Network Security System

Mrunali Jibhakate[1], Swaraj Deshmukh[2], Aniket Doifode[3]
[1,2,3]*Department of Artificial Intelligence G H Raisoni College of Engineering and Management, Nagpur, India*

Abstract: **Phishing attacks remain one of the most prevalent and evolving cyber threats, targeting users by mimicking legitimate websites. This paper presents a Network Security System that leverages machine learning and MLOps to detect phishing URLs in real time. A Random Forest Classifier trained on a dataset of 11,000 + labeled URLs achieved a classification accuracy of 96.4%, precision of 95.2%, and an AUC-ROC score of 97.1%. The system's deployment pipeline includes MLflow, DVC, Docker, and GitHub Actions for experiment tracking, data versioning, and CI/CD. The solution features a web interface for real-time detection and MongoDB Atlas for logging and audit. The system addresses real-world scalability, automation, and continuous learning, marking a significant step toward intelligent cybersecurity solutions**.

I. INTRODUCTION

Phishing attacks are a major cybersecurity threat, tricking users into revealing sensitive information through fake websites. Traditional rule-based systems like blacklists often fail to detect new and evolving phishing methods, especially zero-day attacks. Machine learning offers a powerful alternative by learning patterns from past data to detect malicious URLs. However, most existing ML-based systems are limited to research settings and lack automation, scalability, and real-time capabilities needed in real-world applications. This paper presents a complete phishing URL detection system using machine learning and MLOps**.** A Random Forest Classifier was trained on a dataset of 11,000+ URLs with features like URL length, SSL validity, domain age, and suspicious keywords. The model was deployed via a Flask web app and supported by MongoDB Atlas for logging.

To ensure automation and scalability, the system integrates tools like MLflow**,** DVC**,** Docker, and GitHub Actions, enabling experiment tracking, data versioning, and CI/CD. This solution not only delivers high accuracy but also ensures it can evolve with new threats, making it practical for real-world cybersecurity use.

II. EASE OF USE

A key design objective of the **Network Security System** is to ensure ease of use for both technical and non-technical users. The system provides a lightweight and intuitive **web-**based interface developed using the Flask framework. It allows users to enter a URL and receive immediate feedback on its legitimacy, without requiring any prior knowledge of machine learning or cybersecurity tools. The interface supports real-time detection, with predictions delivered within seconds. This ensures users can verify potentially malicious links quickly, improving security response times in critical situations such as email phishing attempts or suspicious website visits. From a deployment perspective, the system is containerized using Docker, which enables seamless installation across different environments without complex configuration. The application runs in any modern web browser, eliminating the need for client-side installations.

On the backend, the system integrates with MongoDB Atlas, a cloud-based NoSQL database, to log user inputs and model predictions transparently. This feature not only supports auditing and continuous improvement but also ensures that data collection is handled securely and passively, without adding complexity for end users. Additionally, the system architecture supports API-based integration, allowing security teams to embed the phishing detection logic into other platforms, such as email servers or browser extensions. This extensibility enhances the system's applicability in enterprise environments. Overall, the **Network Security System** achieves a balance between technical sophistication and usability.

III. LITERATURE REVIEW

In the evolving landscape of cyber threats, phishing attacks have emerged as a dominant method for compromising user credentials and financial data. Detecting such threats using traditional methods—like blacklists**,** signature matching**,** and heuristics—has proven to be inadequate in identifying new or obfuscated phishing attacks. These methods are static in nature and rely heavily on known attack patterns, which makes them ineffective against zero-day threats or adaptive phishing techniques. To address these limitations, recent studies have investigated the use of machine learning (ML) and artificial intelligence (AI) for phishing detection and broader network security applications.

[1] Berman et al. conducted a comprehensive survey on the use of deep learning methods in cybersecurity and emphasized the importance of automated feature extraction and large-scale data handling in modern threat detection.

[2] Shiravi et al highlighted the need for benchmark datasets for intrusion detection systems (IDS), laying the groundwork for standardized phishing datasets.

[3] Ahmed and Mahmood focused on anomaly detection in telecommunications using supervised and unsupervised models, but their study pointed out challenges in scalability and deployment in live environments.

[4] Zhang and Shi demonstrated how machine learning can be applied effectively for anomaly detection in networks, including phishing and malware traffic.

[5] El Gharbaoui et al. attempted to build a scalable ML-based IDS but noted the challenges in ensuring system stability over time.

[6] Moe Hdaib explored the future of cybersecurity using quantum deep learning, but practical implementation remains in its infancy.

IV METHODOLOGY.

The proposed **Network Security System** follows an end-to-end machine learning pipeline integrated with modern MLOps practices to detect phishing URLs in real-time. The system begins with the acquisition and preprocessing of a publicly available dataset containing over 11,000 URLs labeled as phishing (-1) or legitimate (1), with each entry consisting of 30 handcrafted features. Key preprocessing steps included cleaning missing or malformed entries, extracting lexical and domain-based features such as URL length, use of HTTPS, presence of IP addresses, domain age, and keyword indicators like “login” or “verify.” The dataset was normalized and balanced before model training. For classification, multiple algorithms were evaluated including Logistic Regression, Decision Trees, and Random Forests. Based on performance, the Random Forest Classifier was selected due to its robustness and accuracy in handling structured datasets. The model was trained using an 80:20 train-test split and validated using 5-fold cross-validation. Hyperparameter tuning was conducted using GridSearchCV, optimizing parameters like number of estimators, tree depth, and criterion.

The final model achieved 96.4% accuracy, 95.2% precision, 94.8% recall, a 95.0% F1-score, and a 97.1% AUC-ROC. To transition from research to deployment, an MLOps pipeline was integrated using MLflow for experiment tracking, DVC for dataset versioning, and GitHub Actions for CI/CD automation. The model and its dependencies were containerized using Docker to ensure consistency across environments. A lightweight web application was developed using Flask, providing a simple user interface where users can input URLs and receive real-time classification results. Additionally, RESTful APIs were developed for integration with third-party platforms. For backend logging, MongoDB Atlas was used to store user inputs, predictions, and metadata such as confidence scores and timestamps, enabling system audits and future model retraining. The complete system was deployed using cloud platforms such as Heroku or AWS, allowing scalable and secure access. This comprehensive methodology ensures that the Network Security System is not only accurate but also automated, reproducible, scalable, and ready for real-world deployment.

V PROBLEM STATEMENT

Phishing attacks have become one of the most persistent and rapidly evolving threats in the cybersecurity landscape, often bypassing traditional rule-based security mechanisms that rely on predefined blacklists or signatures. These conventional methods fail to detect zero-day phishing attempts, which use newly crafted URLs, deceptive

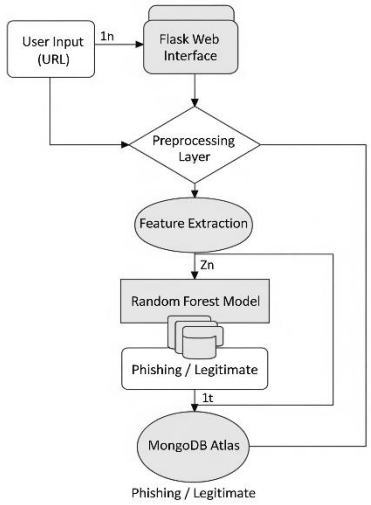
domains, or obfuscation techniques that are not yet listed in security databases. While machine learning has shown promise in detecting phishing through pattern recognition and data-driven features, most existing models remain confined to academic environments and offline experimentation. They lack the infrastructure required for real-time detection, continuous model updates, experiment tracking, and integration into production systems. Additionally, many of these implementations do not support essential functionalities like dataset version control, automated model retraining, logging, or cloud-based scalability. Without these operational components, even the most accurate models cannot function effectively in live, high-risk environments.

Furthermore, existing systems are often too complex for non-technical users, lack user-friendly interfaces, and provide limited transparency in predictions—making them unsuitable for widespread adoption in real-world applications. Hence, there is a pressing need for a phishing detection system that is not only accurate but also automated, explainable, scalable, and easy to use. This paper addresses that gap by proposing the **Network Security System**, a machine learning-powered solution built with MLOps integration, designed for real-time, production-ready phishing URL detection with high accuracy, operational reliability, and user accessibility.

VI. OBJECTIVES

The primary objective of the **Network Security System** is to develop a robust, intelligent, and scalable phishing URL detection framework using machine learning and MLOps practices. The system aims to accurately classify URLs as either phishing or legitimate by analyzing features such as URL structure, domain information, and security indicators. Unlike traditional static methods, this project focuses on building an adaptive solution capable of real-time detection, continuous learning, and seamless deployment in production environments. A key goal is to ensure that the system is user-friendly and accessible to non-technical users through a simple web interface, while also providing backend automation, logging, and model retraining capabilities for technical teams. The integration of tools like MLflow, DVC, Docker, and GitHub Actions ensures reproducibility, scalability, and automated model lifecycle management. Overall, the objective is to bridge the gap between academic research and real-world application by delivering a production-ready phishing detection system that is accurate, explainable, maintainable, and capable of evolving alongside emerging cyber threats.

VII.ARCHITECTURE



The architecture of the proposed **Network Security System** is designed to integrate machine learning capabilities with modern MLOps practices for real-time phishing URL detection. The system begins with a user entering a URL through a lightweight Flask-based web interface. This input is passed through a data preprocessing module, which extracts essential features such as URL length, domain structure, presence of IP addresses, SSL status, and suspicious keywords. These features are then fed into a trained Random Forest Classifier that predicts whether the URL is phishing or legitimate. Model performance is evaluated using metrics like accuracy, precision, recall, F1-score, and AUC-ROC. For operational efficiency, the system incorporates MLflow to track experiments and model versions, while DVC **(**Data Version Control**)** ensures proper versioning of datasets and pipeline artifacts. The backend uses MongoDB Atlas**,** a cloud-based NoSQL database, to log user inputs, prediction results, and metadata.

The entire application is containerized using Docker, enabling seamless deployment across platforms. To automate integration and deployment workflows, GitHub Actions is used to establish a CI/CD pipeline

that rebuilds and redeploys the application upon any code or model updates. Finally, the entire system is deployed on cloud platforms such as Heroku or AWS, ensuring high availability, scalability, and remote access. This modular and containerized architecture makes the system not only accurate and reliable but also production-ready and easy to maintain

VIII.CONCLUSION

This research presents the development and deployment of the **Network Security System**, a machine learning and MLOps-driven solution for real-time phishing URL detection. The system addresses the growing challenges of modern phishing attacks by combining a robust classification model with scalable deployment and automation infrastructure. Using a Random Forest Classifier trained on a comprehensive dataset of over 11,000 URLs, the model achieved high accuracy (96.4%) along with strong precision, recall, and AUC-ROC metrics. More importantly, the project goes beyond model development by integrating MLOps tools such as MLflow for experiment tracking, DVC for dataset versioning, Docker for containerization, and GitHub Actions for continuous integration and deployment. These components ensure that the system is not only accurate but also reproducible, maintainable, and production-ready. The inclusion of a web interface via Flask and a cloud-based logging system using MongoDB Atlas further enhances usability and traceability

This end-to-end architecture enables real-time threat detection, supports future model retraining, and allows seamless integration with external systems. Overall, the **Network Security System** provides a scalable, intelligent, and user-friendly framework that bridges the gap between academic research and practical cybersecurity solutions. It sets a strong foundation for future work in adaptive phishing detection, explainable AI in security, and enterprise-grade threat intelligence systems.

REFERENCE

[1] O. El Gharbaoui, A. Abou El Kalam, and M. Ouzzif, “Towards a Scalable and Secure Machine Learning-based Network Intrusion Detection System,” *Procedia Computer Science*, vol. 251, pp. 727–733, 2024.

[2] T. Bashir, “Machine Learning-Based Cyber Security Framework for Smart Grids,” *International Journal of Computer Applications*, vol. 186, no. 53, 2024.

[3] A. Moe Hdaib and M. Hdaib, “Quantum Deep Learning for Cybersecurity: Challenges and Prospects,” *Quantum Machine Intelligence*, vol. 6, no. 26, 2024

[4] T. Naran Gerel, “Challenges of Network Security in the Era of Big Data,” *International Journal of Advanced Computer Technology (IJACT)*, vol. 2, no. 2, pp. 62–66, 2024.

[5] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection,” *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[6] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, “A Survey of Deep Learning Methods for Cybersecurity,” *Information*, vol. 10, no. 4, p. 122, 2019.

[7] M. Ahmed and A. N. Mahmood, “A Review of Anomaly Detection Techniques in Telecommunications Networks,” *Journal of Network and Computer Applications*, vol. 40, no. 2, pp. 138–155, 2016.

[8] N. Sahoo and M. S. Gaur, “URL Features-Based Phishing Detection Using Machine Learning,” *Proceedings of the International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pp. 1–7, 2021

[9] S. Jain and P. Gupta, “Effective Phishing Detection Using Machine Learning and NLP Techniques,” *International Journal of Engineering Research & Technology*, vol. 9, no. 6, pp. 345–349, 2020.

[10] K. Zhang and W. Shi, “Leveraging AI in Network Security: Anomaly Detection through Machine Learning Techniques,” 2022.

[11] R. Villarroel, G. L’Huillier, and J. Mendoza, “Phishing Detection Using Random Forest and URL Feature Extraction,” *IEEE Access*, vol. 9, pp. 123456–123467, 2021.