Visvesvaraya Technological University

BELAGAVI, KARNATAKA

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Project Report
on

"Machine Learning Based Approach For Phishing Attacks Detection And Prevention"

Submitted by

Pratheeksha N Hampole	**4JN21IS074**
Sowparnika K H	**4JN21IS104**
Sushmitha H S	**4JN21IS111**
T P Keerthi	**4JN21IS115**

Under the guidance of

Mr . Akshay M J B.E, M.Tech

Assistant Professor
Dept. of IS&E,
JNNCE, Shivamogga

Department of Information Science & Engineering
J N N College of Engineering
Shivamogga - 577 204

Visvesvaraya Technological University

BELAGAVI, KARNATAKA

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Project Report
on

" Machine Learning Based Approach For Phishing Attacks Detection And Prevention "

Submitted by

Pratheeksha N Hampole	**4JN21IS074**
Sowparnika K H	**4JN21IS104**
Sushmitha H S	**4JN21IS111**
T P Keerthi	**4JN21IS115**

students of 7th semester B.E. ISE, in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024-25.

Under the guidance of

Mr. Akshay M J B.E, M.Tech

Assistant Professor
Dept. of IS&E,
JNNCE, Shivamogga

Department of Information Science & Engineering
J N N College of Engineering
Shivamogga - 577 204
2024-25

National Education Society ®

J N N COLLEGE OF ENGINEERING
SHIVAMOGGA - 577204.
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Project entitled

" Machine Learning Based Approach For Phishing Attacks Detection And Prevention "

Submitted by

1.	**Pratheeksha N Hampole**	**4JN21IS074**
2.	**Sowparnika K H**	**4JN21IS104**
3.	**Sushmitha H S**	**4JN21IS111**
4.	**T P Keerthi**	**4JN21IS115**

students of 7th semester B.E. ISE, in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024-25.

Signature of Guide

Mr. Akshay M J B.E, M.Tech
Assistant Professor,
Dept. of IS&E,
JNNCE, Shivamogga

Signature of HOD

Dr. Raghavendra R. J.M.Sc (Engg.), Ph.D
Associate Professor & Head,
Dept. of IS&E,
JNNCE, Shivamogga

Signature of Principal

Dr. Y.Vijaya Kumar
Principal,
JNNCE, Shivamogga

1. Examiner ____________________ **2. Examiner** ____________________

ABSTRACT

Phishing attacks remain a significant threat in the cybersecurity landscape, targeting individuals and organizations to steal sensitive information such as usernames, passwords, financial data, and confidential business records. These attacks are typically carried out through deceptive emails, websites, and messages that appear legitimate but aim to trick users into disclosing personal information. Given the sophistication and increasing frequency of phishing schemes, there is an urgent need for automated detection and prevention systems that can quickly identify and mitigate such threats. This project focuses on detecting and preventing phishing attacks using machine learning, specifically leveraging the XGBoost (Extreme Gradient Boosting) algorithm. XGBoost is a state-of-the-art supervised learning model known for its high predictive accuracy and ability to handle imbalanced datasets, which is often the case in phishing detection tasks. The project explores the use of URL-based features, such as domain registration details, URL length, the presence of special characters, and other textual elements, as input to the XGBoost classifier. The process begins by gathering a large dataset containing labeled instances of legitimate and phishing URLs. Various data preprocessing techniques, including feature extraction, normalization, and handling missing values, are applied to ensure the dataset is suitable for model training. The XGBoost model is trained on these features, optimized using hyperparameter tuning, and evaluated using standard performance metrics such as accuracy, precision, recall. The results demonstrate that the XGBoost model outperforms several other machine learning algorithms, such as CNN,KNN, and support vector machines, in terms of accuracy and robustness in phishing detection. Additionally, the system is integrated into a real-time web application that can scan URLs and alert users when they are about to visit a potentially harmful site, thus preventing phishing attempts.

ACKNOWLEDGEMENT

On presenting the Project report on "Machine Learning Based Approach For Phishing Attacks Detection And Prevention", we feel great to express our humble feeling of thanks to all those who have helped us directly or indirectly in the successful completion of the project work.

We would like to thank our respected guide Mr. Akshay M J B.E,M.Tech**,** Assistant Professor, Department of ISE, for her continuous encouragement and guidance.

We would like to thank our project coordinators Dr. Pavan Kumar M. P., Assosciate Professor, Mr. Arun Kumar.P, Assistant Professor , Dr. Deepa V.B., Assistant Professor, Dr. Chethan G. S., Assistant Professor, Department of ISE for all their support and encouragement.

We would like to thank **Dr. Raghavendra R. J.**, Associate Professor and Head of Dept. of ISE, JNNCE, Shimoga and **Dr. Y.Vijaya Kumar**, Principal JNNCE, Shimoga for all their support and encouragement.

We are grateful to Department of Information Science and Engineering and our institution J N N College of Engineering and for imparting us the knowledge with which we can do our best.

Finally, we would like to thank the whole teaching and non-teaching staff of Information Science and Engineering Department.

Thanking You

Pratheeksha N Hampole	4JN21IS074
Sowparnika KH	4JN21IS104
Sushmitha H S	4JN21IS111
T P Keerthi	4JN21IS115

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Artificial intelligence is a new innovative science that reviews and creates hypotheses, strategies, procedures, and applications that recreate, grow and broaden human knowledge. ML is an arm of artificial intelligence and it is analogous to (and frequently overlap with) computational measurements, that also concentrates on making predictions with the use of PCs. Machine leaning has solid relationship with scientific improvement, which tells methods, hypothesis and utilization regions to the field. ML is sometimes, in a while combined with data mining, but the data mining subfield focuses more on preparatory information investigation and is called as unsupervised learning. ML can likewise be unsupervised and be utilized to learn and set up pattern profiles for various entities and then used to find important anomalies.

Phishing attacks represent a significant cybersecurity challenge, leveraging social engineering tactics to manipulate individuals into divulging sensitive information. As these attacks become increasingly sophisticated, traditional detection methods often fall short, highlighting the need for advanced solutions. Machine learning (ML) has emerged as a powerful tool in the fight against phishing, offering innovative techniques for identifying and mitigating these threats.

Machine learning algorithms can analyse vast amounts of data, learning to recognize patterns and anomalies associated with phishing attempts. By employing techniques such as classification, clustering, and natural language processing, ML can effectively differentiate between legitimate communications and potential threats, thereby enhancing detection rates and reducing false positives.

Machine learning (ML) offers a promising solution to enhance phishing detection and prevention efforts. By leveraging algorithms that can learn from data, ML models can identify patterns and anomalies that signify phishing attempts with greater accuracy and speed than conventional methods. Techniques such as supervised learning, unsupervised learning, and deep learning allow for the analysis of vast datasets, including email content, URLs, and user behavior. This incorporating ML into phishing attack prevention strategies not only improves the speed and accuracy of threat identification but also enables adaptive responses to evolving tactics used by attackers. This approach allows organizations to stay one step ahead of cybercriminals.

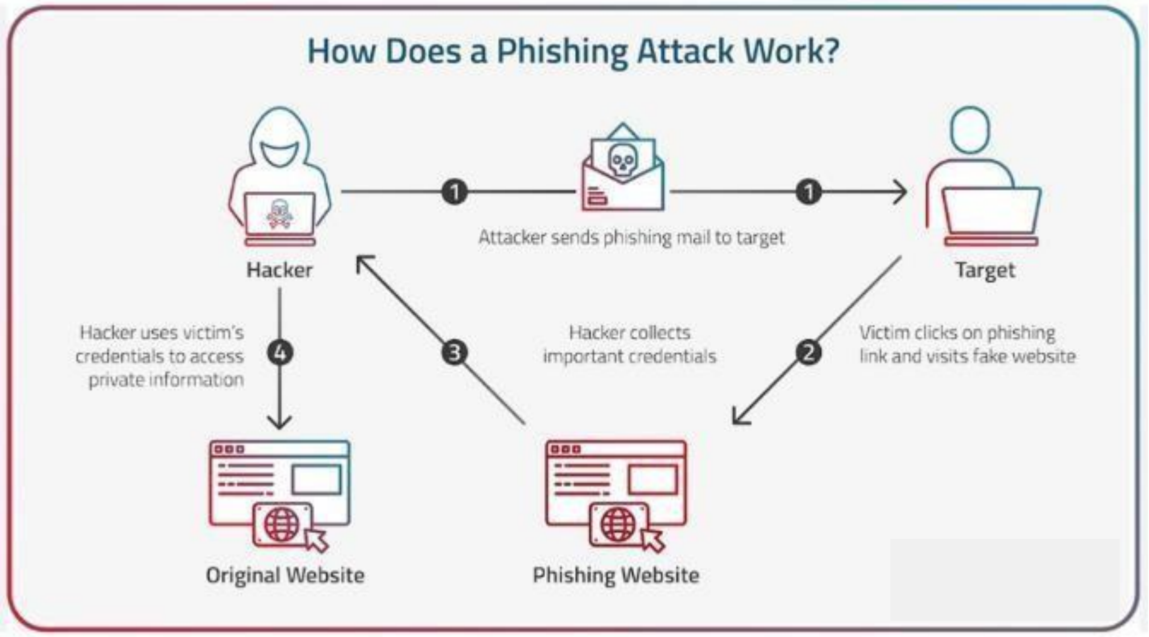

How Does a Phishing Attack Work?
Hacker
1
1
Attacker sends phishing mail to target
Target
Hacker uses victim's credentials to access private information
4
3
Hacker collects important credentials
2
Victim clicks on phishing link and visits fake website
Original Website
Phishing Website

Fig 1.1: Working of Phishing attack

A phishing attack is a carefully designed cybercrime that involves tricking individuals or organizations into divulging sensitive information or performing actions that compromise their security.

Introduction to XGBoost:

XGBoost (eXtreme Gradient Boosting) is a powerful and efficient machine learning library designed for gradient boosting, a technique that combines the predictions of weak learners (typically decision trees) to create a strong predictive model. Developed with a focus on speed, flexibility, and performance, XGBoost has become a go-to tool for solving supervised learning problems, particularly in structured/tabular data.

Key Features of XGBoost:

1. Gradient Boosting Framework: XGBoost implements an optimized gradient boosting algorithm based on decision trees, making it suitable for both classification and regression tasks.
2. Efficiency: The library is designed for computational efficiency, offering fast training and prediction through parallel processing, optimized memory usage, and sparsity-aware algorithms.
3. Regularization: It includes advanced regularization (L1 and L2), which helps prevent overfitting and enhances model generalization.

4. Handling Missing Values: XGBoost automatically handles missing data by learning the optimal splits based on available data.
5. Customizability: Users can fine-tune various hyperparameters to suit specific problem requirements, offering control over the model's behavior.

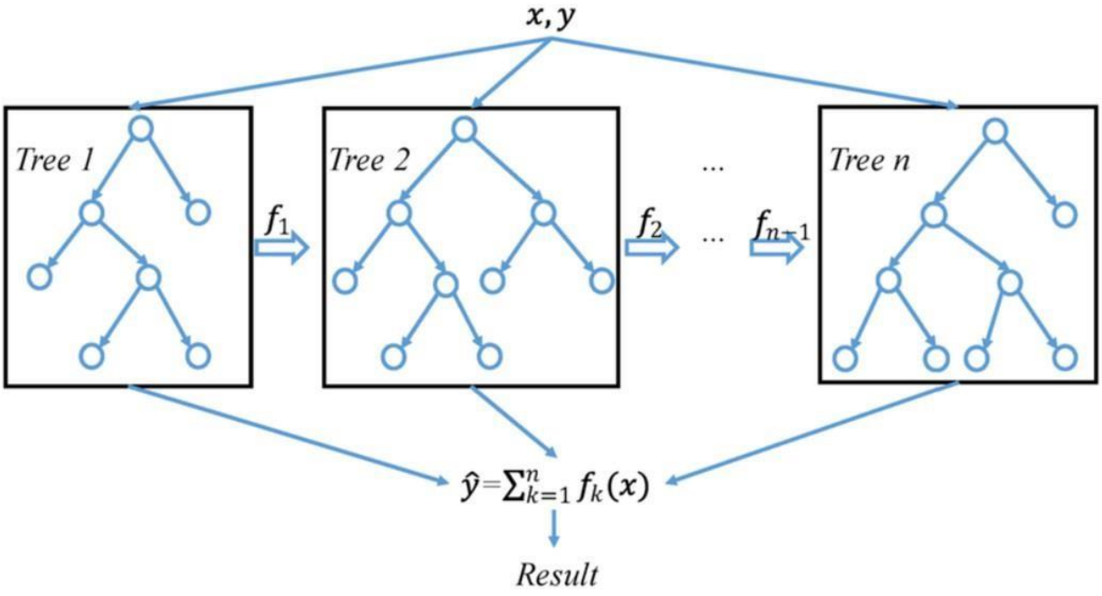


Fig: A general architecture of XgBoost

1.1 Preamble

In the ever-evolving landscape of cybersecurity, phishing attacks continue to pose significant threats to individuals, organizations, and critical infrastructures. These attacks exploit human vulnerabilities and technological loopholes, causing financial losses, data breaches, and reputational damage. The dynamic and adaptive nature of phishing campaigns makes traditional defence mechanisms inadequate, highlighting the urgent need for more sophisticated and proactive solutions.

This project, "Machine Learning-Based Approach for Phishing Attacks Detection and Prevention," seeks to address this challenge by leveraging the power of machine learning. By analysing patterns and anomalies in data, machine learning models can efficiently identify and mitigate phishing threats in real time. The proposed system aims to enhance detection accuracy, reduce false positives, and adapt to emerging attack strategies.Through this research, we aim to contribute to the growing body of knowledge in cybersecurity and provide a robust framework for safeguarding digital environments against phishing attacks.

The integration of advanced machine learning techniques not only strengthens the resilience of security systems but also paves the way for a more secure and trustworthy cyberspace.

The dynamic and adaptive nature of phishing attacks necessitates the adoption of innovative and intelligent solutions. Machine learning (ML) offers a robust approach to this challenge by enabling systems to learn from patterns, detect anomalies, and predict malicious activities in real time. Leveraging ML techniques allows for the analysis of large datasets, including email content, URLs, and website behavior, to identify and prevent phishing attempts more effectively than rule- based systems.

1.2 Problem Description

Phishing attacks are a significant cybersecurity concern, posing risks to individuals, organizations, and governments by exploiting user trust and social engineering techniques. These attacks deceive victims into revealing sensitive information such as login credentials, personal identification numbers (PINs), or financial details. Phishing incidents often occur through deceptive emails, malicious websites, or fraudulent messages, causing financial losses, data breaches, and reputational damage. This project aims to address these limitations by leveraging machine learning (ML) to create a more effective and adaptive solution for phishing detection and prevention. By analyzing diverse features, such as email content, URL characteristics, and web page structures, ML algorithms can detect patterns and anomalies indicative of phishing. The goal is to develop a scalable system capable of identifying phishing attempts in real time, reducing human susceptibility, and enhancing overall cybersecurity measures. This problem encapsulates the critical need for advanced, intelligent, and proactive solutions in the face of an evolving threat landscape, ensuring safer online interactions for users and organizations alike.

1.3 Objectives

1) To collect data and preprocess the data: Identify and gather the relevant datasets and to ensure datasets contain sufficient instances of phishing and legitimate data.

2) Feature Extraction: To automate the process of extracting key features from URLs that indicate phishing. Annotate data points as phishing or legitimate.

3) Model Development: To identify appropriate machine learning algorithm for implementation such as Decision trees, Random forests, Support Vector Machines (SVM), Logistic regression, Xgboost.

4) Model Training and Testing: To Ensure that phishing URLs are detected in real-time or near-real-time, allowing users to avoid interacting with harmful websites.

1.4 Organization of the report

The further report includes following contents, Chapter 2 consists of Literature survey of different reference papers, Chapter 3 represents System Design and Implementation of the proposed work, Chapter 4 consists of Results and Analysis, Chapter 5 consists of Conclusion, Future scope followed by References.

CHAPTER 2

LITERATURE SURVEY

[1] Enhancing Cyber security: A Comprehensive Analysis of Machine Learning Techniques in Detecting and Preventing Phishing Attacks with a Focus on Xgboost Algorithm

Authors: Madan Patil, Nitin Shivsharan , Yashwant Naik , Harshal Yeram, ,Abhishek Gawade.

Year of Publication: 2024

The paper focuses on using machine learning (ML) models to identify and prevent phishing attacks, a significant cyber security threat. It presents a comparative analysis of various ML techniques, including Support Vector Machines (SVM) and XGBoost, to detect phishing websites. The research evaluates models using performance metrics such as accuracy and precision. The XGBoost algorithm is highlighted for its superior accuracy, precision, and computational efficiency across multiple datasets. The study also introduces a novel approach of integrating an API extension for phishing detection.

Advantages

1. High Accuracy and Precision: XGBoost achieves high accuracy (up to 99.17%) and precision compared to other models.
2. Efficient Computation: The XGBoost algorithm demonstrates a fast computation time (0.00894 seconds).
3. Comprehensive Dataset Analysis: Utilizes multiple datasets, such as "Phishing Dataset for Machine Learning," to ensure robust evaluation.

Disadvantages:

1. Complexity of Implementation: Implementing XGBoost and integrating API extensions may require significant expertise and resources.
2. Dataset Dependence: Performance might vary with different datasets or in real-world conditions.

[2] Modified Genetic Algorithm for Feature Selection and Hyper Parameter Optimization: Case of XGBoost in Spam Prediction

Authors: Nazeeh Ghatasheh , Ismail Altaharwa , Kaled Aldebei.

Year of Publication: 2022

The paper presents a novel approach combining feature selection and hyperparameter tuning for the XGBoost algorithm using a modified genetic algorithm (GA). It addresses challenges in spam detection, especially in the context of Twitter, by reducing the dimensionality of large datasets while maintaining or improving classification performance. The research achieves significant spam classification accuracy and geometric mean using less than 10% of the total feature space. The method was validated against datasets including Twitter and SMS spam, outperforming traditional methods such as Chi2 and PCA for feature selection. Spam remains one of the long lasting security threats. E-mail spams represent a true challenge against mail service providers at the early stages of the Internet. Web spams exploit social engineering to lure a privileged user to login into a deceptive service.

Advantages

1. Improved Accuracy: Achieved high accuracy (up to 95.88%) and geometric mean (82.32%) in spam prediction tasks.
2. Effective Dimensionality Reduction: Reduced feature space to less than 10% of the original, enabling efficient model training and testing.
3. Optimized Feature Selection and Tuning: Simultaneous optimization of XGBoost hyperparameters and selection of relevant features enhances model performance.
4. Robust Validation: Performance was validated through rigorous 10-fold cross-validation repeated 50 times.

Disadvantages

1. Computational Complexity: The modified genetic algorithm is computationally expensive, requiring significant time for optimization and feature selection.
2. Dependence on Dataset Quality: Results depend heavily on high-quality labeled datasets, which can be challenging to acquire.
3. Resource Intensive: Training requires considerable computational resources, especially for larger datasets and extensive cross-validation.

[3] “The Need for New Antiphishing Measures Against Spear-Phishing Attacks”

Authors: Luca Allodi, Tzouliano Chotza, Ekaterina Panina, and Nicola Zannone

Year of Publication: 2020

The paper provides an extensive review of the literature on phishing and spear-phishing, differentiating the two types of attacks and dissecting their processes. It highlights the increasing reliance on social engineering (SE) as an attack vector, targeting individuals rather than systems. The paper analyzes a real-world, advanced spear-phishing campaign targeting white-collar workers in 32 countries, illustrating the sophisticated tactics used by attackers, including the creation of fake companies and job postings to lure victims into providing sensitive information. The authors emphasize the need for new countermeasures that account for the human-related characteristics exploited in spear-phishing attacks, calling for increased awareness and research into the peculiarities of spear-phishing to develop and test more effective countermeasures.

Advantages:

1. In-depth Analysis: Provides an extensive review of the literature, dissecting the attack process and characteristics of spear-phishing attacks compared to regular phishing, offering valuable insights into the specificities of spear-phishing.
2. Identification of Gaps: Highlights the foundational gaps between existing countermeasures and the features of spear-phishing attacks, setting the stage for the development of more effective defenses.
3. Multidisciplinary Approach: Integrates perspectives from computer science, psychology, and organizational behavior, providing a holistic view of the spear-phishing phenomenon and its impact on organizations.

Disadvantages:

1. Limited Practical Solutions: Focuses more on the need for future research, offering fewer concrete, immediately applicable solutions.
2. Scope of Case Study: The detailed case study, while informative, is limited to a specific campaign targeting white-collar workers, which may not fully represent the diversity of spear-phishing attacks across different sectors and demographics.
3. Complexity: The multidisciplinary approach, while comprehensive, can be complex and may require readers to have a background in multiple fields to fully understand all aspects of the analysis.

[4] “PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System”

Authors: Maria Sameen, Kyunghyun Han, and Seong Oun Hwang Year of Publication: 2020

The paper presents a novel approach for detecting phishing URLs, particularly those generated by AI systems like DeepPhish. The authors highlight the limitations of traditional phishing detection methods, which often rely on blacklists or simple heuristic rules that are easily evaded by sophisticated attackers. PhishHaven addresses these limitations by employing an ensemble machine learning-based detection system that analyzes the lexical features of URLs. The system is designed to detect both AI-generated and human-crafted phishing URLs. PhishHaven introduces several innovative techniques, including URL HTML Encoding for on-the-fly classification and a URL Hit approach to handle tiny URLs. The ensemble-based machine learning models are executed in parallel using a multi- threading approach, which allows for real-time detection. Theoretical and experimental evaluations demonstrate that PhishHaven achieves high accuracy and efficiency, outperforming existing lexical-based phishing detection systems.

Advantages:

1. Robust Detection: PhishHaven is designed to detect both AI-generated and human-crafted phishing URLs, providing a comprehensive solution to various types of phishing attacks.

2. High Accuracy: The system achieves a high accuracy rate of 98%, significantly outperforming existing lexical-based phishing detection methods.

3. Real-Time Detection: By employing a multi-threading approach, PhishHaven performs real-time detection of phishing URLs, making it practical for everyday use.

Disadvantages:

1. Complex Implementation: The use of ensemble machine learning models and multi-threading adds complexity to the implementation and maintenance of PhishHaven.

2. Scope of URL Types: While effective for many types of phishing URLs, the system's performance may vary for URLs that do not conform to the patterns analyzed in the study.

3. Dependency on Lexical Features: The system relies heavily on lexical features, which means it might miss phishing URLs that use sophisticated encoding or other obfuscation techniques to disguise their malicious intent.

[5] Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity

Authors: Jian Mao, Wenqian Tian, Pei Li, Tao Wei, and Zhenkai Liang Year of Publication: 2017

The paper introduces a new solution for detecting phishing attacks by analyzing the visual similarity between web pages. The authors argue that traditional phishing detection methods, which rely on URL analysis or blacklists, are insufficient as attackers can easily modify these features to evade detection. By evaluating the similarity of page components based on their visual representation, Phishing-Alarm can more accurately and robustly detect phishing attempts. The approach involves three phases: feature extraction, similarity computation, and phishing decision. The authors implemented their solution as a Google Chrome browser extension and evaluated it using a large set of real-world phishing samples. The results demonstrate that Phishing-Alarm is both effective and efficient, achieving high accuracy with a relatively low performance overhead.

Advantages:

1. Robustness: The method focuses on visual features that are difficult for attackers to alter without compromising the effectiveness of their phishing attempts, making it more robust against evasion techniques.
2. High Accuracy: The paper reports high precision and recall rates, indicating that Phishing-Alarm accurately identifies phishing sites with minimal false positives and negatives.
3. Efficiency: The approach efficiently computes similarity scores without the need for rendering web pages, reducing computational overhead and improving detection speed.

Disadvantage:

1. Limited Scope of Evaluation: The evaluation is based on a set of phishing samples from PhishTank, which may not fully represent the diversity of phishing attacks encountered in the wild.
2. Prototype Implementation: The current implementation as a browser extension for Google Chrome may limit its applicability to other browsers or platforms without further development and adaptation.

[6] Phishing Website Detection Using Machine Learning

Authors: Adarsh Mandadi, Vishnu Ravella, Saikiran Boppana, and Prof. Dr. R. Kavitha Year of Publication: 2022

This paper focuses on leveraging machine learning techniques to detect phishing websites, which aim to steal sensitive user information through deceptive means. The proposed approach uses supervised machine learning algorithms—Random Forest and Decision Tree—to classify URLs as phishing or legitimate. The study utilizes a dataset of 10,000 URLs (5,000 phishing and 5,000 legitimate), with extracted features from URL-based (e.g., length, depth), domain-based (e.g., DNS records, domain age), and HTML/JavaScript-based attributes (e.g., iframe redirection, right-click disablement).

The model achieved an accuracy of 87.0% with Random Forest and 82.4% with the Decision Tree algorithm. A web-based application, built using Flask, enables real-time URL classification by users, providing outputs as either “Phishing” or “Legitimate.” This paper highlights a practical, automated solution for identifying and blocking phishing websites.

Advantages:

1. High Accuracy: The Random Forest algorithm delivers 87.0% accuracy, making it highly reliable for phishing detection.
2. Comprehensive Feature Extraction: The model analyzes URL, domain, and HTML/JavaScript-based features for robust classification.
3. Zero-Hour Detection: It can handle zero-hour phishing attacks better than traditional blacklist-based approaches.
4. Scalability: The approach is suitable for scaling to larger datasets and evolving phishing threats.

Disadvantages:

1. Lower Decision Tree Performance: The Decision Tree algorithm has a lower accuracy (82.4%) compared to Random Forest.
2. Dependence on Historical Data: Performance may degrade with insufficiently diverse training data, particularly against novel phishing methods.
3. Computational Complexity: Training the model and extracting features requires significant computational resources.

[7] A Study on Adversarial Sample Resistance and Defense Mechanism for Multimodal Learning-Based Phishing Website Detection

Authors: Phan The Duy, Vo Quang Minh, Bui Tan Hai Dang, Ngo Duc Hoang Son , Nguyen Huu Quyen , And Van-Hau Pham

Year of Publication: 2024

The paper explores innovative approaches to phishing website detection using multimodal learning techniques. It introduces an Adversarial Website Generation (AWG) framework, leveraging Generative Adversarial Networks (GANs) and transfer-based black-box attacks to simulate real-world phishing attacks. The study assesses 15 learning-based models, including machine learning (ML), deep learning (DL), ensemble learning (EL), and multimodal models (MM), focusing on their resistance to adversarial examples (AEs). It also proposes defense strategies such as adversarial training to enhance model robustness against phishing and adversarial websites. Experiments use diverse datasets collected from platforms like OpenPhish, PhishTank, and Alexa to evaluate model performance.

Advantages:

1. Advanced Detection Approach: Utilizes multimodal learning to combine textual, structural, and visual features, resulting in higher detection accuracy.
2. Robust Defense Mechanisms: Proposes effective techniques like adversarial training to significantly improve model resilience.
3. Comprehensive Evaluation: Includes rigorous testing with diverse datasets to ensure practical applicability.
4. High Model Performance: Demonstrates that models like Shark-Eyes achieve up to 99% detection accuracy against adversarial examples.

Disadvantages:

1. Overfitting Risks: Adversarial training could lead to overfitting on specific types of adversarial samples, limiting adaptability.
2. Evolving Threats: The methods may struggle to keep pace with the rapid evolution of phishing and adversarial attack techniques.
3. Scalability Issues: Resource and time requirements could limit scalability for large-scale or low-power systems.

[8] PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification

Authors: Abu A. Sanusi, Darshil Modi, Isaac A. Akintoye, Abu T. Khayami, Nor B. Anuar, and Mohammad F. D. Naser

Year of Publication: 2020

The paper presents a novel dataset specifically designed to address the increasing sophistication and prevalence of phishing attacks. The authors emphasize that phishing kits, which are pre-packaged software used to create phishing websites, pose a significant threat due to their ease of use and distribution. These kits allow attackers with minimal technical knowledge to launch effective phishing campaigns. The PhiKitA dataset aims to enhance the identification and understanding of phishing websites by providing a comprehensive collection of phishing kits and associated metadata. The dataset includes a variety of features extracted from phishing websites, such as URL structures, HTML content, and visual similarity measures.

Advantages:

1. Comprehensive Dataset: PhiKitA provides a detailed and diverse collection of phishing kits, which can significantly aid researchers and practitioners in understanding and identifying phishing websites.
2. Enhanced Detection: The dataset's rich feature set, including URL structures and HTML content, allows for more effective development and testing of machine learning models for phishing detection.
3. Support for Research: By offering a standardized dataset, PhiKitA facilitates benchmarking and comparison of different phishing detection approaches, promoting advancements in the field.

Disadvantages:

1. Maintenance and Updates: Continuously updating and maintaining the dataset to keep pace with evolving phishing tactics can be resource-intensive and challenging.
2. Dependency on Features: The effectiveness of detection models developed using PhiKitA depends heavily on the selected features, which may not always generalize well to all types of phishing attacks.
3. Limited Real-World Application: The dataset, while comprehensive, may not fully capture the diversity of phishing attacks encountered in real-world scenarios.

[9] A URL-Based Social Semantic Attacks Detection with Character-Aware Language Model

Authors: May Almousa And Mohd Anwar Year of Publication: 2023

The paper addresses the challenge of detecting social semantic attacks, a subset of social engineering attacks. These attacks exploit human behavioral and psychological vulnerabilities by creating deceptive elements such as URLs or webpages that mimic legitimate ones. The study focuses on detecting malicious URLs associated with four types of attacks: phishing, spamming, defacement, and malware.The authors propose using deep learning models to analyze URL structures without requiring additional contextual data from web pages. This approach simplifies detection while preserving user privacy. The key contribution of the paper is the introduction of CharacterBERT, a character-aware embedding model that learns URL patterns at the character level, overcoming limitations of traditional word-based embeddings like those in standard BERT models.

Advantages:

5. High Accuracy: The CharacterBERT model outperforms traditional models, achieving nearly 100% accuracy across various attack types.

6. Innovative Embedding Method: The character-aware embedding (CharacterBERT) effectively handles non-standardized URL vocabulary, capturing intricate patterns at the character level.

7. Comprehensive Evaluation: Conducts rigorous testing using a large dataset and evaluates performance across multiple attack types.

Disadvantages:

1. Maintenance and Updates: Continuously updating and maintaining the dataset to keep pace with evolving phishing tactics can be resource-intensive and challenging.

2. Dependency on Features: The effectiveness of detection models developed using PhiKitA depends heavily on the selected features, which may not always generalize well to all types of phishing attacks.

3. Limited Real-World Application: The dataset, while comprehensive, may not fully capture the diversity of phishing attacks encountered in real-world scenarios.

[10] An Evaluation and Comparison for Phishing Attack Detection using Machine Learning Approaches

Authors: Ajeet Kumar Sharma, Anushree , Nitin Rakesh , Pawan Kumar Verma Year of Publication: 2024

This paper investigates the issue of phishing attacks and evaluates the effectiveness of various machine learning algorithms in detecting them. It systematically analyzes five algorithms—Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, and Extreme Gradient Boosting (XGBoost)—using a large dataset of URLs. The paper likely explores and evaluates various machine learning techniques for detecting phishing attacks. Phishing, a social engineering technique, tricks users into divulging sensitive information by mimicking trusted entities. Machine learning approaches have become a crucial tool in identifying phishing attempts by analyzing patterns in URLs, email headers, content, or metadata. In conclusion, phishing attacks are posting a threat to individuals, organizations, and society as a whole. Understanding recent trends of phishing attack is critical to implement effective security measures.

Advantages:

8. Comprehensive Evaluation: Provides a detailed analysis of multiple machine learning models, offering insights into their effectiveness for phishing detection.
9. Real-World Relevance: Addresses a significant cybersecurity threat by proposing automated solutions that can adapt to evolving phishing techniques.
10. Benchmarking: Likely benchmarks models using publicly available datasets, ensuring reproducibility and comparability.

Disadvantages:

1. High Computational Cost: Advanced models (e.g., Neural Networks) may require significant computational resources for training and inference.
2. Generalization Issues: Phishing tactics evolve rapidly, and models trained on historical data may struggle to adapt to new strategies.
3. Feature Engineering Challenges: Extracting meaningful features (e.g., from URLs or content) may require domain expertise and can be error-prone.

[11] Improving Phishing Website Detection Using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning

Authors: Luka Jovanovic1, Dijan Jovanovic2, Milos Antonijevic1, Bosko Nikolic3, Nebojsa Bacanin , Miodrag Zivkovic1 and Ivana Strumberger1 1Singidunum University, Danijelova 32, 11000, Belgrade, Serb

Year of Publication: 2023

This paper presents a novel hybrid two-level framework designed to enhance the detection of phishing websites by optimizing feature selection and XGBoost model performance. The framework integrates advanced feature selection techniques to identify the most relevant attributes, reducing noise and computational overhead. Subsequently, it applies an iterative hyperparameter tuning process to fine-tune the XGBoost classifier for improved accuracy and robustness. Experimental evaluations demonstrate that the proposed approach outperforms traditional methods, achieving higher detection rates and lower false positives. The study highlights the potential of hybrid frameworks in advancing phishing detection and contributes valuable insights into leveraging machine learning for cybersecurity.

Advantages:

1. Efficient Feature Selection: The two-level framework ensures that only the most relevant features are used, minimizing noise and optimizing computational resources.
2. Enhanced Model Performance: XGBoost is known for its robust performance in handling classification tasks. Fine-tuning its parameters further improves its ability to generalize across datasets.

Disadvantages:

1. Complexity in Implementation: The hybrid two-level approach and XGBoost tuning require expertise in feature engineering and machine learning, posing a steep learning curve for some developers.
2. Dependency on Quality of Data: The framework's success heavily relies on the quality and diversity of the input data. Poorly prepared or biased datasets can limit its effectiveness.

[12] An Intelligent System for Phishing Attack Detection and Prevention

Author: Megha N, K R Ramesh Babu, Elizabeth Sherly Year of Publication: 2019

This research paper presents a multi-agent intelligent system for detecting and preventing phishing attacks and malicious scripts using machine learning. It incorporates four agents: a monitoring agent for URL extraction, two decision-making agents utilizing classifiers (SVM and ANN), and an action-performing agent to block malicious pages or scripts. The system employs the Extensible Messaging and Presence Protocol (XMPP) for agent communication and integrates features like URL length, IP addresses, and DNS records for analysis. The proposed approach demonstrated superior accuracy and precision compared to traditional methods, particularly with the SVM classifier.

Advantages:

1. High Accuracy: The system achieves superior detection rates, particularly with the SVM classifier, ensuring reliable identification of phishing websites.

2. Multi-Agent Architecture: The modular and cooperative nature of the agents enhances the system's performance and scalability.

3. Comprehensive Detection: Capable of detecting both phishing attacks and malicious scripts, offering a dual-layered security approach.

4. Adaptability: The system learns and adapts to evolving phishing techniques, staying relevant against new threats.

Disadvantages:

1. Complex Implementation: The use of multiple agents increases the complexity of system design and maintenance.

2. Computational Overhead: Multi-agent coordination and processing add to the system's computational demands.

3. Dataset Dependency: The effectiveness relies on the availability and quality of training datasets for machine learning classifiers.

4. Limited Scope: May struggle to detect highly sophisticated or unknown attacks beyond the scope of its training.

[13] Phishing Attacks and Protection Against Them

Authors: Michael A. Ivanov, Ilya V. Chugunkov, Bogdana V. Kliuchnikova, Anna M. Plaksina

Year of Publication: 2021

This research paper explores the persistent cybersecurity threat of phishing attacks, detailing their various forms, including spear phishing, vishing, and smishing. It explains how attackers exploit social engineering techniques to manipulate users into divulging personal information or installing malware. Specific attention is given to emerging threats like ransomware, banking trojans, and cryptojacking. The article also outlines the stages of phishing attacks, from preparation to execution. Practical strategies for protection, such as updating software, using two-factor authentication, and recognizing suspicious communication, are discussed to equip users with tools to mitigate risks.

Advantages:

1. Comprehensive Overview: Covers various types of phishing attacks and their implications.
2. Educational: Provides detailed insights into social engineering tactics.
3. Practical Advice: Offers actionable measures to protect against phishing.
4. Up-to-Date Examples: Highlights recent real-world cases and threats.
5. Technical Depth: Explains vulnerabilities like XSS and CSRF in detail.

Disadvantages:

1. Technical Jargon: May be difficult for non-technical readers to follow.
2. Limited Scope: Focuses primarily on phishing, without extensive coverage of broader cybersecurity issues.
3. Regional Bias: Some examples are specific to certain countries or industries.
4. No Quantitative Data: Lacks statistical analysis to support the severity of claims.
5. Dependent on References: Relies heavily on prior research without introducing significantly new findings.

[14] Hide and Seek: An Adversarial Hiding Approach Against Phishing Detection on Ethereum

Authors: Haixian Wen, Junyuan Fang , Student Member, IEEE, JiajingWu Year of

Publication: 2023

This paper investigates a novel adversarial hiding approach aimed at evading phishing detection mechanisms within the Ethereum blockchain ecosystem. It explores how malicious actors exploit Ethereum's decentralized infrastructure and smart contract features to conceal phishing activities from detection systems. The proposed approach leverages adversarial techniques, such as obfuscation and manipulation of transaction patterns, to bypass traditional and machine learning-based phishing detection algorithms. The study further evaluates the impact of these evasion strategies on existing detection systems and proposes potential countermeasures to mitigate such threats. By shedding light on this emerging challenge, the paper provides critical insights into enhancing security within blockchain networks.

Advantages:

1. Enhanced Resistance to Detection Systems: By using adversarial hiding techniques, phishing schemes may evade detection mechanisms, challenging even advanced algorithms used for monitoring Ethereum transactions and smart contracts.
2. Innovation in Strategy Design: Adversarial methods push the boundaries of existing hishing tactics, offering insights into how threat actors adapt to evolving detection techniques. This can inadvertently advance research in cybersecurity countermeasures.
3. Exploration of System Weaknesses: The approach reveals potential vulnerabilities in Ethereum's transaction monitoring and anti-phishing systems, allowing for improved robustness in future iterations.

Disadvantages:

1. Resource-Intensive: Implementing and maintaining adversarial techniques require significant technical expertise and computational resources, potentially making them less accessible to less sophisticated attackers.
2. Potential for Systemic Harm: These techniques can lead to widespread financial losses for users and negatively impact Ethereum's reputation, reducing user trust and adoption.

[15] An In-Depth Benchmarking and Evaluation of Phishing Detection

Authors: Ayman El Aassal, Shahryar Baki , Avisha Das, and Rakesh. M. Verma Year of Publication: 2020

This paper provides a comprehensive benchmarking and evaluation of phishing detection techniques, addressing their effectiveness in meeting modern security requirements. It surveys state-of-the-art methodologies, highlighting their strengths, limitations, and practical applicability in real-world scenarios. By analyzing a wide range of phishing detection algorithms and systems, the study identifies critical gaps in existing research and offers insights into improving detection accuracy, scalability, and robustness. The findings aim to guide researchers and practitioners in enhancing security frameworks to combat evolving phishing threats effectively. This study offers a thorough benchmarking and evaluation of existing phishing detection techniques, providing valuable insights into their capabilities and limitations in addressing current security challenges.

Advantages:

1. Diverse Dataset Use: The use of varied and extensive datasets, including diverse legitimate and phishing emails and URLs, enhances the study's applicability and generalizability.
2. Evaluation of Realistic Scenarios: The paper addresses practical challenges like imbalanced datasets and evaluates classifiers in scenarios mimicking real-world distributions of legitimate and phishing samples.
3. Insights into Feature Importance: The study uses multiple feature ranking methods to identify the most significant features, improving understanding of what contributes most to detection performance.

Disadvantages:

1. Complexity for Non-Experts: Despite its modular design, PhishBench may still require significant expertise to use effectively, especially for adding new features or classifiers.
2. Heavy Reliance on Specific Datasets: While diverse datasets are used, the paper acknowledges potential biases, such as over-reliance on specific data sources like Spam Assassin, which may limit generalizability.

[16] Evasion Attacks and Defense Mechanisms for Machine Learning-Based Web Phishing Classifiers

Authors: Manuj. Pillai, S. Remya and Yongyun Cho ,V. Devika1, Somula Ramasubbareddy

Year of Publication: 2023

This paper explores the vulnerabilities of machine learning-based web phishing classifiers to evasion attacks, where adversaries manipulate inputs to bypass detection systems. It provides an in-depth analysis of various evasion attack strategies, such as feature manipulation, adversarial examples, and mimicry techniques, which undermine the effectiveness of these classifiers. The study also reviews state-of-the-art defense mechanisms, including adversarial training, feature hardening, and robust model architectures, aimed at enhancing the resilience of phishing detection systems. By evaluating the interplay between attacks and defenses, the paper offers practical insights into strengthening machine learning models against adversarial threats in the domain of web.

Advantages:

1. Adaptation to Real-World Threats: The study of evasion attacks ensures classifiers remain effective against evolving tactics used by attackers, making them more reliable in real-world scenarios.

2. Improved Detection Accuracy: Defense mechanisms can reduce false negatives by detecting sophisticated phishing attempts, even those crafted to bypass conventional ML systems.

3. Proactive Risk Mitigation: Understanding evasion tactics enables organizations to anticipate and counteract potential threats before they cause significant harm.

Disadvantages:

1. High Complexity: evasion attacks often requires sophisticated techniques and significant expertise, increasing the complexity of designing and maintaining ML classifiers.

2. Resource-Intensive: Implementing and continuously updating defenses against evasion attacks can be computationally expensive and time-consuming.

3. Reduced Model Efficiency: Adding layers of defense may increase processing time, potentially impacting the classifier’s speed and efficiency in real-time apps.

[17] Learning from the Ones that Got Away: Detecting New Forms of Phishing Attacks

Authors: Christopher N. Gutierrez, Taegyu Kim, Raffaele Della Corte, Jeffrey Avery, Dan Goldwasser, Marcello Cinque, Saurabh Bagchi

Year of Publication: 2018

This paper addresses the challenge of detecting new forms of phishing attacks that evade traditional filters. It introduces SAFE-PC, a machine learning system for phishing email detection. SAFE-PC extracts and processes email features, leveraging techniques like Natural Language Processing and Named Entity Recognition. It uses an ensemble classifier for high detection accuracy. Evaluations using a dataset of over 425,000 phishing emails reveal that SAFE-PC detects 71% of phishing emails undetected by state-of-the-art tools like Sophos. Additionally, SAFE-PC's online variant incrementally adapts to new phishing tactics without complete retraining. The paper highlights common phishing strategies, limitations of existing tools, and areas for improvement.

Advantages:

1. High Detection Rate: Detects 71% of phishing emails that bypass traditional filters like Sophos.
2. Adaptability: Online learning enables SAFE-PC to update incrementally with new data.
3. Versatile Application: Performs better than other tools like Spam Assassin and Sophos in phishing detection.
4. Insightful Analysis: Provides detailed understanding of phishing strategies and campaign evolution.

Disadvantages:

1. High False Positive Rate: 15% false positives may lead to legitimate emails being flagged.
2. Slower Performance: Training and classification times are significantly slower than traditional tools.
3. Resource Intensive: Requires substantial computational resources for large datasets.
4. Dependence on Manual Effort: Needs labeled datasets for effective training, which can be time-consuming.

[18] An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework

Authors: Ehsan Nowroozi Abhishek Mohammadreza Mohammad and Mauro Conti, Abhishek

Year of Publication: 2023

The paper investigates the vulnerabilities of machine learning (ML) models in detecting malicious advertisement URLs. The study develops a framework leveraging lexical and web-scrapped features for ML-based classification and clustering. It evaluates four ML models (Random Forest, Gradient Boost, XGBoost, and AdaBoost) for their detection accuracy and robustness against adversarial attacks, specifically the Zeroth Order Optimization (ZOO) attack. The research focuses on improving detection accuracy and analyzing the susceptibility of ensemble-based models to adversarial manipulations.

Advantages:

1. High Detection Accuracy: Achieved a remarkable detection accuracy of up to 99.63% with XGBoost, demonstrating the efficacy of the proposed feature engineering and classification methods.

2. Novel Features: Integration of advanced lexical and web-scrapped features provides a robust approach to distinguishing malicious URLs from benign ones.

3. Adversarial Robustness Analysis: Explores the vulnerability of ML models against ZOO attacks, an important contribution to adversarial machine learning research. Comprehensive

4. Evaluation: Tests on 12 diverse datasets provide a reliable assessment of the model's performance in real-world scenarios.

Disadvantages:

1. Time-consuming Feature Extraction: Web-scrapped feature collection is computationally expensive, potentially limiting scalability in real-time applications.

2. Adversarial Attack Limitations: The study primarily focuses on ZOO attacks during the testing phase and does not explore comprehensive defensive strategies.

3. Limited Dataset Types: Although diverse, the datasets may not cover all potential real-world scenerios.

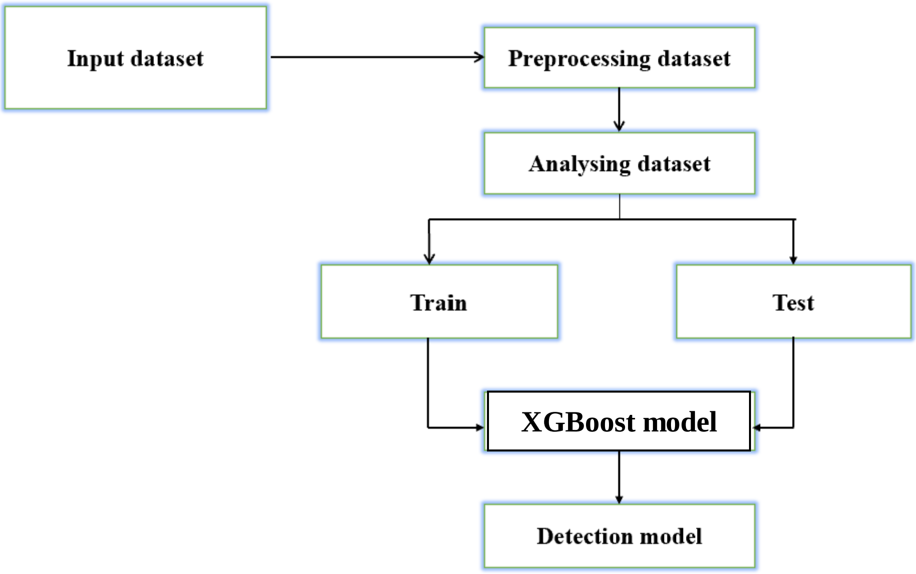
CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 System Design:

The system begins with the input dataset, which contains information relevant to various phishing attacks including numerous features. This dataset serves as the foundation for subsequent stages. Preprocessing steps are then applied to clean, normalize, and prepare the data for analysis. This ensures that the dataset is in a suitable format for the next stages. Following preprocessing, the data undergoes analysis to uncover insights and patterns through techniques such as exploratory data analysis (EDA). This step helps in understanding the characteristics of the dataset and identifying key features for phishing detection. Subsequently, the dataset is split into training and testing sets to facilitate model training and evaluation. The training set is utilized to train machine learning algorithms, while the testing set is used to assess the performance and generalization of the models. This iterative process allows for the development of accurate detection models for the various URL based phishing attacks. The Fig 3.1 represents the block diagram of Phishing attack.

Figure 3.1 Block diagram of Phishing attack detection



3.2 Proposed Methodology:

The proposed methodology begins with the collection of data which contains the information like ‘s’ at the end of https, long URL, short URL and other related details that are required for detecting phishing in a website, followed by data preprocessing preparing the raw data for analysis using various techniques like data completion, data noise reduction, data transformation and validation, feature selection is then performed to identify the most relevant attributes in predicting the disease effectively, later the pre-processed data is splitted into training and testing of the data. 80% of the data is considered for training and 20% of data is considered for testing. The next step involves choosing the most accurate model, which will be used for prediction using user input. The Fig 3.2 represents the proposed methodology of human multiple disease prediction each stages are discussed below one after the other.

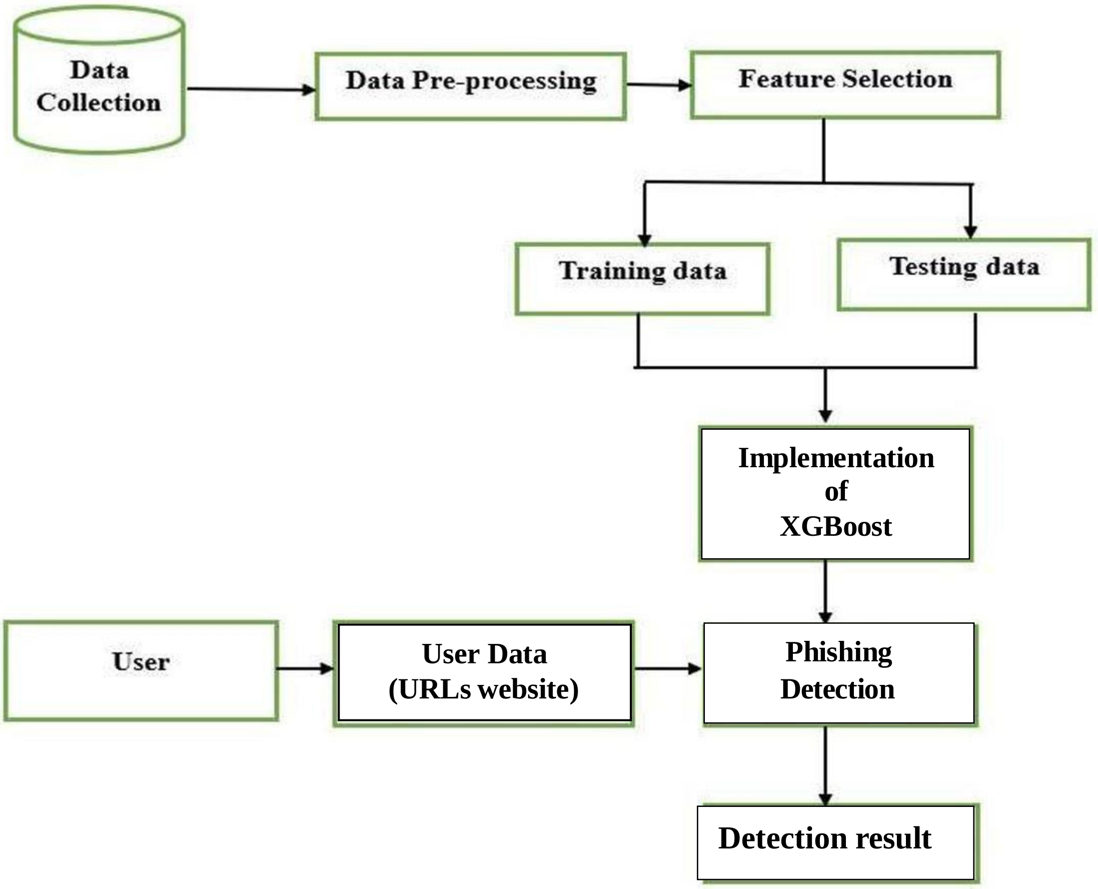


Fig 3.2 Proposed methodology of Phishing attack detection and prevention

3.2.1 Data Collection:

Data collection involves gathering relevant information for a machine learning task. This could be in the form of structured data from databases or unstructured data from various sources. The quality and quantity of a data significantly impact the performance of a model. Distribution Sites offer a pool of phishing kits to be downloaded. Distribution sites can be found on Internet Relay Chat (IRC) channels, underground communities, GitHub, Kaggle or web forums [8].

The project employed the datasets mainly from Kaggle.

The figure 3.3 illustrates a comprehensive pipeline for collecting datasets related to phishing URLs and kits, divided into four main steps. The process begins with URL collection (step i), where various sources are tapped into by a URL collector script designed to gather URLs. Following this, the phishing kit collection (step ii) is managed by a component named Kitphishr, which collects pre-packaged files used by attackers to create phishing websites. In the third step (step iii), the collected URLs are filtered using a URL Crawler filtering script, which identifies potential phishing URLs. This is complemented by a web crawler that systematically browses the web to gather more data and validate the URLs. A crawler, also known as a web crawler or spider, is a type of automated script or program that systematically browses the internet. The primary function of a crawler is to navigate the web, visiting web pages and extracting information from them. This process is often referred to as "crawling" or "spidering". The final step (step iv) involves post-processing filters that refine and clean the dataset, resulting in a final, processed dataset ready for analysis or use in detecting phishing threats. Each step is crucial in ensuring that the data collected is relevant, accurate, and useful for further research or practical application.[8]

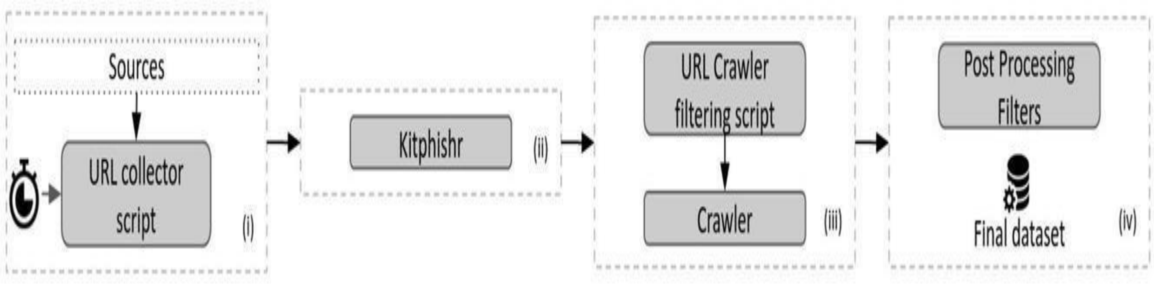


Fig 3.3 Pipeline of the methodology for collecting datasets: (i) URL collection, (ii) phishing kit collection, (iii) URL filter and crawler script, and (iv) post-processing filters.

3.2.2 Data Pre-processing:

Data preprocessing refers to the preliminary steps and procedures carried out on incoming data before feeding it into a detection algorithm or system. It constitutes a fundamental initial phase in constructing a machine-learning model. One crucial task is handling missing values, where techniques such as removal, imputation, or advanced methods are employed. Categorical variables need encoding for numerical representation, achieved through methods like one-hot encoding or label encoding. Feature scaling ensures numerical features are on a consistent scale, either through standardization or normalization. Outliers may be addressed by removal, transformation, or separate treatment. Feature engineering involves creating or modifying features for improved model performance.

The goal is to ensure that your data is in a suitable format for training a machine learning model. Data Completion involves filling in missing or incomplete data to ensure a comprehensive dataset. Data Noise Reduction refers to the process of identifying and minimizing irrelevant or erroneous data. Data Transformation involves converting and restructuring data from one format or representation to another. Data Reduction refers to techniques that reduce the volume but produce the same or similar analytical results. Data Validation is the process of ensuring that data is accurate, consistent, and meets specified requirements or standards.

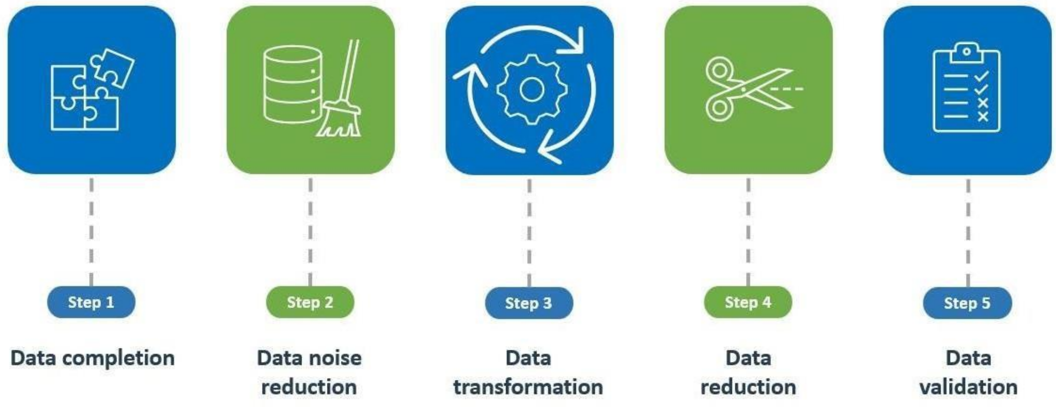


Fig 3.4 Steps in Data pre-processing

3.2.3 Analysing Data:

In machine learning, data analysis involves a systematic examination of datasets to uncover patterns, trends, and relationships that can inform model development. This process typically begins with exploratory data analysis (EDA), where techniques such as statistical summaries, data visualization, and correlation analysis are employed to gain insights into the dataset's characteristics. EDA helps in understanding the distribution of data, identifying outliers, and assessing the relationship between variables. Visualizations like histograms, scatter plots, and heatmaps provide intuitive representations of the data, aiding in identifying potential patterns or anomalies. Additionally, domain knowledge often guides the analysis, helping to uncover meaningful insights relevant to the problem at hand. Data analysis serves as a critical precursor to model building, ensuring that the data is understood thoroughly and appropriately prepared for subsequent machine learning tasks.

3.2.4 Feature Selection:

In phishing website detection, classifying URL features entails scrutinizing different characteristics or attributes of a URL to ascertain its potential association with a phishing attempt. Various URL features serve as classification criteria to distinguish between phishing and legitimate websites. Examples of such URL features include SFH, having IP address, double slash redirecting, URL length, URL of anchor, on mouse hover, and more [1].

URL-based detection techniques analyse URL features of web pages to filter out suspicious malicious websites [5]. Feature selection in machine learning is the process of choosing the most relevant attributes from a dataset to improve the model's performance and avoid unnecessary complexity as

shown in the Fig 3.4. Various techniques are employed for this purpose. Filter methods evaluate individual features based on statistical measures like correlation or mutual information. The goal of feature selection is to enhance model accuracy, reduce overfitting, and streamline the learning process by focusing on the most informative aspects of the data.

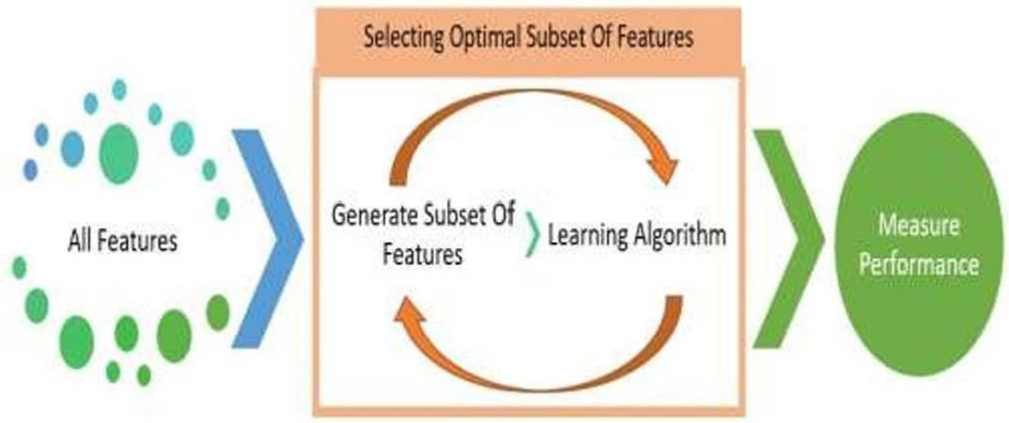

Selecting Optimal Subset Of Features
All Features
Generate Subset Of Features
Learning Algorithm
Measure Performance

Fig 3.5 Selection of the Optimal Features

3.2.5 Training Data and Testing Data:

Training data in a machine learning model consists of a set of examples used to teach the model how to make detection and prevention of phishing attacks. This data includes input features and corresponding target labels. We divided the dataset into two parts to set the thresholds of each algorithm. The 20% of the data was used to find the thresholds. Then, we used the remaining 80% of the data to evaluate the performance of each algorithm. We designed a grid search and divided it into intervals of 0.01, taking into account that the result of the algorithms is a float from 0 to 1, representing the similarity of the sample [8].

By creating the combination of various features selected, we identified whether the website that is URL is whether phishing or not by putting the final result under “class” which should be either 1 or -1. Here 1 represented the website is under phishing and -1 for non-phishing. Then we selected the threshold value where the algorithm achieved the best performance on the 20% of the data and evaluated it on the remaining samples to report the actual result.

During the training phase, the model adjusts its internal parameters based on this data to learn the patterns and relationships within it. The goal is to enable the model to make accurate predictions on new, unseen data. The quality and diversity of the training data significantly influence the model's performance and generalization capabilities. The Fig 3.5 represents the training and testing of data.

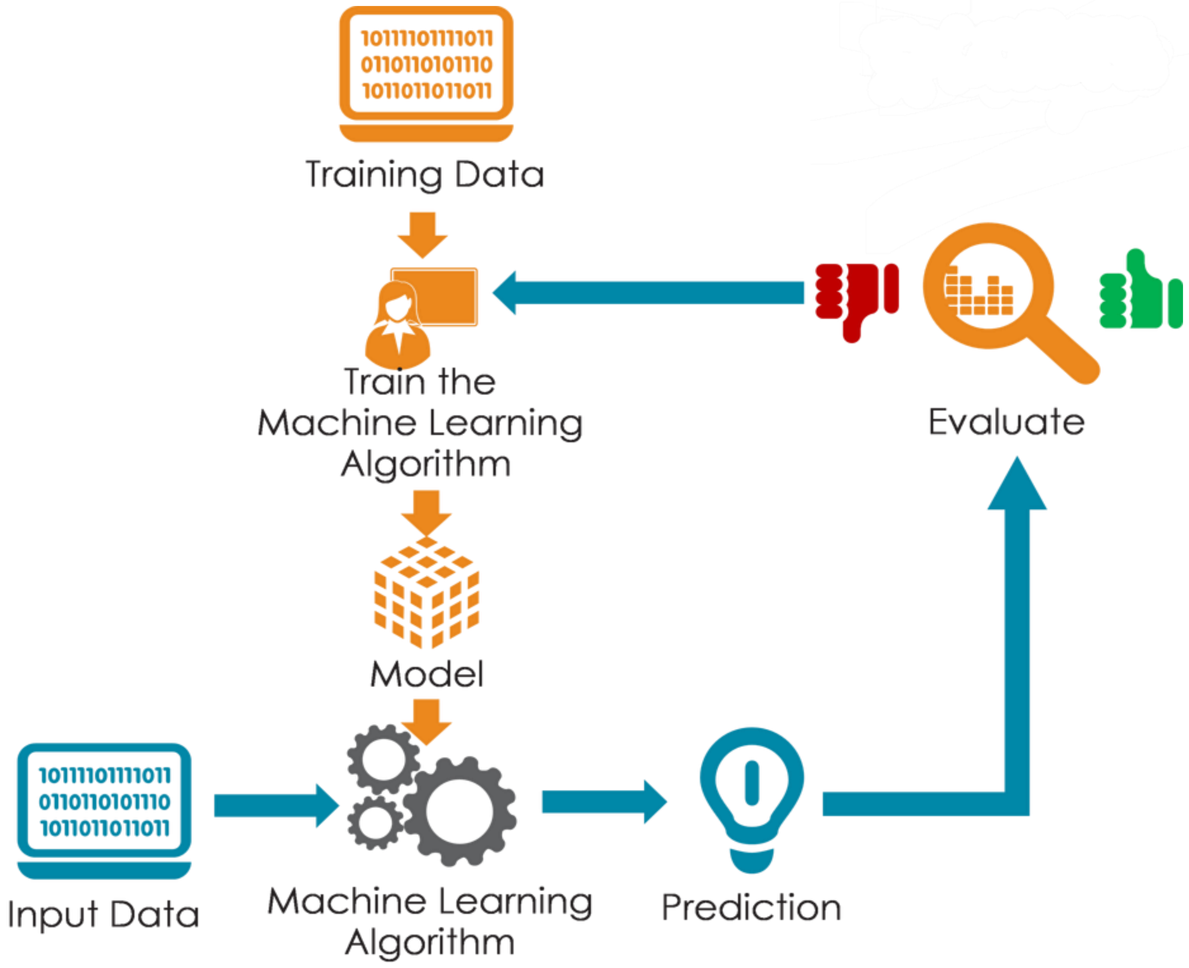


Fig 3.6 Training and Testing of data

3.2.6 Machine Learning Algorithms considered for model creation:

Machine learning algorithms are very important in decision-making processes based on data in current times. With the help of these algorithms, computers can analyse information and independently make decisions or predictions without precise instructions for each individual task. Here is a details of implemented machine learning algorithms.

3.2.6.1 K-Nearest Neighbour (KNN):

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

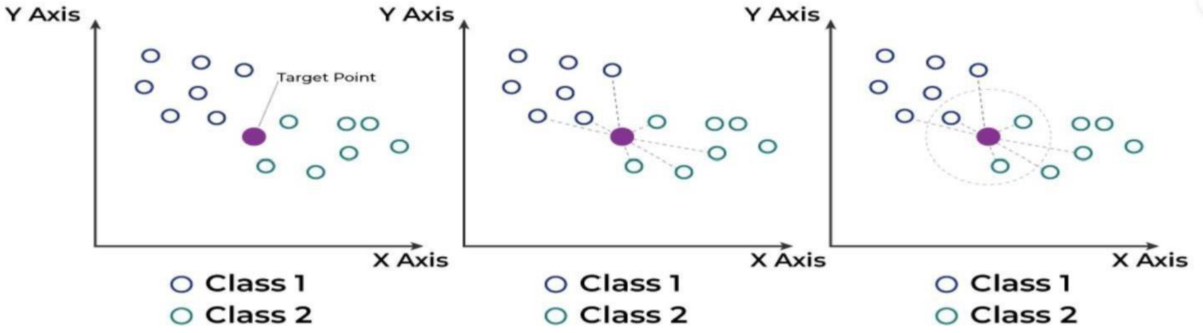


Fig 3.7 Working of the KNN Machine Learning Algorithm

KNN is widely used in pattern recognition, anomaly detection, and recommendation systems due to its simplicity and effectiveness, particularly in situations where data is non linearly separable as shown in the Fig 3.7. Variables used in KNN include:

• **Training data:** Set of labelled data points used to train the model.

• **Test data:** Unlabelled data points for which we want to predict the class or value.

• **Distance metric:** Measure used to calculate the distance between data points.

• **K:** Number of nearest neighbours to consider when making predictions.

KNN operates by storing all available cases and classifying new cases based on a similarity measure (e.g., distance functions). To classify a new data point, the algorithm calculates the distance between that point and all other points in the training data. It then selects the k nearest neighbours (data points with the smallest distances) and assigns the majority class (for classification) or average value (for regression) as the prediction for the new data point. The choice of k is crucial, as smaller values lead to more flexible models prone to overfitting, while larger values lead to smoother decision boundaries but may miss local patterns.

Distance Calculation (e.g., Euclidean Distance):

d(p, q) = sqrt(Σ(q_i - p_i)^2) → Equation 1

where:

d(p,q) : Distance between points p and q,

p_i, q_i : ith component if points p and q,

Class(x): Predicted class for data point x

{y_i}: Classes of k nearest neighbours of x.

Regression Rule (Average):

zValue(x) = mean({y_i}), where: → Equation 2

Value(x): Predicted value for data point x

{y_i}: Values of k nearest neighbors of x

KNN assigns labels or values to new data points based on their proximity to existing data points, making it a versatile and intuitive algorithm for various machine learning tasks.

The algorithm is as follows:

1. Store all available cases with their class labels.

2. Calculate the distance between the query instance and all instances in the dataset using a chosen distance metric (e.g., Euclidean distance).

3. Identify the k nearest neighbors based on the calculated distances.

4. For classification, assign the majority class among the k neighbors to the query instance.

5. For regression, calculate the average of the target values of the k nearest neighbors and assign it to the query instance.

3.2.6.1 Convolutional Neural Networks (CNN):

CNN is a deep learning algorithm used primarily for analysing visual imagery, such as images and videos. It's designed to automatically and adaptively learn spatial hierarchies of features from the input data. Convolutional Neural Network is one of the technique to do image classification and image recognition in neural networks. It is designed to process the data by multiple layers of arrays. The primary difference between CNN and other neural network is that CNN takes input as a two-dimensional array.And it operates directly on the images rather than focusing on feature extraction which other neural networks do. Convolutional Neural Networks have the following 4 layers as shown in Fig 3.8.

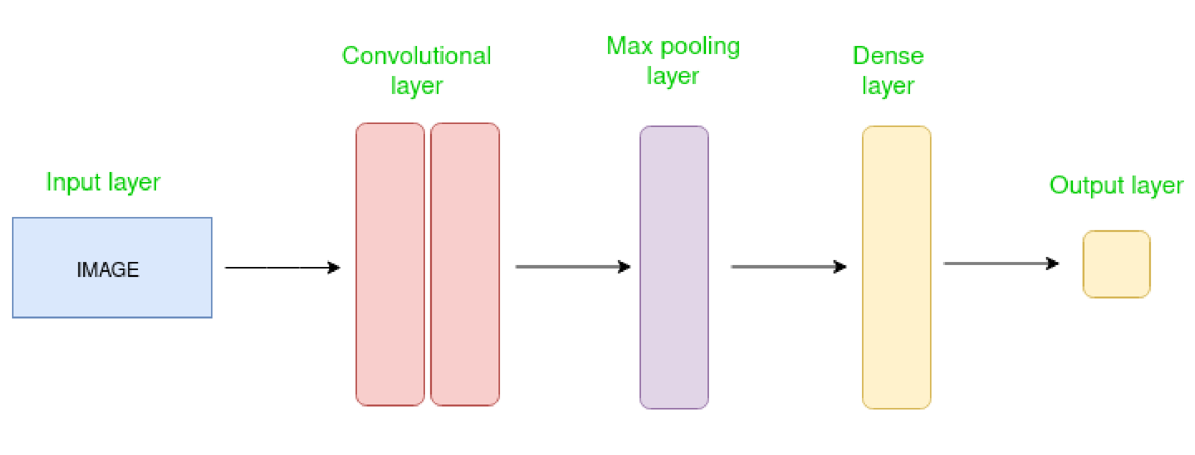
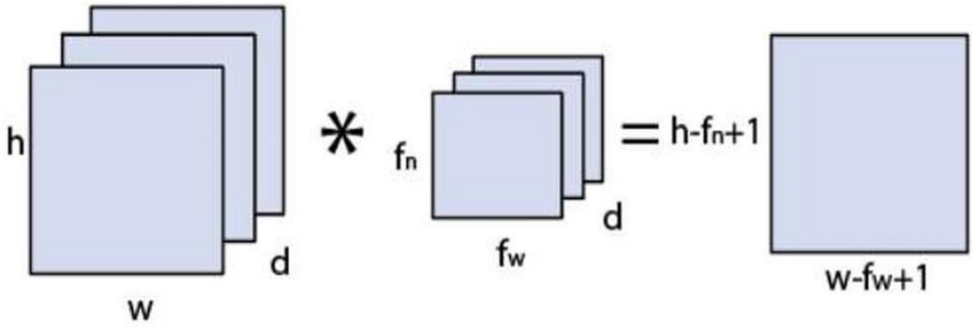


Fig 3.8 Simple Architecture of CNN Algorithm in Machine Learning

Convolutional layer

Convolution layer is the first layer to derive features from the input image. The convolutional layer conserves the relationship between pixels by learning image features using a small square of input data. It is the mathematical operation which takes two inputs such as image matrix and kernel or any filter as shown in Fig 3.9.

- The dimension of image matrix is h×w×d. → Equation 3
- The dimension of any filter is fh×fw×d. → Equation 4
- The dimension of output is (h-fh+1)×(w-fw+1)×1 → Equation 5



.

Fig 3.9 Convolution Layer

3.2.6.2 SVM(Support Vector Machine):

A Support Vector Machine (SVM) is a powerful machine learning algorithm widely used for both linear and nonlinear classification, as well as regression and outlier detection tasks. SVMs are highly adaptable, making them suitable for various applications such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection.

SVMs are particularly effective because they focus on finding the maximum separating hyperplane between the different classes in the target feature, making them robust for both binary and multiclass classification. In this outline, we will explore the Support Vector Machine (SVM) algorithm, its applications, and how it effectively handles both linear and nonlinear classification, as well as regression and outlier detection tasks.

A **Support Vector Machine (SVM)** is a supervised machine learning **algorithm** used for both **classification** and **regression** tasks. While it can be applied to regression problems, SVM is best suited for **classification** tasks. The primary objective of the **SVM algorithm** is to identify the **optimal hyperplane** in an N-dimensional space that can effectively separate data points into different classes in the feature space. The algorithm ensures that the margin between the closest points of different classes, known as **support vectors**, is maximized.

The dimension of the hyperplane depends on the number of features. For instance, if there are two input features, the hyperplane is simply a line, and if there are three input features, the hyperplane becomes a 2-D plane. As the number of features increases beyond three, the complexity of visualizing the hyperplane also increases.

Mathematical Computation

Consider a binary classification problem with two classes, labelled as +1 and -1. We have a training dataset consisting of input feature vectors X and their corresponding class labels Y.

The equation for the linear hyperplane can be written as:

wTx+b=0*wTx*+*b*=0 → Equation 6

The vector W represents the normal vector to the hyperplane. i.e the direction perpendicular to the hyperplane. The parameter **b** in the equation represents the offset or distance of the hyperplane from the origin along the normal vector **w**.

The distance between a data point x_i and the decision boundary can be calculated as:

di=wTxi+b||w||*di*=||*w*||*wTxi*+*b* → Equation 7

where ||w|| represents the Euclidean norm of the weight vector w. Euclidean norm of the normal vector W

For Linear SVM classifier:

$$\hat{y} = \begin{cases} 1 & : \ w^T x + b \geq 0 \\ 0 & : \ w^T x + b \ < 0 \end{cases}$$

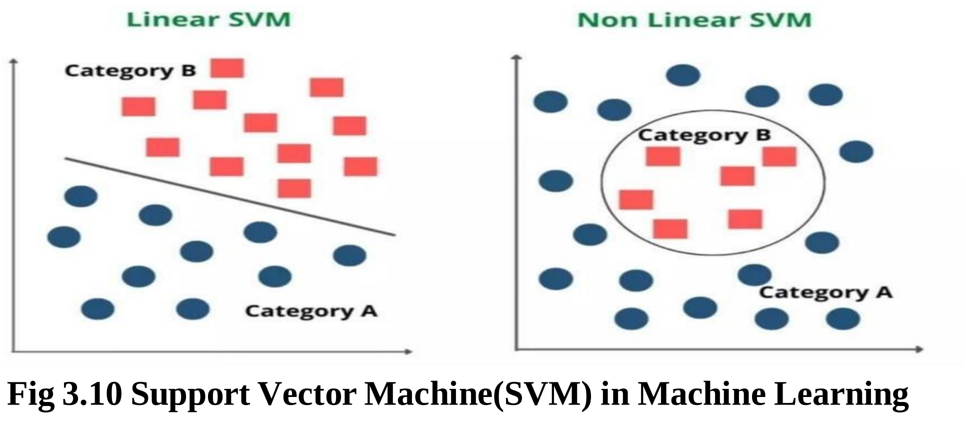


Fig 3.10 Support Vector Machine(SVM) in Machine Learning

In conclusion, Support Vector Machines (SVM) are powerful algorithms in machine learning, ideal for both classification and regression tasks. They excel at finding the optimal hyperplane for separating data, making them suitable for applications like image classification and anomaly detection.

SVM's adaptability through kernel functions allows it to handle both linear and nonlinear data effectively. However, challenges like parameter tuning and potential slow training times on large datasets must be considered.

Understanding SVM is crucial for data scientists, as it enhances predictive accuracy and decision- making across various domains, including data mining and artificial intelligence.

3.2.6.4 XGBoost (eXtreme Gradient Boosting):

Chenand Guestrinin introduced a powerful tree boosting algorithm, which is named as the eXtream Gradient Boosting algorithm. The algorithm is claimed to be scalable; sparsity-aware; takes into consideration data compression and sharding; and cache-aware access [2]. It belongs to the family of boosting algorithms, which are ensemble learning techniques that combine the predictions of multiple weak learners. XGBoost, or Extreme Gradient Boosting, is a state-of-the-art machine learning algorithm renowned for its exceptional predictive performance. It is the gold standard in ensemble learning, especially when it comes to gradient-boosting algorithms. It develops a series of weak learners one after the other to produce a reliable and accurate predictive model. Fundamentally, XGBoost builds a strong predictive model by aggregating the predictions of several weak learners, usually decision trees. It uses a boosting technique to create an extremely accurate ensemble model by having each weak learner after it correct the mistakes of its predecessors.

Key Features of XGBoost

1. **Regularization**: XGBoost includes regularization terms in the objective function to prevent overfitting and improve generalization.
2. **Handling Missing Data**: It employs a "Sparsity Aware Split Finding" algorithm to handle missing values effectively.
3. **Cache-Aware Access**: XGBoost optimizes memory access times during training by utilizing the CPU's cache memory.

Parameters in XGBoost

- **Learning Rate (eta):** An important variable that modifies how much each tree contributes to the final prediction. While more trees are needed, smaller values frequently result in more accurate models.
- **Max Depth:** This parameter controls the depth of every tree, avoiding overfitting and being essential to controlling the model’s complexity.
- **Gamma**: Based on the decrease in loss, it determines when a node in the tree will split. The algorithm becomes more conservative with a higher gamma value, avoiding splits that don’t appreciably lower the loss. It aids in managing tree complexity.
- **Colsample Bytree**: Establishes the percentage of features that will be sampled at random for growing each tree.
- **n_estimators**: Specifies the number of boosting rounds.
- **lambda (L2 regularization term) and alpha (L1 regularization term)**: Control the strength of L2 and L1 regularization, respectively. A higher value results in stronger regularization.
- **min_child_weight**: Influences the tree structure by controlling the minimum amount of data required to create a new node.
- **scale_pos_weight**: Useful in imbalanced class scenarios to control the balance of positive and negative weights.

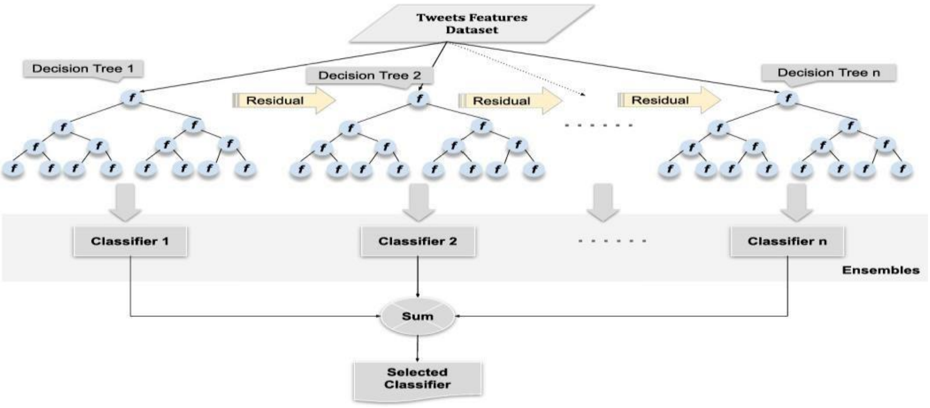


Fig 3.11 eXtreme gradient boosting (XGBoost)

Steps in XGBoost Algorithm:

1. Initialization: Start with a base prediction (e.g., the mean for regression or uniform probability for classification)
2. Compute Residuals: Calculate the gradient (negative derivative of the loss function) representing the residual errors.
3. Fit a Weak Learner: Train a decision tree to predict residuals (gradients).
4. Update Predictions: Add the weighted predictions of the new tree to the model's overall prediction.
5. Optimize the Objective: Minimize the loss function (e.g., mean squared error for regression, log-loss for classification).
6. Repeat: Continue iteratively adding trees until convergence or the maximum number of trees is reached.

XGBoost’s robustness is demonstrated by its novel approach to tree construction, cache-aware access, and handling of missing data. Because of its adaptability, scalability, and effectiveness, the algorithm is a great option for a variety of machine learning tasks, especially ones that require a large volume of training data.

3.3 Implementation:

3.3.4 Tools:

Python is a versatile and powerful programming language that offers a wide range of tools and libraries for various purposes, making it a popular choice among developers, data scientists, and researchers. Following are the libraries employed in our project.

Pandas

Pandas is a powerful data manipulation and analysis library that provides data structures like Data Frames to handle tabular data efficiently. It is widely used for data cleaning, preparation, and exploration.

NumPy

NumPy is the foundational package for numerical computing in Python. It offers support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

Scikit-Learn

Scikit-learn is a library for machine learning in Python that offers simple and efficient tools for data mining and data analysis. It includes a wide range of supervised and unsupervised learning algorithms.

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

re (Regular Expressions)

The re module provides support for regular expressions, which are powerful tools for pattern matching and text processing. Regular expressions allow you to search for, match, and manipulate strings based on complex patterns.

urllib.parse

The urllib.parse module is part of Python's standard library for URL manipulation. It provides functions for parsing URLs into components (such as scheme, netloc, path, etc.), and for constructing or modifying URL strings. This is useful for handling web addresses in web scraping, crawling, or networking applications.

socket

The socket module provides low-level networking interfaces, allowing you to create and manage network connections. It supports various types of network communication protocols, including TCP and UDP. This module is essential for tasks such as creating servers, clients, and managing network connections.

ipaddress

The ip address module provides the capability to create, manipulate, and operate on IPv4 and IPv6 addresses and networks. It is useful for tasks related to IP address validation, subnetting, and network configuration.

RandomForest Classifier

The Random Forest Classifier is a machine learning algorithm used for classification tasks. It is part of the ensemble learning methods, combining the predictions of multiple base estimators (usually decision trees) to improve the overall performance and robustness.

Tfidf Vectorizer

The Tfidf Vectorizer is a tool in natural language processing (NLP) that transforms text data into numerical representations using the Term Frequency-Inverse Document Frequency (TF-IDF) approach. This method helps in converting text data into a matrix of TF-IDF features, which can then be used for various machine learning tasks like classification, clustering, and more.

train_test_split

The train_test_split function in Scikit-learn is a useful tool for splitting a dataset into training and testing subsets. This allows you to train a machine learning model on one part of the data and
evaluate its performance on another part, ensuring that the model can generalize well to unseen data.

3.3.2 Features

Using IP:
Legitimate websites rarely use IP addresses directly in the URL because domain names are easier to remember and more professional. Phishing sites often use IP addresses to mask their identity and avoid detection by simple domain-based filters. The presence of an IP address in a URL is a strong indicator of a potential phishing attempt. Users should be wary of URLs with numeric IPs.

Long URL:

Phishing URLs tend to be longer to include misleading information and obscure the actual domain name. This length can be used to hide suspicious parts of the URL or to mimic legitimate URLs more closely. Long URLs may include additional directories, parameters, or random strings to evade detection. Always scrutinize lengthy URLs for legitimacy.

Short URL:

URL shorteners can be used by phishers to hide the true destination of a link, making it easier to disguise malicious content. While short URLs are convenient for sharing, they prevent users from seeing the full URL before clicking. Phishing attacks often leverage this by redirecting users to malicious sites. Verify shortened URLs through trusted preview tools before visiting.

Symbol @:

The '@' symbol in a URL can be used to obscure the true URL by creating a false sense of security. Everything before the '@' symbol is disregarded by the browser, which can trick users into thinking they are on a legitimate site. Phishing sites use this tactic to hide the real destination. Avoid clicking on URLs containing the '@' symbol.

Redirecting //:

Multiple instances of '//' in a URL path are unusual and can indicate redirection tricks used in phishing attacks. This can be a method to confuse users or manipulate the path structure. Legitimate URLs typically do not use double slashes beyond the initial protocol specification. Be cautious of URLs with unconventional structures.

Prefix/Suffix (-):

The presence of hyphens in the domain name (e.g., www.bank-example.com) is often a sign of phishing attempts. Cybercriminals use hyphens to create domains that look similar to legitimate ones. This can trick users into thinking they are on a trusted site. Verify the domain carefully, especially if it contains unusual hyphenation.

Subdomains:

Phishing sites often use multiple subdomains to make the URL appear similar to a legitimate one (e.g., www.login.bank.com.phishing.com). The actual domain may be misleadingly embedded within numerous subdomains. This tactic aims to confuse users and lend an air of authenticity. Always focus on the main domain when verifying URL authenticity.

HTTPS:

While HTTPS is a sign of security, many phishing sites now use HTTPS to appear legitimate. The presence of HTTPS alone is not a guarantee of safety, as obtaining a certificate has become relatively easy and inexpensive. Users should consider other factors in addition to HTTPS when evaluating a site's legitimacy. Always verify the certificate's validity and the site's overall trustworthiness.

Domain Registration Length:

Newly registered domains are often used in phishing attacks because they have not been blacklisted yet. Phishing sites tend to have a short domain age, as they are quickly set up and discarded. Checking the domain registration date can provide insights into its potential risk. Be cautious of domains registered very recently.

Favicon:

Inconsistency between the favicon of the site and that of the legitimate site can be a sign of phishing. Phishing sites may use generic or mismatched favicons, as creating an exact replica requires additional effort. The favicon is a small but significant detail that can indicate authenticity. Verify the favicon along with other site elements.

Non-standard Port:

Phishing websites may use uncommon ports to avoid detection by standard security measures. Legitimate websites typically use standard ports like 80 (HTTP) and 443 (HTTPS). Non-standard ports can be a red flag, indicating potential malicious activity. Avoid accessing websites that use unusual port numbers.

HTTPS in Domain URL:

Legitimate sites usually don’t include 'HTTPS' as part of the domain name itself; phishing sites may use it to trick users. This tactic exploits users' trust in HTTPS by embedding it within the domain
name. Always verify that 'HTTPS' is part of the protocol and not the domain. Scrutinize URLs carefully for such deceptive practices.

Anchor URL:

Links with misleading text can hide phishing URLs, making the link text appear legitimate while pointing to a malicious site. This tactic relies on users not checking the actual URL behind the link. Hover over links to see the real destination before clicking. Be cautious of anchor text that seems too enticing or urgent.

Links in Script Tags:

Legitimate sites rarely include clickable links in script tags; phishing sites might use this technique to evade detection. Embedding links within scripts can hide them from simple URL checks and scanners. Always inspect the source code of suspicious websites for hidden links. Avoid interacting with script-based links.

Server Form Handler:

Phishing sites often use external form handlers to collect user data, diverting the input data to malicious servers. This can bypass security measures that protect legitimate forms. Check the form action URL to ensure it points to a trusted domain. Be wary of forms that submit data to unfamiliar or unrelated domains.

Info Email:

Presence of email addresses in the URL can be indicative of phishing, as legitimate sites rarely include direct email links in URLs. This can be a tactic to harvest email addresses or redirect to malicious sites. Avoid clicking on URLs that contain email addresses. Use known and trusted communication channels for contact.

Abnormal URL:

URLs that deviate significantly from common patterns associated with legitimate websites can indicate phishing. This includes strange domain names, excessive parameters, or unusual structures. Anomalies in the URL should raise suspicion. Always cross-check suspicious URLs with official sources.

Website Forwarding:

Frequent redirections (more than 2) in the URL can indicate phishing, as attackers often use multiple forwards to obscure the final destination. This can also be a technique to evade security measures. Limit redirections and inspect the final URL carefully. Be cautious of sites that redirect multiple times.

Status Bar Customization:

Phishers may manipulate the status bar to hide the true destination of a link, making it appear legitimate. This can involve changing the status bar text or using JavaScript to display fake URLs. Always verify the actual link destination by inspecting the URL. Disable status bar customizations in browser settings.

Using Pop-up Window:

Phishing sites often use pop-up windows to capture user credentials, presenting them as legitimate login prompts. Pop-ups can bypass certain browser security features and trick users into entering sensitive information. Avoid entering credentials into pop-up windows. Use the main site interface for secure interactions.

Iframe Redirection:

Phishing sites may use iframes to load content from another site, hiding the true source of the content. This can be used to display legitimate-looking pages while capturing user data. Inspect the source code for iframes and avoid sites that use them suspiciously. Ensure the content is directly from the trusted domain.

Age of Domain:

Domains that have been registered recently are often used in phishing attacks because they have not been blacklisted yet. Checking the domain age can provide insights into its credibility. Older domains are generally more trustworthy. Use domain age lookup tools to verify the registration date.

DNS Recording:

Phishing sites may have incomplete or suspicious DNS records, such as missing MX records or mismatched WHOIS information. Analyzing DNS records can reveal potential red flags. Ensure the DNS setup is consistent with legitimate practices. Use DNS lookup tools to check the details.

Website Traffic:

Low website traffic can be an indicator of a phishing site, as legitimate sites tend to have higher and more consistent traffic. Traffic analysis can help determine the popularity and trustworthiness of a site. Use tools like Alexa or Similar. Web to check website traffic. Be cautious of sites with minimal or erratic traffic patterns.

Pagerank:

Phishing sites usually have a low PageRank, indicating they are not well-trusted or linked by reputable sites. PageRank measures the importance of a site based on inbound links. Check the PageRank to assess site credibility. Avoid sites with unusually low PageRank scores.

Google Index:

Legitimate sites are typically indexed by Google, while phishing sites may not be. Checking if a site is indexed can provide a quick credibility assessment. Use the "site:" search operator in Google to see if the site appears in search results. Be cautious of sites that are not indexed.

Links Pointing to Page:

Few or no backlinks can be a sign of a phishing site, as legitimate sites usually have many inbound links from reputable sources. Backlink analysis can help assess the site's reputation. Use tools like Ahrefs or Moz to check backlinks. Be wary of sites with minimal or suspicious backlink profiles.

Stats Report:

Analyzing various statistics and reports, such as WHOIS information, SSL certificates, and historical data, can provide a comprehensive view of the site's legitimacy. Use multiple sources to gather and verify information. A thorough stats report can reveal inconsistencies and potential red flags. Always cross-reference data from trusted tools and databases.

3.3.3 Classification Model

To develop a robust classification model, we first fine-tune the model’s settings, known as classifier parameters, to achieve the best possible performance. This process is akin to adjusting a recipe’s ingredients and baking time to perfect it. Next, we select the most relevant features, or pieces of information, from the dataset that will help the model make accurate predictions. This is similar to choosing the best ingredients for a dish.

By maximizing the objective function, which is a measure of the model’s performance, we ensure the model is as accurate as possible. This objective function could be accuracy, precision, recall, or another metric suitable for the problem at hand. Using these optimized parameters and carefully selected features, we train the classification model, ensuring it is robust.

A robust model performs well even when faced with new, unseen data, making it reliable and consistent in various situations. In summary, this comprehensive approach of fine-tuning

parameters, selecting relevant features, and maximizing performance measures ensures the development of an accurate and efficient classification model [2].

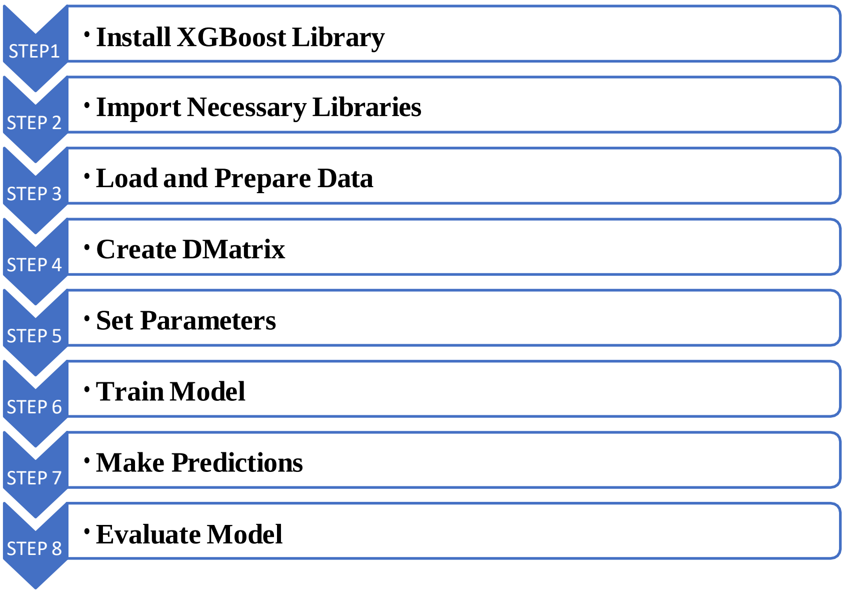


Fig 3.12 Flowchart of XGBoost algorithm model development

Objective Function

The **objective function** in XGBoost is like a scorecard that helps the algorithm decide how good a model is. This scorecard has two main parts:

1. **Loss Function**: This measures how far off the model's predictions are from the actual values. It's like checking how close your dart is to the bullseye on a dartboard. The smaller the difference, the better the score. The loss function measures how well the model's predictions match the actual data. This is a crucial part of the objective function, as it helps the algorithm understand how to improve its predictions.

Simplified Formula:

$$\sum_{i=1}^{n} l(y_{i,}^{\wedge} y_i) \longrightarrow \text{Equation 8}$$

Breaking Down the Formula:

- $\sum_{i=1}^{n}$: This symbol means "sum up" the following terms for all instances i from 1 to n (where n is the total number of instances in the dataset).

- $l(\hat{y}_i, y_i)$: This represents the loss function applied to the predicted value yi and the actual value yi for the i-th instance.

2. **Regularization Term**: This part prevents the model from becoming too complex and overfitting (like memorizing the training data rather than learning to generalize). It's like adding a rule that you can't use too many darts to hit the bullseye; you need to hit it in fewer, well- placed throws.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \longrightarrow \text{Equation 9}$$

- γT:

  - γ: A parameter that controls the penalty for the number of leaves in the tree. Higher values of γ\gamma will make the model simpler by discouraging large trees.

  - T: The total number of leaves in the tree. Each leaf represents a decision point in the tree.

- $\frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$:

  - λ: A parameter that controls the penalty for the leaf weights. Higher values of λ\lambda will result in smaller weights, helping to prevent overfitting.

  - $\sum_{j=1}^{T} w_j^2$: The sum of the squares of the weights of all the leaves. Each 'wj' represents the weight of a leaf.

When building the model, XGBoost tries to minimize (make as small as possible) this objective function. It wants the model to predict accurately (small loss) while also keeping it simple (controlled by regularization).

So, in simple terms:

- **Loss Function** = Measures prediction error.
- **Regularization Term** = Keeps the model simple to avoid over fitting.
- **Objective Function** = The combination of these two to guide the model building process.

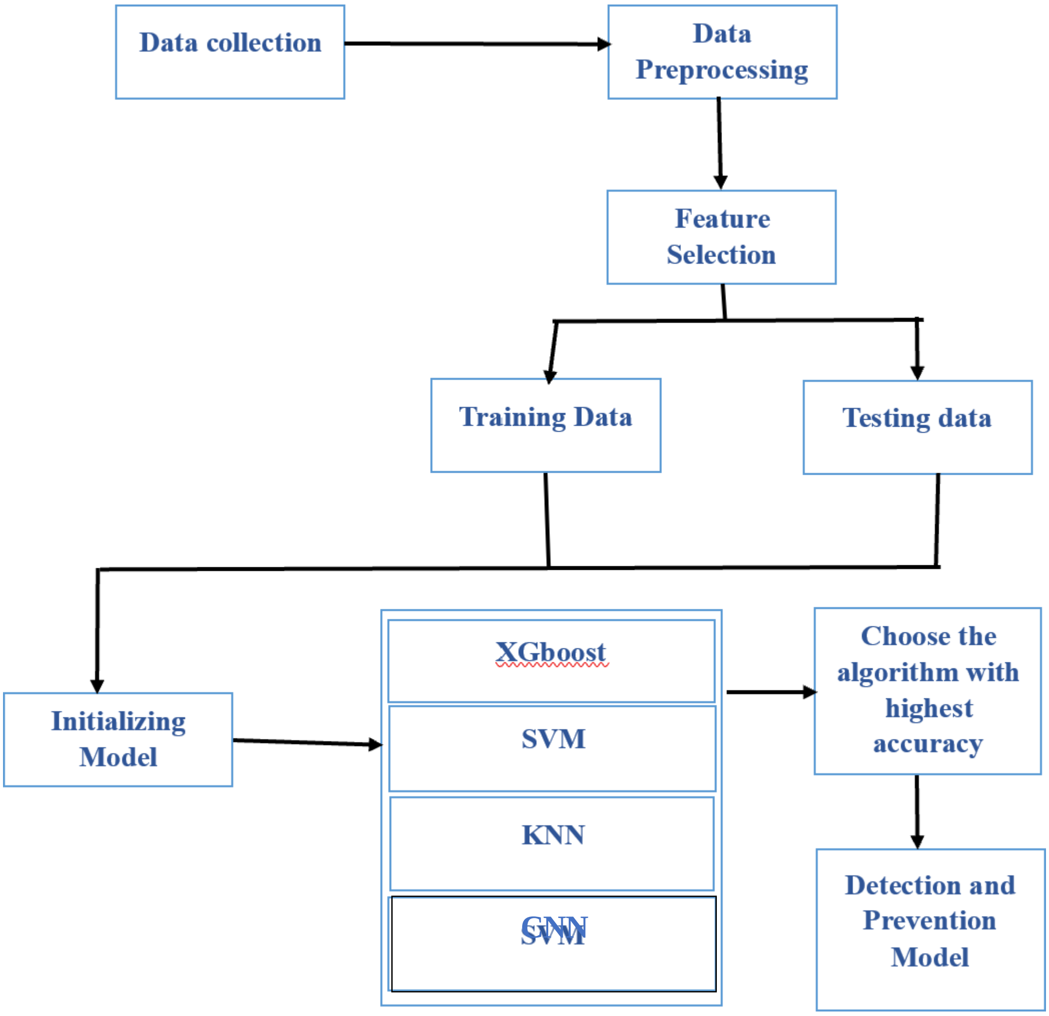


Fig 3.13 Flowchart of Implementation

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Results of the proposed work:

The Result section encompasses a correlation matrix displaying the relationships between variables, bar graphs illustrating the accuracy of implemented algorithms, and a detailed analysis of the accuracy outcomes achieved by the implemented algorithms. Additionally, snapshots of the proposed work are provided to offer visual insights into the study's findings.

4.1.1 Correlation matrix:

Correlation matrix displays the relationships between various elements within a set. Different aspects like how correlation heatmap reveals intricate relationships between various website features and the target variable ‘class’. The table provides a visual representation of the similarities or connections between these items. If the number is close to 1, it indicates a high degree of correlation between them, if a number close to -1 signifies an inverse relationship between two variables, where one increases as the other decreases. If it's close to 0, there is hardly any connection. So, when we look at this table, we can see which things are connected in our data, helping us understand how they impact each other.

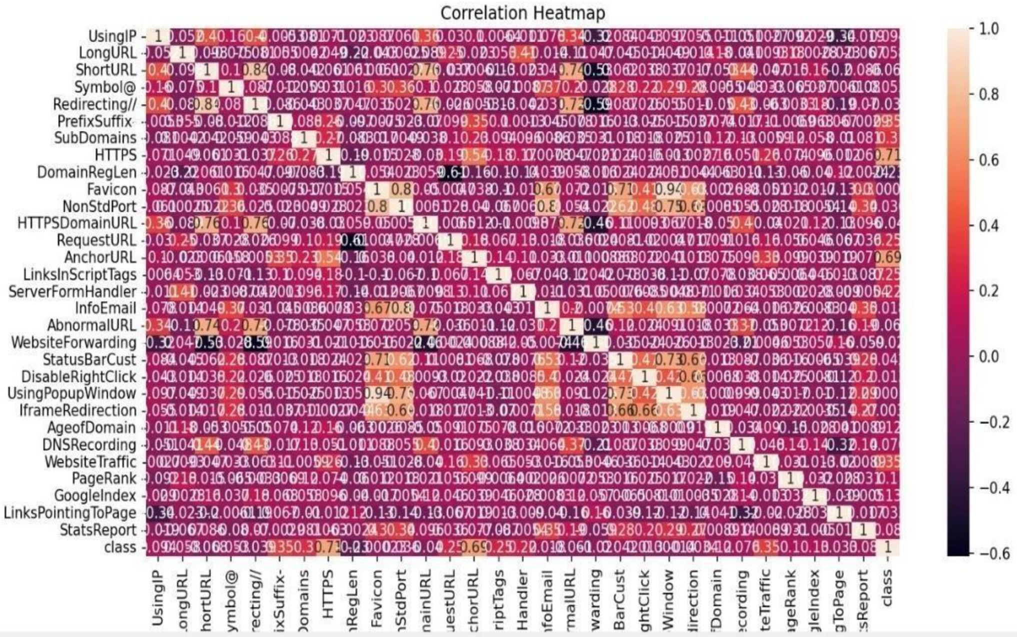


Fig 4.1 Correlation matrix for phishing website prediction

4.1.2 Bar graph accuracy of implemented algorithms:

Bar graph illustrates the performance of different machine learning models(XGBoost, CNN, KNN, SVM) in terms of accuracy is shown in Figure 4.2,showing both the highest and lowest achieved accuracies. The results indicate that XGBoost achieved the highest accuracy, followed by SVM and CNN, with KNN demonstrating the lowest accuracy in this comparison.

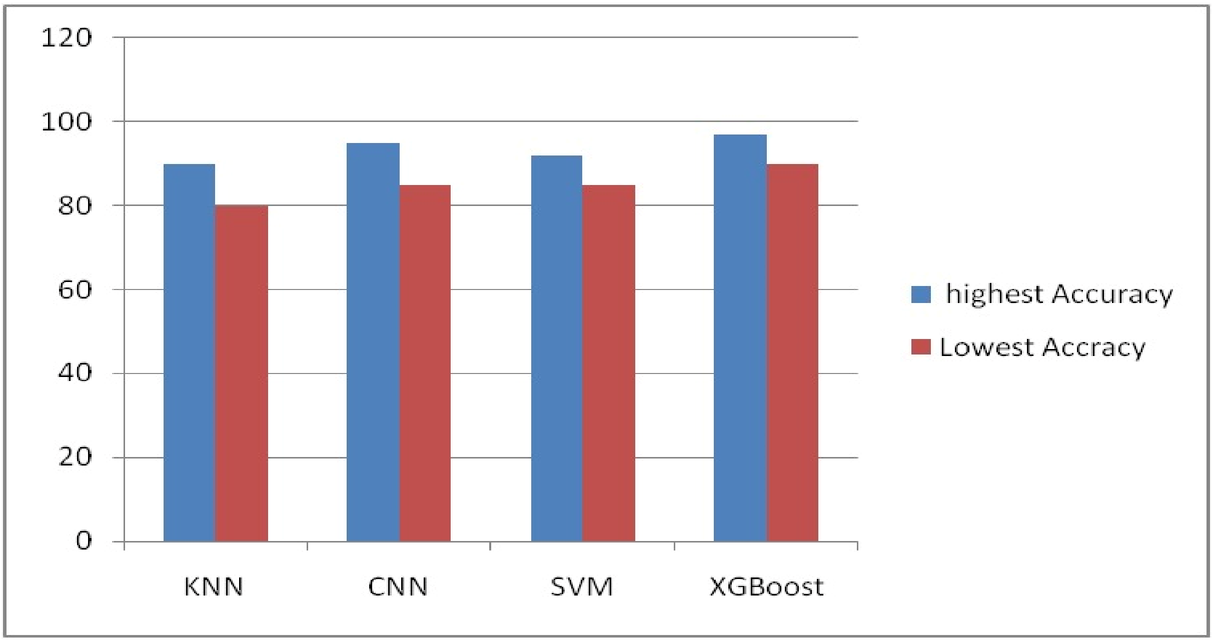


Fig 4.2 Bar graph accuracy of machine learning algorithms

4.1.3 Accuracy outcomes of machine learning algorithms:

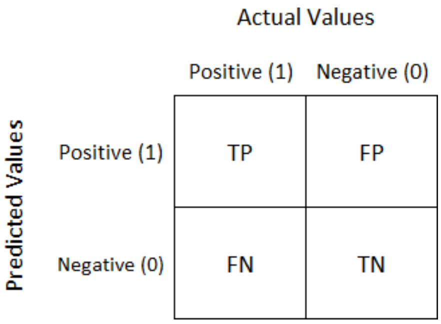
Accuracy outcomes of machine learning algorithms is depicted in Table 4.1, where XGBoost, KNN, SVN, CNN models were evaluated across multiple website features.In all phishing website detection conditions, XGBoost usually performed better than KNN, CNN and SVM.The results suggest that XGBoost may be the most suitable model for the phishing website detection due to its superior accuracy.

Table 4.1 Accuracy outcomes of machine learning algorithms

Algorithms	Highest Accuracy	Lowest Accuracy
KNN	**90**	**80**
CNN	**95**	**85**
SVM	**92**	**85**
XGBoost	**97**	**90**

4.1.4 Confusion Matrix:

Confusion matrix(CM) is a graphical summary of the correct predictions and incorrect predictions that is made by a classifier that can be used to determine the performance. In abstract terms, the CM is as shown in fig 4.3:



Predicted Values	Actual Values Positive (1)	Actual Values Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Fig 4.3 Confusion matrix

In the above figure TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

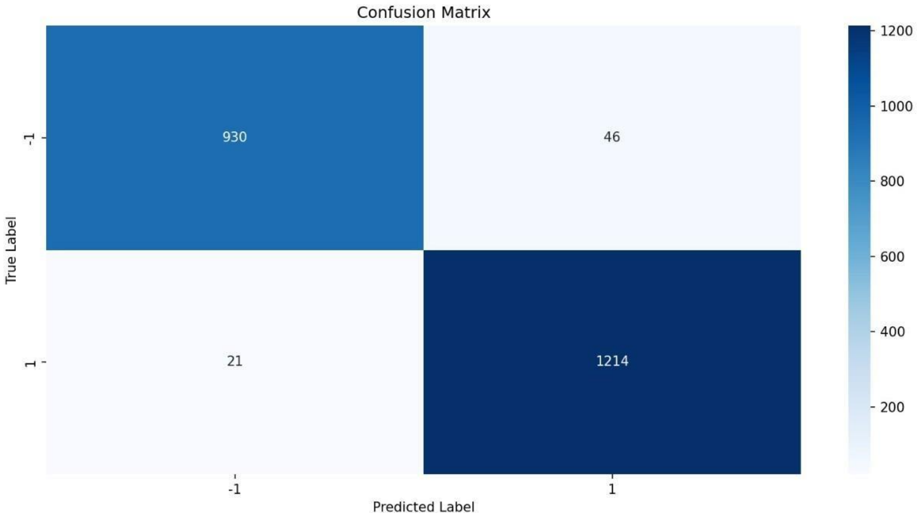
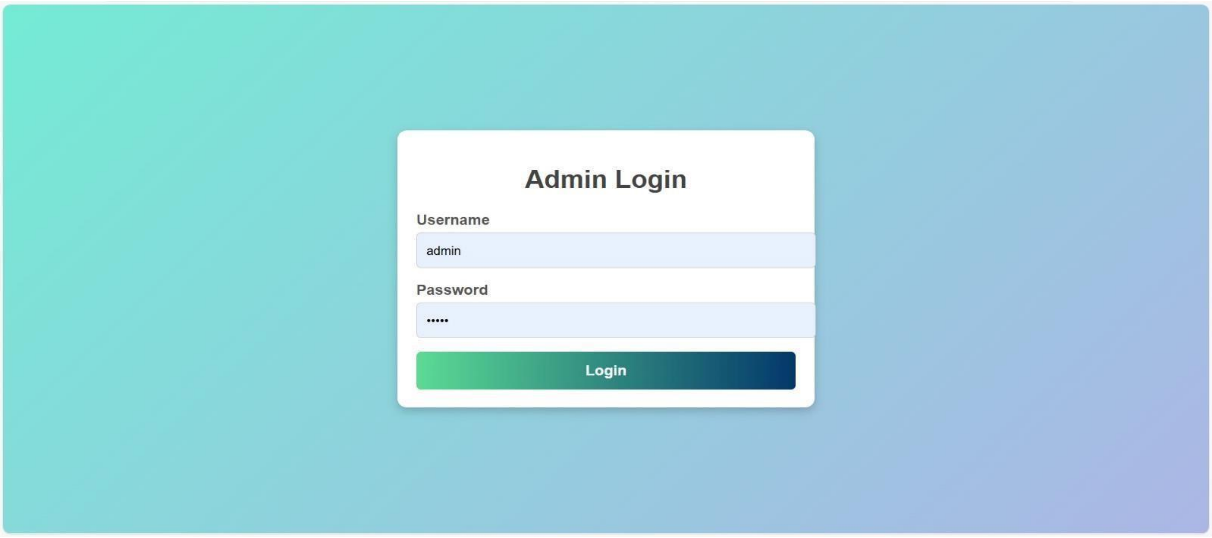


Fig 4.4 XGBoost Confusion matrix

4.1.5 Snapshots of the proposed work:

Admin Login page:

Fig 4.5 represents the Admin Login page where it allows the user to enter the username and password.



Admin Login

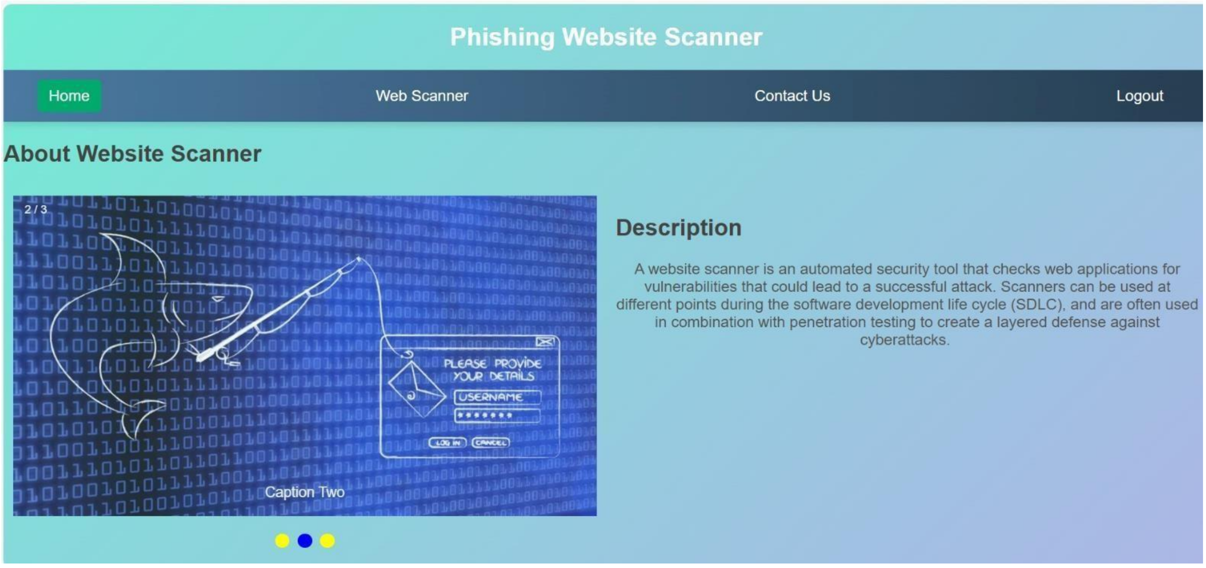
Username: admin

Password: •••••

Fig 4.5 Admin Login page

Home page:

Figure 4.6 represents the Home page where it consists of description about website scanner.



Phishing Website Scanner

About Website Scanner

Description

A website scanner is an automated security tool that checks web applications for vulnerabilities that could lead to a successful attack. Scanners can be used at different points during the software development life cycle (SDLC), and are often used in combination with penetration testing to create a layered defense against cyberattacks.

Fig 4.6 Home page

Website Prediction page:

Fig 4.7 represents the website predictor page which allows users to enter relevant data.The users input there data and press “check here” to determine whether the entered website is phishing or legitimate.

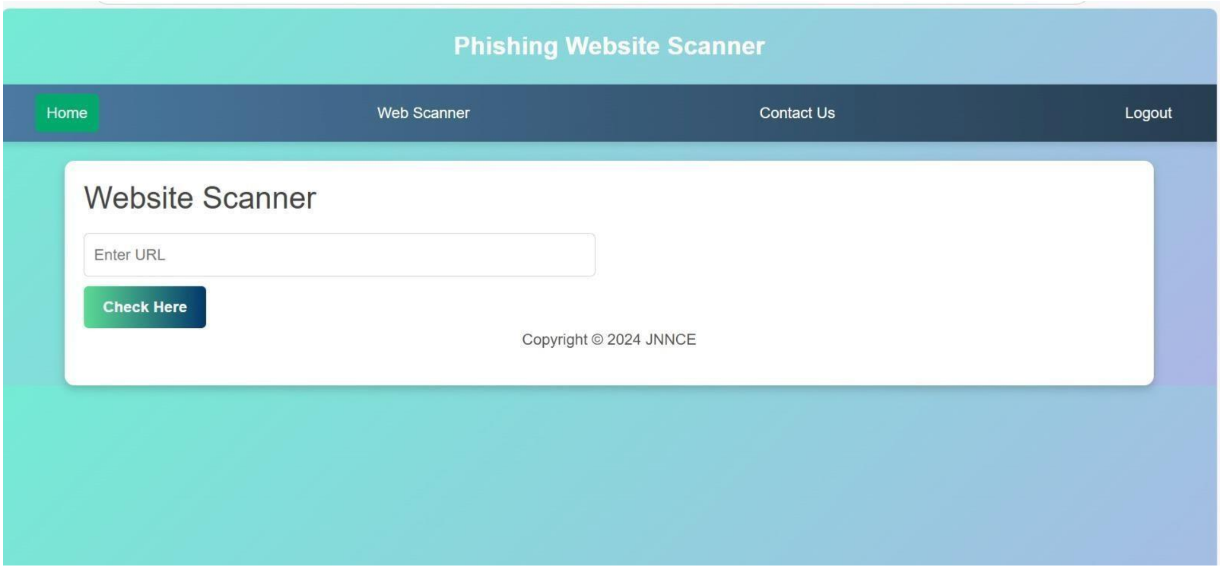
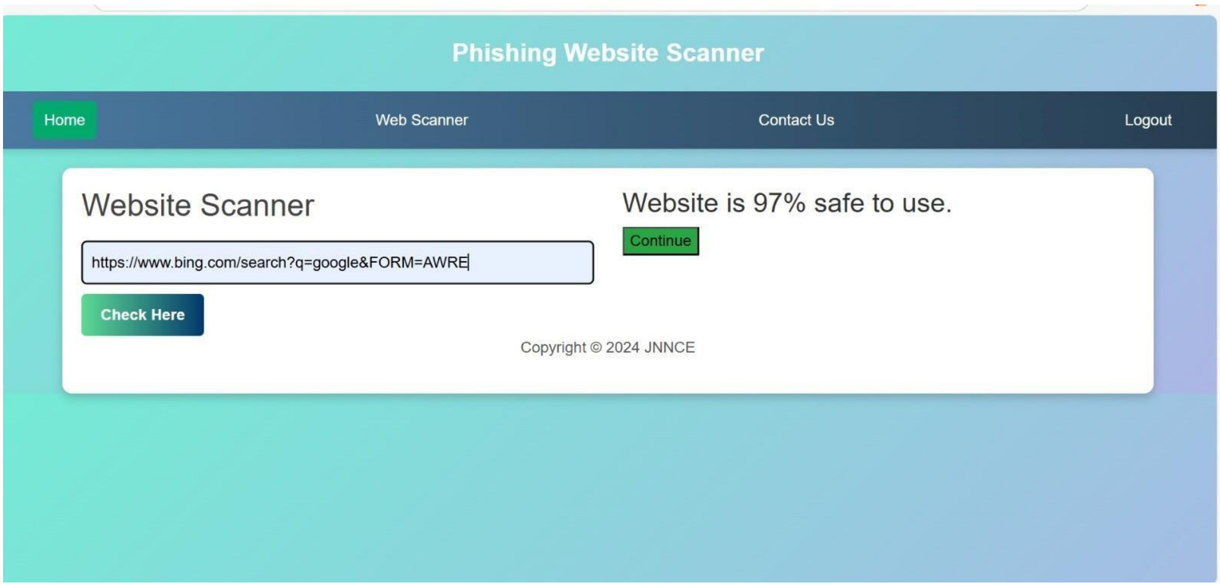


Fig 4.7 Website Prediction page

Detection of legitimate website:

Fig 4.8 represents the detection of legitimate website ,after the user have entered the URL in predictor page it will predict legitimate website based on the features.



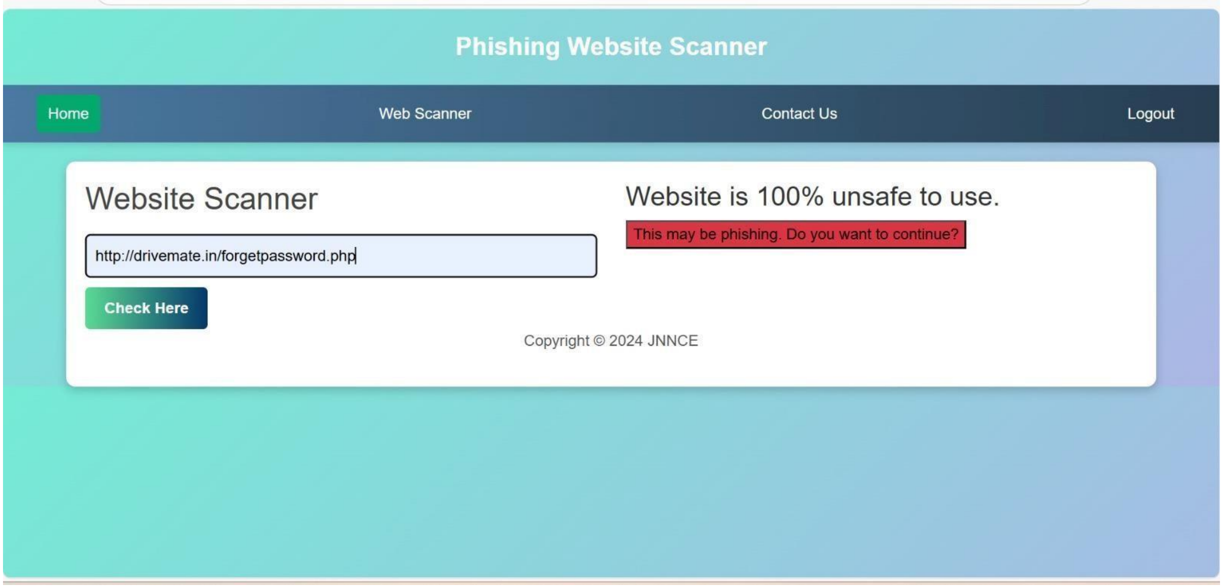
https://www.bing.com/search?q=google&FORM=AWRE

Website is 97% safe to use.

Fig 4.8 Detection of legitimate website

Detection of phishing website:

Fig 4.9 represents the detection of phishing website ,after the user have entered the URL in predictor page it will predict phishing website based on the features and alerts the user that the website is unsafe and asks the user whether they want to continue although it is unsafe.



Website Scanner

http://drivemate.in/forgetpassword.php

Website is 100% unsafe to use.

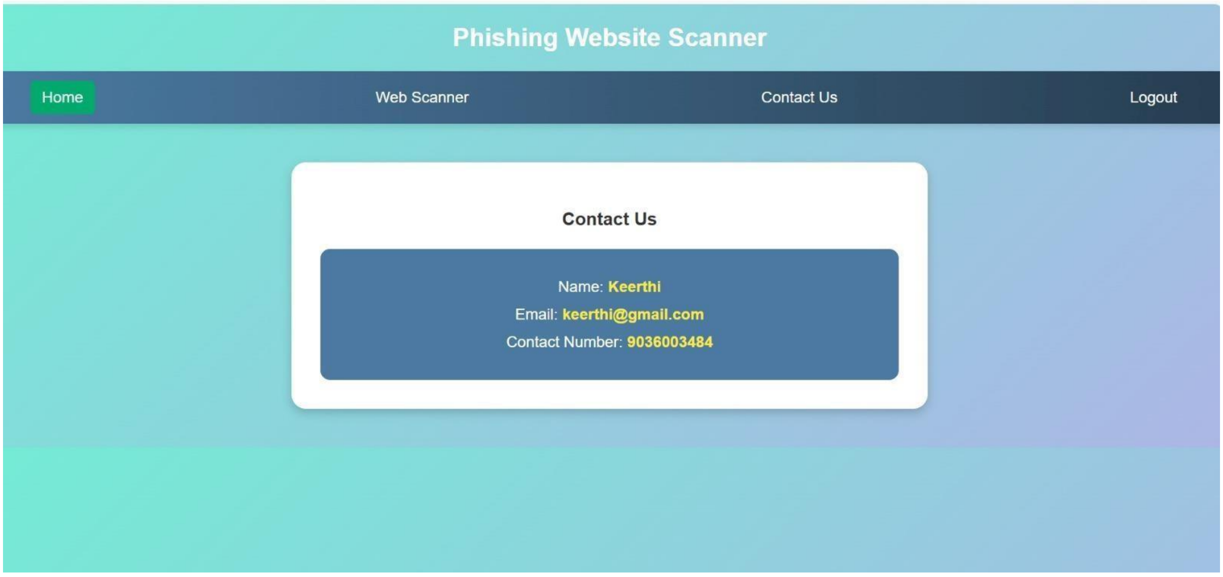
This may be phishing. Do you want to continue?

Copyright © 2024 JNNCE

Fig 4.9 Detection of phishing website

Contact Us page:

Fig 4.10 represents the contact us page, where the admin details is stored like admin name, email and contact number through which the user can contact the admin.



Contact Us

Name: Keerthi
Email: keerthi@gmail.com
Contact Number: 9036003484

Fig 4.10 Contact Us page

CHAPTER 5

CONCLUSION

The demonstration of phishing is turning into an advanced danger to this quickly developing universe of innovation. Today, every nation is focusing on cashless exchanges, business online, tickets that are paperless and so on to update with the growing world. Yet phishing is turning into an impediment to this advancement. Individuals are not feeling web is dependable now. The project means to investigate this region by indicating an utilization instance of recognizing phishing site utilizing ML. It is aimed to build the a phishing detection mechanism using machine learning tools and techniques which is efficient, accurate and cost effective.The project was carried out in the pycharm and was written in python. The proposed method used for machine learning classifiers to achieve this and a comparative study of algorithms was made. Out of all the classifiers XGBoost algorithm has the best accuracy score of 97%. This model can be deployed in real time to detect the URLs as phishing or legitimate.

Future scope of project:

Further work can be done to enhance the model by using ensembling models to get greater accuracy score. Ensemble methods is a ML technique that combines many base models to generate an optimal predictive model. Further reaching future work would be combining multiple classifiers, trained on different aspects of the same training set, into a single classifier that may provide a more robust prediction than any of the single classifiers on their own.

The project can also include other variants of phishing like smishing, vishing, etc. to complete the system. Looking even further out, the methodology needs to be evaluated on how it might handle collection growth. The collections will ideally grow incrementally over time so there will need to be a way to apply a classifier incrementally to the new data, but also potentially have this classifier receive feedback that might modify it over time.

REFERENCES

[1]. M. Patil, N. Shivsharan, Y. Naik, H. Yeram and A. Gawade, "Enhancing Cybersecurity: A Comprehensive Analysis of Machine Learning Techniques in Detecting and Preventing Phishing Attacks with a Focus on Xgboost Algorithm," *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, Gurugram, India, 2024

[2]. N. Ghatasheh, I. Altaharwa and K. Aldebei, "Modified Genetic Algorithm for Feature Selection and Hyper Parameter Optimization: Case of XGBoost in Spam Prediction," in *IEEE Access*, vol. 10, pp. 84365-84383, 2022

[3]. L. Allodi, T. Chotza,E.Panina and N. Zannone, "The Need for New Antiphishing Measures Against Spear-Phishing Attacks," in *IEEE Security & Privacy*, vol. 18, no. 2, pp. 23-34, March-April 2020

[4]. M. Sameen, K. Han and S. O. Hwang, "PhishHaven-An Efficient Real-Time AI Phishing URLs Detection System," in *IEEE Access*, vol. 8, pp. 83425-83443, 2020

[5]. J. Mao, W. Tian, P. Li, T. Wei and Z. Liang, "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity," in *IEEE Access*, vol. 5, pp. 17020-17030, 2017

[6]. A. Mandadi, S. Boppana, V. Ravella and R. Kavitha, "Phishing Website Detection Using Machine Learning," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai,

India, 2022, pp. 1-4

[7]. P. T. Duy, V. Q. Minh, B. T. H. Dang, N. D. H. Son, N. H. Quyen and V. -H. Pham, "A Study on Adversarial Sample Resistance and Defense Mechanism for Multimodal Learning-Based Phishing Website Detection," in IEEE Access, vol. 12, pp. 137805-137824, 2024

[8]. F. Castaño, E. F. Fernañdez, R. Alaiz-Rodríguez and E. Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," in *IEEE Access*, vol. 11, pp. 40779-40789, 2023

[9]. M. Almousa and M. Anwar, "A URL-Based Social Semantic Attacks Detection With Character-Aware Language Model," in *IEEE Access*, vol. 11, pp. 10654-10663, 2023,

[10]. H. Shirazi, S. R. Muramudalige, I. Ray, A. P. Jayasumana and H. Wang, "Adversarial Autoencoder Data Synthesis for Enhancing Machine Learning-Based Phishing Detection Algorithms," in IEEE Transactions on Services Computing, vol. 16, no. 4, pp. 2411-2422, 1 July-Aug. 2023

[11]. L. Jovanovic *et al.*, "Improving Phishing Website Detection using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning," in *Journal of Web Engineering*, vol. 22, no. 3, pp. 543- 574, May 2023

[12]. N. Megha, K. R. Remesh Babu and E. Sherly, "An Intelligent System for Phishing Attack Detection and Prevention," *2019 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2019, pp. 1577-1582

[13]. M. A. Ivanov, B. V. Kliuchnikova, I. V. Chugunkov and A. M. Plaksina, "Phishing Attacks and Protection Against Them," 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia

[14]. H. Wen, J. Fang, J. Wu and Z. Zheng, "Hide and Seek: An Adversarial Hiding Approach Against Phishing Detection on Ethereum," in *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 3512-3523, Dec. 2023

[15]. A. El Aassal, S. Baki, A. Das and R. M. Verma, "An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs," in *IEEE Access*, vol. 8, pp. 22170-22192, 2020

[16]. M. J. Pillai, S. Remya, V. Devika, S. Ramasubbareddy and Y. Cho, "Evasion Attacks and Defense Mechanisms for Machine Learning-Based Web Phishing Classifiers," in IEEE Access, vol. 12, pp. 19375-19387, 2024, doi: 10.1109/ACCESS.2023.3342840

[17]. C.N.Gutierrez *et al*.,"Learning from the Ones that Got Away: Detecting New Forms of Phishing Attacks," in *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 988-1001, 1 Nov.-Dec. 2018

[18]. E.Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," in *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332-1344, June 2023