Unveiling the Power of CNNs with Attention for URL Phishing Detection

MSc Research Project
MSc Cybersecurity

Tanmay Dharmaraj Shukla
Student ID: x22112421

School of Computing
National College of Ireland

Supervisor: Eugene McLaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Tanmay Dharmaraj Shukla
Student ID:	x22112421
Programme:	MSc Cybersecurity **Year:** 2023-2024
Module:	MSc Research Project
Supervisor:	Eugene McLaughlin
Submission Due Date:	25th April 2024
Project Title:	Unveiling the Power of CNNs with Attention for URL Phishing Detection.
Word Count:	6263 **Page Count** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research 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 references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Tanmay Dharmaraj Shukla

Date: 25th April 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. **Please attach a completed copy of this sheet to each project (including multiple copies).**
2. **Projects should be submitted to your Programme Coordinator.**
3. **You must ensure that you retain a HARD COPY of ALL projects, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.**
4. **You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. Late submissions will incur penalties.**
5. **All projects must be submitted and passed in order to successfully complete the year. Any project/assignment not submitted will be marked as a fail.**

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

AI Acknowledgement Supplement

[MSc Research Project]
[Unveiling the Power of CNNs with Attention for URL Phishing Detection]

Your Name/Student Number	Course	Date
Tanmay Dharmaraj Shukla/x22112421	MSc Cybersecurity	25th April 2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. One table should be used for each tool used.

[Insert Tool Name]	
NA	
NA	NA

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Unveiling the Power of CNNs with Attention Mechanism for URL Phishing Detection

Tanmay Dharmaraj Shukla
x22112421

Abstract

91% of cybercrimes is initiated using phishing emails where the victim's personal sensitive information is achieved by the attacker.URL poses prime threats to online security which leads to financial losses and privacy which motivates us to investigate and propose a robust solution. This study is conducted to investigate URL phishing detection methods that traditional methods didn’t achieve and focuses on the comparison between machine learning and deep learning approaches. This study will explore the effectiveness of both ML and DL models using datasets containing benign and phishing URLs sourced from online repositories. The dataset has been split into training and testing stages in a ratio of 90:10 which is applied to 3 models. Random Forest Classifier and Extra Tree Classifier form Machine learning models, alongside Deep learning models such as Convolutional Neural Network with Attention Mechanism were implemented. Performance evaluation was done with the help of a confusion matrix and classification report. Further, an application using Flask is developed for testing phishing URLs. This web application will show URL phishing detection systems which will identify whether the entered URL is phishing or safe. Lastly, it was observed that CNN model performs best with superior and good accuracy of 91%.

1 Introduction

1.1 Background and Motivation

The background and motivation for this study is the increasing prevalence of online phishing attacks, which pose a significant threat to internet user’s security and privacy. Phishing is the practice of pretending to be a reliable source in emails, messages, or websites to trick people into disclosing private sensitive information, such as passwords or bank account information. Because contemporary phishing strategies are so sophisticated, conventional detection measures like Heuristic-based methods and Blacklisting/Whitelisting techniques are insufficient. Thus, to prevent consumers from becoming victims of these nefarious schemes, there is an urgent need for automated systems that can quickly and reliably identify phishing URLs.

This motivates us to propose this study which intends to improve cybersecurity measures and protect people's online activities by creating and optimizing machine learning and deep learning models especially designed for URL phishing detection. The ultimate objective is to develop a strong and trustworthy detection system that can actively recognize phishing attacks, which encourages everyone to use the internet more safely and securely.

1.2 Research Objectives

The research objectives of this study are as follows:

1. Develop and optimize machine learning and deep learning models to accurately identify and classify phishing URLs.

2. Investigate and identify the most informative and discriminative features that will be important variables for detecting phishing URLs and develop effective techniques for extracting and utilizing them.

1.3 Research Questions

Research questions that need to be addressed in this study are as follows:

1. Which features are most indicative of phishing URLs, and how can they be effectively extracted and utilized for classification?
2. What are the strengths and weaknesses of different machine learning and deep learning models in detecting phishing URLs, and how do they compare in terms of accuracy, efficiency, and robustness?
3. How can the dataset be curated and expanded to encompass a broader range of URL categories and sources, ensuring comprehensive training and evaluation of detection models?
4. What preprocessing techniques are most effective in cleaning and preparing the dataset for modeling, and how do they impact the performance of the detection system?

1.4 Assumption and limitation:

After extracting top impacting features which is related to target variables, we can expect good outcome. In this study limited dataset is used which will train and test ML and DL models. The major contribution of this study is our novelty feature Convolutional Neural Network with an Attention Mechanism.

1.5 Structure of the Document:

Existing academic research related to this research issue is reviewed in the second section, titled Related work. section 3 is about Research Method which will describe the procedure and structured approach, section 4 describes Design Specification states of architecture, development processes, and technical specifications for creating the URL phishing detection system, section 5 consists of Implementation which explains the working of models and accuracy graphs, section 6 comprises of Evaluation where comparative analysis of machine learning models and deep learning models is explained and the last section 7 consist of conclusion and future work.

2 Related Work

2.1 Introduction to URL Phishing Detection

This chapter will explain the introduction of URL phishing detection in the cybersecurity field which will identify and mitigate and explain malicious types of web links which has been to deceive the users.

It emerged in response to the increasing prevalence of phishing attacks, which pose significant threats to online security and privacy. The goal of URL phishing detection is to create automated systems that can reliably recognize and block phishing URLs, shielding consumers from being victims of scams. Although URL phishing detection is not ascribed to a single person, it has developed over time thanks to the combined efforts of cybersecurity practitioners and researchers. The field includes a range of methods and strategies, such as heuristic-based approaches, deep learning, and machine learning, that are intended to efficiently identify and reduce the risks related to phishing assaults.

Table 2.1: Table of Types of phishing

Types of Phishing	Description
Spear Phishing	Targeted phishing attacks aimed at specific individuals or organizations, often using personalized information.
Whaling	Phishing attacks that specifically target high-profile individuals or executives within organizations.
Pharming	Redirects users to fraudulent websites without their knowledge by exploiting DNS vulnerabilities.
Vishing	Phishing attacks that use voice or telephone-based communication to trick users into revealing sensitive information.
Smishing	Phishing attacks that use SMS or text messages to deceive users into clicking on malicious links.

2.2 Traditional Approaches to URL Phishing Detection

There are various traditional Approaches earlier in URL Phishing Detection. This includes Heuristic-based methods, Blacklisting and whitelisting techniques and Signature-based detection. Traditional approaches in URL phishing detection basically involve methods that have been developed and used over time so that it can identify and mitigate some sort of phishing attacks.

For evaluating the efficacy of a static type of features (Silva et al., 2020)proposed a study for identifying phishing attacks. Their approach has been involved in various 12 features of different surveys from prior studies if having both self-generated and sources across three distinct samples of phishing and legitimate websites in 2018. The study indicates that although phishing prediction research has advanced, some aspects remain relatively irrelevant, and others are unable to adjust to new attack techniques, indicating that they should be improved upon or removed. The findings draw attention to particular characteristics that are frequently found in phishing URLs, suggesting opportunities for more successful exploitation and emphasizing the need for future research. In addition to quantitative research, the study uses qualitative evaluations to find behavioral patterns and parallels and interrelationships between features. In the end, it is expected that the results will contribute to the creation of improved heuristic strategies or strengthen current techniques in the fight against phishing attempts.

To address the increasing and rapidly growing threat of phishing attacks in the sector of e-commerce (Bhadani, 2023)has proposed a comprehensive approach. Their research, which uses data from the University of New Brunswick, looks at algorithm performance indicators like accuracy, precision, false positive and negative rates, and recall to create a reliable model for phishing URL identification. They present a web application that classifies URLs according to parameters including content and domain characteristics using machine learning methods, such as Gaussian Naïve Bayes, K-Neighbors Classifier, Random Forest, Decision Tree, AdaBoost Classifier, and XGBoost. Their strategy aims to decrease runtime mistakes and vulnerability to cyber surfers by concentrating on URL analysis instead of webpage content. Their hybrid strategy, which uses stacking and other strategies, outperforms more established techniques like blacklisting and visual similarity-based approaches, producing better outcomes concerning both content and generality.

For detecting phishing type of websites (Babagoli et al., 2019)and (Huang et al., 2019)have presented a novel method and they integrated a meta-heuristic-based nonlinear regression algorithm with a feature selection strategy. Twenty features are collected for analysis from a dataset comprising 11,055 instances of authentic webpages and phishing attempts by (Babagoli et al., 2019). With decision tree and wrapper feature selection techniques, the latter attains an impressive 96.32% detection accuracy rate. Then, for website prediction and fraud detection, two meta-heuristic algorithms are used: harmony search (HS) and support vector machine (SVM). The HS algorithm is used to optimize the nonlinear regression model's parameters by creating new harmonies and being driven by dynamic pitch adjustment rates. This method outperforms SVM, producing accuracy rates for the training and testing phases of 94.13% and 92.80%, respectively.

The proposed method given by (Huang et al., 2019)focuses on detecting the Uniform Resource Locator (URL) of a website, leveraging a capsule-based neural network for efficient and effective detection. The network is made up of several parallel branches. The first convolutional layer extracts shallow characteristics from URLs, while the later capsule layers produce accurate feature representations and determine whether a URL is legitimate. The methodology combines the results from every branch to obtain the ultimate detection result. Extensive tests conducted on an Internet-sourced validated dataset show that the method outperforms current state-of-the-art detection algorithms while requiring a reasonable amount of processing overhead.

(Raja et al., 2022)explores a variety of the effects of technology development, emphasizing how important it is to several industries, including banking, business, education, and entertainment, and how it makes time- and money-saving solutions possible. Technology does, however, come with hazards in addition to benefits. This is especially true with malicious URL-based assaults like phishing, spamming, and drive-by downloads, which take advantage of security holes to steal confidential information and carry out harmful tasks. This paper does categorize prior and existing approaches on detection into three major groups which includes Blacklist-based, Heuristic-based, and Machine Learning-based methods. It contends that while heuristic approaches achieve equivalent accuracy to machine learning techniques, they offer higher generalization as compared to blacklist methods. To identify malicious URLs, the paper presents a novel method that extracts and analyzes the most important elements that are extracted from URLs. Despite advancements, the challenge remains in developing methods capable of effectively detecting and mitigating emerging threats in real time.

There is a study which is using blacklisting URL detection which is a traditional approach by (Hong et al., 2020)This study has employed a good approach by combining lexical features gathered through the literature review with blacklisted domains to enhance detection performance. The researchers gathered up-to-date phishing URLs for examination because many of the datasets that are currently available are old. Even though machine learning techniques are widely used in this field, the suggested approach is unique in that it incorporates a variety of features and up-to-date data. Real time detection and thwarting of phishing tactics that are always developing presents the biggest challenge. The method has achieved very good and promisingF-1 score which is around 0.84 which indicates its efficacy in mitigating the risk posed by phishing attacks and fortifying enterprise cybersecurity measures.

The measurement study presented in (Bell & Komisarczuk, 2020)highlights the vital role blacklistsplay in shielding internet users from phishing assaults by analyzing three well-known phishing blacklists: Google Safe Browsing (GSB), OpenPhish (OP) and PhishTank (PT). Despite this,consumers are still exposed to resurfacing threats because none of the blacklists implements aone-time-only URL regulation. The analysis also reveals that a sizable portion of URLs from all three blacklists resurface soon after being removed, raising the possibility of problems withhasty removal or resurfacing threats. Remarkably, OP detected more than 90% of these overlapped URLs before PT and the researchers find a 12% overlap of unique URLs between PT and OP.

There is another study by (Tupsamudre et al., 2019)focuses on enhancing URL-based detection techniques, which leverage machine learning models trained on features extracted directly from URLs. They validate the efficacy of these feature sets by training a logistic regression classifier on a sizable dataset of 100,000 URLs.

Table 2.4: Comparative analysis table for traditional methods

Study	Purpose	Key Features	Strengths	Weaknesses	Proposed Algorithm/Approach
(Silva et al., 2020)	Phishing prediction based on static features	Survey of 12 features, qualitative analysis	Identifies relevance of static features, qualitative insights	Limited focus on static features and potential need for refinement	Survey-based analysis and qualitative assessment
(Bhadani, 2023)	Propose a phishing prediction method using a meta-heuristic-based nonlinear regression algorithm	Meta-heuristic algorithm, feature selection, nonlinear regression	Effective detection approach, utilization of recent data	Lack of real-time adaptability, potential complexity in implementation	Meta-heuristic-based nonlinear regression algorithm

(Babagoli et al., 2019)	Novel phishing website detection using a capsule-based neural network	Capsule-based neural network, parallel branches, URL analysis	Efficient detection approach, utilization of neural network architecture	Potential complexity in training neural networks, computational requirements	Capsule-based neural network
(Huang et al., 2019)	Proposed approach for detecting malicious URLs combining lexical features and blacklisted domains	Lexical features, blacklisted domains, contemporary data	Integration of diverse features, focus on recent data	Potential reliance on specific datasets, limited discussion on real-time adaptability	Combination of lexical features and blacklisted domains
(Raja et al., 2022)	Exploration of recent works in malicious URL detection and novel technique using important features derived from URL	Important features derived from URL, exploration of recent works	Focus on novel detection approach, consideration of recent data	Limited discussion on specific features and methodologies	Utilization of important features derived from URL
(Hong et al., 2020)	Proposal of a method combining lexical features and blacklisted domains to improve phishing URL detection	Lexical features, blacklisted domains, recent phishing URLs	Comprehensive approach, focus on recent data	Potential reliance on specific datasets, limited discussion on real-time adaptability	Combination of lexical features and blacklisted domains
(Bell et al., 2020)	Measurement study analyzing key phishing blacklists	Analysis of Google Safe Browsing, OpenPhish, PhishTank	Insight into blacklist characteristics, comparison of blacklists	Limited discussion on detection methods, focus on blacklist analysis	Measurement study and analysis of phishing blacklists

.

3 Research Methodology

3.1 Methodology

This chapter of Crisp DM is going to establish a structured approach for guiding machine learning models and for deep learning as well. CRISP-DM refers to the Cross-Industry Standard Process for Data Mining. This methodology has been divided into six phases which include Business Understanding, Data Understanding, Modelling, Data Preparation, Deployment and Evaluation. (Alaba, 2021.)

- **Business Understanding**: The very first phase of CRISP-DM is Business Understanding of this study which aims to establish a clear form of understanding for the problem at hand and define the overarching goals of the report. Through collaborative discussions with stakeholders, this study will identify the critical need to detect and categorize malicious URLs to bolster up the measures of cybersecurity. By comprehensively assessing online threats, this study is going to develop a proactive system capable of identifying potential risks before they escalate. It also involves some technical aspects of URL phishing but also grasps some broader implications for the user’s security and also organizational security. Additionally, evaluate available resources, constraints, and potential risks to ensure the feasibility and effectiveness of the proposed solution.

- **Data Understanding**: The second phase is the data understanding, this research is going to explore of datasets which contain URLs whose main aim is to gain insights into the structure, quality, and characteristics of the data. This involves collecting an initial set of URLs from diverse sources, including repositories of both benign and phishing URLs. Subsequently, it examine the data, assessing its completeness, consistency, and potential biases. Through descriptive statistics, visualization techniques, and data profiling and also aim to uncover patterns, trends, and anomalies within the datasets.
- **Data Preparation**: The next phase the data preparation which focuses on a format suitable for transforming raw types of URLs for analysis and modeling. This involves different key steps like data cleaning, feature engineering, and transformation. Initially, the dataset undergoes cleansing to address some of the missing values, duplicates, and inconsistencies, which ensure data quality and integrity.

  After that relevant features are extracted from the URLs, such as length, presence of specific characters, and domain-related attributes, to capture meaningful information for the detection task.

  Then categorical features have been encoded into some sort of numerical representation to facilitate model training. Additionally, the dataset is split into two sets to enable robust model evaluation for training and testing sets, Intrinsic characteristics and integrity of data are maintained.

  By preparing the data in a well-structured and in well-standardized manner, this is the foundation for subsequent stages of the project, including model building and evaluation. Ultimately, the data preparation phase do plays a important role in ensuring the accuracy, reliability, and effectiveness of the URL phishing detection system.
- **Modelling**: For selecting correct techniques and algorithms for training the respective ML and DL models for URL phishing detection, the modelling phase is used.

  This phase includes several key steps, like model selection, training, and evaluation. Initially, suitable machine learning and deep learning models has been chosen based on the data characteristics and the requirements of the task.

  These models are then trained using the prepared dataset, with parameters optimized to maximize performance. After training, the models are tested for accuracy, precision, recall, and F1-score using several measures. This assessment sheds light on how well the models differentiate between legitimate and fraudulent URLs. Furthermore, model interpretability strategies could be used to better comprehend the underlying mechanisms guiding the categorization choices. The modeling step seeks to determine the most efficient and trustworthy models for URL phishing detection through thorough testing and assessment. The project's latter stages, like model deployment and refining, are informed by the results of this phase, which ultimately helps to create a reliable and accurate detection system.
- **Evaluation**: For determining the performance of the trained machine and deep learning models is rigorously assessed for classifying URLs to be either benign or phishing, this can be done through the evaluation phase. A range of metrics, including recall, accuracy, precision, and F1-score, are computed to assess the models' performance in quantitative terms. Techniques for qualitative analysis can also be used to learn more about the behavior and decision-making processes of the models. The outputs of the model are carefully examined to find any biases or anomalies that can compromise its

dependability in practical situations. The advantages and disadvantages of any model are determined by methodical assessment, which helps with judgments about model selection and improvement.

- **Deployment**: During the deployment stage, real-time URL phishing detection is facilitated by integrating the trained models into an actual environment. This entails integrating the models with appropriate frameworks and technologies—like Flask—in a web application. The deployment procedure makes sure that end users may easily test URLs for possible phishing risks by making the detection system available to them. This makes it possible to identify problems or abnormalities quickly.

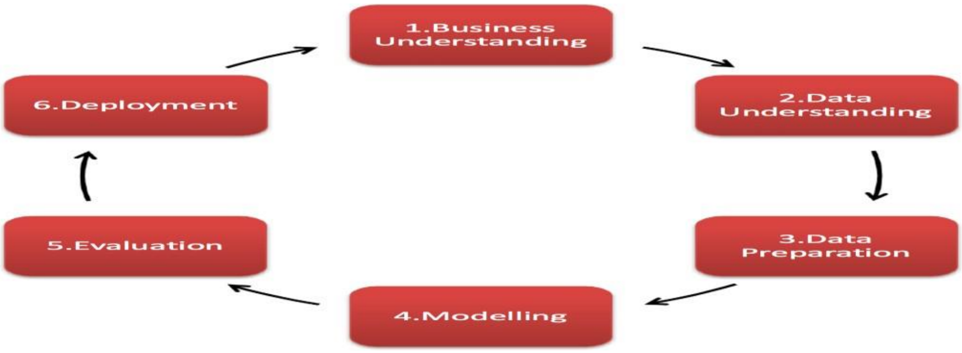


Figure 3.1: CRISP-DM Methodology

3.2 Libraries Imported

There are various libraries which have been imported into this project and implemented as well. These libraries serve different purposes in this project in phishing of URL detection and model development. The first one is the “pickle” library which do facilitates the serialization and deserialization of Python objects, enabling the storage and retrieval of trained models. For analysis and data manipulation two important libraries which have been used include NumPy ("np") and Pandas ("pd") allowing for efficient handling of datasets. Seaborn ("sns") and Matplotlib ("plt") are utilized for data visualization, enabling the creation of informative plots and charts to gain insights into the data. Although Plotly Express ("px") and Plotly Graph Objects ("go") offer interactive visualization capabilities for more complicated data representations, Keras backend ("K") offers functionality for neural network operations and customisation.(*Coursera*, (2024).*9 Best Python Libraries for Machine Learning*)

For model training and evaluation, Scikit-learn ("ensemble") offers machine learning algorithms and tools; RandomForestClassifier and ExtraTreesClassifier are used for classification tasks. For data preprocessing and categorical variable encoding, Scikit-learn's LabelBinarizer, LabelEncoder, and MinMaxScaler are used. Deep learning models are constructed using the tensorflow.keras library; layers such as Conv1D, Dropout, Flatten, Dense, Bidirectional, and LSTM are used to construct different neural network topologies, while Sequential defines the model architecture.(*Introduction to Deep Learning: Advanced Layer Types,* 2024.)

"train_test_split" from Scikit-learn, which divides the dataset into training and testing sets, "to_categorical" from tensorflow.keras.utils for one-hot encoding categorical labels, and "precision_recall_fscore_support" from Scikit-learn for calculating evaluation metrics are among the utility functions imported. Additionally, to evaluate model performance and provide evaluation reports, Scikit-learn's "confusion_matrix," "accuracy_score," and "classification_report" are employed. At last, there is a warning library which has been used for suppressing any kind of warning messages during execution for cleaner output. (*GeeksforGeeks*,(2022).*Compute Classification Report and Confusion Matrix in Python*)

3.3 Feature Extraction

Feature extraction is a fundamental step in the process of preparing data for machine learning models. In URL phishing detection, feature extraction involves transforming raw URL data into a structured format that captures relevant information for classification tasks. To extract properties directly observable from the URL itself, **address bar-based** features are used. One of these elements is **"length_of_url**" which calculates the URL's length. Longer URLs can be an indication of attempts to hide harmful content. **"http_has"** verifies whether "http/https" is present in the URL's domain portion since phishers might trick users by using HTTPS tokens. Special characters like "@" in the URL, which are frequently used to conceal the true address, are identified by **"suspicious_char"**. **"prefix_suffix"** looks for unusual characters like "-" in the domain portion, which is a trick phishers use to look like real URLs. Furthermore, the "dots" and "slash" features examine the URL's structure to find patterns linked to suspicious activity or redirection.

The **domain-based** features focus on some attributes which have been derived from the domain portion of the URL. There is a phishing activity that ensures the popularity of the website, with low or nonexistent traffic which is the "Web Traffic check. **"Domain Age"** and "Domain End" quantify the lifespan and current status of the domain, as phishing websites often have short lifespans or irregular termination dates.

Features **based on HTML and JavaScript** examine the behavior and content of web pages that are linked to the URL. **"IFrameRedirection"** is frequently used in phishing attacks to identify hidden web pages within a URL. **"Disabling Right Click"** and **"Status Bar Customization"** alert users to attempts to trick them by changing the behavior of the browser. The terms **"Website Forwarding"** and **"LinksPointingToPage"** evaluate internal and external linking patterns and redirection patterns that may be signs of phishing efforts. Through the extraction and integration of these varied properties into the dataset, the model acquires important insights to efficiently differentiate between legitimate and fraudulent URLs.

3.4 Data Splitting (Training and Testing the Model)

The dataset in this section will be divided into two states including training and testing to facilitate model training and evaluation. The customary procedure entails dividing the dataset in half, usually allocating 90% to the training set and keeping the remaining 10% for testing. This guarantees that the model is trained on enough data to identify patterns and correlations efficiently and offers a separate dataset for assessing the model's performance on unobserved samples. The algorithm learns to identify patterns and generate predictions based on the input features by using the training set as input for the model. In contrast, the testing set is used to compare the model's predictions to the ground truth labels in order to determine how well the model performs. This makes it possible to estimate how well the model generalizes and how well it can predict fresh, untested data. The data splitting phase helps to prevent overfitting, which occurs when a model performs well on training data but badly on unseen data, by dividing the dataset into distinct training and testing sets. Developing dependable and strong URL phishing detection models requires this technique.

3.5 Dataset Description

The dataset which has been used in this study contains various collections of URLs which has been sourced from online repository, which have both types of instances benign and phishing. This diverse dataset enables two URL phishing detection methods like development and evaluation of it which enables the identification of malicious URLs and proactive threat mitigation. This dataset contains four main types of URLs categories includes: spam, benign, malware and phishing. The benign URLs are taken from over 35,300 samples of the top websites according to Alexa. These URLs are accepted as authentic and act as a standard by which other URLs that might be harmful are measured. Furthermore, almost 12,000 spam URLs—which are linked to undesirable and unsolicited content—are taken from the publicly accessible WEBSPAM-UK2007 dataset.(*URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB,* n.d.)Furthermore, around 10,000 phishing URLs have been sourced from OpenPhish, which is a repository of active phishing sites. It also highlights URLs which designed to deceive users into divulging sensitive information or engaging in fraudulent activities. Furthermore, the dataset includes more than 11,500 phishing URLs which is kind of related to malware websites and obtained from the DNS-BH project, which maintains a list of known malware sites.

4 Design Specification

The architecture, development processes, and technical specifications for creating the URL phishing detection system are described in the design chapter. The needs of the system, including its functionality, performance, and user interface criteria, are covered in detail at the beginning of the chapter. This provides the framework upon which the latter design choices are made. The system architecture is then shown, outlining the many parts, modules, and ways in which they work together. This comprises the web application interface for user interaction, the model training and assessment pipeline, and the data pipeline for preprocessing and feature extraction. The architecture is made to be adaptable, modular, and scalable in order to handle upcoming additions and changes.

After that, the study dives into the specifics of implementation, going over the tools, frameworks, and technologies used to create the system. For machine learning and deep learning applications, this includes libraries like TensorFlow and Scikit-learn and programming languages like Python. Furthermore, web development frameworks such as Flask are utilized in the user interface development process. In addition, the chapter discusses implementation-related issues and the solutions that were developed to resolve them. This could involve problems with system integration, model optimization, or data pretreatment.
Making sure the system is dependable, effective, and easy to use is a priority during the design and implementation phases. Testing, validation, and performance optimization are examples of quality assurance techniques that are used to confirm the accuracy and efficacy of the system.

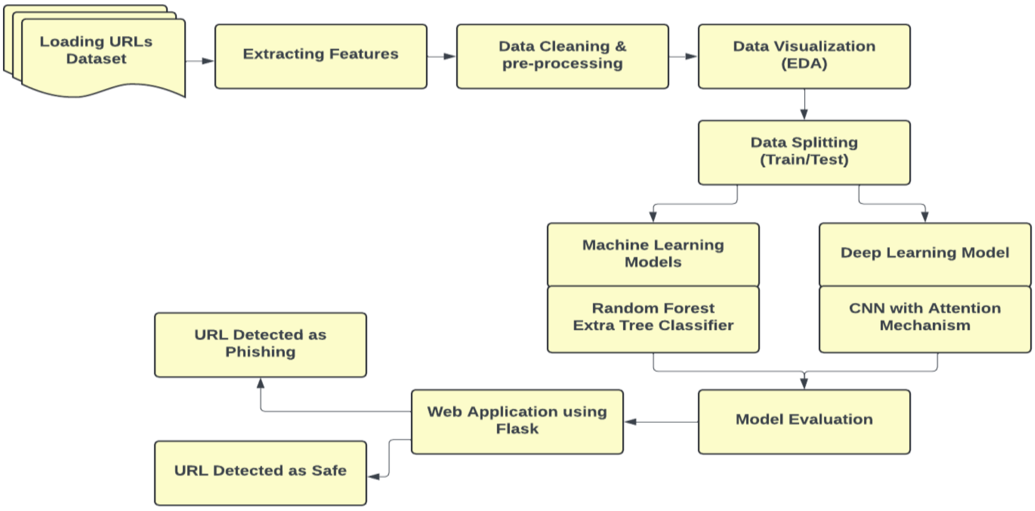


Figure 4.1: Proposed Workflow

5 Implementation

5.1 Random Forest Classifier

Random Forest Classifier is an ensemble learning method used for classification tasks in machine learning.

This model operates with constructing various decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. To decorrelate the trees and lessen overfitting, each tree in the random forest is trained on a random subset of the training data and chooses a random subset of characteristics at each node to split on. The Random Forest Classifier is used to detect phishing URL due to its ability to differentiate between malicious and genuine URLs by training on features retrieved from both types of URLs phishing and benign datasets. To accurately and consistently classify URLs, the Random Forest Classifier makes use of the diversity of decision trees to capture intricate correlations between characteristics and class labels. It can handle various types of high-dimensional data and mitigate overfitting to make it suitable for doing the task of identifying various harmful URLs.(Random Forest Algorithm in Machine Learning,GeeksforGeeks, (2024).)

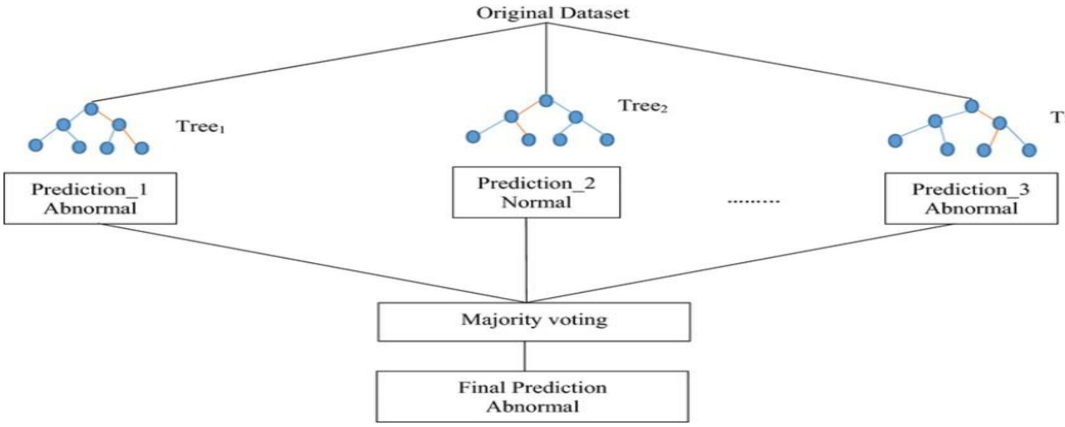
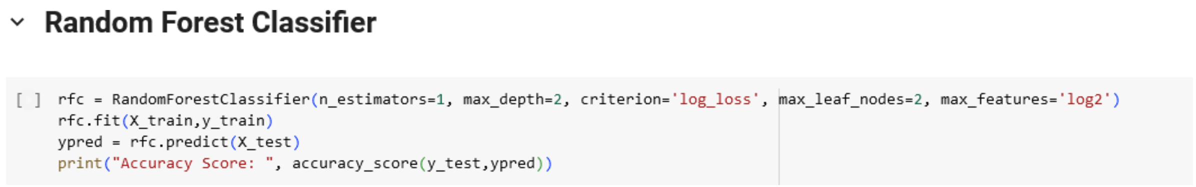


Figure 5.1: Random Forest Classifier architecture (Shafi et al., (2020))



Random Forest Classifier

```
[ ] rfc = RandomForestClassifier(n_estimators=1, max_depth=2, criterion='log_loss', max_leaf_nodes=2, max_features='log2')
    rfc.fit(X_train,y_train)
    ypred = rfc.predict(X_test)
    print("Accuracy Score: ", accuracy_score(y_test,ypred))
```

Figure 5.2: Random Forest Classifier Model Training

This figure 5.3 shows the confusion matrix of the random forest classifier which is showing TP, TN, FP and FN instances. Value of TP is 100, TN is 69, FP is 0 and FN is 31.

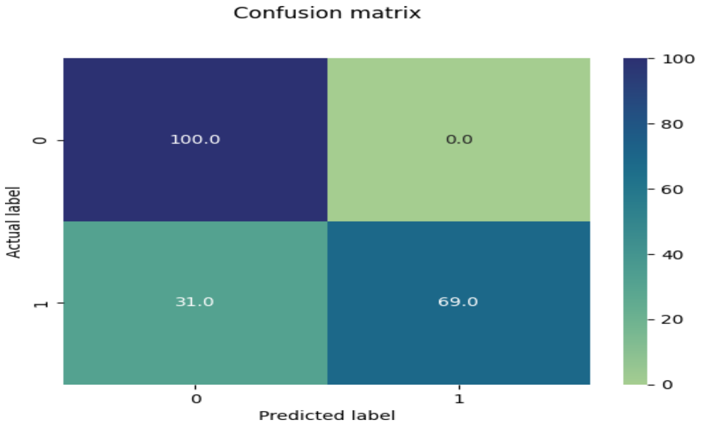


Figure 5.3: Confusion Matrix

5.2 Extra Tree Classifier

Another ensemble learning method is the extra tree classifier which is similar to the random forest, which is used for doing classification tasks in machine learning. It constructs multiple decision trees which is used for during training and outputs the mode of the classes for classification. However, the Extra Trees Classifier introduces some sort of additional randomness by selecting random thresholds for each feature at every split, without bootstrapping or feature selection. The extra tree classifier is an alternative model for URL phishing detection. The classifier gains the ability to differentiate between malicious and genuine URLs by training on a dataset that includes features retrieved from both types of URLs: phishing and benign. The Extra Trees Classifier further minimizes overfitting and enhances generalization performance by taking advantage of the randomization in feature and threshold selection. It is a useful addition to the collection of models used in the project because of its capacity to handle high-dimensional data and take use of the diversity of decision trees.

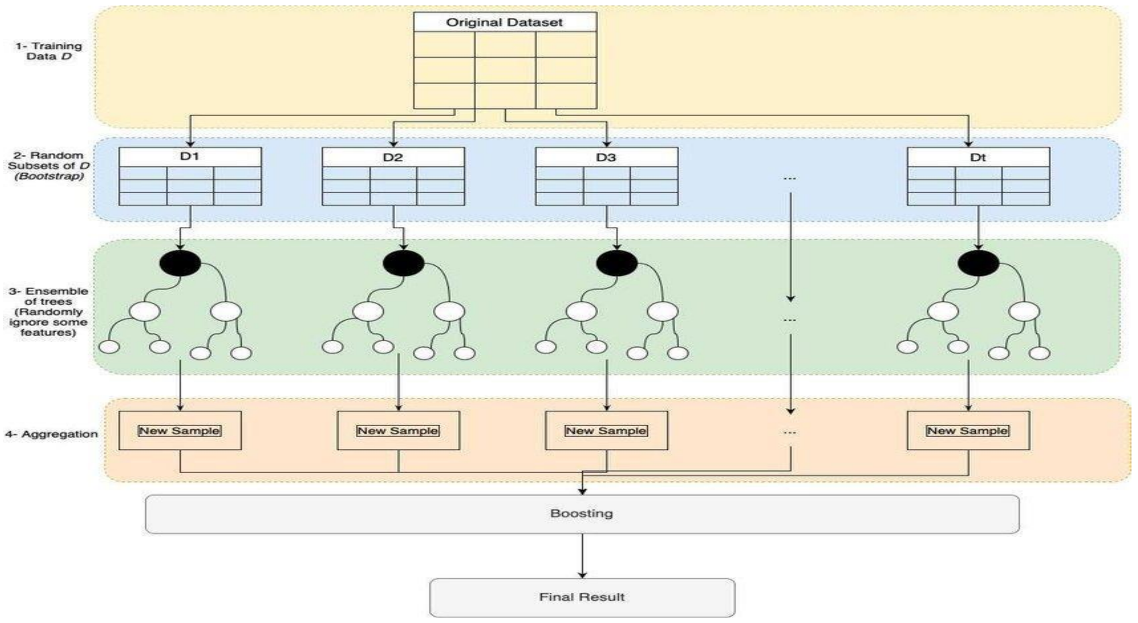
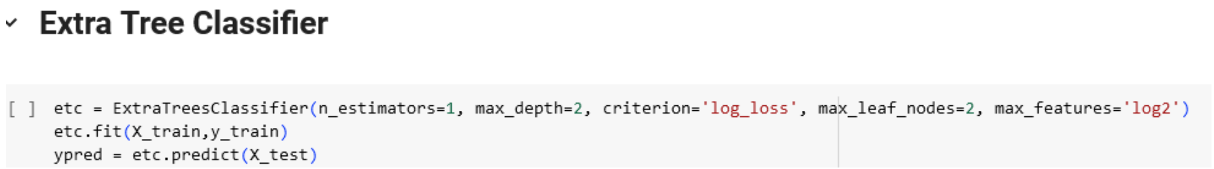


Figure 5.4: Extra Tree Classifier architecture. (Zaher,Ghoneem, Abdelhamid, Ezzat.(2023))



✓ Extra Tree Classifier

```
[ ] etc = ExtraTreesClassifier(n_estimators=1, max_depth=2, criterion='log_loss', max_leaf_nodes=2, max_features='log2')
    etc.fit(X_train,y_train)
    ypred = etc.predict(X_test)
```

Figure 5.5: Extra Tree Classifier Model Training

Figure 5.6 shows the confusion matrix of the **Extra Tree Classifier** which shows TP, TN, FP and FN instances. Value of TP is 86, TN is 89, FP is 14 and FN is 11.

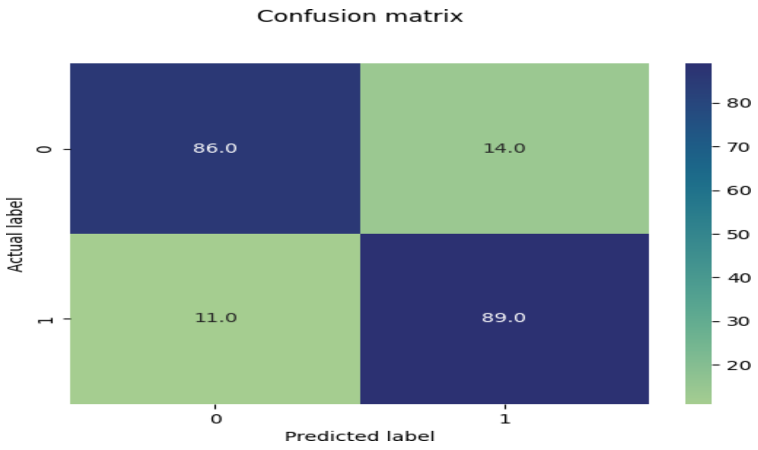


Figure 5.6: Confusion Matrix

5.3 CNN with an attention mechanism

Convolutional Neural Networks (CNNs) with Attention Mechanism is deep learning models which actually combine the strengths of CNNs model for capturing the data to focus on relevant type of information for doing spatial patterns with ability of attention mechanism. In CNNs with attention, attention mechanisms are integrated into the architecture to dynamically weigh the importance of different spatial locations within the input data. The model gains the ability to automatically identify pertinent spatial patterns indicative of malicious URLs by training on a dataset that includes features taken from both benign and phishing URLs. This allows the model to concentrate on important aspects of the URL that play a major role in the classification decision. The attention mechanism helps the model distinguish between benign and phishing URLs more successfully by helping it to prioritize significant elements and disregard irrelevant ones.

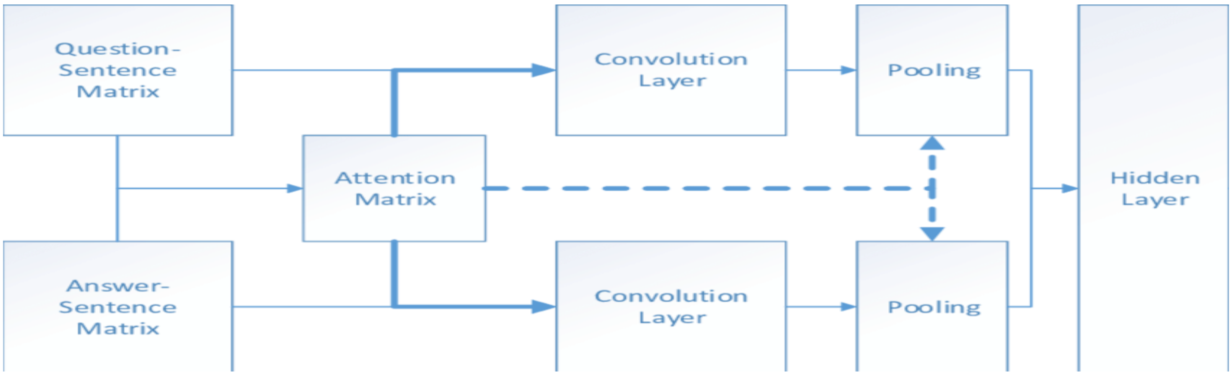
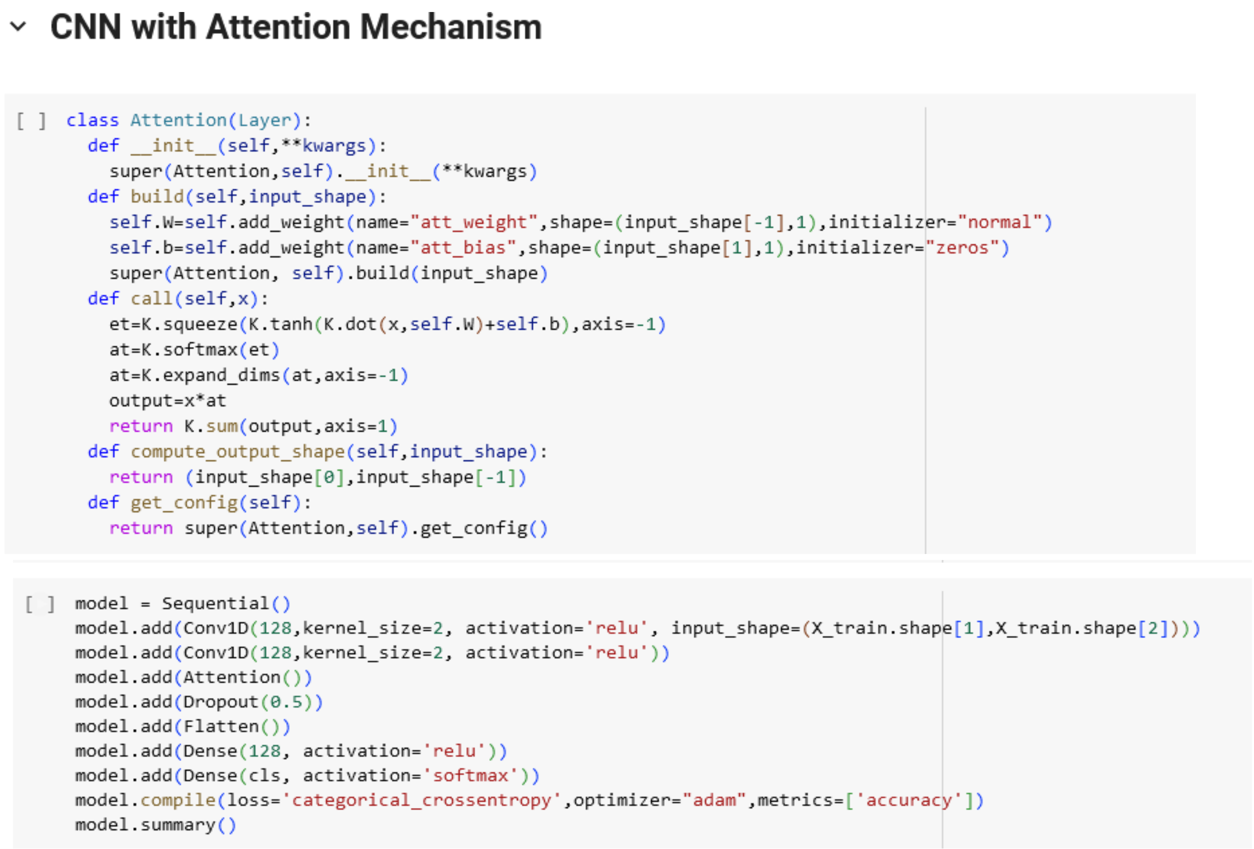


Figure 5.7: CNN with attention mechanism architecture (Deriu & Cieliebak, 2017)



CNN with Attention Mechanism

```
[ ] class Attention(Layer):
      def __init__(self,**kwargs):
        super(Attention,self).__init__(**kwargs)
      def build(self,input_shape):
        self.W=self.add_weight(name="att_weight",shape=(input_shape[-1],1),initializer="normal")
        self.b=self.add_weight(name="att_bias",shape=(input_shape[1],1),initializer="zeros")
        super(Attention, self).build(input_shape)
      def call(self,x):
        et=K.squeeze(K.tanh(K.dot(x,self.W)+self.b),axis=-1)
        at=K.softmax(et)
        at=K.expand_dims(at,axis=-1)
        output=x*at
        return K.sum(output,axis=1)
      def compute_output_shape(self,input_shape):
        return (input_shape[0],input_shape[-1])
      def get_config(self):
        return super(Attention,self).get_config()
```

```
[ ] model = Sequential()
    model.add(Conv1D(128,kernel_size=2, activation='relu', input_shape=(X_train.shape[1],X_train.shape[2])))
    model.add(Conv1D(128,kernel_size=2, activation='relu'))
    model.add(Attention())
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(cls, activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=['accuracy'])
    model.summary()
```

Figure 5.8: CNN with attention mechanism Model Training

Figure 5.9 shows the confusion matrix of **CNN with attention mechanism** which is showing TP, TN, FP and FN instances. Value of TP is 90, TN is 93, FP is 10 and FN is 7.

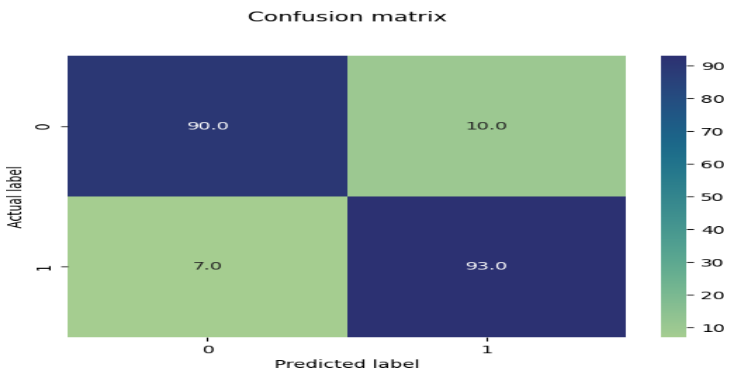


Figure 5.9: Confusion Matrix

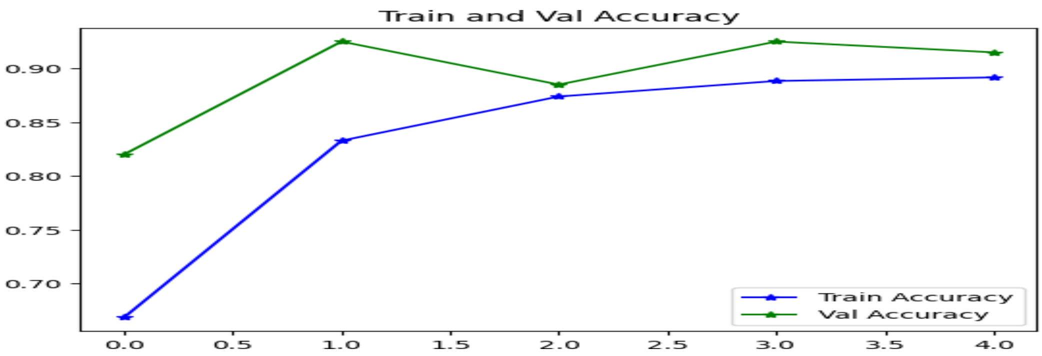


Figure 5.10: Accuracy Graph

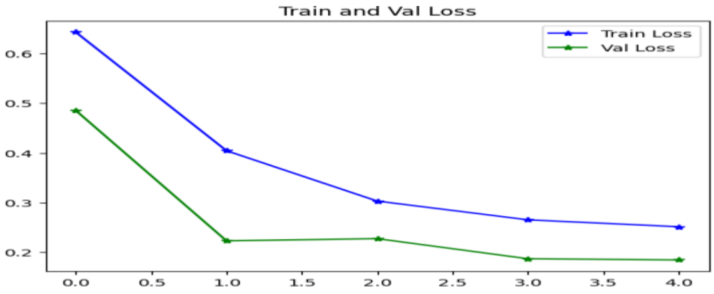


Figure 5.11: Loss Graph

5.4 Data Visualization

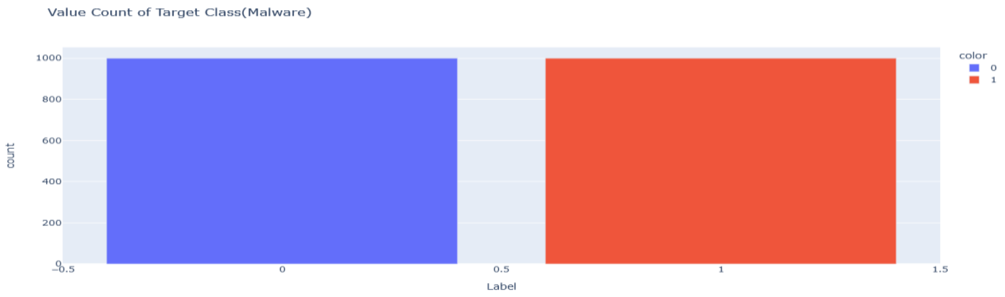


Figure 5.12: Bar Graph

Figure 5.12 depicts a bar graph showing the distribution of the target class "malware." In this graph, the blue bars represent instances labeled as 0, while the orange bars represent instances labeled as 1. Each bar corresponds to the count of instances belonging to the respective class. Specifically, the blue bars, labeled as 0, indicate instances classified as non-malware, with a count of 1000. Conversely, the orange bars, labeled as 1, represent instances classified as malware, also with a count of 1000.

Figure 5.13: Pie chart

Figure 5.13 presents a pie chart that represents the count distribution of the "url_depth" column. Each segment of the pie chart corresponds to a specific depth level of URLs, while the size of each segment represents the proportion of URLs at that depth level.

Like for example, there is a segment colored in orange, representing URLs with a depth of 1, has the highest count of 755 instances, accounting for 37.8% of the total URLs. Similarly, other depth levels are represented by different colors in the pie chart with their respective percentages.

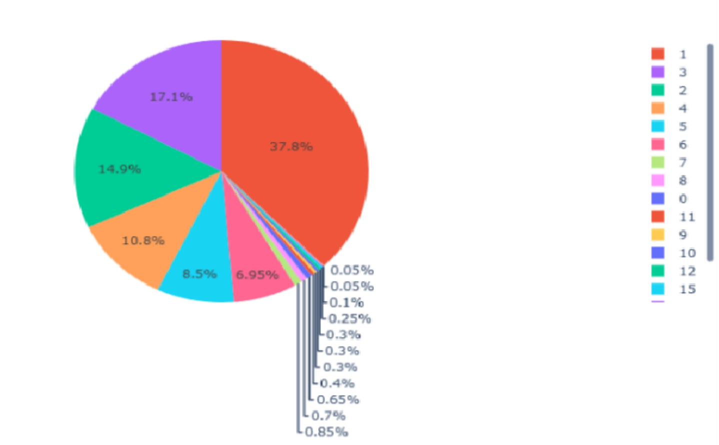


Figure 5.13: Pie chart URL depth

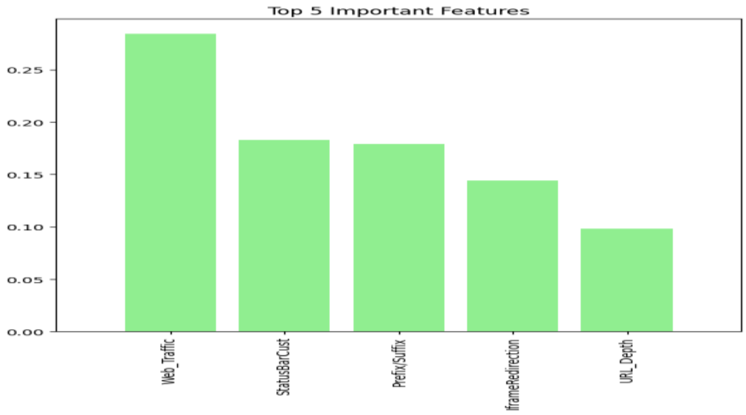


Figure 5.14: Top 5 important features

6 Evaluation

6.1 Comparative Analysis of Evaluation of Machine Learning and Deep Learning Models

This section will give a comparative analysis of machine learning models and deep learning models as well. Below table 6.1 shows a comparative analysis of all 3 models on the basis of their respective accuracy scores**. CNN with an attention mechanism achieved the highest accuracy score of 0.915, outperforming both Random Forest Classifier (0.845) and Extra Trees Classifier (0.875).**

Title 6.1: Comparative Analysis of Models Accuracy Score

Model	Accuracy
Random Forest Classifier	0.845
Extra Trees Classifier	0.875
CNN with an attention mechanism	0.915

6.2 Comparative Analysis

.

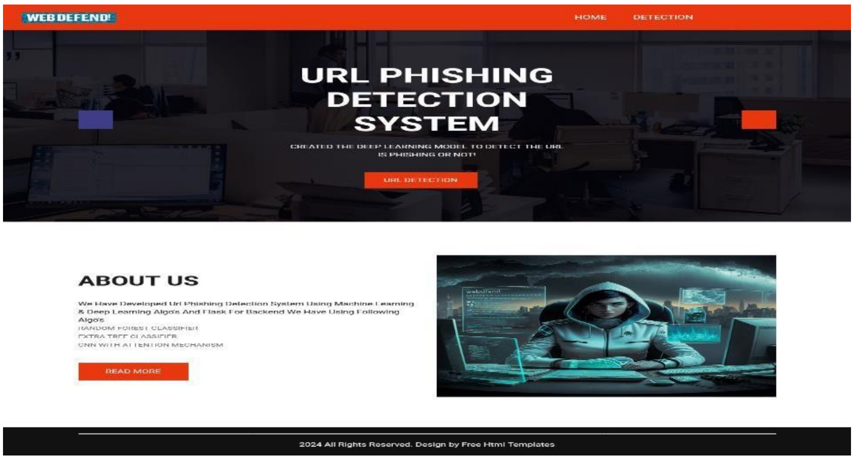
This section will explain comparative analysis based on the performance. So there is a study given by (Jagdale & Chavan, 2022) who have done work on different machine learning algorithms in URL phishing detection. They have used several ML algorithms like Naive Bayes, Decision Tree, also a Hybrid Ensemble Stacking algorithm, achieving accuracies of 85.59%, 87.04%, and 89.25%, respectively. But I have used both machine learning and deep learning models which is achieving higher accuracy. Notably, my focus is primarily on deep learning, which has already demonstrated very good performance compared to this study and other prior work as well. Furthermore, I have implemented graphical user interface (GUI) application for practical usability and used own dataset rather than any publicly available datasets like those from Kaggle and achieving 91% accuracy using CNN with attention mechanism this answers our third question.

Table 6.2: Comparison of the ML/DL Models with prior work(Answers second question)

Model	Accuracy
Naïve Bayes (Prior work)	85.59%
Decision Tree (Prior work)	87.04%
CNN with Attention Mechanism (Our Best Model)	**91%**

6.3 GUI Flasks

Figure 6.1 represents the homepage of a web application designed for phishing URL detection. When a user accesses the system, they interact with it through the interface. With features like a navigation bar, a central area for content display, and a logo or header, the design seems straightforward and user-friendly. After some investigation, it appears that the homepage has features that allow users to enter or paste URLs for analysis. Users are able to submit URLs they believe to be phishing attempts through this input option. Descriptive language or prompts that inform users of the system's features and offer guidance on how to utilize it successfully may also be present. The web application's navigation bar probably includes links to other areas or features, like user preferences, support files, and extra security and URL analysis tools.(*GitHub - FixedOctocat/Phishing-Website-Detector: Website with UI/API on Python Flask for Checking Urls on Phishing*, n.d.)



URL PHISHING DETECTION SYSTEM

CREATED THE DEEP LEARNING MODEL TO DETECT THE URL IS PHISHING OR NOT!

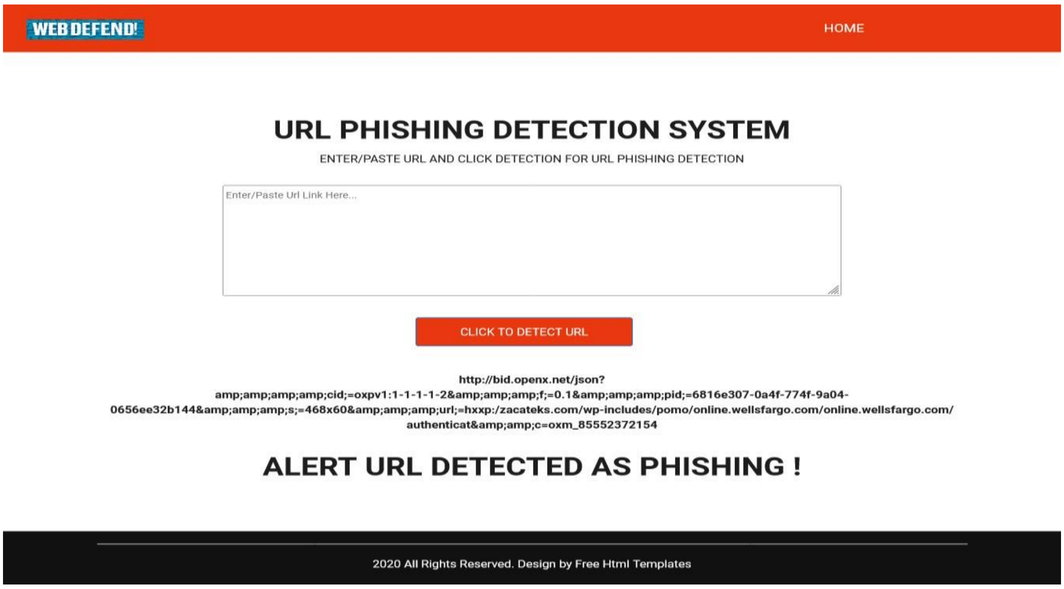
ABOUT US

We Have Developed Url Phishing Detection System Using Machine Learning & Deep Learning Algo's And Flask For Backend We Have Using Following Algo's
RANDOM FOREST CLASSIFIER
EXTRA TREE CLASSIFIER
CNN WITH ATTENTION MECHANISM

2024 All Rights Reserved. Design by Free Html Templates

Figure 6.1: Web Application Home Screen

Figure 6.3 shows the page where the user needs to enter or paste the respective URL to check out whether it is phishing or not. So, enter the URL in the message box and click detection for URL detection system and after that click to detect URL.shows the respective URL is phishing and there is alert message as well which shows “Alert URL detected as Phishing!”.



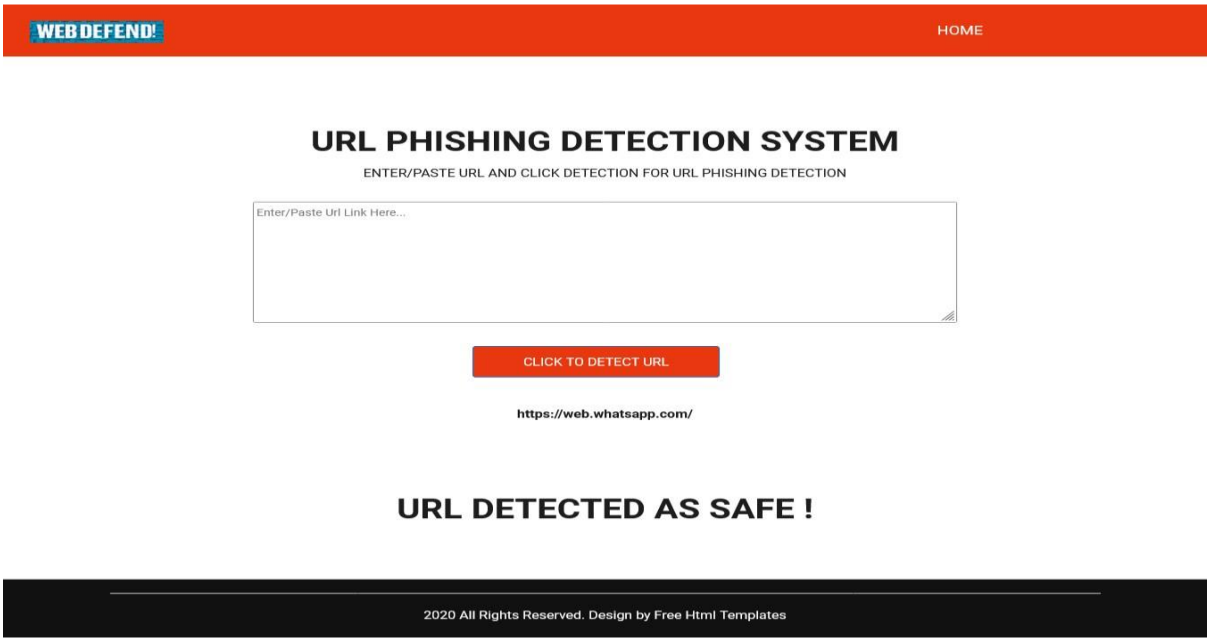
WEB DEFEND!
HOME
URL PHISHING DETECTION SYSTEM
ENTER/PASTE URL AND CLICK DETECTION FOR URL PHISHING DETECTION
Enter/Paste Url Link Here...
CLICK TO DETECT URL

http://bid.openx.net/json?amp;amp;amp;amp;cid;=oxpv1:1-1-1-1-2&amp;amp;f;=0.1&amp;amp;pid;=6816e307-0a4f-774f-9a04-0656ee32b144&amp;amp;s;=468x60&amp;amp;url;=hxxp:/zacateks.com/wp-includes/pomo/online.wellsfargo.com/online.wellsfargo.com/authenticat&amp;c=oxm_85552372154

ALERT URL DETECTED AS PHISHING !

2020 All Rights Reserved. Design by Free Html Templates

Figure 6.3: Alert URL detected as phishing



WEB DEFEND!
HOME
URL PHISHING DETECTION SYSTEM
ENTER/PASTE URL AND CLICK DETECTION FOR URL PHISHING DETECTION
Enter/Paste Url Link Here...
CLICK TO DETECT URL

https://web.whatsapp.com/

URL DETECTED AS SAFE !

2020 All Rights Reserved. Design by Free Html Templates

Figure 6.4 shows the respective URL is not phishing and there is the message as well that shows “URL detected as Safe!”.

Figure 6.3: Alert URL detected as safe

7 Conclusion and Future Work

7.1 Conclusions

In a nutshell up, the creation and assessment of the URL phishing detection system have been a noteworthy undertaking with the goal of improving cybersecurity protocols and shielding consumers from virtual dangers. We have methodically investigated a range of approaches, strategies, and models during this study in order to accurately detect and categorize bad URLs. The phase of data understanding yielded significant insights into the properties and distribution of URLs, establishing the foundation for the ensuing stages of data preparation and feature extraction. Through the process of extracting pertinent features from the dataset, such as features based on the address bar, domain, HTML, and JavaScript, we were able to provide our models with the data they needed to distinguish between legitimate and fraudulent URLs,We also found top 5 impacting features, this answers **first question**.

Three different models were implemented and assessed throughout the modeling phase: CNN with an attention mechanism, Random Forest Classifier, and Extra Trees Classifier. We found that the CNN with an attention mechanism was the best-performing model, attaining the maximum accuracy score of 0.915, after conducting extensive testing and evaluation. This emphasizes how crucial it is to use attention mechanisms and deep learning approaches in order to identify intricate patterns in URL data. **In addition, a user-friendly web application for real-time phishing URL identification was developed during the implementation phase.** With the help of this program, users may evaluate the legality of URLs with speed and confidence and take the necessary precautions to reduce any potential risks.

7.2 Future Works

Future research can investigate many paths for enhancing and broadening the scope of the URL phishing detection system. First off, broadening and varying the dataset may improve the system's capacity to adapt to novel and developing threats. A more complete training set for the models would be produced by adding more recent data and a wider variety of URL categories and sources. Further increases in detection efficiency and accuracy may result from further model optimization and refinement. Testing various architectures, hyperparameters, and preprocessing methods may assist in resolving issues and improving the system's resilience.

Incorporating real-time data monitoring and updating processes would also allow the system to quickly adjust to changing phishing tactics and patterns. To ensure the system continues to be successful against emerging threats, this may entail putting in place systems for ongoing data gathering, model retraining, and automated updates.

Ultimately, improving user interface elements and offering instructional materials may enable users to engage with URLs in a more knowledgeable manner. This can entail adding interactive elements, explaining detection outcomes, and giving advice on the safest ways to use the internet.

References

Coursera.(2024).9 Best Python Libraries for Machine Learning. Retrieved from https://www.coursera.org/articles/python-machine-learning-library

Alaba, A. (2021). Detecting Spam Campaigns on Twitter Using Machine Learning Approach MSc Internship Cyber Security. https://norma.ncirl.ie/5098/1/adedoyinalaba.pdf

Babagoli, M., Aghababa, M. P., & Solouk, V. (2019). Heuristic nonlinear regression strategy for detecting phishing websites. Soft Computing, 23(12), 4315–4327. https://doi.org/10.1007/S00500-018-3084-2

Bell, S., & Komisarczuk, P. (2020). An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. ACM International Conference Proceeding Series. https://doi.org/10.1145/3373017.3373020

Bhadani, D. (2023). Heuristic-Based Phishing Site Detection. https://scholarworks.calstate.edu/downloads/v692td67g

GeeksforGeeks.(2022).Compute Classification Report and Confusion Matrix in Python. Retrieved from https://www.geeksforgeeks.org/compute-classification-report-and-confusion-matrix-in-python/

Deriu, J., & Cieliebak, M. (2017). SwissAlps at SemEval-2017 Task 3: Attention-based Convolutional Neural Network for Community Question Answering. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 334–338. https://doi.org/10.18653/V1/S17-2054

GitHub - FixedOctocat/Phishing-website-detector: Website with UI/API on python flask for checking urls on phishing. Retrieved April 25, 2024, from https://github.com/FixedOctocat/Phishing-website-detector

Hong, J., Kim, T., Liu, J., Park, N., & Kim, S. W. (2020). Phishing URL Detection with Lexical Features and Blacklisted Domains. Adaptive Autonomous Secure Cyber Systems, 253– 267. https://doi.org/10.1007/978-3-030-33432-1_12

Huang, Y., Qin, J., & Wen, W. (2019). Phishing URL Detection Via Capsule-Based Neural Network. Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification, ASID, 2019-October, 22–26. https://doi.org/10.1109/ICASID.2019.8925000

Introduction to deep learning: Advanced layer types. (2024). Retrieved April 25, 2024, from https://carpentries-incubator.github.io/deep-learning-intro/4-advanced-layer-types.html

Jagdale, N., & Chavan, P. (2022). Hybrid Ensemble Machine Learning Approach for URL Phishing Detection. 2022 2nd Asian Conference on Innovation in Technology,
ASIANCON 2022. https://doi.org/10.1109/ASIANCON55314.2022.9908667

Zaher,Ghoneem, Abdelhamid, Ezzat.(2023)Comparative Study Between Machine learning algorithms and feature ranking techniques on UI-PRMD dataset.Retrieved April 25, 2024, from https://www.researchgate.net/publication/370242370_Comparative_Study_Between_Machine_learning_algorithms_and_feature_ranking_techniques_on_UI-PRMD_dataset

Raja, A. S., Pradeepa, G., & Arulkumar, N. (2022). Mudhr: Malicious URL detection using heuristic rules based approach. AIP Conference Proceedings, 2393(1). https://doi.org/10.1063/5.0074077

Random Forest Algorithm in Machine Learning - GeeksforGeeks. (2024). Retrieved April 25, 2024, from https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

Shafi, A. S. M., Molla, M. M. I., Jui, J. J., & Rahman, M. M. (2020). Detection of colon cancer based on microarray dataset using machine learning as a feature selection and classification techniques. SN Applied Sciences, 2(7), 1–8. https://link.springer.com/article/10.1007/s42452-020-3051-2

Silva, C. M. R. da, Feitosa, E. L., & Garcia, V. C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. Computers and Security, 88. https://doi.org/10.1016/J.COSE.2019.101613

Tupsamudre, H., Singh, A. K., & Lodha, S. (2019). Everything is in the name – a URL based approach for phishing detection. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11527 LNCS, 231–248. https://doi.org/10.1007/978-3-030-20951-3_21

URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d). Retrieved April 25, 2024, from https://www.unb.ca/cic/datasets/url-2016.html

Link:For presentation and demo video is mentioned below.
https://studentncirl-my.sharepoint.com/:p:/r/personal/x22112421_student_ncirl_ie/_layouts/15/Doc.aspx?sourcedoc=%7B27CCD51B-4B95-4B21-978E-AE1B1F4C7265%7D&file=Unveiling%20the%20Power%20of%20CNNs%20with%20Attention%20for.pptx&wdLOR=c8EA6CADC-F115-4F45-AA93-3FF14A94502F&action=edit&mobileredirect=true

Password:22112421@2023