Phishing Detection Site Using ML-Algorithms

Aayush Gupta
Department of Computer Science & Engineering (AI & ML)
Dayananda Sagar University
Bengaluru, India
aayushgupta860@gmail.com

Abinash Dubey
Department of Computer Science & Engineering (AI & ML)
Dayananda Sagar University
Bengaluru, India
dabinash318@gmail.com

Aman Anand
Department of Computer Science & Engineering (AI & ML)
Dayananda Sagar University
Bengaluru, India
info.amananand@gmail.com

K Chakradhar Naidu.
Department of Computer Science & Engineering (AI & ML)
Dayananda Sagar University
Bengaluru, India
chakradhar2010.cse20@gmail.com

Prof Raja Lakshmi
Department of Computer Science & Engineering (AI & ML)
Dayananda Sagar University
Bengaluru, India

***Abstract*— To combat the growing threat of online phishing scams, this project investigates the effectiveness of machine learning (ML) algorithms in accurately identifying malicious websites. Phishing attacks pose a significant threat to user privacy and security, causing financial losses and identity theft. Traditional detection methods struggle to keep pace with evolving scams. ML offers a promising approach due to its ability to learn from large datasets and adapt to new patterns. Existing research has explored various ML algorithms for phishing detection, with supervised learning methods like Support Vector Machines (SVM) and Random Forests demonstrating promising results. However, the optimal choice and feature selection remain open questions**

Keywords—HTTPS,SVM,ML-MODEL,URL

1. INTRODUCTION

In the digital era, technology and the internet offer numerous benefits, yet they also pose cybersecurity challenges, notably the rise of phishing attacks. This project introduces a groundbreaking solution using machine learning (ML) to detect phishing websites in real time. Traditional methods struggle to keep up with evolving tactics, prompting us to move beyond static rules.

Our ML model analyzes diverse features such as URL structure, SSL certificates, domain age, user behavior, and content for anomalies. By holistically considering these factors, our system provides a comprehensive defense against phishing. The project focuses on a binary classification task—malicious versus benign websites. We employ Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) to enhance real-time detection, leveraging their capacity to store and process data efficiently.

This project aims to proactively identify and mitigate phishing risks by integrating advanced ML algorithms. The scalable architecture ensures adaptability to the growing digital threat landscape. The significance lies not only in immediate accuracy but also in future-proofing against emerging threats. Our approach represents a paradigm shift in phishing detection, contributing to a safer online environment. Subsequent chapters will detail methodologies, ML algorithm selection, data collection, and the potential impact on cybersecurity measures, emphasizing the project's contribution to a secure digital future.

2. PROBLEM DEFINITION AND OBJECTIVES

2.1 PROBLEM

Phishing attacks pose a significant and evolving threat to individuals and organizations, as malicious actors create deceptive websites to steal sensitive information. Traditional rule-based and signature-based methods are insufficient, struggling to adapt to the dynamic tactics employed by phishers. There is a pressing need for a more effective and adaptive detection mechanism to safeguard against the growing sophistication of phishing attacks

2.2 SOLUTION

This project addresses the need for a proactive and adaptive solution by leveraging machine learning (ML) algorithms to detect phishing websites. Unlike traditional rule-based methods, ML offers the capability to analyze website features and user behavior patterns dynamically, allowing for a more accurate and responsive detection system. The proposed solution aims to move beyond static approaches, providing a comprehensive defense against phishing attacks by continuously learning from emerging threats and evolving tactics

2.3 OBJECTIVE

1. Develop a machine learning-based system capable of autonomously analyzing website features and user behavior patterns.
2. Enhance phishing detection accuracy by moving beyond traditional rule-based methods to adapt to the dynamic nature of cyber threats.
3. Create a proactive system that can identify emerging phishing tactics and patterns in real time.
4. Implement a scalable architecture to accommodate the increasing volume and complexity of phishing attacks in the digital landscape.
5. Provide a user-friendly interface for organizations and individuals to easily integrate and utilize the phishing detection system.
6. Contribute to the overall improvement of cybersecurity measures by offering a more advanced and effective defense against phishing attacks.
7. Evaluate and validate the effectiveness of the ML-based phishing detection system through rigorous testing and real-world scenarios.

LITERATURE SURVEY

The project "phishing site detection using ML algorithms" reveals a growing body of research focused on leveraging machine learning (ML) to enhance the identification of phishing websites. Various methodologies have been explored, including intelligent phishing detection using fuzzy data mining and the application of ML approaches tailored to detecting phishing attacks. Researchers have investigated discrepancies within website identity, structural features, and HTTP transactions to develop effective models for identifying mock websites. The literature highlights the significance of identity and feature extraction processes, and several studies detail experiments conducted to assess the performance of ML models in phishing site detection. Commonly employed supervised learning algorithms in this context include Multi-layer Perceptron (MLP), Decision Tree Induction (DT), and Naïve Bayes (NB) classification. Overall, the literature underscores the promising role of ML algorithms in proactively and accurately identifying phishing websites, contributing to advancements in cybersecurity.

[1] Phishing, a prevalent cybercrime, involves creating deceptive websites to steal sensitive user information. Traditional blacklist-based methods fall short against sophisticated attacks, prompting increased research into machine learning (ML) for automated phishing website detection. Studies, such as Mohammad et al. (2014), proposed a self-structuring neural network with 95% accuracy. Altyeb Taha (2021) developed a 98% accurate ensemble model using weighted soft voting. Sahoo et al. (2017) surveyed ML techniques, including SVM and neural networks, for malicious website detection. Maini et al. (2021) compared eight ML algorithms, with an ensemble model outperforming individual ones, especially XGBoost. Bentéjac et al. (2021) analyzed gradient boosting algorithms like XGBoost, LightGBM, and CatBoost for phishing detection based on URL and content features. These studies collectively highlight the efficacy of ML in addressing the evolving challenges of phishing detection.

[2] Explores the use of machine learning for phishing website detection due to the limitations of traditional blacklisting against advanced phishing attacks. Various studies, including Mohammad et al. (2014) and Sahoo et al. (2017), have applied machine learning models such as neural networks and SVM for effective detection. Alarbi et al. (2020) proposed a multilayer perceptron (MLP) model using content, URL, and domain-based features, achieving 99.1% accuracy. Maini et al. (2021) highlighted the effectiveness of ensemble learning with XGBoost, outperforming individual models. Key techniques include neural networks, ensemble learning, and tree-based models, with feature selection to enhance relevance and reduce overfitting. Evaluation metrics such as accuracy, precision, recall, and F1-score are crucial, emphasizing the importance of minimizing false negatives for reliable phishing detection.

[3] Phishing, a significant cyber threat, involves creating deceptive websites to steal sensitive information, leading to billions in financial losses. Researchers focus on machine learning (ML) for automatic phishing website detection. Aburrous et al. (2008) achieved 83.7% accuracy using fuzzy logic, while Pan and Ding (2006) employed SVM with 84% accuracy. Xiang and Hong (2009) combined identity discovery and keyword retrieval for 89% accuracy. Altaher (2017) proposed a KNN-SVM hybrid model with 90.04% accuracy, and Maini et al. (2021) favored ensemble methods, with XGBoost performing best. SVM, ensemble models,

[4] Provides a comprehensive summary of methodologies employed in identifying phishing websites, encompassing intelligent phishing detection using fuzzy data mining, machine learning for phishing attack detection, and scrutiny of discrepancies in website identity, structural features, and HTTP transactions. The paper delves into identity and feature extraction processes, detailing experiments conducted to evaluate model performance. Supervised learning algorithms, specifically Multi-layer Perceptron (MLP), Decision Tree Induction (DT), and Naïve Bayes (NB) classification, are employed for learning purposes in the context of phishing website detection.

[5] Improving Phishing Website Detection Using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning gives information on machine learning approaches for phishing website detection. It introduces a hybrid framework employing the eXtreme Gradient Boosting (XGBoost) model, optimized through an enhanced metaheuristics algorithm. The paper highlights the significance of feature selection and XGBoost hyper-parameter tuning. Through extensive experiments on three prominent phishing website datasets, the hybrid model demonstrates superior performance compared to other methods, positioning it as a promising solution for enhancing web security.

[6] Phishing website detection using machine learning techniques reveals insights into several approaches. A pruned decision tree demonstrated the highest accuracy at 90.39%, emphasizing the potential for ensemble methods or increased feature sets to enhance performance. Support Vector Machines (SVM) achieved 86.69% accuracy, suggesting parameter tuning or alternative kernel functions for improvement. Naïve Bayes' Classifier, with 86.14% accuracy, faced challenges due to discrete feature values, suggesting potential enhancement through deep learning techniques. Neural networks, with 84.87% accuracy, also highlighted the need for improved decision boundaries. Associative classification achieved an accuracy of 91.5% by generating rules from frequent item sets. Fuzzy Inference System, using fuzzy membership functions, offered another avenue for rule extraction and optimization. Future research could explore these techniques with larger datasets, additional features, and comparative analyses with other classification methods.

4.METHODOLOGY

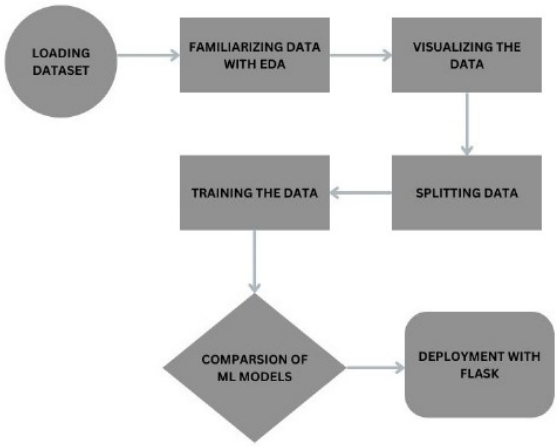
The proposed methodology with the six steps is a solid foundation, here's an alternative methodology incorporating additional considerations and potential optimizations:

Load the dataset: This involves bringing your data into your chosen environment for analysis. This could be a CSV file, a database, or an API.

Familiarize the data with EDA: Exploratory data analysis (EDA) is a crucial step in any machine learning project. It involves getting to know your data by looking at its basic statistics, distributions, and relationships between variables. This helps you understand what you're working with and identify any potential problems.

Train the data: This is where you build your machine learning model. You'll need to choose a model that's appropriate for your task (e.g., regression, classification), and then train it on your data. This involves feeding the model your data and letting it learn from it.

Splitting data: Before you can train your model, you need to split your data into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance on unseen data. This helps to prevent overfitting,
which is when your model memorizes the training data but doesn't generalize well to new data.
Comparison of ML models: This step involves training and comparing different machine learning models to find the one that performs best on your task. You can use metrics like accuracy, precision, and recall to evaluate the performance of your models.
Deployment with Flask: Once you've found the best model, you can deploy it into production using a web framework like Flask. This allows you to use your model to make predictions on new data.



5. EXPERIMENTATION

In the experimentation phase of the "Phishing Site Detection using Machine Learning Algorithms" project, critical steps are taken to fine-tune models and ensure the system's robustness. Algorithm selection involves training and evaluating various ML algorithms, comparing their performance metrics such as accuracy and precision. Benchmarking against established models situates the project within the broader context of phishing detection. Feature engineering explores the impact of different dataset features and transformation techniques on model performance. Hyperparameter tuning optimizes algorithm parameters for optimal results. Cross-validation techniques like K-Fold and Stratified Cross-validation ensure robust model generalizability. Real-world testing involves deploying a prototype system, incorporating the best-performing model and features, and assessing effectiveness on a small set of phishing and legitimate websites. User feedback, if applicable, guides further refinement. Thorough documentation of experiments facilitates comprehensive analysis, enabling insights into system strengths and weaknesses for continuous improvement and enhanced user protection against online scams.

6. RESULT AND ANALYSIS

The performance metrics of the nine models are analyzed and the best-performing model is chosen to be deployed which is XGBOOST.
The provided performance metrics for various machine learning models offer valuable insights into their effectiveness on a given task. Firstly, Logistic Regression, K-Nearest Neighbor, Support Vector Machine (SVM), Decision Tree, Random Forest, Gradient Boosting, CatBoost, XGBoost, and Multilayer Perceptron have all been evaluated based on key metrics such as accuracy, F1 score, recall, and precision.
The highest accuracy is achieved by Gradient Boosting with an impressive 97.4%, closely followed by XGBoost at 96.9%. These models demonstrate strong overall predictive capabilities. However, it's crucial to consider other metrics to comprehensively assess performance. For instance, Naïve Bayes exhibits a significantly lower accuracy of 60.5%, indicating potential limitations in its ability to correctly classify instances. Nevertheless, its precision is remarkably high at 99.5%, suggesting that when it predicts a positive outcome, it is highly likely to be accurate.
The F1 score, which balances precision and recall, is particularly important when dealing with imbalanced datasets. In this regard, Multilayer Perceptron stands out with an F1 score of 98.0%, emphasizing its ability to achieve a harmonious balance between precision and recall. On the other hand, Naïve Bayes has a lower F1 score of 54.0%, reflecting its challenges in handling both false positives and false negatives effectively.

MODELS	ACCURACY	F1_SCORE	RECALL	PRECISION
LOGESTIC REGRESSION	0.934	0.941	0.953	0.930
K–NEAREST NEIGHBOR	0.956	0.961	0.962	0.960
SVM	0.964	0.968	0.980	0.957
NAÏVE BAYES	0.605	0.54	0.294	0.995
DECISION TREE	0.961	0.965	0.964	0.966
RANDOM FOREST	0.967	0.970	0.972	0.966
GRADIENT BOOSTING	0.974	0.977	0.989	0.966
CATBOOST	0.911	0.975	0.982	0.969
XGBOOST	0.969	0.973	0.993	0.984
MULTILAYER PERCEPTRON	0.954	0.980	0.967	0.923

7. CONCLUSION

In conclusion, the evaluation of machine learning models for phishing website detection underscores the importance of considering specific requirements and trade-offs. XGBoost, known for its ensemble learning approach, emerges as a highly effective choice for this task. Its ability to handle imbalanced data, utilize ensemble learning to improve generalization, and employ tree pruning and regularization for preventing overfitting makes it well-suited for detecting phishing websites. XGBoost's widespread use, robust implementations in various languages, and scalability further contribute to its suitability for handling large datasets across diverse websites.

8. REFERENCES

[1] Wenqian Tian; Pei Li; Tao Wei; Zhenkai Liang, 23 August 2022 "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity" 17020 – 17030.

[2] B. Deekshitha, Ch. Aswitha, Ch. Shyam Sundar, A. Kavya Deepthi, , June 2022"URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms "IJRASET, Volume 10.

[3] Ammar Odeh, Abdalraouf Alarbi, Ismail Keshta, 31st August 2022 "efficient prediction of phishing websites using multilayer perceptron (MLP) ", Journal of Theoretical and Applied Information Technology, vol.98.

[4] Luka Jovanovic, Dijana Jovanovic, Milos Antonijevic, 18 April 2023" Improving Phishing Website Detection Using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning", Journal of Web Engineering, Vol. 22 3.

[5] Rishikesh Mahajan, Irffan Siddavatam, October 2018, "Phishing Website Detection using Machine Learning Algorithms ", International Journal of Computer Applications (0975 – 8887) ,Volume 181 – No. 23.