving contexts where the statistical features of the target parameter fluctuate. To address this issue, (Tan *et al.*, 2018) has employed a pipeline that utilizes an adaptive learning mechanism to identify malicious URL's. Real-time HTTP request

traffic for 44 days transiting at their campus network's Points of Presence (PoPs) was analysed, along with a malignant dataset from the Phish tank. To distinguish conceptual drifts, a measurable test procedure centred on the (WRST) Wilcoxon-Rank-Sum-Test was utilized and various machine learning classifiers were deployed to achieve high efficiency. WRST is a non-parametric test employed to determine if two independent tests were obtained from populations with similar distribution. This pipeline yielded a precision of 96.9% and a FRP of 3.5%. The project was tested with an imbalance dataset between original and non-malicious URLs.

Zero-day Phishing attacks had increased rapidly during the Covid-19 pandemic. (Yadollahi *et al.*, 2019) suggested a smart phishing identification solution which extracts URL features based on HTML Document Object Model (DOM) and employs XCS which is a web-oriented learning framework with rules. XCS is a semi-supervised framework that can update the rules defined when the parameters of the web page change. This framework was tested against 4021 non-malicious web pages and 3983 malicious web pages. This framework achieved an efficency of 98.1% and a 1.59% FPR. The approach is mainly a client-side solution that concentrates on features related to analysis the ones that exploit the HTML Document Object Model (DOM).

(Ortiz Garces, Cazares and Andrade, 2019) suggested a pre-emptive approach that analyses contaminated data and then determines if the output is legitimate or not. Kaggle dataset around 11000 with a data matrix of (420464 x 2) was applied to examine the efficiency of the model. This simulation was purely based on cognitive security architecture that was implemented by combining AI-based procedures and machine learning approaches. The model predicted the output by analysing 2 features, Google index and having subdomain. This experiment has provided results for just short length URLs.

In recent years, a growing number of academics have acquired the knowledge of these linked, multi-typed datasets akin to heterogeneous information networks, and therefore have created analysis tools that take advantage of the high-level semantic significance of architectural sorts of objects and linkages in the network infrastructure. HinPhish extracts diverse connection associations from websites and constructs heterogeneous network information integrating domains and resource items (Guo *et al.*, 2021). HinPhish employs customized ML algorithms to capitalize

just on the peculiarities of various link kinds in determining the phish-score of a user domain on the web application. This experiment was conducted on datasets retrieved from Phishpedia (Lin *et al.*, 2021) and OpenPhish(*OpenPhish - Phishing Intelligence*, 2021) and Alexa(*Alexa*, 2021). This model acquired an accuracy of 98.5% for the Random Forest algorithm. This study mainly focuses on Domain and network information.

2.2.2. Research using Deep Learning Approaches

Deep Learning already had a significant influence in fields like cancer detection, pharmacogenomics, self-driving vehicles, futuristic predictions, and speech synthesis. (Lin *et al.*, 2021) Differentiated instruction, categorisation, and information processing algorithms require carefully built feature extractors that are not sustainable for massive datasets In many circumstances, contingent on the issue complexity, DL can also remediate the restrictions of prior shallow systems, which impeded training program and synthesis of hierarchical depictions of the multivariate training dataset. Deep neural network (DNN) employs numerous (deep) tiers of modules that are significantly optimized in terms of methodologies and structures. (Shrestha and Mahmood, 2019)

This study (Yi *et al.*, 2018) proposes dual categories of web-based phishing characteristics: unique parameters and interactive parameters. Feature extraction was done by implementing an SVM model with a hidden layer. A dynamic graphical model (DNB) Deep-belief network, a type of deep learning model was constructed of several tiers of latent variables with contacts across levels but not across modules in every unit. This model analyses the real network traffic data from ISP for 40 minutes and 24 hours and predicts malicious web pages. This deep learning model achieved an correctness of 89.6% and a FRP of 0.6%. The data availability requires access to real-time ISP network traffic data.

The features retrieved from various dimensions are highly definitive; nevertheless, extracting these features is time-consuming. (Yang, Zhao and Zeng, 2019) recommended a deep-learning simulation that identifies fake web applications using multi-dimensional features extraction. Initially, character pattern features from the given URL were retrieved and fed into a deep learning

model for efficient classification. Then the URLs are merged with statistical parameters such as website code and text parameters. Phishtank(*PhishTank | Join the fight against phishing*,2021) and dmoztools(*The Directory of the Web*, 2021) are the sources for datasets used in this experiment. This research acquired a FRP of 0.59% and an effectiveness of 98.7%.

Another study by (Saha *et al.*, 2020) describes a data-driven methodology for identifying malicious URLs by employing a deep learning technique. To be more specific, a multilayer perceptron, termed a feed-forward neural network, has been leveraged to anticipate phishing websites. A dataset from Kaggle with eight features was tested using this model. This model obtained a test efficiency of 93% and training accuracy of 95%.

2.3 Summary

Authors	**Methodology/Algorithm**	**Dataset sources**	**Observations**
Shantanu; B Janet; R Joshua Arul Kumar (Tan *et al.*, 2018)	Wilcoxon Rank-Sum Test (WRST for brief) with ML algorithms	Real-time HTTPS dataset from their campus network and Phish tank	Dataset imbalance
Afsaneh Madani; Mohammad; Farzaneh Shoeleh; Elham Serkani; Hossein Gharaee Mehdi Yadollahi (Yadollahi *et al.*, 2019)	HTML-(DOM) Document-Object-Model characteristics and XCS	Source not mentioned	Mainly just checks for HTML-(DOM) Document-Object-Model characteristics
Maria Fernada Cazares; Roberto Omar Andrade: Ivan Ortiz Garcés (Ortiz Garces, Cazares and Andrade, 2019)	Combination of AI-based procedures and machine learning approaches	Kaggle dataset	Tested only for short length URL's
Fan Shi, Yunyi Zhang, Chengxi Xu, Min Zhang, Bingyang Guo,	Modified ML algorithms	Phishpedia, Open-Phish, Alexa	Predominantly checks for Domain and network

Yuwei Li (Guo *et al.*, 2021)			information in the URL
Wei Wang, Ping Yi, Ting Zhu, Yao, Futai Zou, Yuxiang Guan, (Yi *et al.*, 2018)	Deep Learning framework using SVM classifier and Deep Belief networks	Real-time data from ISP for 40 minutes and 24 hours	Requires access to real-time ISP data.
Peng Yang; Guangzhen Zhao; Peng Zeng (Yang, Zhao and Zeng, 2019)	Deep Learning – convolutional neural network (CNN)	Phish tank and dmoztools	This model works only for datasets with multi-dimension features
Sohrab Hossain; Dhiman Sarma; Asma Sultana; Mohammad Nazmul Alam; Rana Joyti Chakma; Ishita Saha; (Saha *et al.*, 2020)	Deep Learning	Kaggle	Less number of features are tested.

Table 1: Related Work Summary

CHAPTER 3: METHODOLOGY

3.1 Agile Methodology

Phishing is an ongoing cyber-attack that keeps on enhancing with improvement in technology. Agile methodology (*Agile Methodology*, *Figure 4)* provides room for improvements in every stage of the process thus this particular methodology has been used for the research.

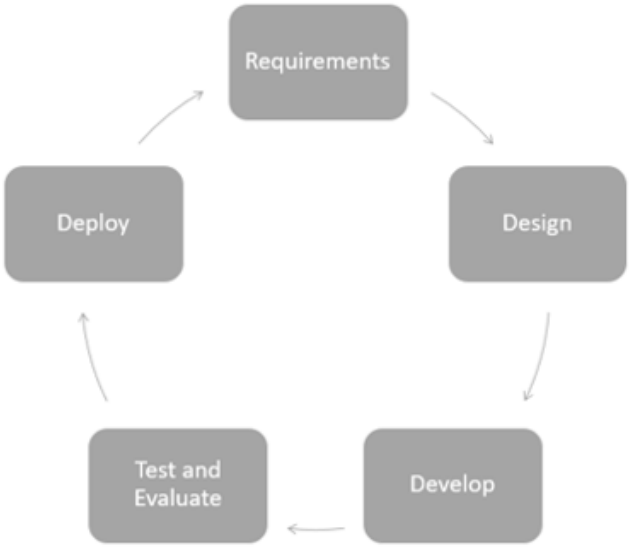


Figure 4: Agile Methodology

3.2 Proposed Model Design

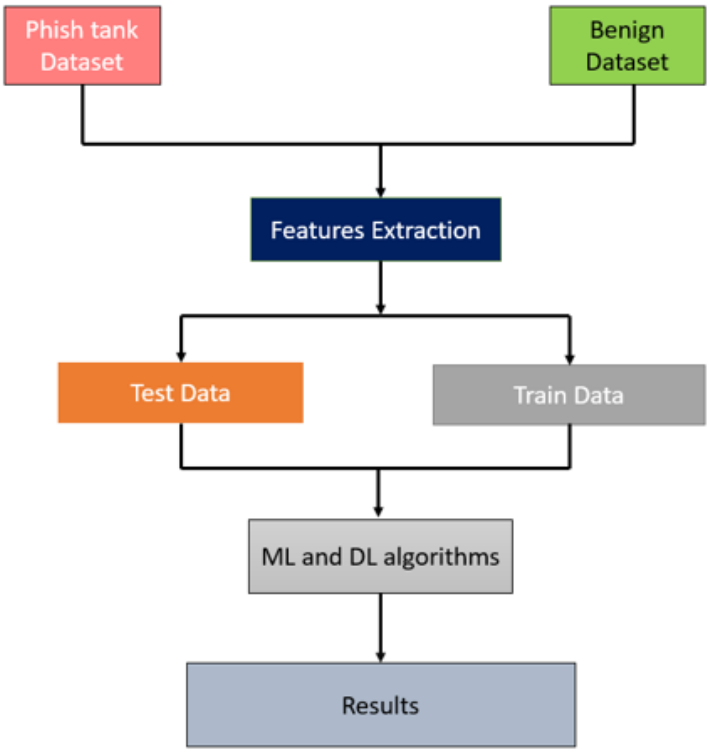


Figure 5: Proposed Design

The proposed design can be specified as steps required for the whole process. The concerning machine learning system processes start with accessing the webpage by the user. Phishing attacks generally start with an email that is sent by the fraudster. That email carries a link, and the user is enticed to access it. By accessing the link, the end user is then led to a phishing website. That website must be taken as the input to the system. The webpage has some features that look the same as a legitimate website.

3.2.1 Dataset:

This is the part of the project where an explanation will be presented on the methods that are chosen to execute the study. The required data is collected from the Phish tank(*PhishTank*, 2021) for malicious URL's and the Benign dataset is extracted from the University of New Brunswick (UNB)(*University of New Brunswick*, 2021). Based on the requirements, tools and techniques required for the analysis are finalized. After that all the chosen techniques are analysed before implementation, to predict the outcome and their probable effects on analysis. In this part, all the data is analysed with the chosen tools and techniques and the results are evaluated.

3.2.2 Features extraction:

As the features of the website, domain name structure has been analyzed. Length of the website, numbers of characters, and letter to digit ratio have been measured. A domain is a unique element and that cannot be copied. (Tan *et al.*, 2018). A column of the dataset consists of domain names where all the possible website names have been saved. A python code is programmed to extract the 30 features of the dataset. The purpose of the proposed method was to implement a system that can predict a phishing website. The chosen data set has been imported into a data frame, which has been separated into two tasks: training and testing. The model must be thoroughly trained in order to forecast and make judgments for a fake website (Patil and Dhage, 2019). Eighty percent of the data frame has been used for the training purpose and the rest twenty percent has been used for testing purposes. It has been seen that if the training of the machine is done with equal data distribution, then the result of accuracy increases systematically. In that case, the data frame must be divided into different segments and tested as experiment 1 (Exp1) and experiment 2 (Exp2).

3.2.3 Algorithms:

The CSV dataset file is fed into the below algorithms

- **Machine Learning algorithms**
  - Random forest
  - Logistic Regression
  - Decision Trees

- **Deep Learning algorithms**
  - FastAi
  - CNN using Kera TensorFlow

These algorithms analyse the test and train data and predict for the selected 30 features if the provided URL legit or not. (Yadollahi *et al.*, 2019)(Yi *et al.*, 2018) this prediction aids in the detection of phishing URLs. The output is measured in the accuracy of the model. The results will be evaluated in Chapter 5 of this report.

CHAPTER 4: DESIGN SPECIFICATION AND IMPLEMENTATION

This chapter describes all the requirements and specifications that are used to implement the model.

4.1 Requirements

4.1.1 Hardware

This design was built using an HP laptop featuring the corresponding hardware compatibility:

- Storage: 256GB SSD with 1 Terabyte
- Graphical Processing Unit: Iris Plus Graphics
- Central Processing Unit: Intel 10th Gen i7 Processor with 2.4 GHz

- RAM: 16GB DDR4

4.1.2 Software

- **Google Collab:** an open-source online compiler is used for all the coding tasks required for this project. (*Google Colaboratory*, 2021)
- **Libraries**: Pandas, Seaborn, Keras, request, OpenCV, urllib, Beautiful Soup, whois, matplotlib, TensorFlow, SKLearn, Glob, OS, Matplotlib, Numpy, Fastai and PyTorch.
- **Python3:** programming language utilised to develop the model

4.2 Implementation

4.2.1 Collection of Data

The prime motive of this phase is to describe how the data has been collected and the importance of this data in different stages of the project. For malicious URLs, the relevant data is obtained from Phish Tank (PhishTank, 2021), while the benign dataset is obtained from the University of New Brunswick (UNB) (University of New Brunswick, 2021). Entirely 11000 URL's are analysed in this experiment.

4.2.2. Feature Extraction:

The URL features extraction is done using the python code and a binary output file is obtained. In the data set there are some columns also used for data collection, few of them are, consisting of an URL IP address, thus the user must check if the URL contains any IP addresses. Length of the phishing URL will be embedded with various characters it is crucial to examine the length thus this feature is extracted. The lifespan of the phishing URL is short hence domain age feature is extracted which gives us lifespan of the URL by subtracting expiration date and creation date. In this project the domain age is considered to 12 months. If the URL's domain age is shorter than twelve months, it is recognized as phishing. Security sensitive is a feature where certain token words are defined and the program checks for those token words and classify the URL accordingly.

Phishing URL's also contains executable files that are the users are unaware of, so the python code also checks for any ".exe" files in the URL. Likewise the program generates the list and gets the feature list. (Aydin and Baykal, 2015)(Paliath, Qbeitah and Aldwairi, 2020).

Features	
Abnormal_URL	Page_Rank
age_of_domain	popUpWidnow
DNSRecord	port
Domain_registeration_length	Prefix_Suffix
double_slash_redirecting	Redirect
exe	Request_URL
Google_Index	RightClick
having_At_Symbol	Security_sensitive
having_IP_Address	SFH
having_Sub_Domain	Shortining_Service
HTTPS_token	SSLfinal_State
Iframe	Submitting_to_email
Links_in_tags	URL_Length
Links_pointing_to_page	URL_of_Anchor
on_mouseover	web_traffic

Figure 6: Extracted Features

The data is generated as an ".csv" file by the python code and will be accessible to download. An analogous method is used for the phishing URL list. Then the obtained dataset is separated into test and train and then supplied to deep learning and machine learning algorithms(Mohammad, Thabtah and McCluskey, 2014). After the calculation, the result column is showing all the ways most commonly taken by the phishers to build a phishing site and trap all the unconcerned persons and steal much relevant information from their devices(Aydin and Baykal, 2015). After getting the result the decision is made on the ways of predicting the phishing websites and differentiating them from the real ones.

Correlation Matrix:

A python code was developed to determine the correlation matrix of the features extracted to examine the correlation between the various characteristics. The acquired correlation matrix will be used to examine how each attribute complements the other.

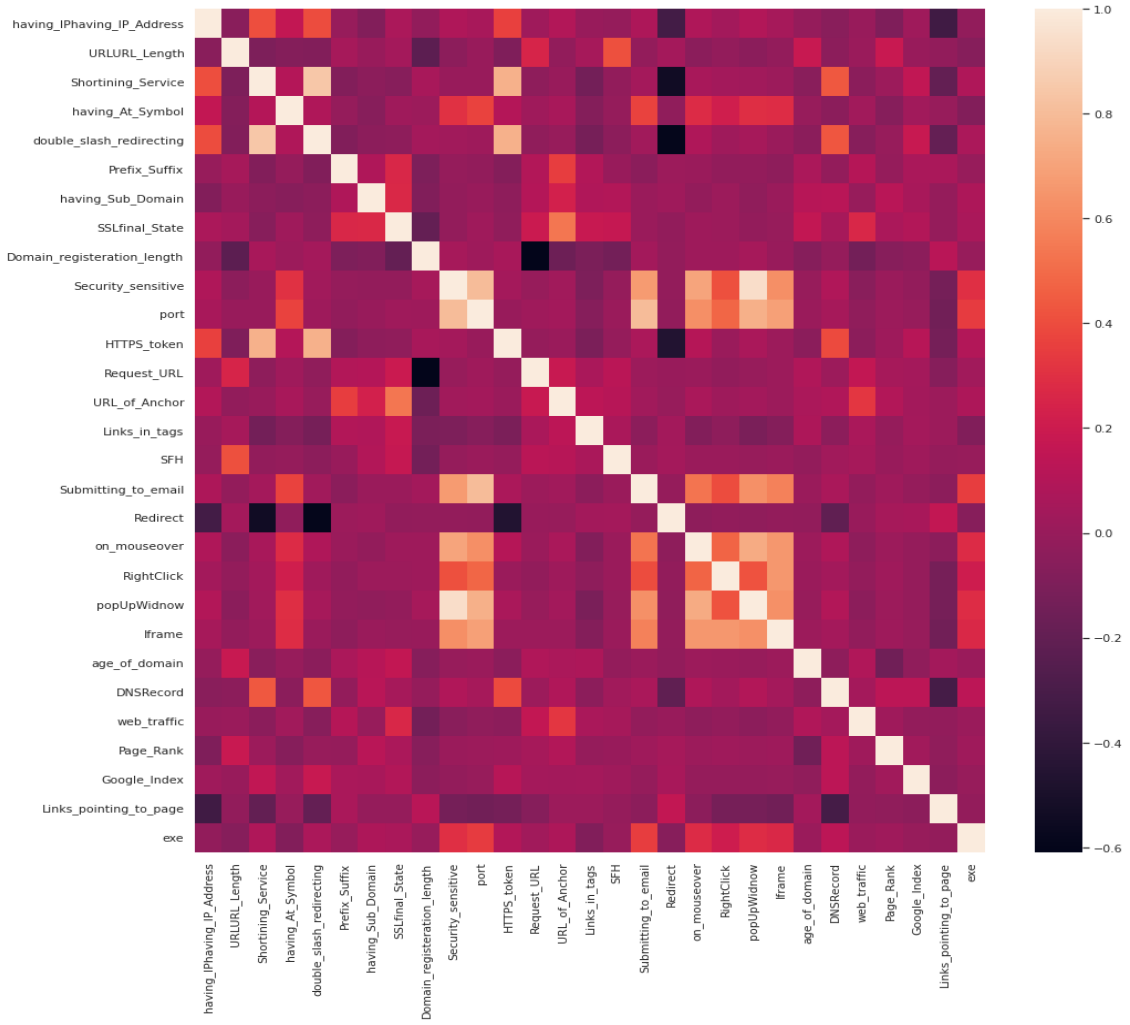


Figure 7: Correlation Matrix

The correlation between the extracted URL features is depicted in the matrix above. The measure functions best with variables that have a linear connection with one another. A heatmap is leveraged to graphically display the data's fit. The above matrix illustrates the positive and negative correlations among the independent parameters, allowing to depict how they interact. As the number of variables employed in the correlation research were substantial in number, a separate function in Python was employed to extract the characteristics that had a robust relationship in

predicting the result. The filtration criteria for these extracted attributes were set at 0.7. *Abnormal_URL, HTTPS_token, Result, Submitting_to_email, double_slash_redirecting, on_mouseover, popUpWidnow, port* these variables are said to have positive correlation coefficient value which implies these features have strong positive correlation in predicting the result.

Data Distribution Matrix:

Two experiments were conducted with the different numbers of Legit(1) and phishing (-1) URLs. The below pie chart provides the overview of how our data is distributed for both experiment1 (Exp1) and expermient2 (Exp2). In Exp1, the dataset of 11000 URL's 55.7% is legit URL and 44.3% is malicious URL. Later, the ensemble model was also tested for Exp2 with 50% legit URLs and 50% malicious URLs for 10000 total URLs. In the below, pie chart (1) is legit and (-1) is malicious (-1) URLs.

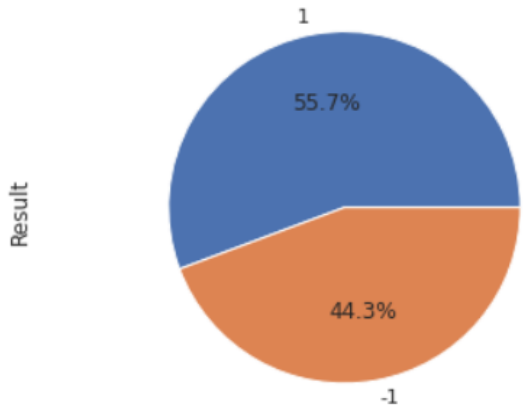


Figure 8: Data Distribution- Exp1

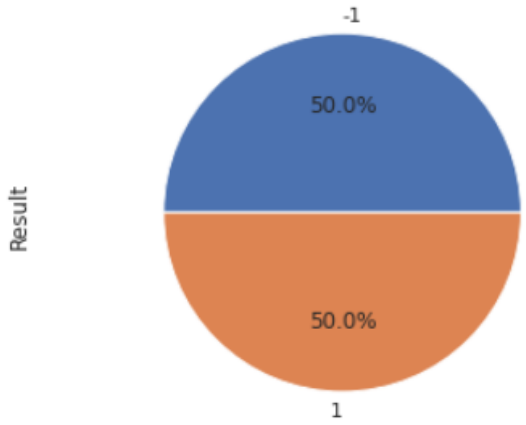


Figure 9: Data Distribution - Exp2

4.2.3 Training and Testing

URL analysis has been done on the chosen dataset by selecting appropriate columns of it as target and predicted variable through proper algorithm and techniques. Train and test mechanisms have been used for phishing website detection. For data splitting, the *train_test_split()* function from the *SKLearn* library was leveraged and train-test dataset was generated (Medar, Rajpurohit and Rashmi, 2018). In Deep Learning, *the StratifiedShuffleSplit* and *StratifiedKfold* function was deployed for the same (Yang, Zhao and Zeng, 2019). Both the spilt functions use 80:20 as the main to test ratio. Train models have been performed analysis whose result can be given by the test

model. In simple words, the models that have been developed in the training part has been evaluated in the test part. The test dataset has never been used for training purposes. Predictions have been made based on the model using python coding and it has returned predicted values of target variables.

4.3 Proposed Model

4.3.1 Machine Learning Algorithms:

Logistic Regression Logistic regression can be stated as the technique of predictive analysis in python. Since the output is dichotomous, it has been used to constrain or limit the output between -1 to 1 in the research. The evaluation is mainly on the association between independent and dependent variables. The analysis of logistic regression has predicted the outcome in a variable that has just two possible outputs. The "dependent variable" can be categorical and is also known as the target variable. The variables that have been considered as independent are actual predictors. It has predicted the event probability using the "log function"(Khurma *et al*., 2021).

Decision tree algorithms have handled data with high dimensions and good accuracy(MacHado and Gadge, 2018). All the internal nodes in the "decision tree" have a rule of decision that splits the datasets. For this experiment categorial variable decision tree has been deployed as all the variables are categorial. Decision trees can be stated as easy to visualize and interpret. Besides that, it can also easily capture nonlinear patterns. It needs fewer amounts of "data pre-processes" from a user. It can also be used to predict missing values.

Random Forest classifier has been used by selecting random samples from the provided dataset. Random forest solves categorization and regression difficulties using ensemble approaches (bagging). At the training phase, the method creates a large decision tree-based and outputs the mean/mode of the prognosis. It has taken the average of each prediction that has cancelled out the biases. (Rashid *et al.*, 2020).

All the above classifiers will be trained, tested on the obtained dataset and the prediction is measured in accuracy.

4.3.2 Deep Learning Algorithms:

Fast Ai: fastai is a library(Jawade and Ghosh, 2021) that provides practitioners with sophisticated features that can yield ground-breaking results in deep learning applications in a timely and expedient manner, as well as restricted features that can be merged and synced to construct unique methodologies. (Howard and Gugger, 2020). Initially, the data is normalised and split using the *StratifiedShuffleSplit* function. In this module, cyc_len defines the Epoch which determines the number of cycles the dataset was interpreted by the model. Batch size defines that size of the data set divided as samples and loaded to the model at one time. Due to the computational power of the host machine batch size is set to 64. The dataset for the model was overfitting as the validation loss was increasing hence to address this concern the Epoch value is set to 10 to maintain the quality of the model and faster execution. Then the model starts its prediction and yields the results.

CNN using [Keras-Tensorflow]: CNN(Singh, Singh and Pandey, 2020) is a robust research topology; the principle underlying CNN is employed in the article, the convolution layer learns better to represent data input in a high-level automated method. The application of CNN contours the information to obtain the categorization model. StratifiedKFold is a KFold variant. *StratifiedKFold* rearranges the data initially, then splits the data into n_splits portions and finishes. It then utilizes each component as a test set. This function shuffles data just once before the split occurs. In the proposed model the n_split value is assigned as 10 which indicates cross-validation of 10-folds. To avoid overfitting of the data DropOut and activation functions are utilised. Rectified Linear Unit (Relu) activation function is employed as it does not stimulate all of the neurons at once(Al-Milli and Hammo, 2020). As a result, the biases, as well as the weights for specific neurons, are not revised throughout the back-propagation operation. To normalize the inputs automatically into layers and to avoid internal covariate shift *BatchNormalization()* function is deployed. Once data is treated then the designed model predicts the outcome.

CHAPTER 5: EVALUATION

For evaluating the implementation of the proposed model, 2 experiments were conducted to predict using URL features to recognize the server problems to host the web pages that include different aspects of research perspectives. In the first experiment (Exp1), 55.7% is legit URL and 44.3% is

malicious URL with a total of 11000 URLs. In Exp2 with 50% legit URL and 50%, malicious URL's for 10000 total URL's in the data set. There are some important successive factors to initiate its overall analysis in an improved way. This chapter has proposed all its segments to analyse its phases to understand the internal mechanism.

5. Results

5.1 Logistic regression

Accuracy values of this logistic regression have been stated as per its predictive after applying this on data frame to proceed with the analysis. Specification of data analysis majorly factorises the detection levels to develop the implementation procedure for phishing detection here on the analysis to enhance the level of its analysis. Exp1 yielded an accuracy of 95.6% and Exp2 gave 95.6% accuracy.

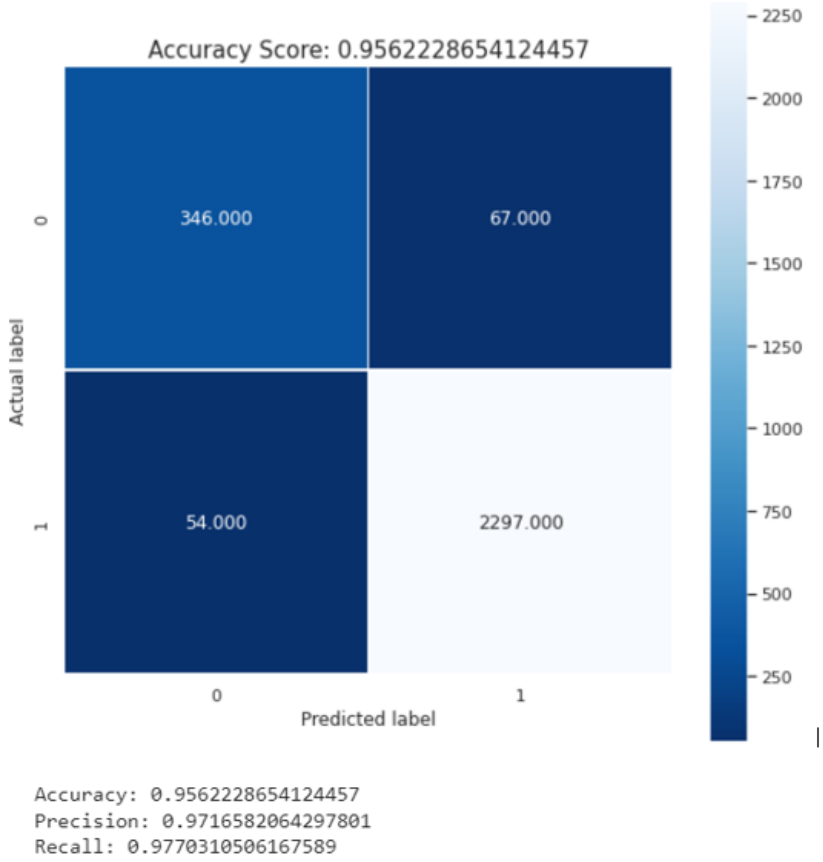
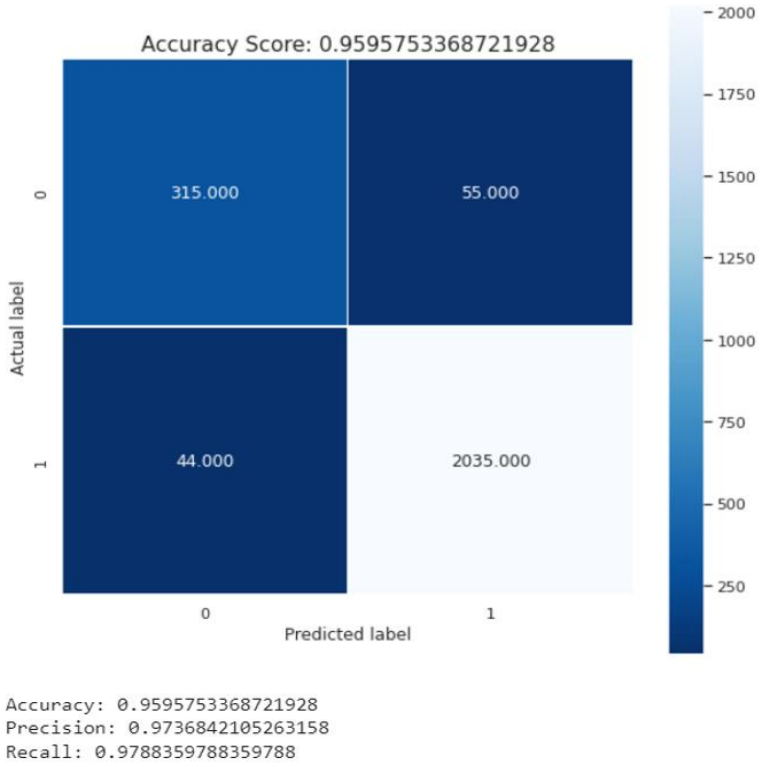


Figure 10: Exp1 Logistic Regression results



Accuracy: 0.9595753368721928
Precision: 0.9736842105263158
Recall: 0.9788359788359788

Figure 11: Exp2 Logistic Regression Result

5.2 Decision tree

For the effective method, the algorithm must be applied here to perform the possible real-time application. For the design of this overall work procedure, this decision tree has been plotted to present the conditional approaches to perform the statistical factors over here as its variables have been created here. There is an accurate score of 98.3 % for Exp1 and 98.5 % for Exp2 has been formed to get an understanding of the prediction.

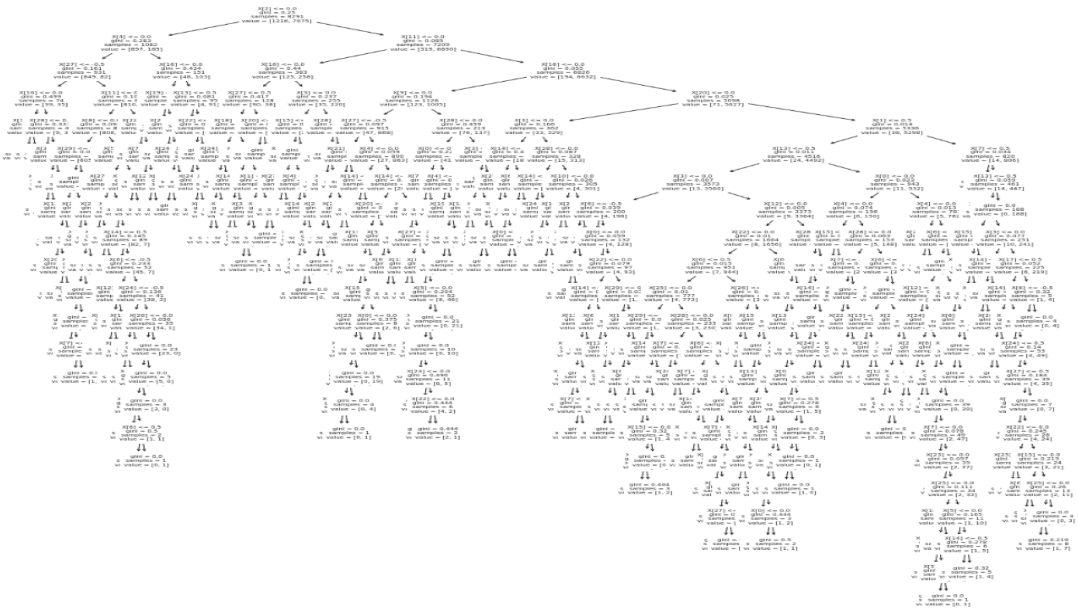


Figure 12: Exp1 Plot for Decision Tree

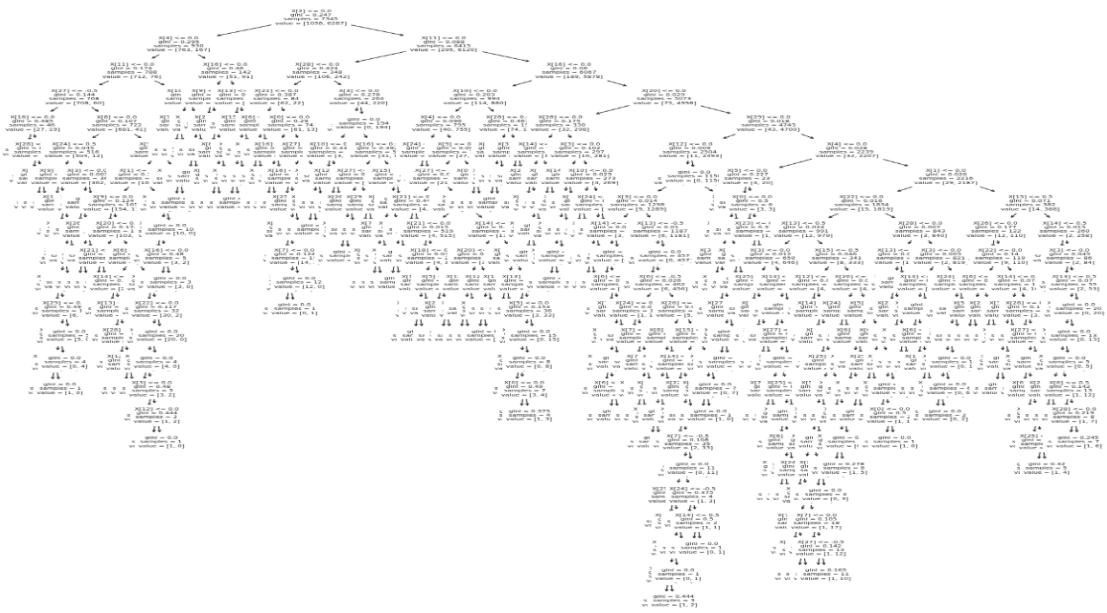
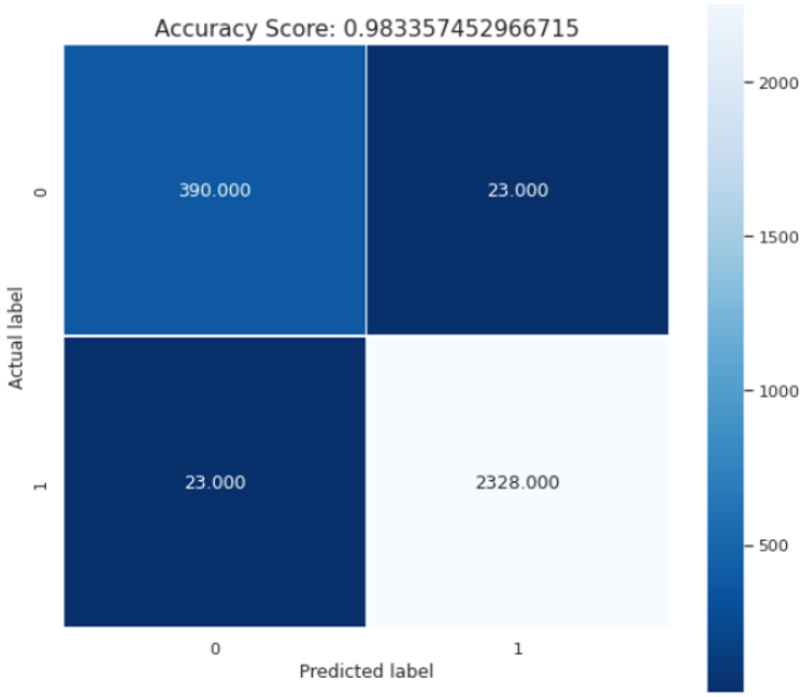
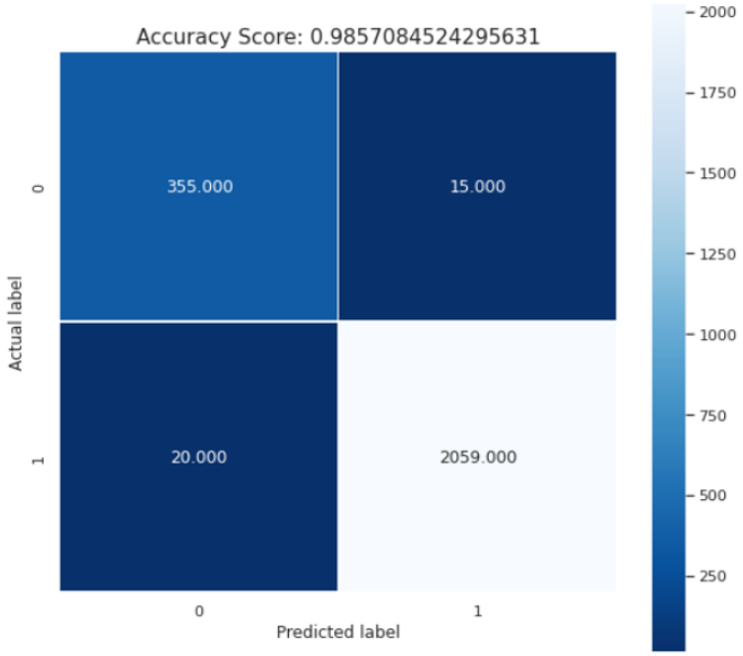


Figure 13: Exp2 Plot for Decision Tree




```
Accuracy: 0.983357452966715
Precision: 0.9902169289663972
Recall: 0.9902169289663972
```

Figure 14: Exp1 Results for Decision Tree




```
Accuracy: 0.9857084524295631
Precision: 0.9927675988428158
Recall: 0.9903799903799904
```

Figure 15: Exp2 Results for Decision Tree

5.3 Random Forest classifier

In the Google Collab, there are each of the algorithm performances to get fitted the models. After this phishing detection, an accuracy of 98.9 % for Exp1 and 99.1 % for Exp2 has been gained here to deliver the internal specification to measure the accuracy and its efficiency. With python, an overall process has been solved to capture the considerable factors of phishing websites. The features of the implementation process have been selected here to get an accuracy score here.

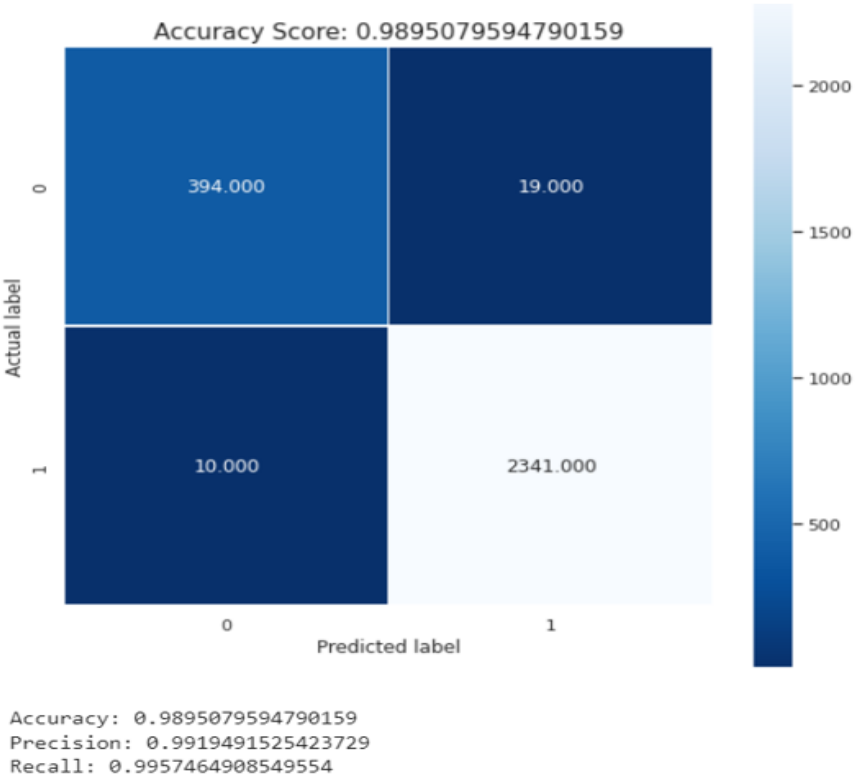
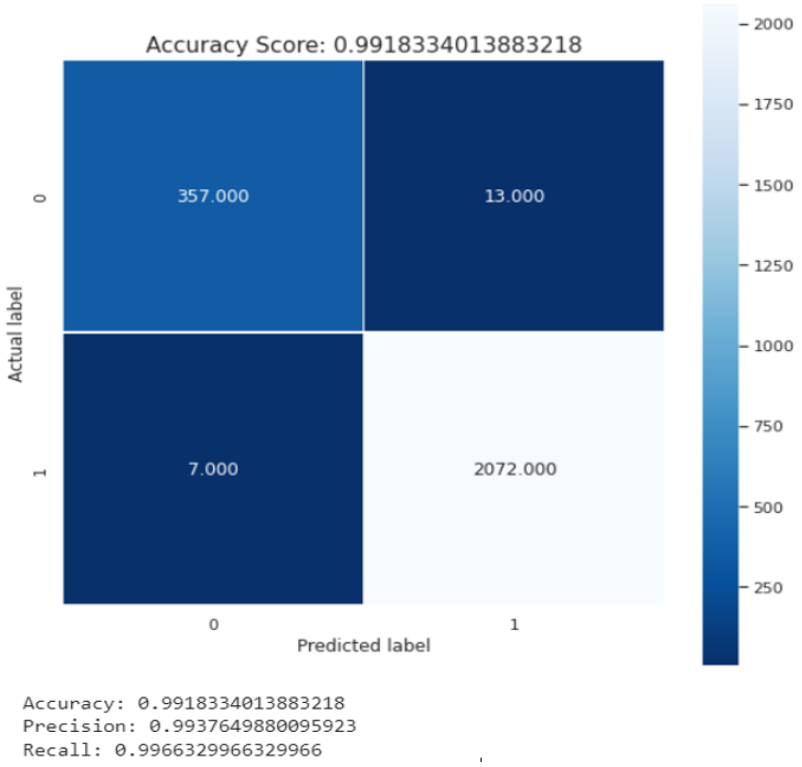


Figure 16: Exp1 Random Forest results



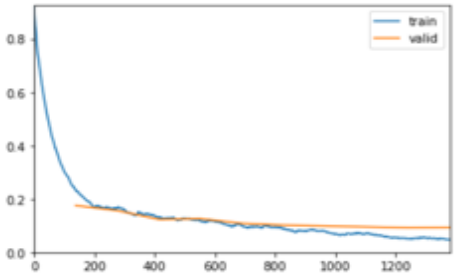
Accuracy: 0.9918334013883218
Precision: 0.9937649880095923
Recall: 0.9966329966329966

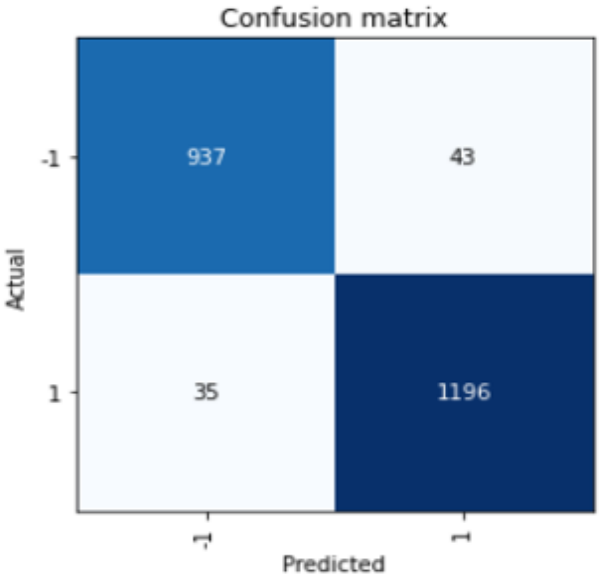
Figure 17: Exp2 Random Forest results

5.4 FastAi

In this segment, the results obtained using Deep Learning Models is interpreted. Fastai and CNN model using Keras Framework. It is noticed that Fastai tabular algorithm yielded an average accuracy of 96.5% with 10 epochs and an average loss of 0.092%. Both values are concerning validation loss and validation accuracy. When the model was evaluated against the Test dataset, the accuracy for Exp1 is 96.4% and for Exp2 is 96.27%.

epoch	train_loss	valid_loss	accuracy	time
0	0.235703	0.175077	0.928539	00:01
1	0.163145	0.156254	0.933967	00:01
2	0.132317	0.123112	0.951153	00:01
3	0.118776	0.125701	0.952058	00:01
4	0.099791	0.108089	0.958390	00:01
5	0.090042	0.101566	0.958842	00:01
6	0.079052	0.098282	0.960199	00:01
7	0.070648	0.095126	0.967436	00:01
8	0.051430	0.091843	0.965626	00:01
9	0.047622	0.092749	0.964722	00:01

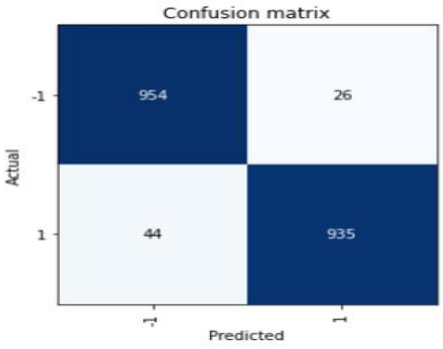


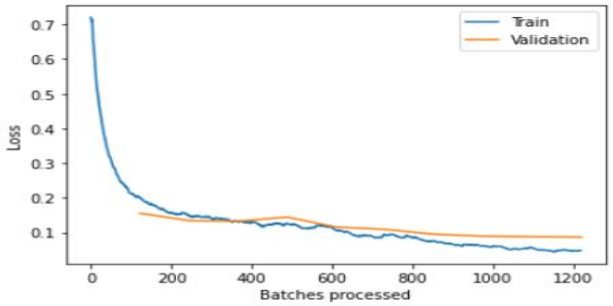



```
loss 0.09274942427873611: accuracy: 96.47%
```

Figure 18: Exp1 FastAi results

epoch	train_loss	valid_loss	accuracy	time
0	0.201680	0.155047	0.940276	00:01
1	0.147524	0.134152	0.947422	00:01
2	0.130001	0.131977	0.947422	00:01
3	0.125591	0.144183	0.945891	00:01
4	0.106072	0.115262	0.953548	00:01
5	0.093490	0.108794	0.958142	00:01
6	0.075584	0.094487	0.963247	00:01
7	0.061650	0.089193	0.964778	00:01
8	0.055591	0.087428	0.963757	00:01
9	0.047716	0.086328	0.964267	00:01



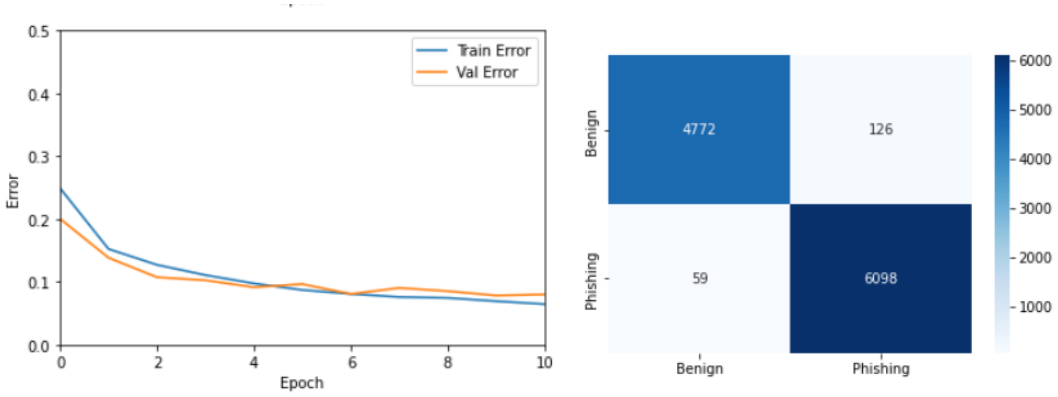



```
loss 0.0863279402256012: accuracy: 96.43%
```

Figure 19: Exp2 FastAi results

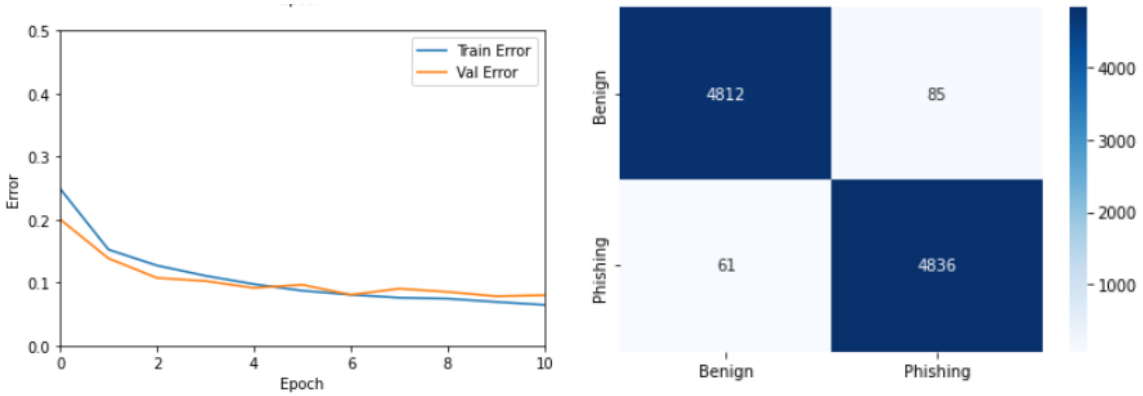
5.5 CNN-Keras TensorFlow

Further experiments were conducted using CNN developed using Keras framework and the results obtained from this Deep Learning model is presented in this section. CNN model using Keras Framework employs a cross-validation of K-fold technique to learn the dataset and evaluate the data for the given URL. This algorithm produced an accuracy of 98.33% for Exp1 and 98.51% for Exp2 with 10 epochs and an average loss of 0.04%. The confusion matrix shows how the data is represented for each fold.




```
173/173 [==============================] - 0s 2ms/step - loss: 0.0436 - accuracy: 0.9833
['loss', 'accuracy']
Accuracy: 98.33%: loss: 0.04
```

Figure 20: Exp1 CNN-Keras TensorFlow results




```
154/154 [==============================] - 0s 2ms/step - loss: 0.0396 - accuracy: 0.9851
['loss', 'accuracy']
Accuracy: 98.51%: loss: 0.04
```

Figure 21: Exp2 CNN-Keras TensorFlow results

5.6 Summary of Result:

The ensemble model provides an accuracy range of 95.6% to 99.1% for the chosen classifiers. The outcome for both experiments Exp1 and Exp2 infers that for Exp2 the accuracy was slightly higher except for FastAi deep learning algorithm. Associating the results implies, among machine learning and deep learning algorithms, Random Forest provides the highest accuracy 99.1% and logistic regression provided the least efficiency of 95.9% for experiment 2 where the data distribution is 50% -50%. The result summaries that the prediction capability is marginally higher and more efficient for the equal data distribution of the dataset.

	Algorithms	**Exp1 Accuracy in %**	**Exp2 Accuracy in %**
Machine Learning	Logistic Regression	95.6	95.9
	Decision Tree	98.3	98.5
	Random Forest	98.9	99.1
Deep Learning	FastAi	96.4	96.2
	CNN-Keras Tensorflow	98.3	98.5

Table 2: Result Synopsis

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

Phishing detection considers several factors, objectives according to technological advancement. For significant phishing detection for URL analysis to distinguish between the fake and the original websites based on a different basis for the challenges to develop the factors as per the authorized and unauthorized segments. For establishing the mitigation process of the overall analysis, an ensemble procedure has been formed to get versatile aspects of the techniques to ensure the solutions for combining the preventive measures of this analysis process in a better way. This URL analysis is performed to get an accurate result to form a basic method to classify the attributes

capability. By employing on the DL and ML models, calculations have proceeded to phase the training on this dataset to perform the factors so that in future there would be a more efficient analysis procedure in a relevant way. As the accuracy attained by the ensemble model is 99.1% this implies a significant value which ensures that the result based on the consideration of this overall procedure to achieve an optimal model.

6.2 Challenges

Dataset: Phishtank is an opensource tool that updates phishing URL hourly, thus the URL's download from this source should be within 60 minutes.

6.3 Future work

As a Plugin

The model has been tested with various machine learning and deep learning algorithms which has provided good accuracy in predicting the malicious URL. This model can be used to develop a browser extension that can be compatible with all browsers. The browser extension can alert online users when accessing a fake website.

Independent software

For this evaluation of the phishing websites, the use of Google Collab has been used based on Python language to modify the ensured factors of the overall method. In future instead of Google Collab, there would be a use of MATLAB rather than the Google Collab. There would be some more advantages as to get more accuracy here so that in future there would be some advantages as to perform all its accessed factors so that easily, the result would be gained. With MATLAB simulation and obtained calculation would have to be achieved to perform this phishing website prediction. Validation of the model has different restrictions so that a confidential design method would work in a better way.

Dataset

This model can be enhanced to test on a larger dataset of legit and fake websites.

References

Abroshan, H. *et al.* (2021) 'COVID-19 and Phishing: Effects of Human Emotions, Behavior, and Demographics on the Success of Phishing Attempts during the Pandemic', *IEEE Access*, 9, pp. 121916–121929. doi: 10.1109/ACCESS.2021.3109091.

Agile Methodology (2021). Available at: https://www.nvisia.com/insights/agile-methodology (Accessed: 13 December 2021).

Al-Milli, N. and Hammo, B. H. (2020) 'A Convolutional Neural Network Model to Detect Illegitimate URLs', *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, pp. 220–225. doi: 10.1109/ICICS49469.2020.239536.

Alkhalil, Z. *et al.* (2021) 'Phishing Attacks: A Recent Comprehensive Study and a New Anatomy', *Frontiers in Computer Science*, 3, p. 6. doi: 10.3389/FCOMP.2021.563060/BIBTEX.

Alzuwaini, M. and Yassin, A. (2021) 'An Efficient Mechanism to Prevent the Phishing Attacks', *Iraqi Journal for Electrical and Electronic Engineering*, 17(1), pp. 1–11. doi: 10.37917/IJEEE.17.1.15.

Aydin, M. and Baykal, N. (2015) 'Feature extraction and classification phishing websites based on URL', *2015 IEEE Conference on Communications and NetworkSecurity, CNS 2015*, pp. 769–770. doi: 10.1109/CNS.2015.7346927.

Bahnsen, A. C. *et al.* (2017) 'Classifying phishing URLs using recurrent neural networks', *eCrime Researchers Summit, eCrime*, pp. 1–8. doi: 10.1109/ECRIME.2017.7945048.

Balogun, A. O. *et al.* (2021) 'Improving the phishing website detection using empirical analysis of Function Tree and its variants', *Heliyon*, 7(7), p. e07437. doi: 10.1016/J.HELIYON.2021.E07437.

Basit, A. *et al.* (2020) 'A Novel Ensemble Machine Learning Method to Detect Phishing Attack', *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*. doi: 10.1109/INMIC50486.2020.9318210.

Blum, A. *et al.* (2010) 'Lexical feature based phishing URL detection using online learning', *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 54–60. doi: 10.1145/1866423.1866434.

Google Colaboratory (2021). Available at: https://colab.research.google.com/?utm_source=scs-index (Accessed: 11 December 2021).

Gowtham, R. and Krishnamurthi, I. (2014) 'A comprehensive and efficacious architecture for detecting phishing webpages', *Computers and Security*, 40, pp. 23–37. doi: 10.1016/J.COSE.2013.10.004.

Guo, B. *et al.* (2021) 'HinPhish: An Effective Phishing Detection Approach Based on Heterogeneous Information Networks', *Applied Sciences 2021, Vol. 11, Page 9733*, 11(20), p.

9733. doi: 10.3390/APP11209733.

Howard, J. and Gugger, S. (2020) ‘fastai: A Layered API for Deep Learning’, *Information (Switzerland)*, 11(2). doi: 10.3390/info11020108.

Jawade, J. V and Ghosh, S. N. (2021) ‘Phishing Website Detection Using Fast.ai library’, pp. 1–5. doi: 10.1109/ICCICT50803.2021.9510059.

Keyword Research, Competitive Analysis, & Website Ranking | Alexa (2021). Available at: https://www.alexa.com/ (Accessed: 6 December 2021).

Khurma, R. A. *et al.* (2021) ‘Salp Swarm Optimization Search Based Feature Selection for Enhanced Phishing Websites Detection’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12694 LNCS, pp. 146–161. doi: 10.1007/978-3-030-72699-7_10.

Korkmaz, M., Sahingoz, O. K. and DIri, B. (2020) ‘Detection of Phishing Websites by Using Machine Learning-Based URL Analysis’, *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*. doi: 10.1109/ICCCNT49239.2020.9225561.

Lin, Y. *et al.* (2021) ‘Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages’, *undefined*.

Lin, Y. *et al.* (no date) ‘Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages’.

MacHado, L. and Gadge, J. (2018) ‘Phishing Sites Detection Based on C4.5 Decision Tree Algorithm’, *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*. doi: 10.1109/ICCUBEA.2017.8463818.

Mao, J. *et al.* (2017) ‘Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity’, *IEEE Access*, 5, pp. 17020–17030. doi: 10.1109/ACCESS.2017.2743528.

Medar, R., Rajpurohit, V. S. and Rashmi, B. (2018) ‘Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning’, *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*. doi: 10.1109/ICCUBEA.2017.8463779.

Mohammad, R. M., Thabtah, F. and McCluskey, L. (2014) ‘Intelligent rule-based phishing websites classification’, *IET Information Security*, 8(3), pp. 153–160. doi: 10.1049/IET-IFS.2013.0202.

Moreno-Fernández, M. M. *et al.* (2017) ‘Fishing for phishers. Improving Internet users’ sensitivity to visual deception cues to prevent electronic fraud’, *Computers in Human Behavior*, 69, pp. 421–436. doi: 10.1016/J.CHB.2016.12.044.

OpenPhish - Phishing Intelligence (no date). Available at: https://www.openphish.com/ (Accessed: 6 December 2021).

Ortiz Garces, I., Cazares, M. F. and Andrade, R. O. (2019) 'Detection of phishing attacks with machine learning techniques in cognitive security architecture', *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, pp. 366–370. doi: 10.1109/CSCI49370.2019.00071.

Paliath, S., Qbeitah, M. A. and Aldwairi, M. (2020) 'Phishout: Effective phishing detection using selected features', *Proceedings of the 2020 27th International Conference on Telecommunications, ICT 2020*. doi: 10.1109/ICT49546.2020.9239589.

Patil, S. and Dhage, S. (2019) 'A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework', in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE. Available at: https://ieeexplore.ieee.org/document/8728356/ (Accessed: 10 December 2021).

Phishing Report (2021). Available at: https://www.prnewswire.com/news-releases/apwg-q1-2021-report-detected-phishing-websites-maintain-historic-high-in-q1-2021-after-doubling-in-2020-301309187.html (Accessed: 14 December 2021).

Phishing Reports (no date). Available at: https://blog.knowbe4.com/apwg-q3-report-phishing-attacks-at-highest-level-in-three-years (Accessed: 12 December 2021).

PhishTank (no date). Available at: https://www.phishtank.com/developer_info.php (Accessed: 6 December 2021).

PhishTank | Join the fight against phishing (no date). Available at: https://phishtank.org/index.php (Accessed: 6 December 2021).

Rashid, J. *et al.* (2020) 'Phishing Detection Using Machine Learning Technique', *Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020*, pp. 43–46. doi: 10.1109/SMART-TECH49988.2020.00026.

Saha, I. *et al.* (2020) 'Phishing attacks detection using deep learning approach', *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, pp. 1180–1185. doi: 10.1109/ICSSIT48917.2020.9214132.

Shrestha, A. and Mahmood, A. (2019) 'Review of deep learning algorithms and architectures', *IEEE Access*, 7, pp. 53040–53065. doi: 10.1109/ACCESS.2019.2912200.

Singh, S., Singh, M. P. and Pandey, R. (2020) 'Phishing detection from URLs using deep learning approach', *Proceedings of the 2020 International Conference on Computing, Communication and Security, ICCCS 2020*. doi: 10.1109/ICCCS49678.2020.9277459.

Tan, G. *et al.* (2018) 'Adaptive Malicious URL Detection: Learning in the Presence of Concept Drifts', *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science*

and Engineering, Trustcom/BigDataSE 2018, pp. 737–743. doi: 10.1109/TRUSTCOM/BIGDATASE.2018.00107.

The Directory of the Web (no date). Available at: https://dmoztools.net/ (Accessed: 6 December 2021).

University of New Brunswick (no date). Available at: https://www.unb.ca/cic/datasets/url-2016.html (Accessed: 6 December 2021).

Yadollahi, M. M. *et al.* (2019) 'An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features', *2019 5th International Conference on Web Research, ICWR 2019*, pp. 281–286. doi: 10.1109/ICWR.2019.8765265.

Yang, P., Zhao, G. and Zeng, P. (2019) 'Phishing website detection based on multidimensional features driven by deep learning', *IEEE Access*, 7, pp. 15196–15209. doi: 10.1109/ACCESS.2019.2892066.

Yi, P. *et al.* (2018) 'Web phishing detection using a deep learning framework', *Wireless Communications and Mobile Computing*, 2018. doi: 10.1155/2018/4678746.