URL Phishing Detection Using Machine Learning

MSc Research Project
MSc Cybersecurity

Sconal Gonsalves
Student ID: x23235551

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sconal Godfrey Gonsalves

Student ID: X23235551

Programme: MSc Cybersecurity **Year:** 2024-25

Module: Practicum/ internship part-2

Supervisor: Arghir Nicolae Moldovan

Submission Due Date: 12th December 2024

Project Title: URL Phishing Detection Using Machine Learning

Word Count: 7572 **Page Count** 20

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 bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sconal Godfrey Gonsalves

Date: 12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

URL Phishing Detection Using Machine Learning

Sconal Godfrey Gonsalves
X23235551

Abstract

The overall performance of all the machine learning models on three different datasets that are been selected are examined in this research work which is been done, with keeping an aim of feature selection and the strategies of reducing the dimensionality. After thorough examination and testing of the models like Random Forest and Gradient Boosting out performed and showed a very exceptional accuracy and toughness. In order to balance the accuracy and the computational efficiency recursive Feature Elimination was used and it emerged as very effective method for doing it, while at the same time enhanced the ensemble techniques and maintained performance of the model. The main role was played by correlation matrix in improving the generalization for the models which were simpler by overcoming the problem of plurality but at the same time by overcoming that problem it created a problem of overfitting when all the features which are highly correlated were retained. The results which were achieved provides an approach which can be followed for implementing a scalable and accessible solutions using machine learning in it.

1 Introduction

Background:

In this evolving world that are full of technologies there is a significant threat of phishing attacks. Which are targeting the normal users to steal the sensitive information through the sharing of fraud URLs. Due to the advancement in the phishing techniques which are been developed by the attackers, the traditional methods which were been developed like blocklists and manual verification technique they usually fail in detecting this advance phishing URLs. Due to the dependency which is growing on the online services of each user there is an immediate need of cybersecurity measures which needs to be implemented. (Tamal *et al.*, 2024)

As the machine learning techniques are been emerging as a very powerful tool for detection of Phishing URL which is offering a diverse solution that can help in analysis and also predict any malicious behavior based on various features like length of URL, how old the domain is, special characters present and also attributes that are host based. These studies demonstrate that high accuracy rates can be achievable by the use of machine learning models like Random Forest, KNN, Naïve Bayes, Support Vector Machine, Gradient Boosting and Multi-layer perceptron Classifier. Although there has been a remarkable improvement, but there a few numbers of important challenges that are still existing in the machine learning field based on phishing detection.

There is various work that are been done in this field which highlights a great advancement. Studies which are done have demonstrated the performance of different algorithms like Random Forest and Frameworks used like DARTH which helps in combining ML and Natural language Processing to get a very prefect metrics of detection.(Mittal *et al.*, 2022b)

But there are some gaps in these approaches. Methodologies used for feature selection vary widely during the studies which leads to inconsistent results and brings a limit to the findings. There is problem in scalability and distribution in the dataset.

Research Questions:

The goal of this study is to address these questions:

1. How can the false positive rates be reduced by tuning the phishing URL detection machine learning models while at the same time preserving high accuracy?
2. To know which of the machine learning model is the most well founded and effective in detecting the phishing URLs among all of the models like Random Forest, Naive Bayes, KNN, Decision Tree, Gradient Boosting, Multi-layer perceptron (MLP) classifier?
3. What will be effect of different feature selection method on their performance?

Objectives:

The main objectives of this research are as follows:

- To build and implement different models of machine learning for the detection phishing URLs and then evaluating it by using various feature sets.
- To evaluate and differentiate the performance of all the machine learning models which will include Random Forest, Naïve Bayes, KNN, Decision Tree, Gradient Boosting, Multi-layer perceptron (MLP) classifier in the form of their accuracy, precision, recall and the F1-score.
- To find out what is the impact of different feature selection methods like Correlation Matrix and Recursive Feature Elimination on the performance of each model.
- To uncover which one model among all the model is the best one which in future can be implemented in a real-world application of cybersecurity.

Contribution:

The following key contributions are been made in this project.

- A well-defined evaluation of all the machine learning models which includes Random Forest, Naïve Bayes, KNN, Decision Tree, Gradient Boosting and Multi-Layer Perceptron for the detection of phishing URL.
- Implementing some of the feature selection methods like Recursive Feature Elimination and correlation matrix.
- Running models with FS, without FS and Recursive Feature Elimination to do the analysis of what is the importance of feature and what were the tradeoffs done.
- Implemented a well-developed pipeline for the detection of Phishing URL by including steps like pre-processing data, feature engineering, training the models and the evaluation of it.
- Comparing the evaluation of three different dataset with which varies in the complexity and the structure of the it.

The aim of this study is to you the machine learning techniques to make a effective pipeline for the identification of phishing URLs. The aim of the approach that is recommended tries to go beyond the limits of the traditional methods by using the characteristics like metadata, sequential analysis and the structural data. With the goal to get an outstanding performance the study will also use machine learning models like Random Forest, K-nearest Neighbors, Decision Tree, Naïve Bayes, Gradient Boosting and Multi-layer Perceptron classifier.

2 Related Work

The study “A Comparative Analysis of Machine Learning Based Website Phishing Detection Using URL Information” mainly focuses on phishing detection using the machine learning techniques and the features of URL. The main features include URL length, some special characters and also host based detailed like the domain age and also the suspicious keywords. All the ML algorithms which were used like Logistic Regression, SVM, Random Forest, Gradient Boosting, KNN, and Navie Bayes the highest accuracy was achieved by Random Forest which was 98.7%. By combing the lexical and the host-based features help in improving the overall performance of the models. (Uddin *et al.*, 2022)

“A Machine Learning Based Approach for Detecting Phishing URL’s” this paper focuses on the cybersecurity threats that are been raising by the phishing attacks which tricks the user by presenting them the fraud links as legitimate to overcome this challenge the paper discovers different models of machine learning for identifying the URL’s which are phishing. The methodology which was used included extracting the features like the attributes and information about the domain and determining the overall performance of all the different machine learning algorithms like Logistic Regression, Decision Tree, Random Forest, SVM, K-Nearest Neighbors, Naïve Bayes, GBM and ANN. From all of this models Random Forest achieved the highest accuracy which describes the overall capability of it to handle the complex datasets. (Atari and Al-Mousa, 2022)

"An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning," a research paper which help to tackle the growing problem of web phishing attacks, where some of the fake websites tend to steal the private user data. SLA-FS demonstrated a very significant growth in both accuracy as well as efficiency with the help of an enhanced dataset of 11,430 samples of data which had different URL-based, HTML-based, and external features. Random Forest (RF), XGBoost (XGB), and k-Nearest Neighbors (kNN) are the three different ML models which were been used. Random forest was the one which had the highest accuracy 97.41% among all, followed by XGB of 97.21%, while kNN got 84.51%. SLA-FS can be used and also is an affordable approach which can be used in real world as it has improved its accuracy of detecting the Phishing URL. (Shafin, 2024)

A Framework known as DARTH, a multi model method is used for identifying the emails that are phishing with the help of Natural Language Processing (NLP) and Machine Learning (ML) models, as it is been presented in the paper "Phishing Detection Using Natural Language Processing and Machine Learning". As there is a growing complexity of phishing assaults which are being happening and the insufficient effectiveness of the traditional techniques like blocklists URL, the method which is used it utilizes the models which are based on neural networks so that it examines body text of the email, metadata, URLs, and attachments in it. The results that were obtained according to that, the combined models that were used which included the predictions from the given characteristics worked very well, better than the traditional single methods which had a wide margin with got an accuracy of 99.98%, precision of 99.97%, and F-Score of 99.98%. The study that has been done highlights the possibility of the analysis that is been done by the multi-layered for improving the detection of URL phishing and also highlights the important role of the metadata, which is mostly ignored in the traditional detection approaches. (Mittal *et al.*, 2022a)

"Phishing Detection Using NLP and Machine Learning" tries to address that Phishing emails try to indicated a very serious cybersecurity risk to all of them because they try to utilize

the techniques of social engineering so that the users are been tricked which will lead in disclosing the private information of the users. So, to overcome this kind of attacks which are sophisticated the research that is done in the paper tries to introduce a new framework named as DARTH, which is a diverse technique that combines the machine learning techniques and the Natural Language processing for the detection of the Phishing URL. As the traditional methods tries to focus on the main aspect like the analysis of the URL and the email content, the DARTH framework tries to integrate some of the features like the body text of the email, URL's which are embedded and also the metadata. The main techniques that are used include BERT for analyzing the "Masquerade-ness" and "Urgent-ness" of the email that is been received, whereas the neural networks which are been used for metadata and the evaluation of the URL's, also for the analysis of any attachments present in the mail. The model used by combing all the features got a result of 99.98% accuracy, precision and F1-score leaving the traditional methods behind. This kind of approach uncovers the importance of combining the multiple features so that there is an enhancement in the phishing detection and also there is a decrease in false positive and false negative alarms. The future work suggests that there can be an expand which can include multilingual dataset and also an advancement in the analysis for the more complex phishing patterns.(Mittal *et al.*, 2022b)

"The Role of Predictive Analytics in Cybersecurity: Detecting and Preventing Threats", this paper explores the use of forecasting the analysis which plays as a game changing technique for tackling the cyberthreats as they are becoming more and more complex, which may include ransomware attacks, phishing attacks and some advance threats. Some of the preventive measures are required as the traditional methods for this kind of attacks often fall short to address it. Some of the very important machine learning models which were used involve neural networks, decision tree, and support machine followed by deep learning models in it gave a very good accuracy in detecting this kind of complex and the attacks patterns that are evolving. In order to check the efficiency of that model the study that is been done gathers results from different research to highlight the methods used for data preparation, from the sources like network logs, activity of the user data and the evaluation of performance like the accuracy, precision, recall and the F1 score of it. As a way to improve these frameworks of cybersecurity the research that is been done ends by recommending some of the developments that can be done in the adaptive algorithms with the help of Blockchain and IOT and also analysis of real time. (Chowdhury *et al.*, 2023)

By using the Mutual Information technique for feature selection and Logistic Regression for the classification, the study that is been done in "Mutual Information-Based Logistic Regression for Phishing URL Detection" explores a very unique method for overcoming this kind rising danger of the phishing attack. With the use of PhiUSIIL dataset, the dataset which contains 100,945 phishing URLs in it and 134,850 legal URLs, the study that is done tries to implement a very intensive method which includes selection of features, data preprocessing, and training of model in it. URLSimilarityIndex, LineOfCode, NoOfExternalRef, NoOfImage, and NoOfSelfRef were the only five characteristics that were used and helped Logistic Regression model to get a very high accuracy of 99.97% accuracy. When the number of features were increased it was seen that there was a decrease in the performance, whereas this method had outperformed in some of the prior studies which were done and also with other models of machine learning like Random Forest and Gradient Boosting. At the end study says that the approach that was made was not highly accurate but is also practical for the applications of cybersecurity in real-world. In future there could be scope in expanding the dataset, uncovering different deep learning techniques and determining the robustness of this models in real world scenarios. (Vajrobol, Gupta and Gaurav, 2024)

“Phishing webpage Detection via Multi-Modal integration of HTML DOM Graphs and URL Features Based on Graphs Convolutional and Transformer Networks” this paper which is been proposed tries to uncover the increasing threats that are been cause by the phishing attacks and proposes a detection frame that combines HTML DOM graph with the analysis of URL features using deep learning techniques. The traditional methods which were used, they mainly focus on URL analysis which fails against the advance techniques of phishing. There are three advanced methods which are been used in the proposed models: Transformer Networks which tries to capture URL relationships at world level, Convolutional Neural Networks which evaluates characteristics like URL properties and Graph convolutional networks which extracts structural data from HTML DOM graphs. The models that were assembled achieved a great accuracy of 98.12%. The study that has been made mainly highlights the importance of integrating the data that is structural and sequential for effective detection of phishing. In future there can be more progress by expanding the datasets, exploring different real time applications, and also implementing data of user behavior. (Yoon, Buu and Kim, 2024)

The paper tried to discover different applications of machine learning in detection of phishing by highlighting its capability to make cybersecurity more enhanced. The work that is done by them is separated in different categories like supervised approach, unsupervised and hybrid approach. In supervised learning the use of algorithms like decision tree and neural networks in effective with the help of dataset which is labeled, whereas in unsupervised learning anomaly detection is used to detect zero-day attacks. And in Hybrid both the methods are been combined to improve its accuracy and adaptability. The paper tries to highlight the implementation natural language processing so that the phishing content can be analyzed and the deep learning models like CNN and RNN are used to process the data which is complex. Recommendation tries to address the real world challenges like the adversarial attacks and also the computational cost which is high for robust and scalable algorithm.(Huang *et al.*, 2019)

As in this evolving world phishing attack remains as a critical threat in cybersecurity which tries to exploit the vulnerabilities in both humans as well as technology. In this paper Brown and Johnson tries to examine what are the limitations of the traditional methods which were used for phishing detection also including systems that are ruled based, heuristic analysis and models of machine learning. They noted that the rule-based systems are effective but only for the threats that are known but they struggle with the phishing threats that are evolving. The authors mention that the attackers have become adaptable in bypassing all the frameworks that are static by altering email or URL altering. It mainly focusses on implementing a system which can be integrated with technological and strategies that are user focused in the framework. The study mainly highlights the need for advancing in the system of phishing detection by overcoming from the traditional methods. (‘(PDF) A REVIEW OF CYBERSECURITY STRATEGIES IN MODERN ORGANIZATIONS: EXAMINING THE EVOLUTION AND EFFECTIVENESS OF CYBERSECURITY MEASURES FOR DATA PROTECTION’, 2024)

3 Research Methodology

The approach which was been take to develop and implement a robust system for detection of Phishing URL is been demonstrated in this part.
The study which is been made it makes the use of three datasets which are different from each other, in which each of the dataset includes URLs which are classified as either legitimate one or as phishing URL, in order to make sure that the evaluation is done

properly of the proposed method. As the datasets may differ in many of the terms like the size of the dataset, features and the complexity of the dataset, it is easy to compare all the machine learning models and the strategies for feature selection. The main aim of his methodology is to develop a system which can detect the phishing URL.

1. Data selection:

In total three datasets were been chosen to make sure there was a diversity in the structure of the dataset, size and the complexity of the dataset. Which will help in making a very comprehensive evaluation of the methodology which is been proposed. After the data selection dataset was been loaded using the panda's framework for data. After the loading of dataset was done the structure of it was seen such as the names of the columns, few rows from the top and the overall values how they were distributed.

Dataset 1- This dataset has 11,430 URLs with extracted features counted till 87. It has a balance data of 50% phishing URL and 50% Legitimate URL.(Hannousse and Yahiouche, 2021)

Dataset 2- This dataset has 41 features and there is only one target variable. There are total 247950 instances from which 128541 are phishing and the remaining are legitimate.(Tamal, 2023)

Dataset 3- The URLs present in this data are latest URLs. Some of the features present in this data are been derived from the existing features. (Prasad and Chandra, 2024)

Summary Table:

Dataset	Total URLs	Phishing URLs	Legitimate URLs	Features
Dataset 1	11430	5715	5715	87
Dataset 2	247950	128541	119409	41
Dataset 3	235795	134850	100945	54

Table 1- Dataset Summary Table

2. Data Pre-Processing

Pre-processing of the data was done to make sure that the quality and consistency of the data is been managed across the whole dataset. Following are the steps which were followed:

2.1 Data Cleaning:

Cleaning of the Data is very important step in the preprocessing data pipeline, to make sure that the data is ready for the accurate analysis. This data cleaning process involves finding out and fixing the mistakes that are present, problems present in the dataset.

2.2 Managing the missing Values:
Depending on if there were any missing values were then identified and solves depending on how frequently they were occurring and also how it had an effect on the dataset.

2.3 Dropping duplicate columns:
All the columns which were not of use like the URL columns which had no contribution in the task carried out of phishing detection were all dropped.

2.4 Finding Exceptions:
The present Statistical techniques like the box plots and z-scores were been used to find out the exceptions. It all Depended on how they were affecting the whole of the dataset, all of these exceptions were either been removed or limiting was been applied to it.

3. Transformation

Converting the data to its correct form so that the machine learning models can process it as well as extracting the features to optimize the machine learning models.

3.1 Feature Engineering

3.1.1 Standardization:

The standard scaling algorithm known as Standard Scaler was implemented for standardizing the numerical features which were present. So that to make sure that every feature which were present were been transformed and scaled to have a median of 0 and an average of 1, the scaler algorithm was implemented in the training data which was used and after that was then was implemented in the dataset of training and testing.

3.1.2 Discretization:

To convert all the constant variables which were target into a specific discrete class KBinsDiscretizer was implemented.

3.1.3 Factorization:

To encode all the categorical variables which were target factorize command was used.

3.2 Feature Selection

3.2.1 Analysis of Correlation:

A method named as Pearson was been used in the implementation to determine the overall correlation between all the features. To reduce the complexity between the data, features which had a strong correlation above 0.7 threshold were gradually being eliminated.

3.2.2 The Chi-Square Test:

To determine the overall statistical relationship between the categorical features and the target variable, a chi-square test was been carried out. All the features which had low importance during the test were not been included.

3.2.3 Recursive feature elimination:

Was been used by implementing Random Forest Classifier. Which selected the features which were top 10 for the training of the model.

3.3 Data Mining

3.3.1 Data Partitioning

Dividing the data in two sets one for training and the other for testing is a very important step in the pipeline of machine learning, as it will make sure that the machine learning model has been tested on the data that is unseen so that it can provide a fair evaluation on its evaluation. Features which are named as x and target values which are named as y were been separated out of the dataset(df).

a. The targeted variables (y) and the features(x) were been separated from the dataset.
b. The data was been split into subsets of training and testing. These functions allocated 70% of the data for training set and the remaining 30% to the testing set.

3.3.2 Training Model

Random Forest:

It is a type of ensemble learning method which can build up numerous decision tree and can combine the output of all to make a prediction that can be robust.

Decision Tree:

This model offers a interpretability by visualizing the paths in advance of decision making.

Naïve Bayes:

This model is very much effective for the data which is categorical as well as this model is probabilistic.

K-Nearest Neighbors:

This model is very much simple and at the same time is very effective in doing the recognition of the patterns

Gradient Boosting:

It is a ensemble technique of machine learning which helps in building the decision tree in a series and each of it is aimed on correcting the errors which occurred on the previous ones.

Multi-Layer Perceptron (MLP):

It is a neural network-based classifier that mainly uses backpropagation to optimize the weights.

3.4 Evaluation:

1. Accuracy was been calculated using the accuracy score
2. Precision, Recall and the F1-score was been generated for a very detailed information about the performance of the models.
3. Confusion matrix was been implemented and was visualized using the heatmap for all the interpretation of true positives, true negatives, false positives and also false negatives.
4. To illustrate the performance of each model an ROC curve was been drawn for each model.
5. A bar chat was generated for tree-based models which showed which were the most influential variables that were present.

Implementation	Feature Selection Method	Process
Implementation 1	Correlation Matrix and Chi-Square Test	Eliminating highly correlated features
Implementation 2	No Feature Selection; Standardization Only	Standardizing all the features to a common scale
Implementation 3	Recursive Feature Elimination + Discretization + Factorization	RFE to selecting top features + convert all the variables + encoding all the categorical variables

Table 2- Feature selection Implementation

4 Design Specification and Implementation

The design of the system for the project "URL Phishing detection" it mainly focuses on building a very robust pipeline which can easily and effectively process all the data inputted, apply the machine learning models and also give very actionable results as the output. The design which is been made it uses advance techniques for cleaning the data, pre-processing, training the model and for evaluation which will ensure that there is a systematic way to overcome the problem of detecting the phishing URL.

The aim of this project is to build a system which detect the phishing URL with a high accuracy by using machine learning models in it. The design of this system makes sure that it can handle large dataset of URL's. The system which is been built tries to identify the characteristics of the URL that is contributing to the phishing activity, which helps in enhancing the measures of cybersecurity.

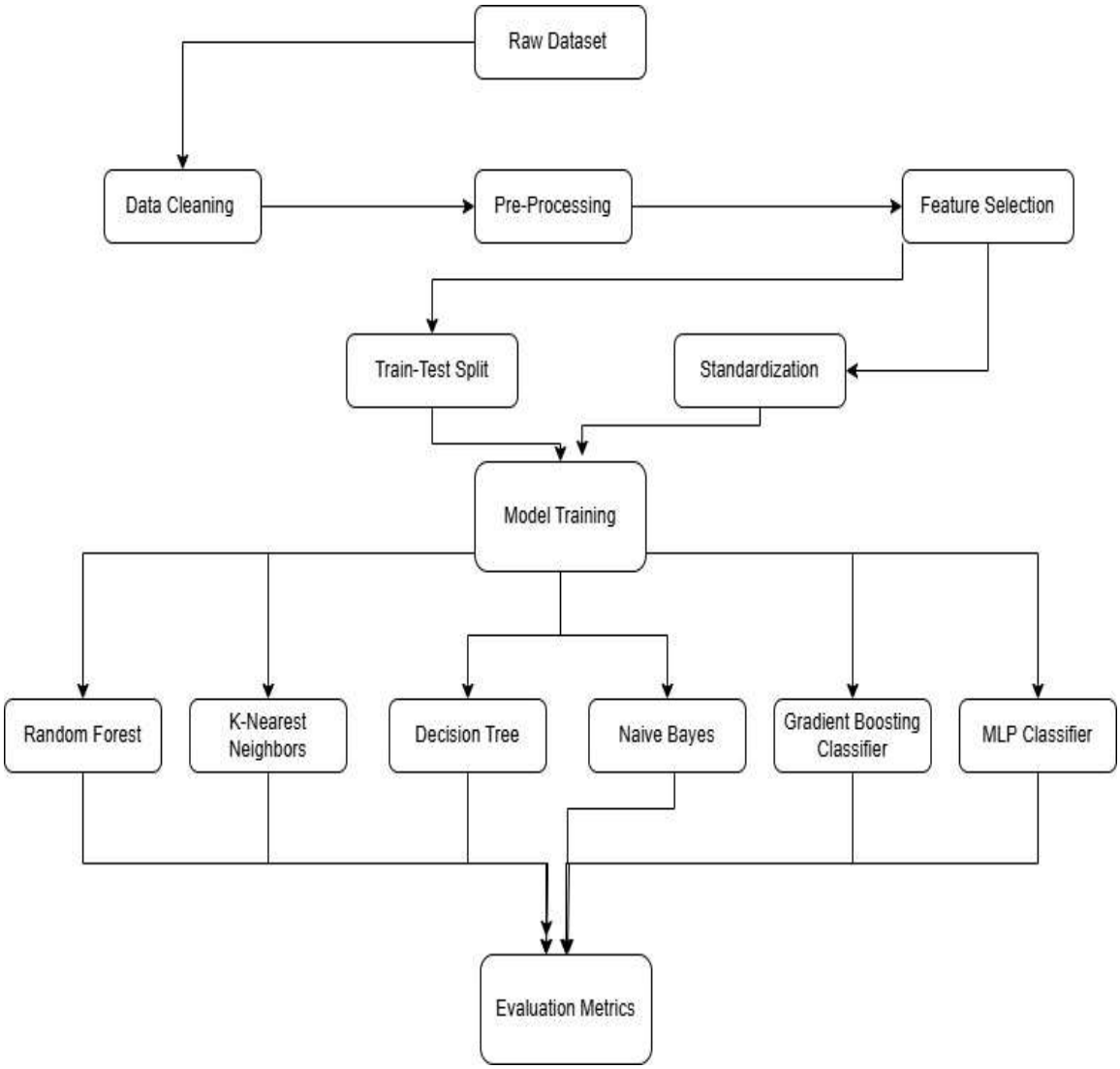

Fig-1 System Work Flow

Category	**Tools, Framework & Languages**	**Purpose**
Programming Language	Python	Essential programming language for creating systems.
Modifying data	Pandas	Loading, cleaning, transforming, and analyzing datasets.
	NumPy	For calculating numbers, especially arrays and numerical data.
Data Preprocessing	Scikit-learn	Handling missing values, one-hot encoding,

		standardization, and RFE.
Feature Engineering	Scikit-learn	Feature selection using Recursive Feature Elimination (RFE).
Machine Learning	Scikit-learn	Implementing machine learning models like Random Forest, Gradient Boosting, MLP classifier, etc.
Model Evaluation	Scikit-learn	Calculation of metrics such as accuracy, precision, recall, F1-score, etc.
Data Visualization	Matplotlib	Creating visualizations like bar charts, ROC curves, and feature plots.
	Seaborn	Enhanced visualizations such as heatmaps for confusion matrices.
Data Storage/Import	CSV Format	Storing and loading raw datasets for the project.

Table 3- Tools Used

4.1 Implementation Details:

4.1.1 Setup of Pipeline:

- A very defined pipeline was been created that included cleaning of the data, feature extraction from the data, scaling, and different model training was been used to process each of the dataset.
- All of the models were trained by using the default parameters at the initial stage with a minimal of tunning that made sure that consistency stays across the datasets.

4.1.2 Replication and Automation

- Python was been used to automate implementation that was been done, by using the libraries like Scikit-learn, pandas and matplotlib.
- Reproducibility of the results was guaranteed across the dataset by the procedure of standard preparation.

4.2 Pipe line Steps:

4.1.1 Data Ingestion:

Dataset: The input that is been done in the system consist of Dataset which has URLs in it with their allocated features like the length of URL, special characters present in it and labels of it if phishing or legitimate.

Data intake section: This section handles with loading of the data and initially validating it. It ensures the adaptability if the system by standardizing the inputs in the form of CSV or Json file.

4.1.2. Pre-processing:

In this the raw data is been prepared for machine learning by cleaning the data, transforming it and then optimizing it.

Data Cleaning: It removes all the irrelevant and the unnecessary data which has no contribution in the process. It handles with the values which are missing by putting the means in the numerical columns and modes for all the categorical columns.

Converting all the categorical data into numerical data by using one-hot encoding or factorization method.

4.1.3. Feature Engineering and Selection

Feature Selection: Implementing Recursive feature Elimination method to identify and remove the feature which are most relevant and also to reduce the computational complexity.

Feature engineering: extracting all the meaning full features like URL length, suspicious TLDs, presence of any '@' symbol and number of any special characters and dots present.

4.1.4 Training and Testing of Model:

In this the machine learning algorithms are been trained and then the models are been evaluated.

Train-Test Split: The dataset which is been used is been split in two different parts of ratio. 70% ratio of the data is been used for training and the remaining 30% is used for testing.

Training Model: Multiple machine learning models are been trained:

a. Random Forest
b. K-Nearest Neighbors
c. Naïve Bayes
d. Decision Tree
e. Gradient Boosting
f. Multi-layer Perceptron (MLP) classifier.

Each of the layer which is been trained is optimized through the hyperparameter so that the performance is improved.

4.1.5 Evaluation and Visualization:

a. Performance Metrics: Key metrics are been calculated such as accuracy, precision, recall, F1-score and ROC-AUC.
b. Analysis of Confusion Matrix: When displaying the classification performance, it highlights false positive, false negative, True positives as well as Ture negatives.
c. Visualization Tools: ROC curves are been generated, bar charts are created for comparison of model and for confusion matrices heatmaps are created. (*scikit-learn: machine learning in Python — scikit-learn 1.6.0 documentation*, no date)
d. Results: Gives a comprehensive result of the output which includes performance of the model and visualization.

5 Evaluation

A very important step in machine learning that helps you to find out what are the successful outcomes of the algorithms and also to resolve any of the particular issue that is present in the inspection of the model. Metrics that are used to evaluate the performance of the various models and their combinations are accuracy, precision, recall and the F1-score of it. By this metrics which are generated it shows the insights on how good the model worked, how good it handles the dataset and generalize to the data that was unseen.

The two main methods used in this research to uncover the evaluation is filter based feature selection wherein it finds outs and removes features which are of least use and have no contribution in the performance and the other is Recursive Feature Elimination (RFE) a different approach which Selects the top 10 features which are very much important for the evaluation and eliminates the remaining ones.

5.1 Results of Dataset 1

Model	Feature Selection Method	Accuracy	Precision	Recall	F1-score
Random Forest	No Feature Selection	0.967	0.967	0.967	0.967
Random Forest	Correlation Matrix	0.965	0.965	0.965	0.965
Random Forest	Recursive Feature Elimination	0.952	0.95	0.95	0.95
KNN	No Feature Selection	0.820	0.821	0.821	0.820
KNN	Correlation Matrix	0.825	0.825	0.825	0.825
KNN	Recursive Feature Elimination	0.935	0.94	0.94	0.94
Decision Tree	No Feature Selection	0.932	0.932	0.932	0.932
Decision Tree	Correlation Matrix	0.932	0.932	0.932	0.932
Decision Tree	Recursive Feature Elimination	0.926	0.93	0.93	0.93
Naïve Bayes	No Feature Selection	0.734	0.743	0.735	0.732
Naïve Bayes	Correlation Matrix	0.726	0.746	0.728	0.722
Naïve Bayes	Recursive Feature Elimination	0.90	0.91	0.91	0.91
Gradient Boosting	No Feature Selection	0.9647	0.9648	0.964	0.964
Gradient	Correlation	0.9644	0.9645	0.9644	0.9644

Boosting	Matrix				
Gradient Boosting	Recursive Feature Elimination	0.955	0.96	0.96	0.96
MLP	No Feature Selection	0.734	0.786	0.736	0.722
MLP	Correlation Matrix	0.722	0.774	0.724	0.709
MLP	Recursive Feature Elimination	0.939	0.94	0.94	0.94

Table 4- Dataset 1 Results

5.2 Results for Dataset 2

Model	Feature Selection Method	Accuracy	Precision	Recall	F1-score
Random Forest	No Feature Selection	0.953	0.953	0.953	0.953
Random Forest	Correlation Matrix	0.9622	0.9625	0.9619	0.962
Random Forest	Recursive Feature Elimination	0.9598	0.96	0.96	0.96
KNN	No Feature Selection	0.9016	0.902	0.9009	0.9013
KNN	Correlation Matrix	0.908	0.9084	0.9075	0.9078
KNN	Recursive Feature Elimination	0.9062	0.91	0.91	0.91
Decision Tree	No Feature Selection	0.9404	0.9405	0.9401	0.9403
Decision Tree	Correlation Matrix	0.945	0.945	0.9455	0.9455
Decision Tree	Recursive Feature Elimination	0.943	0.94	0.94	0.94
Naïve Bayes	No Feature Selection	0.738	0.739	0.730	0.7211
Naïve Bayes	Correlation Matrix	0.738	0.774	0.7311	0.7252
Naïve Bayes	Recursive Feature Elimination	0.7502	0.78	0.74	0.74
Gradient Boosting	No Feature Selection	0.9023	0.9035	0.9013	0.9019

Gradient Boosting	Correlation Matrix	0.9081	0.9091	0.9073	0.9078
Gradient Boosting	Recursive Feature Elimination	0.9057	0.91	0.99	0.91
MLP	No Feature Selection	0.884	0.888	0.8831	0.884
MLP	Correlation Matrix	0.8991	0.8891	0.8989	0.899
MLP	Recursive Feature Elimination	0.8944	0.9	0.89	0.89

Table 5- Dataset 2 Results

5.3 Results for Dataset 3

Model	Feature Selection Method	Accuracy	Precision	Recall	F1-score
Random Forest	No Feature Selection	1.0	1.0	1.0	1.0
Random Forest	Correlation Matrix	1.0	1.0	1.0	1.0
Random Forest	Recursive Feature Elimination	0.9999	1.0	1.0	1.0
KNN	No Feature Selection	0.9966	0.9968	0.9962	0.996
KNN	Correlation Matrix	0.9966	0.9968	0.9962	0.996
KNN	Recursive Feature Elimination	0.9997	1.0	1.0	1.0
Decision Tree	No Feature Selection	1.0	1.0	1.0	1.0
Decision Tree	Correlation Matrix	1.0	1.0	1.0	1.0
Decision Tree	Recursive Feature Elimination	0.9998	1.0	1.0	1.0
Naïve Bayes	No Feature Selection	0.9826	0.9805	0.9848	0.982
Naïve Bayes	Correlation Matrix	0.9684	0.9655	0.9724	0.968
Naïve Bayes	Recursive Feature Elimination	0.9955	0.99	0.99	0.99
Gradient Boosting	No Feature Selection	1.0	1.0	1.0	1.0

Gradient Boosting	Correlation Matrix	1.0	1.0	1.0	1.0
Gradient Boosting	Recursive Feature Elimination	0.9998	1.0	1.0	1.0
MLP	No Feature Selection	0.9996	0.9996	0.9995	0.9996
MLP	Correlation Matrix	0.9985	0.9985	0.9983	0.9984
MLP	Recursive Feature Elimination	0.9999	1.0	1.0	1.0

Table 6- Dataset 3 Results

Results Of All the Dataset:

Random Forest:
Among all the datasets, Random Forest was found out to be the most performing model for repeatedly achieving a great accuracy.
The system without feature selection and with feature selection configuration worked almost similarly in most of the cases, which suggested that the model can work very well with irrelevant and unnecessary features if present.
In Recursive Feature Selection method, the accuracy for Dataset 1 and 3 was decreased minimally but had strong performance on Dataset 2, which indicated that there is not much need in the dimension reduction in a very partitioned data for Random Forest.

5.4 Discussion

Dataset 1:
The complexity of Dataset 1 was moderate for the simpler models which was challenging for the models. Models like logistic regression and basic neural networks couldn't capture the nature of the data. But the ensemble models like random forest and Gradient Boosting had a great performance as it combined many weak learners effectively. However, feature selection was enough for the robust classifier.

Dataset 2:
The structural feature of URL was dominated with the help of Random Forest and Gradient Boosting which helped them in performing the best. Variation in the neural networks may need some more preprocessing and tuning. This indicates that the ensemble can perform very well when out of the zone but at the same time neural networks may need some more adjustments to achieve similar performance.

Dataset 3:

All models which were implemented performed very well due to the high separation and the content-focused features, with Random Forest achieving almost the perfect accuracy. Which tends to say that due to well-structured data and the strength of features it helped in effective learning. As every model got high performance it indicates that the characteristics of that dataset was favorable towards the accuracy.

5.4.1 Selection of feature and reduction of dimensionality:

The computing efficiency and the performance was balanced across the dataset by RFE. Without sacrificing the accuracy, the reduction that was done in dimensionality it greatly improved the ensemble techniques, especially in the case of Random Forest and Gradient Boosting.

5.4.2Impact of correlation matrix

In the case of robust classifiers, removing the most strongly correlated features helped maintained the competitive performance while improving generality at the same time for the simpler models.

Although adding the linked characteristics had increased the overall accuracy, but there was a chance of overfitting which could have happened.

5.4.3Difficulties and Restrictions

Unbalanced Data: some of the Simpler models had an impact due to the class imbalances present in some of the datasets.

Risks of Overfitting: There was a need for an external validation which was shown by the 3 dataset near-perfect accuracy, which tries to points on overfitting which was caused by the separable data.

Hyperparameter Tuning: In order to go ahead with head with the ensemble approaches, MLP had to be very significantly tuned.

5.4.4Real World Applications

Models like Random Forest and Gradient boosting are the well-suited models which can be deployed in the systems which are used in real world, as it offers high accuracy and many robust feature importance insights.

RFE and some of the other feature selection methods that can help the domain experts prioritize some important predictors, which will help in improving the decision-making in applications like as preventions from the fraud and detection of phishing attacks.

5.4.5 Role of Feature Selection Technique

Without Feature Selection:

Without the feature selection technique, it provided the best performance in all of the majority scenarios for Random Forest and Gradient Boosting, which showed that these models can seamlessly handle a large set of features.

MLP struggled and performed poorly in Dataset 1 and 2 without the feature selection, which demonstrated that there is a need to reduce the features so that performance is maximized.

With feature Selection:

When the feature selection was implemented, there was a slight decrease in the accuracy in several cases, which suggested that removing the features may some eliminate some of the relevant features which are the predictors, this occurred only for Random Forest and Gradient Boosting.

In the case of MLP it showed mixed results, but there were some minor improvements in some of the cases but it gave consistent underperformance as compared to the Recursive Feature Elimination.

Recursive Feature Elimination:

It proved to be very effective in getting a balance between the accuracy and the computational economy, especially for the MLP model, which produced a very noticeable improvements in its performance for 3 datasets.

It maintained a accuracy which was very competitive for both the ML models i.e. Gradient Boosting and Random Forest it proved that MLP model can also be a very good option for datasets which are of high dimensions.

6 Conclusion and Future Work

6.1 Performance of Models:

- Among all of the datasets, Random Forest demonstrated that it is the most trustworthy and stable model, it continuously achieved a very balanced metrics and good accuracy at the same time. The adaptability of this model can be seen by the capacity of it to handle the both reduced and high-dimensional feature sets.

- Random Forest was the one model which was the most effective model which emerged as the highest achieving accuracy model across all the 3 datasets where as in some scenarios gradient boosting models also performed very well as specially while interacting with complex features.

- Gradient boosting at the same time gave a performance which was comparable, particularly performing very well in the dataset which had complex interactions of features.

- Potential of MLP classifier was been seen in dataset 3 where the features were content focused and were dominant. It nevertheless managed to show an inconsistency in the datasets, requiring further an hyperparameter adjustment which was to be made.

- Recursive Feature Elimination method which was used for feature selection helped in balancing by reducing the complexity of the model while not compromising the accuracy.

6.2 Usefulness of Recursive Feature Selection method:

All datasets underwent through the Recursive feature elimination method, which indeed improved the understanding of the dataset as well as decreased its complexity. All the Important findings include:

- Dataset 1: The top ten selected features were enough for the models for it to perform very well enough like the Random Forest model and Gradient Boosting which highlighted the complexity of the dataset was moderate.
- Dataset 2: The Recursive Feature Elimination method which was implemented evaluated 10 important features which were related to URL structure and the domain characteristics, which helped in enhancing the classification power of the models.
- Dataset 3: RFE improved its computational productivity while maintaining an high accuracy at the same time, while metrics based on content like the URL Similarity Index was very dominant.

6.3 Results of Correlation matrices:

- Some of the Highly correlated characteristics may lead to the duplication and may have an impact on the interpretability of the model, of the code that use correlation matrices. For some of the certain models, by removing the strongly features that are correlated may have enhanced the generalizability and the clarity of the feature but marginally the accuracy was been decreased.

6.4 Metrics for Evaluation:

- In each and every dataset that was used, the most important parameter was accuracy. More of the comprehensive insights were been offered by precision, recall, and F1-scores, especially for the datasets which were unbalanced.

Future Work

There are still some parts in this project which are not uncovered and need to be worked on, even if for this project of Phishing URL detection using techniques of machine learning the research done provides as strong base.

1. Improving the Datasets Volume and Diversity:
   Adding international, language and or any region-specific URLs to the dataset might help in developing more effective models which can be used internationally. At the same time adding some real-time data from some other sources which will help directly the system to become more generalizable.
2. Detection in Real Time and Flexibility:
   Building and implementing a system which detect phishing URLs in real time with a least delay. In order to adjust the models dynamically with new phishing strategies that are been developed there should be an incremental learning which should be made.
3. Discovering Hybrid Models:
   Building up a model where in the machine learning models re combined with the rule-based systems for a hybrid approach which can help in reducing false positives while the high accuracy is been maintained.

References

Atari, M. and Al-Mousa, A. (2022) 'A Machine-Learning Based Approach for Detecting Phishing URLs', in *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA). 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 82–88. Available at: https://doi.org/10.1109/IDSTA55301.2022.9923050.

Chowdhury, R. *et al.* (2023) *The role of predictive analytics in cybersecurity: Detecting and preventing threats*. Available at: https://doi.org/10.30574/wjarr.2024.23.2.2494.

Hannousse, A. and Yahiouche, S. (2021) 'Web page phishing detection', 3. Available at: https://doi.org/10.17632/c2gw7fy2j4.3.

Huang, Y. *et al.* (2019) 'Phishing URL Detection via CNN and Attention-Based Hierarchical RNN', in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 112–119. Available at: https://doi.org/10.1109/TrustCom/BigDataSE.2019.00024.

Mittal, A. *et al.* (2022a) 'Phishing Detection Using Natural Language Processing and Machine Learning', *SMU Data Science Review*, 6(2). Available at: https://scholar.smu.edu/datasciencereview/vol6/iss2/14.

Mittal, A. *et al.* (2022b) 'Phishing Detection Using Natural Language Processing and Machine Learning', 6(2).

'(PDF) A REVIEW OF CYBERSECURITY STRATEGIES IN MODERN ORGANIZATIONS: EXAMINING THE EVOLUTION AND EFFECTIVENESS OF CYBERSECURITY MEASURES FOR DATA PROTECTION' (2024) *ResearchGate* [Preprint]. Available at: https://doi.org/10.51594/csitrj.v5i1.699.

Prasad, A. and Chandra, S. (2024) 'PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning', *Computers & Security*, 136, p. 103545. Available at: https://doi.org/10.1016/j.cose.2023.103545.

scikit-learn: machine learning in Python — scikit-learn 1.6.0 documentation (no date). Available at: https://scikit-learn.org/stable/ (Accessed: 12 December 2024).

Shafin, S.S. (2024) 'An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning', *Data Science and Management* [Preprint]. Available at: https://doi.org/10.1016/j.dsm.2024.08.004.

Tamal, M. (2023) 'Phishing Detection Dataset', 1. Available at: https://doi.org/10.17632/6tm2d6sz7p.1.

Tamal, M.A. *et al.* (2024) 'Unveiling suspicious phishing attacks: enhancing detection with an optimal feature vectorization algorithm and supervised machine learning', *Frontiers in Computer Science*, 6. Available at: https://doi.org/10.3389/fcomp.2024.1428013.

Uddin, Md.M. *et al.* (2022) 'A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information', in *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pp. 220–224. Available at: https://doi.org/10.1109/PRAI55851.2022.9904055.

Vajrobol, V., Gupta, B.B. and Gaurav, A. (2024) 'Mutual information based logistic regression for phishing URL detection', *Cyber Security and Applications*, 2, p. 100044. Available at: https://doi.org/10.1016/j.csa.2024.100044.

Yoon, J.-H., Buu, S.-J. and Kim, H.-J. (2024) 'Phishing Webpage Detection via Multi-Modal Integration of HTML DOM Graphs and URL Features Based on Graph Convolutional and Transformer Networks', *Electronics*, 13(16), p. 3344. Available at: https://doi.org/10.3390/electronics13163344.