MALICIOUS URL DETECTION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

MOHAMMED KAIF S (810019205064)

SRIIKANTH U S (810019205096)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING, BIT CAMPUS,

TIRUCHIRAPPALLI - 620024

ANNA UNIVERSITY:: CHENNAI - 600 025

MAY 2023

MALICIOUS URL DETECTION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

MOHAMMED KAIF S (810019205064)

SRIIKANTH U S (810019205096)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING, BIT CAMPUS,

TIRUCHIRAPPALLI - 620024

ANNA UNIVERSITY:: CHENNAI - 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"MALICIOUS URL DETECTION USING DEEP LEARNING"** is the work of **"MOHAMMED KAIF S (810019205064), SRIIKANTH U S (810019205096)"** who carried out the project work under my supervision.

SIGNATURE
Dr. G. ANNAPOORANI,
Assistant Professor (Sl. Gr),
HEAD OF THE DEPARTMENT
Department of Information Technology,
University College of Engineering, BIT Campus,
Tiruchirappalli - 620024

SIGNATURE
Mrs. S. CHITRA DEVI,
SUPERVISOR
Assistant Professor,
Department of Computer Science and Engineering,
University College of Engineering, BIT Campus,
Tiruchirappalli - 620024

Submitted for "**IT8811 – Project Work**" in B.Tech. Information Technology Degree Jan – May 2023 Examination held on …….

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that the work entitled **"MALICIOUS URL DETECTION USING DEEP LEARNING"** submitted in partial fulfillment of the requirement for the award of the degree in B.Tech., Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli is a record of my work carried out by me during the academic year 2022- 2023 under the supervision of **Mrs. S. CHITRA DEVI, M.E.,** Assistant Professor, Department of Computer Science and Engineering, University College of Engineering, Anna University, BIT Campus, Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree or diploma, either in this or any other university.

(Signature of the Candidate) (Signature of the Candidate)

MOHAMMED KAIF S (810019205064) SRIIKANTH U S (810019205096)

I certify that the declaration made by the above candidate is true.

(Signature of the Guide)
Mrs. S. CHITRA DEVI, M.E,
Assistant Professor,
Department of Computer Science and Engineering,
University College of Engineering, BIT Campus,
Anna University, Tiruchirappalli.

ACKNOWLEDGEMENT

A truthful heartfelt and deserved acknowledgement comes from one's heart to convey the real influence others have on one's work.

We express our gratitude to our honorable Dean **Dr. T. SENTHIL KUMAR, M.E., Ph.D.,** for giving us chance to complete our education in one of the reputed government institutions running under his leadership. We express our sincere gratitude to our head of the department **Dr. G. ANNAPOORANI, M.Tech., Ph.D.,** for giving us the provision to do the project. We are much obliged to our project coordinators **Dr.K.UMAMAHESHWARI, M.Tech., Ph.D., and Mr.K.SARAVANA KUMAR**, **M.Tech.,** and our class coordinator **Dr. V.M.PRIYADHARSHINI, M.Tech., Ph.D.,** for giving us the opportunity to do the project, hearty thanks for them. We stand even more thankful to our project supervisor **Mrs. S. CHITRA DEVI, M.E.,** for guiding us throughout and giving us the opportunity to present the main project.

We also express our sincere thanks to all other staff members, friends, and our parents for their help and encouragement.

ABSTRACT

The World Wide Web services are essential in our daily lives and are accessed through Uniform Resource Locator URL links. Due to the rapid growth of internet services such as online banking, business, entertainment and government e-services like Aadhar and PAN now became online. This resulted in the increase of cyber crimes. Therefore, detecting malicious URLs is crucially important to prevent the occurrence of many cybercriminal activities. To address this issue, many web phishing and malware detection systems are developed to prevent users from falling victims to online threats such as phishing, malware, and scams. Machine learning techniques have show promising results in identifing malicious URLs. Existing only detects whether the url is malicious or benign. This system proposes a malicious URL detection system using a deep learning algorithm Multi-Layer Perceptron which classifies which type of malicious url it is and it also classifies idn homograph characters. The system is designed to extract feaures from URLs and classify them as benign, defacement, phishing and malware URLs. We experimentally evalute the performance of our system using a huge dataset comprising 651,192 records of URLs. Furthermore, we conduct feature importance analysis to identify the most significiant features that contribute to the system's performance. Experimental results demonstrate that the proposed method achieves an accuracy of 94.95%.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

1.	ML	–	Machine Learning
2.	DL	–	Long Short-Term Memory
3.	URL	–	Universal Language Model Fine tuning
4.	SVM	–	Support Vector Machine
5.	ANN	–	Artificial neural network
6.	RF	–	Random Forest
7.	DT	–	Decision Tree
8.	LR	–	Logistic Regression
9.	NB	–	Naïve Bayes
10.	JRIP	–	Joint Reserve Intelligence Program
11.	CNN	–	Convolution Neural Network
12.	CPU	–	Central Processing Unit
13.	RAM	–	Random Access Memory
14.	GPU	–	Graphics Processing Unit
15.	OS	–	Operating System
16.	IDE	–	Integrated Development Environments
17.	IP	–	Internet Protocol
18.	WWW	–	World Wide Web
19.	HTTPS	–	Hypertext Transfer Protocol Secure
20.	TP	–	True Positive
21.	TN	–	True Negative
22.	FP	–	False Positive
23.	FN	–	False Negative
24.	UML	–	Unified Modelling Language
25.	DFD	–	Data Flow Diagram

CHAPTER 1
INTRODUCTION

1.1 Introduction:

Before technical and application details, the background understating of problem source is very important. If source the background information is understood well, then easier to follow project description. In this chapter, existing state of society, their statistical information, how that issue will affect our future generation were described.

1.2 Background Understanding:

The rapid growth of the internet has provided numerous benefits to individuals and organizations, but it has also given rise to various cyber threats, including malicious URLs. Malicious URLs are web addresses that lead users to websites with harmful intent, such as phishing, malware distribution, or other cyber attacks. Detecting and preventing users from accessing such malicious websites is crucial for ensuring a secure online environment.

This project focuses on developing a malicious URL detection system using Multilayer Perceptron (MLP) neural networks. MLP is a popular type of artificial neural network that consists of multiple layers of interconnected nodes, known as neurons. With its ability to learn complex patterns and make accurate predictions, MLP has proven to be effective in various machine learning tasks.

The goal of this project is to leverage the power of MLP to classify URLs as benign, malware, phishing, defacement. **Benign URLs:** These are safe to browse URLs. Malware urls - These type of URLs inject malware into the victim's system once he/she visit such URLs. Defacement URLs are generally created by hackers with the intention of breaking into a **web server** and replacing the **hosted website** with one of their own, using techniques such as **code**

injection, cross-site scripting, etc. Common targets of **defacement** URLs are religious websites, government websites, bank websites, and corporate websites. **Phishing URLs:** By creating phishing URLs, hackers try to steal sensitive personal or financial information such as login credentials, credit card numbers, internet banking details, etc .By extracting relevant features from the URLs, such as domain information, path structure, and query parameters, we can create a comprehensive dataset for training the MLP model. The model will learn to recognize patterns associated with malicious URLs, allowing it to accurately identify potential threats.

To achieve this, we will first gather a diverse and representative dataset of labeled URLs, including both safe and malicious examples. We will preprocess the data, transforming the URLs into numerical representations suitable for input into the MLP. Next, we will design and train the MLP model, fine-tuning its architecture and hyperparameters to achieve optimal performance.

During the training process, the MLP will learn to generalize from the provided examples, capturing the underlying patterns that distinguish malicious URLs from safe ones. We will utilize techniques such as cross-validation and regularization to prevent overfitting and ensure the model's robustness.

Once the MLP model is trained, we will evaluate its performance using various metrics such as accuracy, precision, recall, and F1 score. Additionally, we will conduct extensive testing to assess the system's ability to detect different types of malicious URLs, including those with advanced obfuscation techniques.The successful implementation of this project will provide a powerful and efficient malicious URL detection system based on MLP. By accurately identifying and blocking malicious URLs, we can mitigate the risks associated with cyber attacks, protecting individuals, organizations, and networks from potential harm.

1.3 Overview of Chapters:

Chapter 2: This chapter gives a detailed view of existing research works, systems, their methodology, algorithms, source of dataset, and result.

Chapter 3: The goals and objectives are defined in this chapter and alogrithms used also described in this chapter.

Chapter 4: Overview of the system configuration needed for execution of Malicious url detection System as a web application, software requirements, frameworks, and libraries used for application development are explained.

Chapter 5: In this chapter how dataset is collected and how it is preprocessed and also extracting the necessary features are extracted.

Chapter 6: In this chapter the evaluation metrics used to evaluate the performance of the deep learning model has been explained.

Chapter 7: This chapter gives a detailed explanation of the design and architecture of malicious url detection system. Also use case,DFD, also been explained in this chapter.

Chapter 8: In this chapter the sample outputs, advantages, disadvantages and future work is explained.

Chapter 9: The resources, and research articles referred for reference are listed.

CHAPTER 2
LITERATURE SURVEY

2.1 Introduction:

A literature survey is a critical component of any research project, as it involves reviewing and analysing the existing literature in the field. It helps to identify the current state-of-the-art, knowledge gaps, and research challenges. The purpose of a literature survey is to provide a comprehensive overview of the existing research, which serves as a foundation for the research project. In the context of malicious URL detection, a literature survey would involve reviewing the existing research on various techniques and approaches for detecting malicious URLs. This would include a review of the different types of features used for extraction, machine learning algorithms, and evaluation metrics used to assess the performance of the detection system. The literature survey would also involve a review of the current state-of-the-art in the field, including recent advances, emerging trends, and challenges. This information can help to identify research gaps and opportunities for future research. Overall, a literature survey is a crucial step in any research project, as it helps to provide a solid foundation for the research and ensure that the research questions are relevant, innovative, and feasible.

2.2 Related Work:

The paper [01] provides a comprehensive review of the current state-of-the-art machine learning techniques used for malicious URL detection. The authors present a detailed survey of different types of machine learning algorithms, including decision trees, support vector machines, artificial neural networks, and others, and analyze their performance in detecting malicious URLs. The paper also highlights the importance of feature selection and extraction techniques in improving the effectiveness of machine learning models for malicious URL

detection. Additionally, the paper discusses the limitations of current techniques and identifies potential directions for future research in the field. Overall, the paper provides valuable insights into the latest developments in machine learning-based malicious URL detection and is a useful resource for researchers and practitioners working in this area.

The paper [02] provides a comprehensive review of the current state-of-the-art machine learning techniques used for malicious URL detection. The authors present a detailed survey of different types of machine learning algorithms, including decision trees, support vector machines, artificial neural networks, and others, and analyze their performance in detecting malicious URLs. The paper also highlights the importance of feature selection and extraction techniques in improving the effectiveness of machine learning models for malicious URL detection. Additionally, the paper discusses the limitations of current techniques and identifies potential directions for future research in the field. Overall, the paper provides valuable insights into the latest developments in machine learning-based malicious URL detection and is a useful resource for researchers and practitioners working in this area.

The authors of [08] discuss various techniques used for malicious URL detection. They classify these techniques into four categories: signature-based, heuristic-based, machine learning-based, and hybrid techniques. The review highlights the strengths and weaknesses of each technique and provides a comparison between them. The authors emphasize that machine learning-based techniques have shown promising results in detecting malicious URLs. They also discuss various datasets used for evaluating these techniques and suggest the need for standardized datasets to compare the effectiveness of different methods. The review concludes by highlighting the importance of combining different techniques to improve the accuracy of malicious URL detection systems.

In this paper [16] the authors provide an overview of various malicious URL detection techniques and categorize them into three types: static, dynamic, and hybrid. The static techniques analyze the content of the URL without actually accessing it, while dynamic techniques use a sandbox environment to execute the URL and analyze its behavior. Hybrid techniques combine both static and dynamic techniques to improve the detection rate and reduce false positives. The review discusses the strengths and weaknesses of each technique and provides a comparison between them. The authors emphasize the importance of using machine learning-based techniques for malicious URL detection and highlight various machine learning algorithms used in the literature. They also discuss various evaluation metrics used to compare the effectiveness of different techniques and datasets used for evaluation. The review concludes by suggesting future research directions to improve the accuracy and efficiency of malicious URL detection techniques.

In this survey [06] the authors provide an overview of various machine learning-based malicious URL detection techniques. They categorize these techniques into three types: feature-based, deep learning-based, and hybrid. The survey discusses the strengths and weaknesses of each technique and provides a comparison between them. The authors emphasize that machine learning-based techniques have shown promising results in detecting malicious URLs. They also discuss various datasets used for evaluating these techniques and suggest the need for standardized datasets to compare the effectiveness of different methods. The survey concludes by highlighting the importance of selecting appropriate features for machine learning-based techniques and the need for further research in this area.

The study of [14] provide an overview of various malware detection techniques and their applications. They discuss the evolution of malware, the various types of malwares, and the stages of the malware attack cycle. The survey

then focuses on various malware detection techniques, such as signature-based, behaviour-based, anomaly-based, and machine learning-based techniques. The authors compare these techniques in terms of their strengths and weaknesses, and discuss the challenges associated with each technique. They also provide a detailed analysis of various malware detection tools, including their features, limitations, and effectiveness. The survey concludes by highlighting the need for a multi-layered approach to malware detection and the importance of continuous research in this area to keep up with the evolving threat landscape

the authors of [24] discuss various URL-based malware detection techniques and their effectiveness in detecting malicious URLs. They provide an overview of the different types of malware that can be delivered through URLs, such as phishing attacks, malware downloads, and drive-by downloads. The survey then focuses on various techniques used for URL-based malware detection, including signature-based, behavior-based, heuristic-based, and machine learning-based techniques. The authors evaluate these techniques in terms of their detection rates, false positive rates, and overall effectiveness. They also discuss the challenges associated with URL-based malware detection, such as obfuscation techniques used by attackers to evade detection. The survey concludes by highlighting the need for a multi-layered approach to URL-based malware detection, and the importance of continuous research to stay ahead of evolving threats.

The paper of [19] provides a comprehensive survey on the different approaches and techniques used for detecting malicious URLs. The authors provide an overview of the different types of threats associated with malicious URLs and discuss the importance of detecting these threats. The paper covers various machine learning techniques, data mining, and feature extraction methods used for detecting malicious URLs. The authors also present a comparative analysis of the different techniques used for malicious URL detection and evaluate the strengths and weaknesses of each approach. Additionally, the paper

provides insights into the future of malicious URL detection and highlights potential research directions. Overall, this paper provides a valuable resource for researchers and practitioners interested in the field of malicious URL detection.

The author of [27] provides a comprehensive review of various techniques and approaches used for detecting malicious URLs. It covers traditional rule-based methods, machine learning-based methods, and hybrid techniques that combine both approaches. The paper also discusses different types of features that can be extracted from URLs, such as lexical, host-based, and content-based features. Furthermore, it compares the performance of various methods on different datasets and discusses their strengths and limitations. Overall, the paper provides a useful resource for researchers and practitioners interested in the area of malicious URL detection.

The paper [15] provides a comprehensive survey of recent research in malicious URL detection techniques using machine learning. The authors categorize the techniques into two groups: feature-based and deep learning-based. The feature-based techniques extract features from URLs and use machine learning algorithms to classify them as malicious or benign. The deep learning-based techniques use neural networks to learn representations of URLs and classify them. The paper also discusses the datasets used in previous studies and compares the performance of various techniques. The authors conclude that deep learning-based techniques have shown promising results in recent studies and are likely to be the future direction of research in this area.

The paper of [03] provides a literature review of various malware detection and analysis techniques. The authors discuss the key challenges in malware detection and analysis, including obfuscation, polymorphism, and anti-analysis techniques. They review traditional signature-based detection, behavior-based detection, and heuristic-based detection techniques. The authors also discuss the role of machine learning in malware detection and analysis and describe various

machine learning algorithms used in the domain. Additionally, the paper discusses the limitations of existing techniques and highlights the future research directions for malware detection and analysis.

The work of [32] presents a survey of various malware detection techniques used for web applications. The article discusses various types of web-based malware and their attack vectors. It then goes on to describe various malware detection techniques such as signature-based detection, anomaly-based detection, and behavior-based detection. The authors also discuss the limitations of existing techniques and propose some future research directions. Overall, the article provides a good overview of malware detection techniques for web applications

The work of [05] provides a comprehensive overview of the state-of-the-art techniques for detecting malicious websites. The paper starts by discussing the characteristics of malicious websites and the various types of attacks that they can launch. Then, it presents the various approaches to detecting malicious websites, such as signature-based, heuristic-based, and machine learning-based methods. The paper also covers the limitations of these techniques, such as false positives and negatives, and the challenges that researchers face in developing effective detection methods. Finally, the paper concludes by highlighting the future directions in this field, such as the need for more real-time detection methods and the use of hybrid approaches that combine multiple techniques.

The study of [12] presents a comprehensive review of malware detection techniques in mobile platforms. The paper discusses the various challenges and limitations in detecting malware in mobile devices. It further reviews the existing techniques and approaches for malware detection in mobile platforms, including signature-based, behavior-based, and anomaly-based detection techniques. The paper also discusses the various features that are used for malware detection, such as permission-based features, API call features, network traffic features, and

system call features. Finally, the paper concludes with some future research directions in mobile malware detection.

The paper of [25] discusses the various malware detection techniques for smartphones. The paper provides an overview of the types of mobile malware, the sources of malware, and the techniques used to detect malware. The paper also provides a detailed analysis of the various detection techniques, including signature-based, anomaly-based, behavior-based, and hybrid techniques. The authors conclude that a combination of techniques is needed to provide effective malware detection on mobile devices and that machine learning techniques show great promise for improving the detection accuracy of these techniques.

SI. NO	Journal Name & Year	Authors	Concepts/ Techniques	Advantages	Disadvantages/ Issues
1.	A Survey on Machine Learning Techniques for Malicious URL Detection	Mohammed Alshehri	Decision Tree, Random Forest, Logistic Regression and Naive Bayes	Provides a detailed review of various machine learning algorithms used for malicious URL detection..	The survey paper does not evaluate the effectiveness of the algorithms in real-world scenarios.
2.	A Literature Review on Malware Detection Techniques(2021)	Akshay Deepak	The survey paper does not propose any new algorithm	Provides a detailed review of various malware detection techniques, including their strengths and weaknesses.	It does not compare the techniques to other existing methods, such as intrusion detection
3.	Detecting malicious web links and their Attack types(2011).	Hyunsang Choi, Heejo Leo, Bin Zhu.	Rule based Anti phishing.	It detects both Malicious URL and Types of Attack.	It uses non machine learning approach.
4.	Malicious Website Detection & Effective issues(2011).	Birhanu Eshete, Adolfo Villafiorita, Komminist.	Offline ML technique.	It is fast in processing.	It does not adaptable to different parameter.

5.	A Survey on URL-Based Malware Detection Techniques(2021)	S. Kumar and N. Bhalaji	Random Forest, Naive Bayes, k-Nearest Neighbors, Support Vector Machines.	Provides an in-depth review of the state-of-the-art machine learning techniques	Focuses only on URL-based malware detection and does not cover other types of malware.
6.	Machine Learning based malicious web detection(2020).	Niranjan Swarap, madhan Kumar.	Logistic Regression. Random Forest.	Data Training is Easier.	It an predict only disease function.
7.	Intelligent malicious URL Determination with feature Analysis.	Yu-Chun chan,Yi-Wei-ma,Jeanes-Lian chen.	XGBoost.	It provides type based on domains based database.	Different to understand & learn.
8.	Detection and analysis of drive by downloads & Malicious website.	Montha Aldwain.	Naive Bayes. JRIP. J48.	It applied via Blocklist content &System based evaluation.	The work include to small dataset and number of Classifier& Actual real time texting.
9.	A convolutional based system for Malicious URL detection(2020).	Sten Su,Zihong Jian.	CNN.	It uses Auto encode to represent the URL .	Thee representation and Discrimination both needes to be Updated.
10.	Detecting Malicious URL using ML techniques	Kerken Salan,Rami Muusthafa.	LR & DT.	It explain common URL feature type & possible types of Attack.	Requires Huge Dataset.

2.3 Result and Conclusion:

In the above, various research works were discussed. Among them, the model of [12], model of [05], model of [15], and model of [26] were performed well and described in more technical detail. Dataset preparation methodology is learned from [30].

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction :

- The proposed model is to develop a web based malicious url detection system which classifies the urls based on the features using deep learning algorithm.
- A huge dataset is used for training the model
- The url is given as input to detect.
- The users can give input url through web page.

3.2 Goal:

The main goal of the proposed model is to develop a web based malicious url detection system which detects the malicious urls to prevent users from visiting malicious sites.

3.3 Objectives:

- To collect datasets which consists of benign , malware, defacement, phishing urls.
- To Cleanse and optimize the dataset for training model.
- To extract the url features based on lexical features.
- To develop the url trained model using deep learning.
- To deploy the model as a web application.

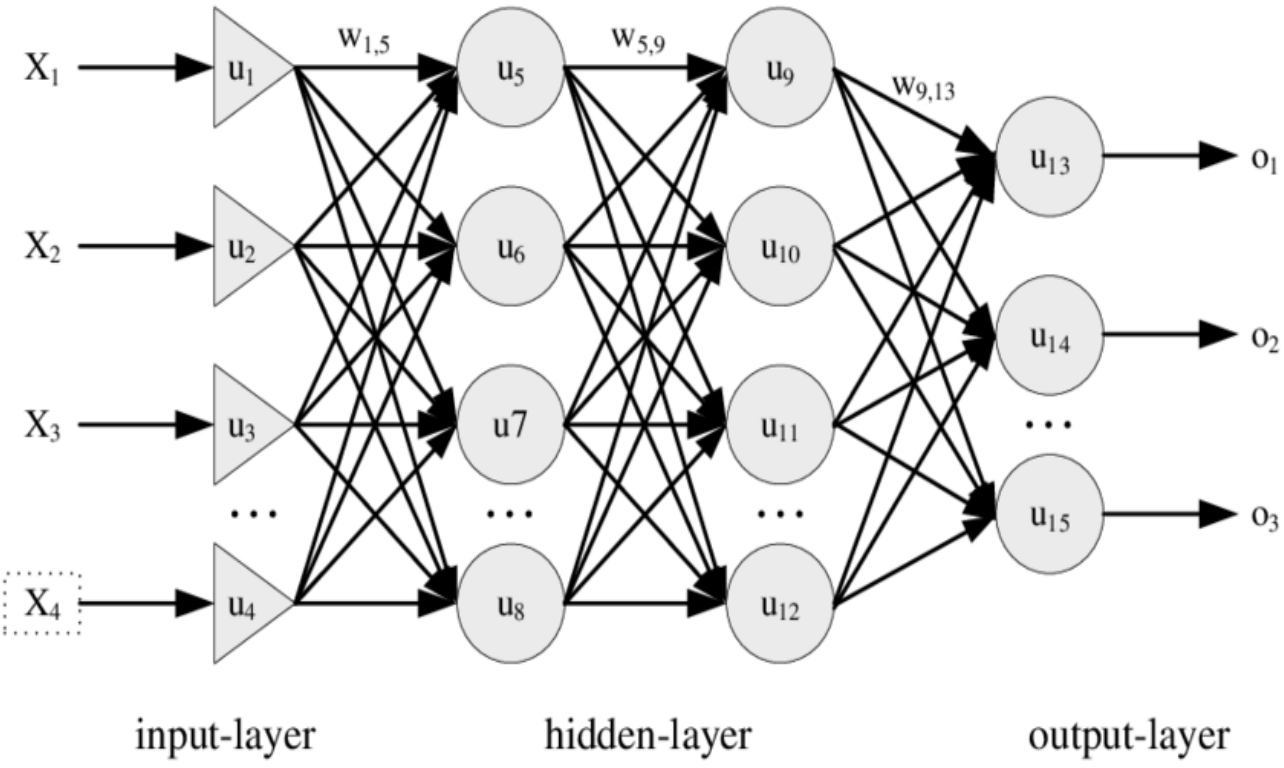
3.4 Algorithm:

Multi Layer Perceptron(MLP)

A multi-layer perceptron (MLP) is a type of artificial neural network (ANN) that consists of multiple layers of interconnected nodes, called neurons. It is a feedforward neural network, meaning that information flows through the network in one direction, from the input layer to the output layer.

Here's a step-by-step explanation of how an MLP works:

Architecture: An MLP typically consists of three types of layers: an input layer, one or more hidden layers, and an output layer. Each layer is composed of multiple neurons, and each neuron is connected to neurons in the adjacent layers. The input layer receives the input data, the hidden layers process the information, and the output layer produces the final predictions or outputs.



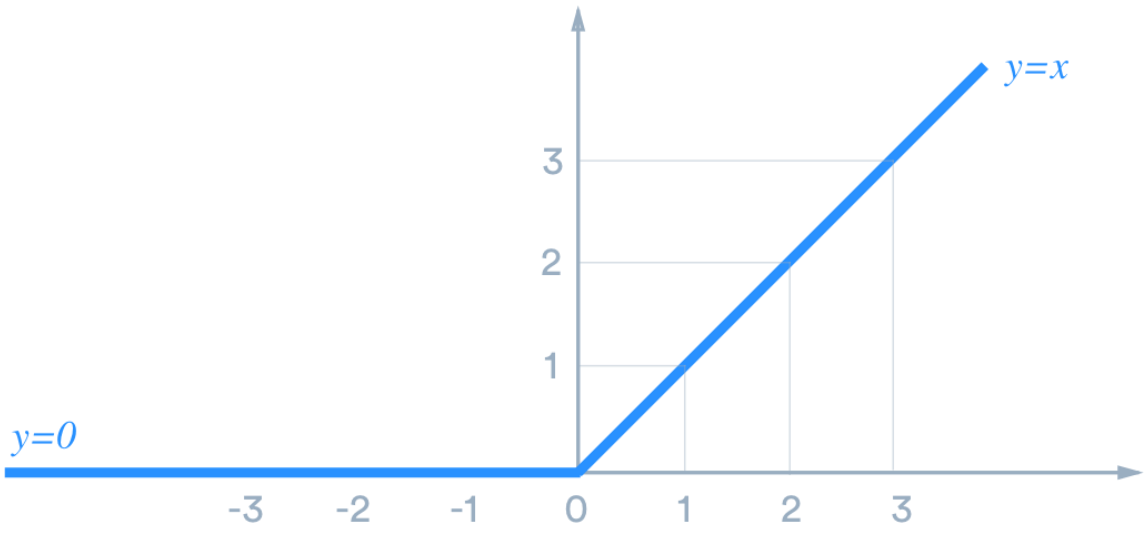
Neurons and Activation Functions: Each neuron in the MLP performs a weighted sum of the inputs it receives, applies an activation function to the sum, and produces an output. The activation function introduces non-linearity into the network, enabling it to model complex relationships in the data. Common activation functions include the sigmoid, tanh, and ReLU functions.

ReLU:

The Rectified Linear Unit (ReLU) activation function is a widely used activation function in neural networks, including multi-layer perceptrons (MLPs). It is a simple yet effective non-linear function that introduces non-linearity to the network and helps it learn complex patterns in the data.

Mathematically it is represented as:

$$f(x)=\max(0,x)$$



Feedforward Propagation: During the feedforward process, the input data is passed through the network layer by layer. The output of each neuron in a layer serves as the input to the neurons in the next layer. The weighted sum of inputs is calculated at each neuron, and the activation function is applied to produce the

output. This process continues until the output layer is reached, and the final predictions are obtained.

Training: The training of an MLP involves adjusting the weights of the connections between neurons to minimize the difference between the predicted outputs and the true outputs. This is done using a technique called backpropagation, which uses the gradient descent optimization algorithm. During backpropagation, the error between the predicted outputs and the true outputs is propagated backward through the network, and the weights are updated in the direction that minimizes the error.

Backpropagation: In the backpropagation phase, the error is first calculated at the output layer and then propagated backward through the hidden layers. The gradient of the error with respect to the weights is computed, and the weights are updated accordingly using the gradient descent algorithm. This process iteratively repeats for a number of epochs or until a convergence criterion is met.

Model Evaluation: Once the MLP is trained, it can be used to make predictions on new, unseen data. The input data is fed through the network, and the output layer produces the predicted outputs. The performance of the model can be evaluated using various evaluation metrics, such as accuracy, precision, recall, or mean squared error, depending on the type of problem.

MLPs are capable of learning complex non-linear relationships in the data and are widely used for tasks such as classification, regression, and pattern recognition. However, they can be prone to overfitting if the model complexity or the number of hidden layers and neurons is too high. Regularization techniques, such as L1 and L2 regularization, are often used to prevent overfitting in MLPs.

3.5 Conclusion:

In this chapter, objectives and goals of Malicious URL detection system, deep learning algorithm and evaluation metrics which are used for measure the perfomance of models are learned.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Introduction

For the development of the system, knowledge about hardware and software configurations must be known, Following of this chapter, hardware requirements such as CPU, RAM, GPU, Storage and Software requirements such as OS, Programming language, IDE, Frameworks such as machine learning and deep learning frameworks, web hosting framework and libraries used for system development will discuss.

4.2Hardware requirements:

- CPU: Laptop or PC with Intel i5 5th Gen or Higher versions is needed.
- RAM: Minimum of 8 GB of RAM is required.
- GPU: Intel UHD Graphics or Higher versions is recommended.
- Storage: 50 GB of storage is required. SSD is more preferable than HDD.

4.3 Software requirements:

- OS- Windows 7 or higher versions but windows 10 is recommended/Minimum Ubuntu 16.04 is required.
- Python (version:3.7.9)-Programming Language is used for machine learning and deep learning.
- IDE-Jupyter Notebook is used as development environment.
- Spyder- Both development and deployment environment.

4.4 Frameworks:

- Keras or Tensorflow- Framework developed by Google for machine learning and deep learning.
- Flask-Python framework used for deploy web applications.

4.5 Libraries:

- Sci-kit: Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It is built on top of other scientific computing packages like NumPy, SciPy, and matplotlib. Scikit-learn provides a wide range of machine learning algorithms and tools for tasks such as classification, regression, clustering, dimensionality reduction, model selection, and pre-processing of data.
- Urllib: This library provides a set of functions and classes for working with urls.
- Tld: This library helps to extract the top-level domain from a given URL.
- Matplotlib: This is a popular plotting library in python that provides a wide range of functionalities to create high-quality visualizations. It is widely used for data visualization.
- Seaborn: It also used for high level interfacing for creating attractive and informative statistical graphics.

4.5 Conclusion

The knowledge about system requirements, software, libraries, frameworks are understood.

CHAPTER 5

FEATURE EXTRACTION

5.1 Introduction

Feature extraction is a critical step in malicious URL detection, as it involves identifying and extracting the most relevant features from the URLs that can be used to distinguish between benign and malicious URLs. In general, these features can be categorized into several types, including domain-based features, path-based features, and lexical features. Domain-based features include characteristics of the domain name such as the length of the domain name, the number of subdomains, and the presence of special characters or numbers. Path-based features capture information about the URL path, such as the length of the path, the number of directories in the path, and the presence of keywords or unusual characters in the path. Lexical features, on the other hand, are based on the URL's text content and include information such as the presence of specific keywords, the use of obfuscation techniques, and the similarity of the URL to known malicious URLs. To perform feature extraction, various techniques such as regular expressions, natural language processing, and pattern matching algorithms can be used. These techniques help to identify and extract the most relevant features from the URLs that can be used to train a machine learning model for malicious URL detection. The feature extraction process is an essential component of the overall malicious URL detection system, and the choice of features and extraction techniques can have a significant impact on the system's accuracy and performance. Data preparation consists of two phases,

i. Data gathering

ii. Feature extraction

5.2 Data gathering:

Dataset is collected from Kaggle website.

Link: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

The url dataset consists of 6,51,191 URLs out of which 4,28,103 benign URLs, 96,457 defacement URLs, 94111 phishing urls, and 32,520 malware URLs.

Out[2]:

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement

5.3 Feature extraction:

In this step, we will extract the following **lexical features** from raw URLs, as these features will be used as the input features for training the machine learning model.

- **having_ip_address:** Generally cyber attackers use an IP address in place of the domain name to hide the identity of the website. this feature will check whether the URL has IP address or not.

- **abnormal_url:** This feature can be extracted from the WHOIS database. For a legitimate website, identity is typically part of its URL.

- **Count. :** The phishing or malware websites generally use more than two sub-domains in the URL. Each domain is separated by dot (.). If any URL contains more than three dots(.), then it increases the probability of a malicious site.

- **Count-www:** Generally most of the safe websites have one www in its URL. This feature helps in detecting malicious websites if the URL has no or more than one www in its URL.

- **count@:** The presence of the "@" symbol in the URL ignores everything previous to it.

- **Count_dir:** The presence of multiple directories in the URL generally indicates suspicious websites.

- **Count_embed_domain:** The number of the embedded domains can be helpful in detecting malicious URLs. It can be done by checking the occurrence of "//" in the URL.

- **Suspicious words in URL:** Malicious URLs generally contain suspicious words in the URL such as PayPal, login, sign in, bank, account, update, bonus, service, ebayisapi, token, etc. We have found the presence of such frequently occurring suspicious words in the URL as a binary variable i.e., whether such words present in the URL or not.

- **Short_url:** This feature is created to identify whether the URL uses URL shortening services like bit. \ly, goo.gl, go2l.ink, etc.

- **Count_https:** Generally malicious URLs do not use HTTPS protocols as it generally requires user credentials and ensures that the website is safe for transactions. So, the presence or absence of HTTPS protocol in the URL is an important feature.

- **Count_http:** Most of the time, phishing or malicious websites have more than one HTTP in their URL whereas safe sites have only one HTTP.

- **Count%:** As we know URLs cannot contain spaces. URL encoding normally replaces spaces with symbol (**%**). Safe sites generally contain less number of spaces whereas malicious websites generally contain more spaces in their URL hence more number of **%**.

- **Count?:** The presence of symbol (***?***) in URL denotes a query string that contains the data to be passed to the server. More number of ? in URL definitely indicates suspicious URL.

- **Count-:** Phishers or cybercriminals generally add dashes(-) in prefix or suffix of the brand name so that it looks genuine URL. For example. www.flipkart-india.com.

- **Count=:** Presence of equal to (=) in URL indicates passing of variable values from one form page t another. It is considered as riskier in URL as anyone can change the values to modify the page.

- **url_length:** Attackers generally use long URLs to hide the domain name. We found the average length of a safe URL is 74.

- **hostname_length:** The length of the hostname is also an important feature for detecting malicious URLs.

- **First directory length:** This feature helps in determining the length of the first directory in the URL. So looking for the first '/' and counting the length of the URL up to this point helps in finding the first directory length of the URL. For accessing directory level information we need to install python library TLD. You can check this link for installing TLD.

- **Length of top-level domains:** A top-level domain (TLD) is one of the domains at the highest level in the hierarchical Domain Name System of the Internet. For example, in the domain name www.example.com, the top-level domain is com. So, the length of TLD is also important in identifying malicious URLs. As most of the URLs have .com extension. TLDs in the range from 2 to 3 generally indicate safe URLs.

- **Count_digits:** The presence of digits in URL generally indicate suspicious URLs. Safe URLs generally do not have digits so counting the number of digits in URL is an important feature for detecting malicious URLs.

- **Count_letters:** The number of letters in the URL also plays a significant role in identifying malicious URLs. As attackers try to increase the length of the URL to hide the domain name and this is generally done by increasing the number of letters and digits in the URL.

5.4 Conclusion:

Feature extraction is a critical step in malicious URL detection, as it involves identifying and extracting the most relevant features from URLs that can be used to distinguish between benign and malicious URLs. The process of feature extraction involves various techniques such as regular expressions, natural language processing, and pattern matching algorithms, which help to identify and extract patterns and relationships within the URL. The choice of features and extraction techniques is crucial to the accuracy and performance of the malicious URL detection system. Therefore, researchers must continuously explore and experiment with different feature extraction techniques to identify the most effective methods for identifying and extracting relevant features from URLs. The success of the feature extraction process determines the effectiveness of the entire malicious URL detection system, which is crucial in preventing online threats such as phishing, malware, and scams.

CHAPTER 6

EVALUATION METRICS

6.1 Introduction:

The detection of malicious URLs is a crucial aspect of cybersecurity, as it helps protect users from various types of online threats, including phishing, malware, and scams. Machine learning techniques have become increasingly popular for malicious URL detection due to their ability to learn complex patterns from large datasets. However, evaluating the effectiveness of a machine learning-based malicious URL detection system is not a trivial task, and requires appropriate evaluation metrics.

6.2 Accuracy:

Machine Learning model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input or training data.

6.3 Classification Report:

The classification report visualizer displays the precise, recall, F1 and support scores for the model. In order to support easier implementation and problem detection, the report integrates numerical scores with a color-coded heatmap. All heatmaps are in the range(0,0,1,0) to facilitate easy comparison of classification models across different classification reports. The classification report always a representation of the main classification metrics on a per-class basis. This gives a deeper intuition of the classifier behaviour over global accuracy which can mask functional weakness in one class of a multiclass problem.

i. Precision

- Precision is the ability of a classifier not to label an instance positive that is negative. For each class, it is defined as the ratio of true positives to the sum of a True Positive and False Positive.
- Precision = TP/ (TP + FP)

ii. Recall

- Recall is the ability of a classifier to find all positive instances. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives.
- Recall = TP/(TP+FN)

iii. F-measure

- The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

	precision	recall	f1-score	support
benign	0.97	0.98	0.97	85621
defacement	0.93	0.98	0.95	19292
phishing	0.93	0.85	0.89	6504
malware	0.90	0.80	0.84	18822
accuracy			0.95	130239
macro avg	0.93	0.90	0.92	130239
weighted avg	0.95	0.95	0.95	130239

6.4 Confusion matrix:

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

True Positives (TP): This represents the number of instances that were correctly predicted as positive by the model. In other words, the model predicted the class correctly when the true class was positive.

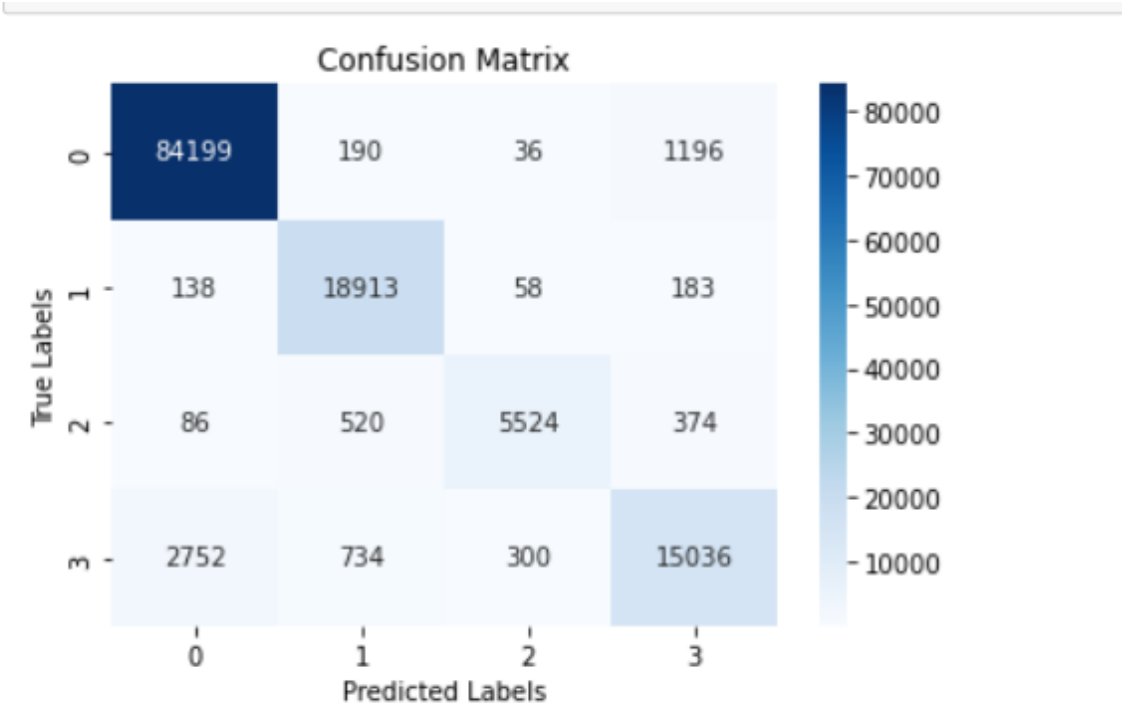
True Negatives (TN): This represents the number of instances that were correctly predicted as negative by the model. It indicates that the model predicted the class correctly when the true class was negative.

False Positives (FP): Also known as Type I error, this represents the number of instances that were incorrectly predicted as positive by the model. It occurs when the model predicts a positive class when the true class is negative.

False Negatives (FN): Also known as Type II error, this represents the number of instances that were incorrectly predicted as negative by the model. It occurs when the model predicts a negative class when the true class is positive.

Actual values

Predicted values	Positive (1)	Negative(0)
Positive (1)	TP	FP
Negative (0)	FN	TN



6.5 Conclusion:

From the above evaluation results it is found that MLP algorithm performs well and has an accuracy score of around 95%.It can detect the malicious urls more accurately.

CHAPTER 7
SYSTEM DESIGN

7.1 Introduction

The detailed explanation of each module such as General Analysis, Learning Required Technology, Dataset Collection and Semantic Analysis, Dataset Pre-processing, Model Development and Implementation, WebApp implementation, Testing Overall system and timeline for each module between 04.02.2023 and 21.04.2023, architectures such as system architecture, technical architecture and other technical details of the system will discussed in following topics.

7.2. Module Split

The complete system development process is split into 7 modules such as

- General Analysis
- Learning Required Technology
- Dataset Collection and Semantic Analysis
- Dataset Pre-processing
- Model Development and Implementation
- WebApp implementation
- Testing Overall system

7.2.1 Modules:

- General Analysis

  The General Analysis module consists of a Literature Survey (study of existing systems, research works, relevant and similar systems with their approaches and used algorithms, gaining experience and knowledge from them), Requirement Analysis (defining the

environments and identifying system needs, finding the required tools, technology, hardware and software with their availability, capabilities, consistency, features, supported platforms and languages then make sure the right fit of them for further development process)

- Learning Required Technology

  After finding the right tools and technology, studying and learning the unknown tools and technologies in-depth which are needed.

- Dataset Collection and Semantic Analysis

  This module consists of planning for collecting data like what data to be collected, how to collect, where to collect, and analysing better choices and fitting them. As per the plan, data must collect and make semantic analysis to categories and get needed data from the pool of data.

- Dataset Pre-processing

  In this module, collected data undergoes various cleansing processes and techniques, then transform into the right format and required features are extracted using existing and own defined algorithms.

- Model Development and Implementation

  The development consists of building a model using different algorithms with creative approaches, training the developed model using train dataset then test and evaluate the models to find their performance using evaluation techniques and metrics. The model which performs best is implemented in the application.

- WebApp implementation

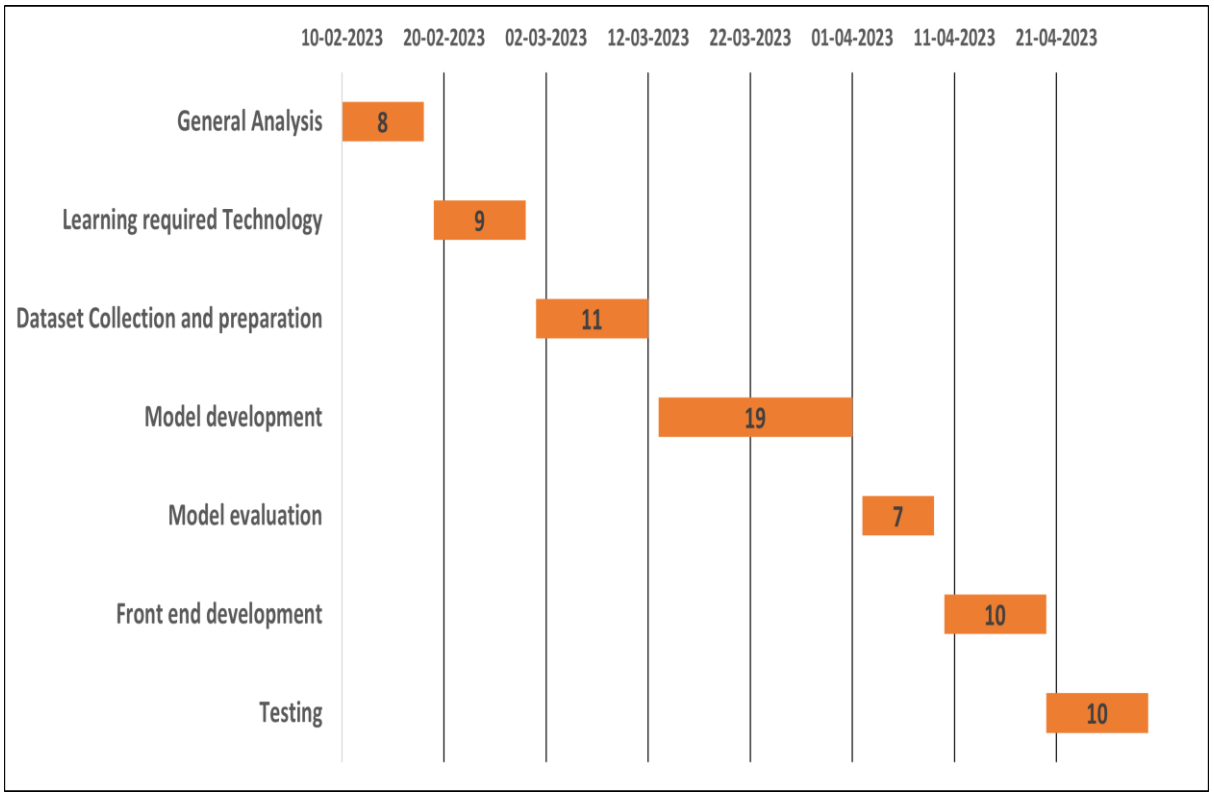
  Creating interactive GUI and implementing the best model in the back-end of the application then deploying it as a web app using Flask framework are came under this module

- Testing Overall system

  Testing each and every component, unit, their function and debugging issues if anything raises and evaluating overall system performance using metrics are done in the testing module.

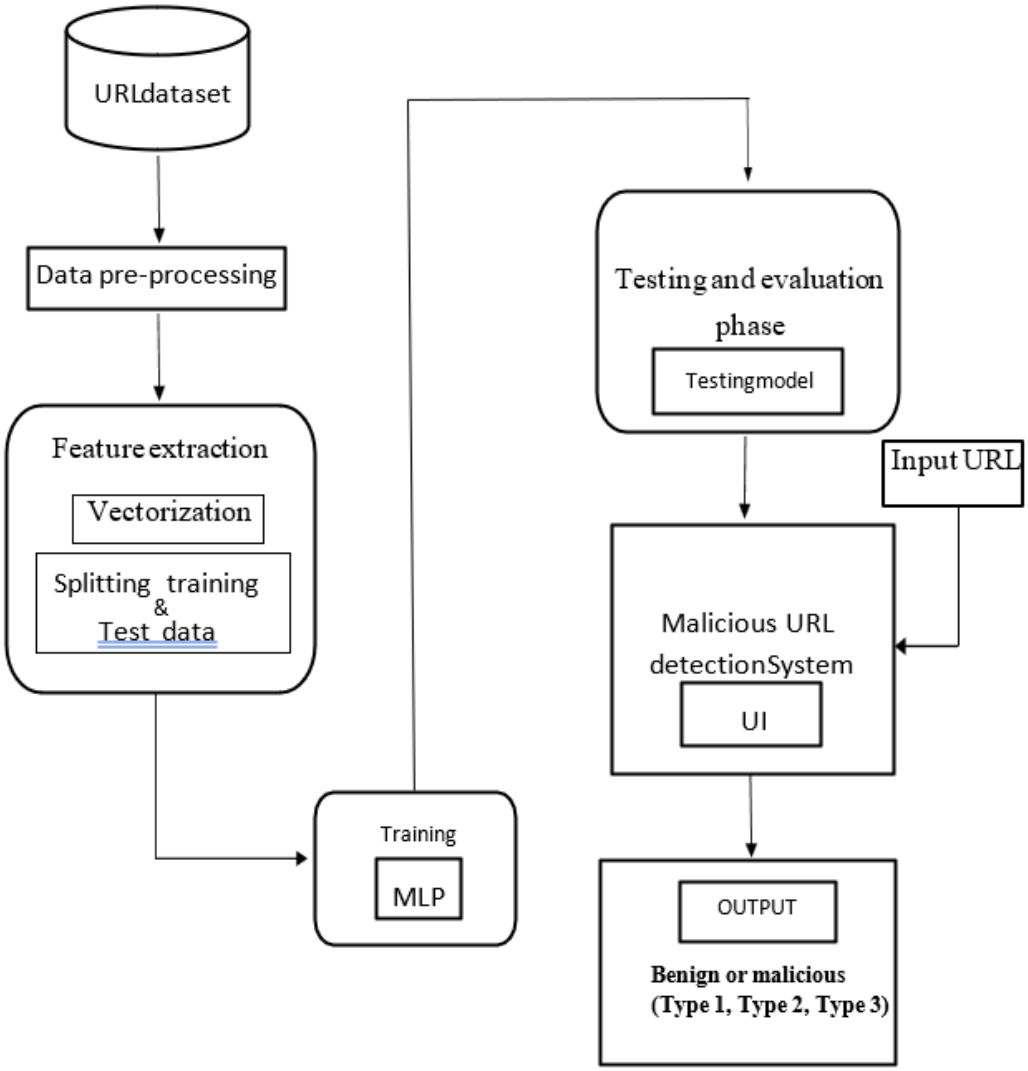
7.2.2 Gantt Chart:

A Gantt chart is a visual tool used in project management to illustrate the scheduling and progress of tasks or activities over time. It provides a graphical representation of the project timeline, task durations, task dependencies, and the overall project schedule. Gantt charts help project managers and team members understand the project's progress, identify critical tasks, and manage resources effectively.



7.3 Architecture Diagram:

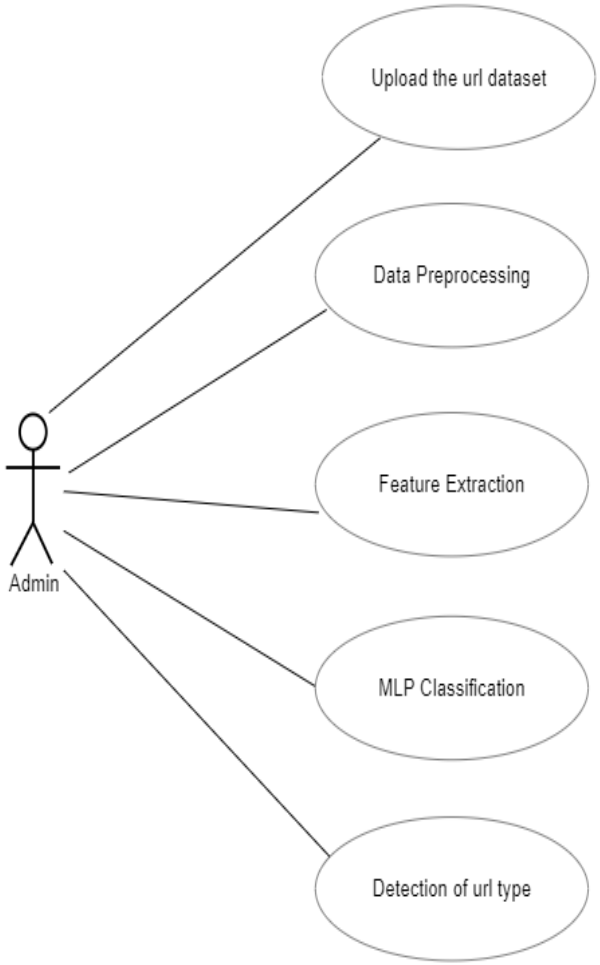
A system architecture diagram is the diagrammatic representation of an overview of the project. It depicts the methodologies used in the layer and the relationship between layers in the system



7.4 UML Diagram:

7.4.1 Use case Diagram:

A use case diagram represents the behavior of the system. It expresses how the user can interact with the system, the services provided by the system, and relationship between them. The components are actors represented by stick figures and use cases represented by ellipse, system and lines. Ellipse represents the role of the actors and whole system functions bounded by rectangle bound which portrays complete system functionality.



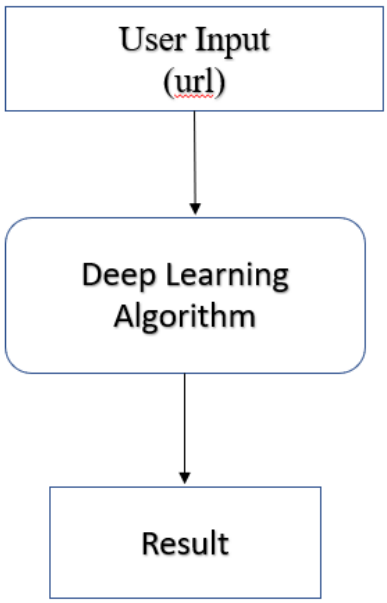
7.4.2 Data Flow Diagram:

Data Flow Diagram (DFD) is the representation of information flows in the system. It shows how data enters, what process takes place and where is data stored in that system. It is also known as data flow graph. It is classified into three different levels based on increasing information and functionality of the system by,

- DFD Level 0
- DFD Level 1
- DFD Level 2

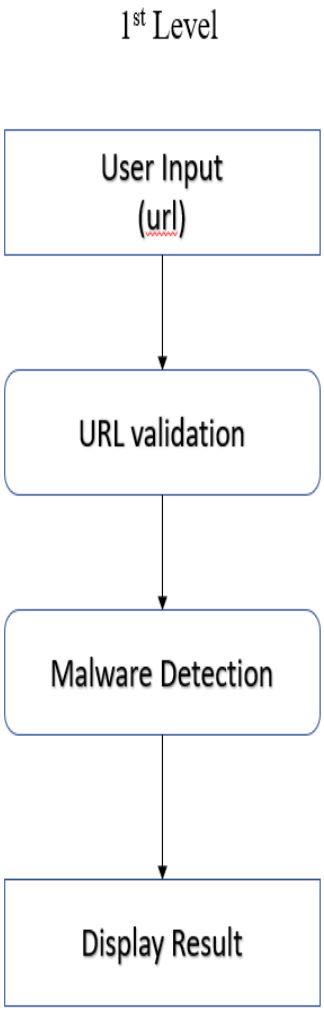
Dataflow Diagram Level 0:

Zeroth level DFD shows the overall data flow of the TAAD model. Data undergoes the subsequence of processes such as audio text conversion, pre-processing, feature extraction, and detecting the output by the model with help of a knowledge base.



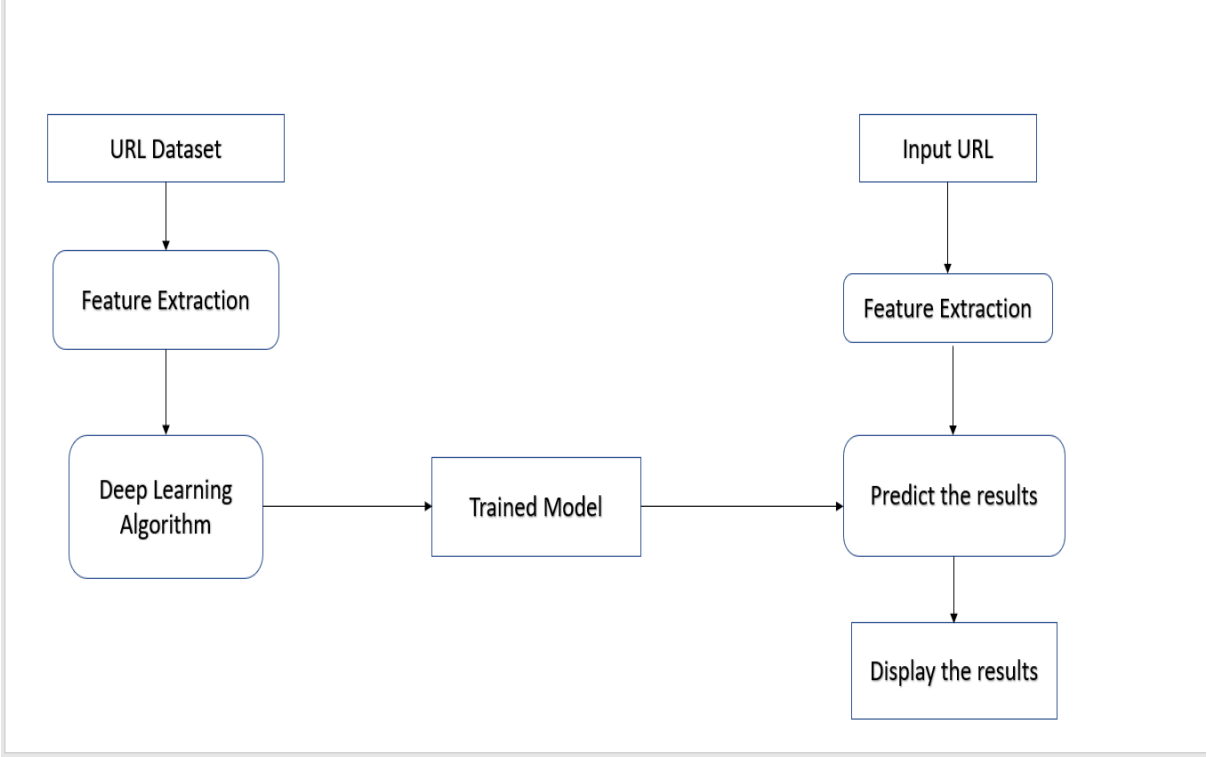
Dataflow Diagram Level 1:

Level one DFD shows the detailed view of dataflow in TAAD model with their techniques. Till feature extraction, all the details are same as level zero DFD but followed process and approaches were given for more depth like removing punctuations, special characters, other language letters etc… Word embedding technique and Lexicon model are used to extract features for model.



Dataflow Diagram Level 2:

Level two DFD gives the complete and detailed view of full TAAD-System as a web application by dividing them into client and server sides. In client side, get input from the user and pass to server by request. TAAD server took input from the client-end and gives output by passing through various processes then give output as a state response to client-side which displays output to the user.



7.5 Conclusion

Working of the system, their data flow and overview of development process are seen in this chapter.

CHAPTER 8

CONCLUSION

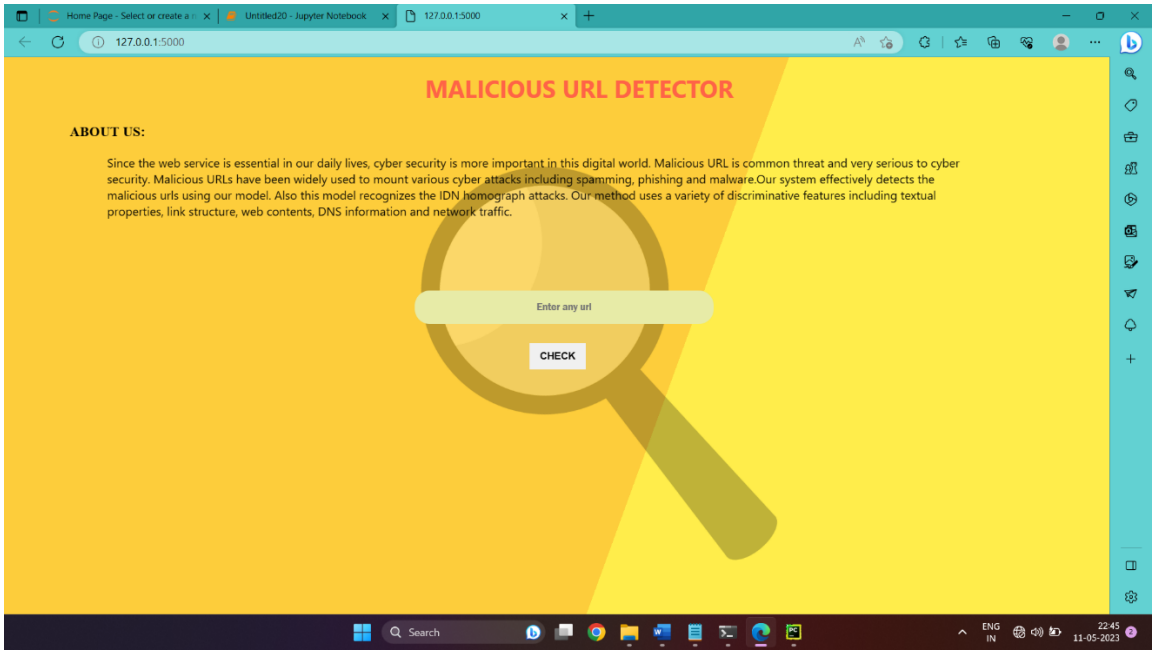
8.1 Introduction

Malicious System model deployed as a web application using Flask web hosting framework, overview of the UI features and conclusion with advantages, disadvantages and future work of the System will be described.

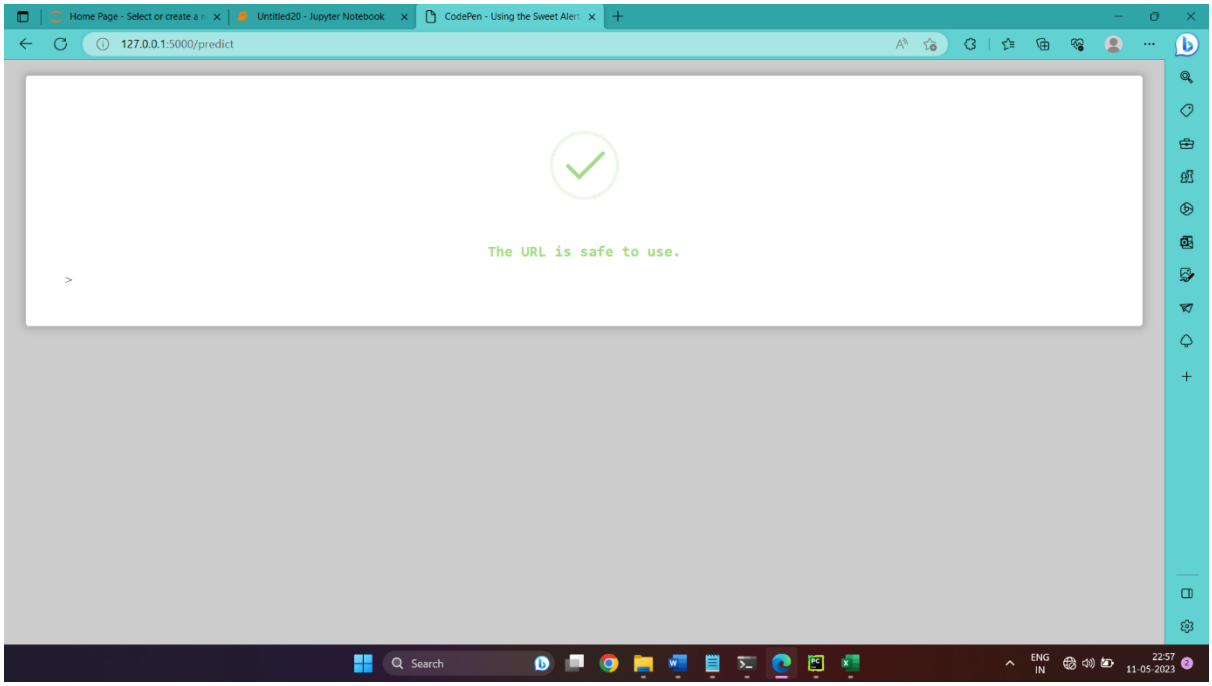
8.2 Web Application:

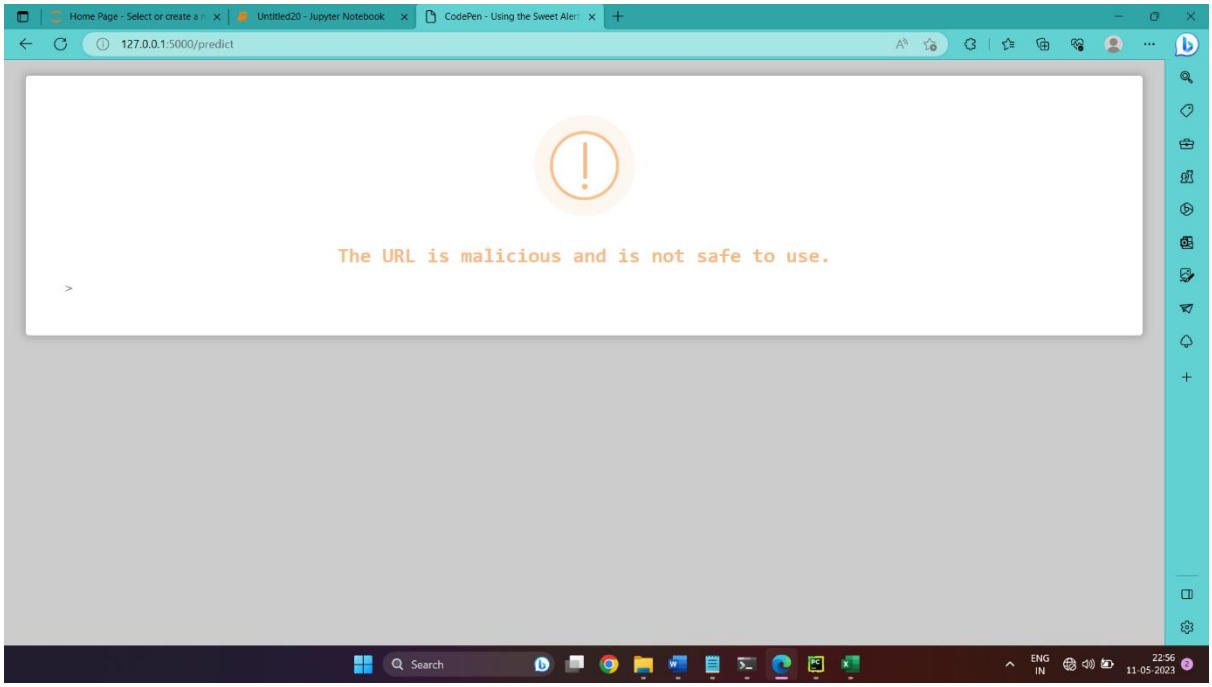
For users to interact with the model, Malicious url detection system was developed as a web application. Flask is used to deploy the Malicious url detection System. It is a popular Python framework used for deploying and hosting web applications using Python language. It uses the jinja2 template for rendering and internal functions.

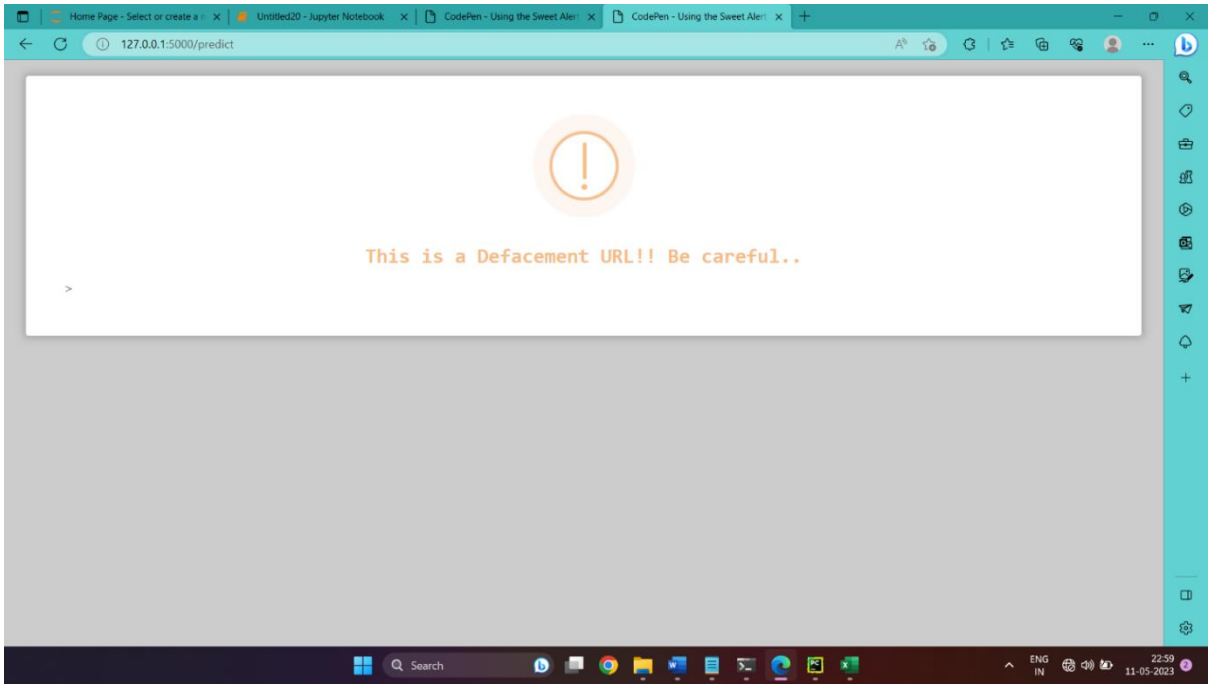
This is the main page of the system. In This page the users can check the url by giving url as input.

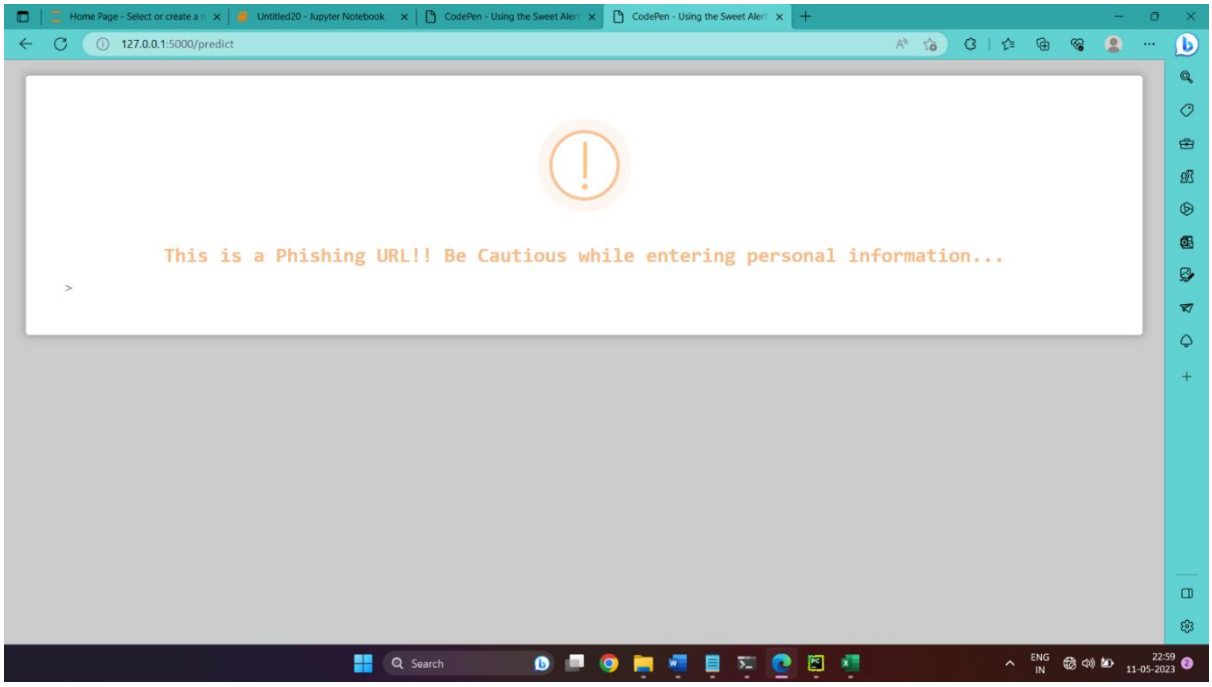


The following figures shows the results obtained by giving input urls:






127.0.0.1:5000/predict
This is a Defacement URL!! Be careful..
>


127.0.0.1:5000/predict
This is a Phishing URL!! Be Cautious while entering personal information...
>

8.3 Conclusion:

From the above techniques and procedures it is obvious that MLP is a good algorithm for detecting the malicious urls with an accuracy of 95%.Still many researches and works needed to be done to develop most efficient systems. Improved datasets can also help in increasing the accuracy.

System Advantages:

- classification model provides precise results in many cases.
- Classifies in depth which type of harmful url it is.
- Detects homograph characters which resembles like normal English Alphabets.

System Disadvantages:

- Need to improve the accuracy.
- More features needed to be extracted and processed so that it can detect accurately.

Future Works:

- Reduce Time complexity.
- To create an browser extension so that it can detect automatically whenever a malicious url is detected.

REFERENCES:

[01] Alshehri, M. et al. (2018) "A Survey on Machine Learning Techniques for Malicious URL Detection", Journal of Cyber Security Technology, Vol.2(2), pp. 77-96.

[02] Deepak, A. et al. (2021) "A Literature Review on Malware Detection Techniques", Journal of Network and Computer Applications, Vol.183, pp. 103036.

[03] A. I. Schein and L. H. Ungar (2007),” Active learning for logistic regression: An evaluation”, vol. 68, no. 3.

[04] Kumar, S. and Bhalaji, N. (2020) "An Overview of Malicious URL Detection Techniques", International Journal of Advanced Science and Technology, Vol.29(9), pp. 1648-1657.

[05] Kumar, S. and Bhalaji, N. (2021) "A Survey on URL-Based Malware Detection Techniques", International Journal of Advanced Science and Technology, Vol.30(2), pp. 3555-3563.

[06] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri (2019), “Machine learning based phishing detection from URLs,” Expert Syst. Appl., vol. 117, pp. 345– 357.

[07] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke (2020), “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” J. Inf. Secur. Appl., vol. 50, p. 102419.

[08] M. Alazab and S. Fellow (2020), “Malicious URL Detection using Deep Learning,” pp. 1–9.

[09] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen (2010), “Malicious web content detection by machine learning,” Expert Systems with Applications, vol. 37:1, pp. 55–60.

[10] Dhamdhere, R. M. et al. (2021) "Malware Detection using Machine Learning Techniques: A Survey", Journal of Computer Science and Applications, Vol.3(1), pp. 25-32.

[11] Ansari, K. M. et al. (2021) "A Survey on Malware Detection and Classification Techniques", Journal of Cyber Security Technology, Vol.5(1), pp. 1-18.

[12] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri(2019), “Machine learning based phishing detection from URLs,” Expert Syst. Appl., vol. 117, pp. 345– 357, 2019.

[13] Y. Wang, R. Agarwal, and B.-Y. Choi (2008) ,”Light weight anti-phihing with user whitelisting in a web browser,” in Region 5 Conference, 20002 IEEE. IEEE, pp.1-4.

[14] Patil, S. S. et al. (2021) "Malware Detection Techniques: A Comprehensive Survey", International Journal of Emerging Trends & Technology in Computer Science, Vol.10(1), pp. 28-35.

[15] Mirajkar, N. S. et al. (2020) "A Review on Machine Learning Techniques for Malware Detection", International Journal of Engineering Research & Technology, Vol.9(11), pp. 906-912.

[16] Aravind, K. B. et al. (2021) "Machine Learning Techniques for Malware Detection: A Survey", International Journal of Advanced Science and Technology, Vol.30(1), pp. 3723-3732.

[17] Al-Akaidi, M. M. et al. (2020) "Malware Detection Techniques: A Review", Journal of Computer Science, Vol.16(7), pp. 952-962.

[18] Al-Dabbagh, A. A. et al. (2020) "A Survey of Machine Learning Techniques for Malware Detection and Classification", International Journal of Emerging Trends in Engineering Research, Vol.8(11), pp. 6529-6535.

[19] Singh, M. and Singh, S. (2021) "A Comprehensive Study on Malware Detection Techniques", International Journal of Advanced Computer Science and Applications, Vol.12(2), pp. 123-133.

[20] Rajput, V. S. et al. (2020) "Machine Learning Techniques for Malware Detection: A Review", International Journal of Emerging Technology and Advanced Engineering, Vol.10(5), pp. 434-441.

[21] Rao, P. B. and Mehta, N. K. (2020) "Malware Detection Techniques: A Review", International Journal of Research in Engineering, Science and Management, Vol.3(9), pp. 230-235.

[22] Choudhary, R. S. et al. (2021) "A Review on Malware Detection Techniques Using Machine Learning Algorithms", Journal of Emerging Technologies and Innovative Research, Vol.8(4), pp. 426-431.

[23] R. Heartfield and G. Loukas(2015), “A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks,” ACM Computing Surveys (CSUR), vol. 48, no. 3, p. 37.

[24] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker (2009), “Identifying suspicious urls: an application of large-scale online learning,” in Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp. 681–688.

[25] C. Seifert, I. Welch, and P. Komisarczuk (2008), “Identification of malicious web pages with static heuristics,” in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, pp. 91–96.

[26] Suyeon Yoo and Sehun Kim (2014),” Two phase malicious web page detection scheme using misuse and anomaly detection”, International Journal of Reliable Information and Assurance, Vol.2, No.1.

[27] Jino S Ganesh, Niranjan Swarup. V, Madhan Kumar.R, Harinisree.A (2020), "Machine Learning Based Malicious Website Detection", International Journal of Engineering Research & Technology, Volume 9, Issue 8, pp. 427-432.

[28] JIANTING YUAN ,YIPENG LIU , LONG YU (2021), "A Novel Approach for Malicious URL Detection Based on the Joint Model", 2021, Hindawi, Volume , Article ID 6615165, pp. 1-10.

[29] Cho Do Xuan, Hoa Dinh Nguyen, Tisenko Victor Nikolaevich (2020), "Malicious URL Detection Based on Machine Learning", International Journal of Advanced Computer Science and Applications (IJACSA), Volume 11, Issue 5, pp. 73-78.

[30] Yongjie Huang, Jinghui Qin, Wushao Wen (2019), "Phishing URL Detection Via Capsule-Based Neural Network", IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification(ASID), pp. 69-73.

[31] VANITHA ANANDKUMAR, "Malicious-URL Detection Using Logistic Regression Technique", 2019, International Journal of Engineering Business Management, Volume 11, pp. 1-9.

[32] Chaochao Luo, Shen Su, Yanbin Sun, Qingji Tan, Meng Han, Zhihong Tian (2020), "A Convolution-Based System for Malicious URLs Detection", Computers, Materials & Continua, Volume 64, Issue 1, pp. 227-239.

[33] C. Seifert, I. Welch, and P. Komisarczuk (2008), “Identification of malicious web pages with static heuristics,” in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, pp. 91–96.

[34] S. Sinha, M. Bailey, and F. Jahanian (2008), “Shades of grey: On the effectiveness of reputation-based “blacklists”,” in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on. IEEE, pp. 57–64.

[35] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker (2009), “Identifying suspicious urls: an application of large-scale online learning,” in Proceedings of the 26th Annual International Conference on Machine Learning. ACM,, pp. 681–688.

[36] B. Eshete, A. Villafiorita, and K. Weldemariam (2013), “Binspect: Holistic analysis and detection of malicious web pages,” in Security and Privacy in Communication Networks. Springer, pp. 149–166.

[37] S. Purkait (2012), “Phishing counter measures and their effectiveness–literature review,” Information Management & Computer Security, vol. 20, no. 5, pp. 382–420.