A Project report on

Malicious URL detection based on machine learning

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

NAMPALLY SIDDHARTHA
(21H51A0591)

BASHAM RAJU
(21H51A05D3)

MOHAMMED ABDUL SAMEER
(21H51A0514)

Under the esteemed guidance of

Mr. B. SIVAIAH
(ASSOCIATE PROFESSOR)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A$^+$ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2021- 2025

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase I report entitled **"Malicious URL detection based on machine learning"** being submitted by N. Siddhartha (21H51A0591), B. Raju (21H51A05D3), Md. Abdul Sameer (21H51A0514) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. B. Sivaiah
Associate Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure, we want to take this opportunity to express my heartfelt gratitude to all the people who helped make this project a grand success.

We are grateful to **Mr. B.Sivaiah, Associate Professor** , Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving force to complete my project work successfully.

We are very grateful to **Dr. J Rajeshwar**, Dean of CS at CMR College of Engineering and Technology, for his constant support and motivation in successfully completing the project.

We are highly indebted to **Major Dr. V A Narayana,** Principal of CMR College of Engineering and Technology, for allowing us to carry out this project successfully and fruitfully.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

N. Siddhartha	21H51A0591
B. Raju	21H51A05D3
Md. Abdul Sameer	21H51A0514

TABLE OF CONTENTS

List of Figures

ABSTRACT

Currently, the risk of network information insecurity is increasing rapidly in number and level of danger. The methods mostly used by hackers today is to attack end-to-end technology and exploit human vulnerabilities. These techniques include social engineering, phishing, pharming, etc. One of the steps in conducting these attacks is to deceive users with malicious Uniform Resource Locators (URLs). As a results, malicious URL detection is of great interest nowadays. There have been several scientific studies showing a number of methods to detect malicious URLs based on machine learning and deep learning techniques. In this paper, we propose a malicious URL detection method using machine learning techniques based on our proposed URL behaviors and attributes. Moreover, bigdata technology is also exploited to improve the capability of detection malicious URLs based on abnormal behaviors. In short, the proposed detection system consists of a new set of URLs features and behaviors, a machine learning algorithm, and a bigdata technology. The experimental results show that the proposed URL attributes and behavior can help improve the ability to detect malicious URL significantly. This is suggested that the proposed system may be considered as an optimized and friendly used solution for malicious URL detection.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1.Problem Statement:

The rise of cyber threats, particularly through malicious URLs used for phishing, malware distribution, and other cyber attacks, poses a significant risk to individuals and organizations alike. Traditional URL detection methods, such as blacklists and rule-based systems, are limited in their ability to keep pace with the rapid evolution and volume of new malicious URLs. These conventional approaches often suffer from high false positive rates, are resource-intensive to maintain, and lack the adaptability to detect newly crafted or obfuscated URLs that may bypass existing security measures.

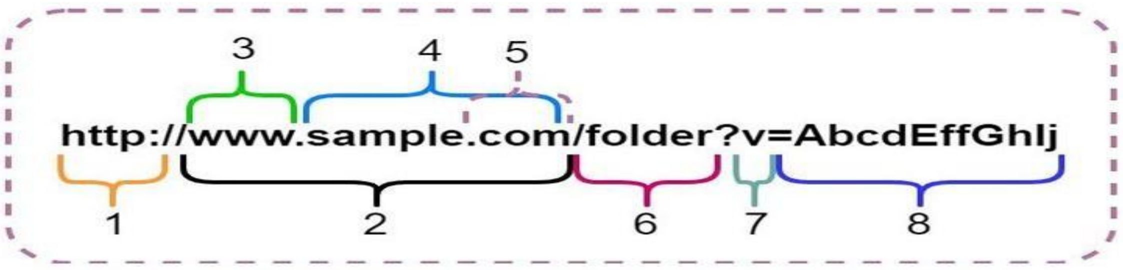

Figure: 1.1-URL

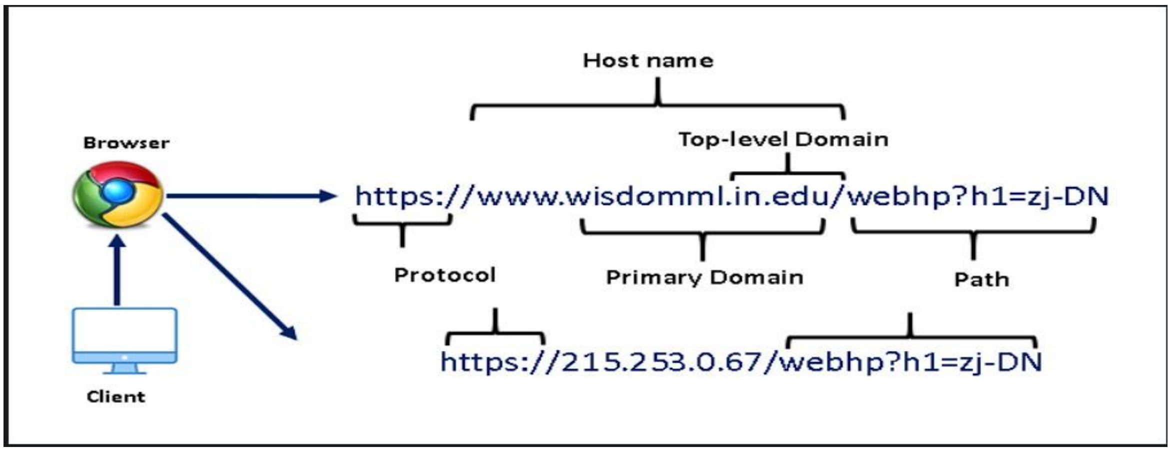
Host name
Browser
Top-level Domain
https://www.wisdomml.in.edu/webhp?h1=zj-DN
Protocol
Primary Domain
Path
https://215.253.0.67/webhp?h1=zj-DN
Client

Figure:1.1.2- Detecting URL

1.2 Research Objective:-

The objective of this research is to develop an effective, scalable, and adaptive machine learning-based framework for detecting malicious URLs. This study aims to investigate and evaluate various machine learning models to determine the most effective algorithms for accurately identifying malicious URLs based on lexical, host-based, network-based, behavioral, content, and historical features. Additionally, the research seeks to explore the optimal set of features that most significantly contribute to the accurate classification of URLs as either benign or malicious. By analyzing real-world URL data, this project strives to achieve a high detection accuracy with minimal false positives, enabling proactive protection against evolving cyber threats.

1. **Evaluate Machine Learning Models**: Investigate and compare the effectiveness of various machine learning models (such as logistic regression, decision trees, support vector machines, and deep learning algorithms) to determine the most suitable algorithms for accurately identifying malicious URLs.
2. **Identify Key Features**: Analyze different types of features—lexical, host-based, network-based, behavioral, content, and historical—to identify which combinations of these attributes contribute most significantly to high-accuracy classification between benign and malicious URLs.
3. **Optimize Feature Engineering**: Explore feature engineering and selection techniques to improve model performance and efficiency by focusing on attributes that enhance predictive accuracy and reduce false positives.
4. **Adapt to Evolving Threats**: Develop methods that enable the framework to adapt to emerging threats by detecting newly crafted or obfuscated URLs, which may bypass traditional detection systems.
5. **Ensure Scalability and Real-Time Performance**: Design a scalable solution capable of operating in real-time, ensuring that it can handle high volumes of URL data without compromising detection accuracy or speed, making it suitable for integration into existing cybersecurity infrastructure.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2
BACKGROUND WORK

2.1. Detection of Malicious URLs using an Efficient Feature-Based Machine Learning Framework

2.1.1 Introduction:

In the modern digital landscape, the internet has become an integral part of everyday life, providing unprecedented access to information, services, and social connections. However, this vast digital ecosystem also exposes users to a growing number of cybersecurity threats, particularly through malicious URLs. These URLs are commonly used in phishing attacks, malware distribution, and various other forms of cybercrime, putting individuals, businesses, and organizations at significant risk. As cybercriminals continue to evolve their techniques, traditional methods of detecting malicious URLs, such as blacklisting or rule-based systems, struggle to keep up with the dynamic nature of online threats.

The emergence of machine learning has opened up new possibilities for tackling the challenges of malicious URL detection. Unlike traditional approaches, machine learning models can analyze large datasets of URLs, learning complex patterns from various features (such as lexical, host-based, and behavioral attributes) to distinguish between benign and harmful links. This allows for a more adaptive and scalable detection system.

2.1.2 Merits , Demerits and Challenges:

Merits

1. **Enhanced Detection Accuracy**: By leveraging machine learning models, the project enables more accurate identification of malicious URLs compared to traditional methods.
2. **Adaptability to New Threats**: Traditional security systems often rely on static rules or blacklists, which can be easily bypassed by new or obfuscated malicious URLs. This project incorporates machine learning algorithms.
3. **Scalability**: The machine learning-based framework is designed to handle large volumes of URLs in real-time, making it scalable for widespread deployment in various cybersecurity systems.

Demerits

1. Dependency on Data Quality: The accuracy and performance of the machine learning models heavily depend on the quality and diversity of the dataset used for training. If the dataset contains biased or incomplete data, the model may struggle to detect certain types of malicious URLs, leading to false negatives or reduced generalization to new attack patterns.
2. Computational Complexity: Training and deploying machine learning models, especially deep learning models, can be computationally intensive. This might require significant hardware resources and could pose challenges for real-time URL detection in environments with limited computational power or in large-scale deployments.
3. Adaptation to Novel Attacks: While machine learning models are adaptive, the system may initially struggle to detect brand new or highly obfuscated malicious URLs that differ significantly from previously seen patterns. These types of attacks can take time to recognize and may lead to a temporary decrease in detection accuracy.

Challenges

1. **Handling Obfuscated URLs**: One of the primary challenges is dealing with obfuscated or camouflaged URLs. Attackers often use techniques like encoding, randomization, or URL shortening to hide malicious intent. These methods can make it difficult for traditional feature-based systems to recognize malicious URLs, requiring advanced techniques and frequent model updates to stay ahead of evolving tactics.
2. **Balancing Accuracy and Performance**: Achieving a balance between high detection accuracy and low latency is challenging. While more complex machine learning models (such as deep learning) may provide better accuracy, they often require more computational resources and time to process URLs, which could hinder real-time detection in high-traffic environments.
3. **Data Imbalance**: The dataset used for training may be imbalanced, with a higher number of benign URLs compared to malicious ones. This imbalance can lead to biased models that perform poorly in detecting malicious URLs, as the model may become biased toward predicting benign URLs, leading to an increase in false negatives.

2.1.3 Implementation:

The implementation of the malicious URL detection system using machine learning involves several key stages, from data collection and preprocessing to model training, evaluation, and real-time deployment. Below is an outline of the steps involved in building and deploying the system:

1. Data Collection

- **URL Dataset**: Collect a large, diverse dataset of URLs that includes both benign and malicious URLs. These datasets can be obtained from open sources like publicly available phishing URL datasets (e.g., PhishTank, Alexa Top Sites) or by scraping URLs from various websites.
- **Feature Extraction**: For each URL, extract a wide range of features that can aid in classification. Common features include:

    Lexical Features: URL length, presence of special characters, use of suspicious keywords (e.g., "login", "free", "verify").

    Host-Based Features: Domain age, WHOIS information, registrar details, IP address reputation.

    Behavioral Features: Number of redirects, traffic patterns, SSL certificate presence.

    Network-Based Features: Geolocation of server, IP reputation, and DNS resolution time.

    Content-Based Features: Presence of scripts, forms, and suspicious content in the webpage linked by the URL.

2. Data Preprocessing

- **Data Cleaning**: Handle missing values, remove duplicates, and ensure data consistency.
- **Feature Engineering**: Select relevant features based on their predictive power. Techniques like feature scaling (normalization, standardization) might be used to bring all features to a common scale for better model performance.

3. Model Selection and Training

- **Model Selection**: Choose suitable machine learning models for URL classification. Common models used for this task include:

    Logistic Regression: A simple yet effective model for binary classification tasks.

    Decision Trees: Can capture nonlinear relationships in the data.

    Random Forest: An ensemble method that aggregates multiple decision trees to reduce overfitting and improve accuracy.

    Support Vector Machine (SVM): Particularly useful for high-dimensional data.

    Neural Networks: Particularly deep learning models for large datasets with complex patterns.

    XGBoost: A highly effective gradient boosting model.

- **Training**: Train the model on the training set using the extracted features. This process involves tuning hyperparameters (e.g., learning rate, depth of trees) using techniques like -validation to ensure the model performs optimally.

2.1.4 Results:

The results of the malicious URL detection system are typically evaluated based on how accurately the model can classify URLs into benign and malicious categories. These results depend on the quality of the dataset, the selected machine learning algorithms, and the features used in the model. Below is an outline of the possible results and performance evaluation metrics that would be used to assess the effectiveness of the system.

2.2 Signature-Based Malicious URL Detection:

2.2.1. Introduction:

Signature-based detection is one of the earliest approaches used to identify malicious URLs. It relies on maintaining a database of known malicious URLs, often referred to as a blacklist. This technique operates by comparing the accessed URL against the stored list of signatures.

2.2.2 Merits , Demerits and Challenges:

Merits (Advantages)

- Simplicity: The approach is straightforward and does not require complex computations.
- Efficiency: For URLs already in the blacklist, detection is almost instantaneous.
- Low computational requirements: Suitable for systems with limited resources.

Demerits (Disadvantages)

- Limited scope: Unable to detect zero-day attacks or new, previously unseen malicious URLs.
- High maintenance cost: Requires frequent updates to the blacklist to stay effective.
- False negatives: Legitimate-looking malicious URLs not in the blacklist go undetected.

2.2.3.Implementation:

- A database containing blacklisted URLs is maintained.
- For each new URL accessed:
    - The URL is queried against the database.
    - If found in the blacklist, it is flagged as malicious.
    - Otherwise, the URL is deemed benign.
- Warnings are generated for flagged URLs.

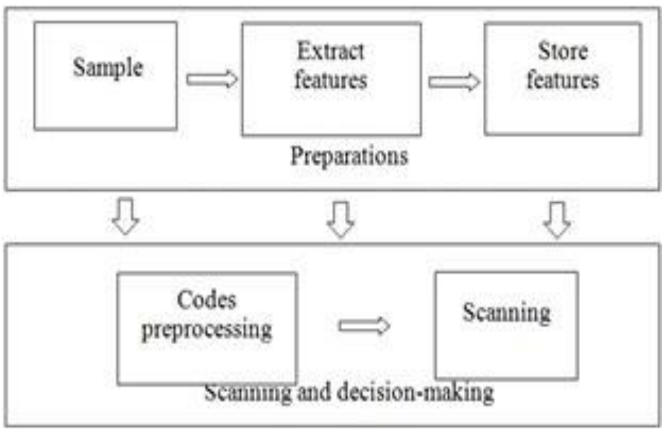

Figure 1. Procedure of signature-based detection

Figure: 2.2.3-Signature based model

2.2.4.Results:

- Signature-based detection can effectively block known malicious URLs.
- However, the approach struggles to keep up with evolving cyber threats, especially when attackers frequently modify URL patterns.

2.3 Machine Learning-Based Malicious URL Detection:

2.3.1.Introduction:

Machine learning-based approaches have gained prominence due to their ability to analyze and classify URLs dynamically based on patterns and behaviors. These methods leverage features extracted from both **static behaviors** (e.g., lexical analysis, content analysis) and **dynamic behaviors** (e.g., runtime monitoring and abnormal activities).

2.3.2 Merits , Demerits:

Merits (Advantages)

- **Adaptability**: Can detect new malicious URLs without relying on a predefined database.

- **Scalability**: Handles large volumes of data effectively, especially with the integration of big data techniques.
- **Feature-rich detection**: Incorporates multiple attributes, such as lexical, host, and behavioral features, for enhanced accuracy.

Demerits (Disadvantages)

- **High computational cost**: Training and deploying machine learning models require significant resources.
- **Complexity**: Requires expertise in feature engineering, model selection, and hyperparameter tuning.
- **False positives**: Legitimate URLs can sometimes be misclassified as malicious.

2.3.3.Implementation:

1. **Feature Extraction**:
    - **Static Features**:
        - **Lexical Analysis**: Examines the URL structure, length, character patterns, etc.
        - **Host Features**: Includes IP address, domain age, and server location.
    - **Dynamic Features**:
        - Behavior-based attributes, such as abnormal patterns in network traffic.
2. **Model Training**:
    - Various machine learning algorithms are used, including:
        - Support Vector Machines (SVM)
        - Random Forest (RF)
        - Logistic Regression, Decision Trees, etc.
    - Training is performed using labeled datasets (URLs classified as benign or malicious).
3. **Classification**:

- The trained model predicts whether a new URL is malicious or benign based on its features.

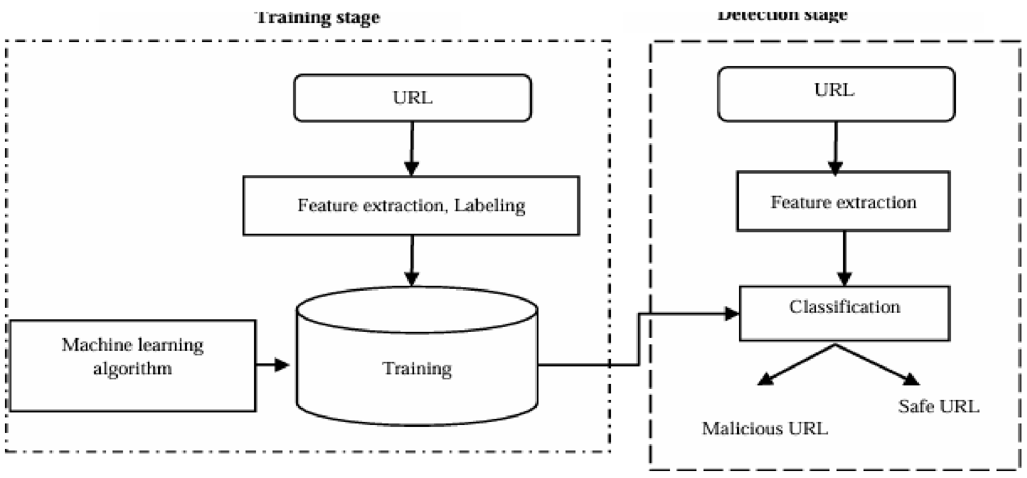

Fig. 1. Malicious URL Detection Model using Machine Learning.

2.3.4.Results:

- Machine learning-based methods outperform signature-based detection in identifying zero-day threats.
- Models like SVM and RF demonstrate high accuracy when trained on both static and dynamic features.
- Big data integration improves the scalability and robustness of detection systems.

TABLE. V. TRAINING PERFORMANCE OF MALICIOUS URL DETECTION SYSTEM

Dataset	Algorithm and parameters	Accuracy (%)	Precision (%)	Recall (%)	Training time (s)	Testing time (s)
10.000 URLs	SVM (100 iterations)	93.39	94.67	92.51	2.32	0.01
	SVM (10 iterations)	93.35	94.84	92.71	3.11	0.01
	RF (10 trees)	99.10	98.43	97.45	2.78	0.01
	RF (100 trees)	99.77	98.75	97.85	3.34	0.01
470.000 URLs	SVM (100 iterations)	90.70	93.43	88.45	272.97	2.12
	SVM (10 iterations)	91.07	93.75	88.85	280.33	2.31
	RF (10 trees)	95.45	90.21	95.12	372.97	2.02
	RF (100 trees)	96.28	91.44	94.42	480.33	2.30

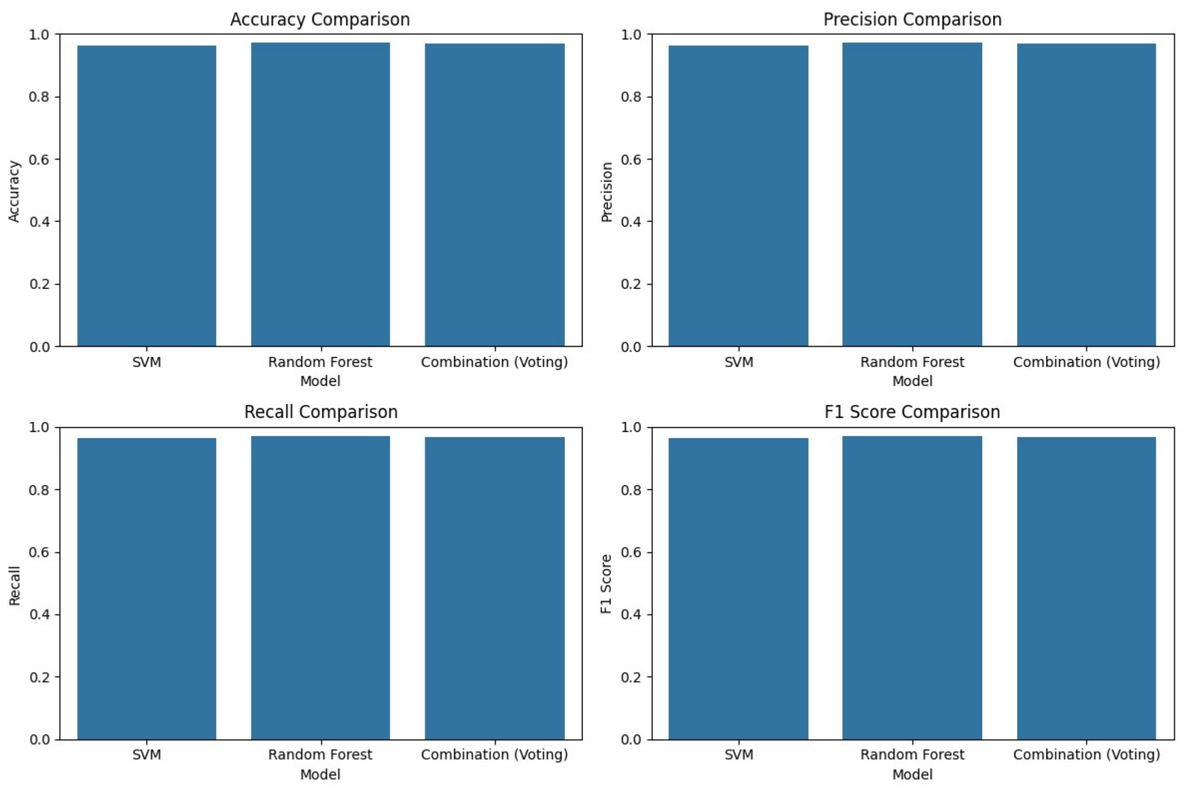

Figure: 2.3.4- Comparison of models

CHAPTER 3
PROPOSED METHOD

CHAPTER 3
Proposed System

3.1 Working

The proposed system is designed to effectively classify URLs into four categories: benign, phishing, malware, and defacement. The system operates by combining advanced machine learning algorithms with comprehensive feature extraction techniques. It integrates static and dynamic URL features, enabling the detection of new and sophisticated malicious URLs.

1. **Data Collection**:

The system uses a labeled dataset comprising both benign and malicious URLs. The dataset includes examples of phishing, malware, and defacement URLs to ensure comprehensive coverage.

2. **Feature Extraction**:

- **Static Features**: These include lexical features (URL length, special characters, domain name) and host-based features (IP address, WHOIS information).
- **Dynamic Features**: These capture runtime behaviors, including abnormal traffic patterns, request headers, and server response times.

3. **Model Training**:

Supervised machine learning algorithms such as Support Vector Machine (SVM) and Random Forest (RF) are used to train models on extracted features.

Cross-validation ensures the model generalizes well to unseen data.

4. **Real-Time Detection**:

Once trained, the models are deployed in a real-time system to classify new URLs.

The system flags malicious URLs, categorizing them based on their type (phishing, malware, or defacement).

3.2 Methodology

The methodology is structured into the following phases:

1. **Data Preprocessing**:

Removal of incomplete and duplicate URLs from the dataset.

Encoding categorical variables and scaling numeric attributes for uniformity.

2. **Feature Selection**:

Diagram Representation

Here is a textual description of the system architecture diagram you can visualize or include in your document:

Input: URL Dataset (Malicious and Benign).

- **Step 1**: Data Preprocessing → Data Cleaning and Feature Extraction (Static and Dynamic Features).
- **Step 2**: Feature Selection → Optimal Features for Classification.
- **Step 3**: Model Training → Train Models (SVM, RF).
- **Step 4**: Deployment → Real-Time Detection and Categorization.
- **Step 5**: Alerts & Reporting → Block malicious URLs, Notify Users.

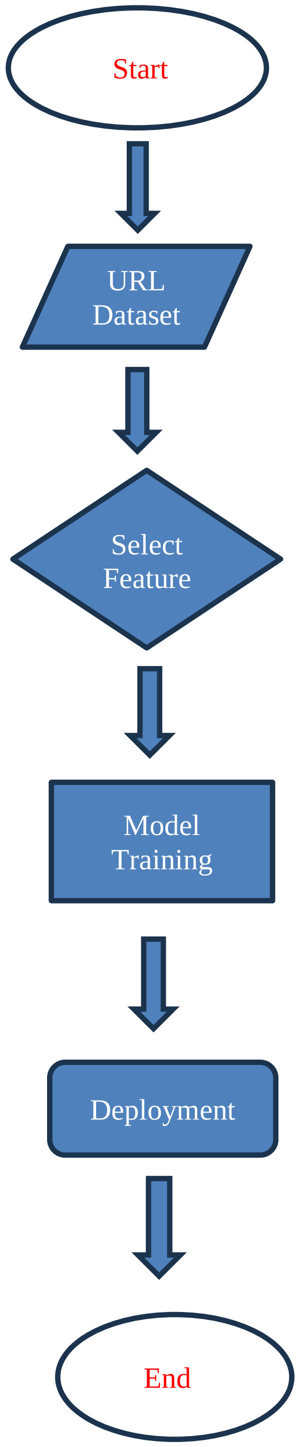

Figure 3.2: Flow Chart

3. **Algorithm Implementation**:

i.Support Vector Machine (SVM):

A supervised algorithm that uses hyperplanes to classify URLs into categories.

Kernel functions (e.g., RBF, polynomial) optimize non-linear classification.

ii.Random Forest (RF):

An ensemble method that constructs multiple decision trees for robust classification.

Out-of-bag (OOB) error estimates are used for internal validation.

iii.System Architecture:

The system is modular, comprising separate components for data ingestion, preprocessing, model training, and prediction.

3.3 Optimization and Validation

1.Hyperparameter Tuning:

Grid Search and Random Search are employed to optimize hyperparameters, such as the number of estimators in RF or the kernel type in SVM.

2. **Cross-Validation**:

k-Fold Cross-Validation is used to evaluate model performance on different subsets of the dataset, ensuring that results are not biased by data splits.

3. **Feature Engineering**:

Additional feature engineering techniques, such as PCA (Principal Component Analysis), are used to reduce noise and improve model efficiency.

4. **Big Data Integration**:

Tools like Apache Spark are used to process large datasets efficiently.

Real-time detection capabilities are implemented using streaming frameworks.

3.4 Evaluation Metrics:

Evaluation metrics are essential for assessing the performance of classification models in machine learning. Each metric provides different insights into how well a model is performing, especially in terms of its effectiveness (how well it identifies positive cases) and reliability (how consistent it is in its predictions). Below are some commonly used evaluation metrics:

1. Accuracy

Definition: Accuracy is the ratio of correctly predicted instances to the total instances in the dataset. It is calculated as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure: 3.4.1- Accuracy

- **Importance**: Accuracy is a straightforward measure of overall performance, especially useful when the class distribution is balanced. However, it can be misleading in cases of class imbalance, where a model could achieve high accuracy by simply predicting the majority class.

2. Precision

Definition: Precision measures the proportion of true positive predictions among all positive predictions. It indicates how many of the predicted positive cases were actually positive. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{TP}{TP + FP}$$

Figure: 3.4.2- Precision

- **Importance**: Precision is crucial in scenarios where the cost of false positives is high (e.g., spam detection, medical diagnosis). It reflects the model's reliability when it predicts a positive class.

3. Recall (Sensitivity)

Definition: Recall measures the proportion of true positive predictions among all actual positive instances. It indicates how well the model identifies positive cases. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{TP}{TP + FN}$$

Figure: 3.4.3- Recall

- **Importance**: Recall is vital when the cost of false negatives is high (e.g., detecting diseases). It reflects the model's effectiveness in identifying all relevant cases.

4. F1-Score

Definition: The F1-score is the harmonic mean of precision and recall, providing a balance between the two. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure: 3.4.4- F1-Score

Importance: The F1-score is particularly useful in situations where precision and recall are in tension, such as imbalanced datasets. It gives a single metric to optimize when you want a balance between capturing positive instances and minimizing false positives.

5. ROC-AUC (Receiver Operating Characteristic – Area Under Curve):

Measures the model’s ability to distinguish between classes across various threshold levels.

6. Confusion Matrix:

Provides insights into true positive, false positive, true negative, and false negative rates.

7. Execution Time:

Evaluates the time taken for feature extraction, model training, and prediction.

REFERENCES

REFERENCES

1. IEEE Xplore -Detecting Malicious URLs Using Machine Learning Techniques
2. Corelight - What Is Signature-Based Detection?
3. arXiv Survey - Malicious URL Detection using Machine Learning
4. International Journal of Advanced Computer Science and Applications
5. IEEE Xplore - Detecting Malicious URLs Using Machine Learning Techniques