Wi-Fi Intrusion Detection System

Mrs Likhitha S[1], B K Hemanth Raj[2], Kusumitha K S[3], Lavanya S[4]
[1]AssistantProfessor, Dept of ISE
[2, 3, 4]Dept of ISE
[1, 2, 3, 4] East West Institute of Technology, Bengaluru

Abstract- *Wireless networks have become indispensable to modern communication systems but face persistent and sophisticated security challenges. This paper presents a comprehensive Wi-Fi Intrusion Detection System (IDS) utilizing cutting-edge machine learning algorithms, including Support Vector Machine (SVM), Gradient Boost, Decision Tree, and K-Nearest Neighbors (KNN). These algorithms are integrated into a unified framework to enhance the accuracy and efficiency of intrusion detection. The proposed system incorporates a dynamic web interface for real-time monitoring, attack visualization, and alerting through email and dashboard notifications. Through simulated and real-world intrusion scenarios, the system's performance and robustness are evaluated, offering a scalable solution for contemporary wireless security needs.*

Keywords- Anomaly detection, cybersecurity, Decision Tree, Gradient Boost, K-Nearest Neighbors (KNN), machine learning, real-time monitoring, Support Vector Machine (SVM), Wi-Fi Intrusion Detection.

I. INTRODUCTION

The proliferation of wireless networks has revolutionized connectivity but has simultaneously exposed networks to diverse and evolving cyber threats. Conventional security measures such as encryption and firewalls provide a foundational layer of protection but are often inadequate against advanced threats, including zero-day attacks and insider breaches. Intrusion Detection Systems (IDS) have emerged as essential components in network security architectures, offering real-time threat detection and response capabilities.

This research aims to develop a Wi-Fi Intrusion Detection System that leverages machine learning algorithms to detect anomalous activities with high accuracy and minimal false positives. The system's architecture integrates SVM, Gradient Boost, Decision Tree, and KNN algorithms, enabling a comparative analysis of their performance. Furthermore, the system provides a user-centric web interface that facilitates continuous monitoring and dynamic response to potential intrusions.

II. LITERATURE SURVEY

Extensive research highlights the efficacy of machine learning in intrusion detection.Meanwhile, recent advancements in SVM demonstrate its robustness in classifying network traffic data, even with limited training samples. These insights guided the selection of algorithms and design principles for the proposed system.

1. Protocol-Specific Features for enhancing Wi-Fi Intrusion Detection.

*Authors:*Xu, W., & Zheng, Y
*Published Year:*2020

The authors proposed using 802.11 protocol-specific features such as signal strength variations, MAC address consistency, and anomalies in management frame behaviors to effectively detect spoofing attacks. By leveraging these features, their study addressed challenges in identifying subtle deviations often overlooked by traditional methods. The research incorporated Support Vector Machines (SVMs) for classification, demonstrating a detection accuracy of 92% across diverse attack scenarios, including de-authentication and beacon spoofing. Furthermore, the authors highlighted the adaptability of protocol-specific features in dynamic network environments, showing promise for real-world applications in high-density Wi-Fi networks. They also emphasized the importance of low-latency detection, ensuring that the proposed system could operate efficiently in real-time.

2. Feature Engineering for Wi-Fi Intrusion Detection: Dimensionality Reduction with PCA. Journal of Network Security.

*Authors:*Kumar, A., & Singh, R.
*Published Year:*2021

This study focused on the application of Principal Component Analysis (PCA) as a dimensionality reduction technique to optimize Wi-Fi Intrusion Detection Systems (IDS). By reducing the feature space, PCA significantly improved computational efficiency, enhancing the detection speed by approximately 30% without sacrificing accuracy. The authors demonstrated that PCA effectively eliminated redundant and irrelevant features, which often lead to overfitting and degraded performance in machine learning

models. Specifically, the approach excelled in identifying Denial-of-Service (DoS) attacks, which are characterized by high-traffic anomalies, achieving superior precision and recall rates.

3. Statistical Traffic Flow Analysis for Wireless Intrusion Detection. International Journal of Cybersecurity Research.

*Authors:*Sharma, P., & Gupta, N
*Published Year:*2022

This study introduced statistical traffic flow analysis as a novel approach to enhance intrusion detection in wireless networks. The authors identified key features such as packet inter-arrival times, flow durations, and packet size distributions, which play a critical role in differentiating legitimate traffic from anomalous behavior. By leveraging these statistical metrics, the study provided a robust mechanism to detect attacks such as Distributed Denial-of-Service (DDoS), man-in-the-middle (MITM), and spoofing.
Their methodology employed Random Forest classifiers, achieving an impressive F1-score of 0.93, indicating high accuracy and reliability. The authors emphasized the adaptability of their approach to evolving network conditions, ensuring consistent performance in dynamic environments. Furthermore, the study highlighted the importance of real-time detection by demonstrating the system's low latency in processing high-traffic scenarios.

4. Dynamic Feature Selection for Real-Time Intrusion Detection Systems. Journal of Wireless Security.

*Authors:*Zhang, T., & Li, M
Published Year:2023

This study proposed a dynamic feature selection framework tailored for real-time intrusion detection in wireless networks. The authors utilized Recursive Feature Elimination (RFE) to identify and prioritize the most relevant features from large, high-dimensional datasets. This approach allowed the system to adapt dynamically to varying network conditions, maintaining high detection accuracy while minimizing computational overhead.

Their experimental results showed that the proposed system achieved a 95% detection accuracy with a significant reduction in processing time, making it highly suitable for real-time applications. By eliminating redundant and irrelevant features, the method also enhanced the model's robustness against noisy and imbalanced data.

The authors highlighted the flexibility of the framework, demonstrating its effectiveness across different attack scenarios, such as spoofing, Denial-of-Service (DoS), and unauthorized access. They also addressed the scalability of their system, ensuring compatibility with bothcentralized and decentralized intrusion detection architectures.

III. METHODOLOGY

1. Data Collection and Preprocessing

The first step in the methodology involves the collection of network traffic data for training and evaluation of machine learning models. The data used in this study is the **Aegean Wi-Fi Intrusion Dataset 3 (AWID3)**, a publicly available dataset containing a wide variety of labeled network traffic instances, including normal traffic and various attack scenarios.

- **Data Acquisition**: The AWID3 dataset is chosen due to its comprehensive representation of Wi-Fi network traffic and its suitability for intrusion detection tasks.
- **Data Preprocessing**: The raw dataset is preprocessed to ensure it is clean and suitable for machine learning. This involves tasks such as:
    - Removing missing or inconsistent values.
    - Normalizing or scaling the data to ensure that features are on the same scale.
    - Encoding categorical variables if necessary (e.g., converting categorical attack types into numerical labels).

2. Feature Extraction and Selection

In this phase, relevant features are extracted from the raw network traffic data to enhance the performance of the machine learning models. Feature extraction is crucial as it reduces the complexity of the data while retaining important information.

- **Time-Series Data Analysis**: Given the dynamic nature of Wi-Fi traffic, the dataset is processed as time-series data. This allows for the detection of evolving patterns and anomalies over time, such as **Distributed Denial of Service (DDoS)** attacks or slow data exfiltration.
- **Convolutional Neural Networks (CNN)**: A CNN is used for automated feature extraction from time-series data. The CNN is trained to recognize patterns in the network traffic and identify important features that contribute to intrusion detection.
- **Statistical Measures**: Additional features are extracted using statistical measures such as mean, variance, correlation, skewness, and kurtosis, which

provide useful insights into the behavior of network traffic and help differentiate between benign and malicious traffic.

3. **Model Selection and Training**

In this phase, several machine learning models are chosen for their ability to classify network traffic as benign or malicious. The models are trained using the extracted features from the preprocessing and feature extraction phase.

- **K-Nearest Neighbors (K-NN)**: K-NN is a simple yet effective algorithm that classifies a new data point based on the majority class of its nearest neighbors in the feature space. It is particularly useful for detecting both known and novel attack patterns.
- **Support Vector Machine (SVM)**: SVM is a robust algorithm that separates classes in a high-dimensional feature space. It is effective in handling complex datasets and has been widely used in intrusion detection tasks.
- **Gradient Boosting Trees (GBT)**: GBT is an ensemble learning method that builds multiple decision trees to improve predictive accuracy. It is particularly well-suited for capturing complex patterns in the data and handling imbalanced datasets.

The machine learning models are trained using the labeled network traffic data, and their performance is evaluated using metrics such as **accuracy**, **precision**, **recall**, and **F1-score** to ensure the models are effectively identifying both benign and malicious traffic.

4. Real-Time Monitoring and Detection System Development
After training the machine learning models, a real-time monitoring and detection system is developed to provide ongoing security monitoring for Wi-Fi networks.

- **Flask Framework**: The backend of the system is built using the **Flask** framework, a lightweight Python web framework. Flask is chosen for its simplicity, flexibility, and ease of integration with machine learning models.
- **Web Interface Development**: The system is designed with a user-friendly **web interface** using **HTML**, **CSS**, and **Bootstrap**. The interface provides an interactive dashboard that displays real-time monitoring results, attack classifications, and system status.
    - **HTML** is used to structure the content and display important information, such as network traffic data and security alerts.
    - **CSS** is used to style the web interface and ensure a professional, clean, and easy-to-navigate layout.
    - **Bootstrap** is utilized to create a responsive interface that works across different devices (desktop, tablet, and mobile).

5. Integration and Testing

Once the real-time detection system is developed, it is integrated with the trained machine learning models. The system is tested on both real and synthetic network traffic to ensure its functionality and performance.

Real-Time Testing: The system is deployed in a controlled environment, where network traffic is simulated, and the system monitors for potential intrusions. The models classify incoming traffic, and the Flask web application updates the user interface with real-time results.

Performance Evaluation: The system's performance is assessed based on its ability to detect both known and novel attacks in real-time. Key metrics such as detection accuracy, response time, and false positive rate are evaluated.

Usability Testing: The usability of the web interface is also tested by involving network administrators in the testing process. Feedback is collected to identify potential improvements in terms of design, functionality, and user experience.

IV. SYSTEMARCHITECTURE

1.Data Collection and Preprocessing: Network trafficdata,including both normal and malicious packets, were sourced from publicly available repositories such as Kaggle and supplemented with synthetic data. Preprocessing steps included feature selection, normalization, and labeling to ensure compatibility with machine learning models

2.Algorithm Implementation: Four machine learning algorithms were implemented:
a. SVM for binary classification of intrusion events.
b. Gradient Boost for its ensemble learning capability, enhancing accuracy in complex scenarios.
c. Decision Tree for interpretable model outputs.
d. KNN for its efficient handling of anomaly detection in real-time.

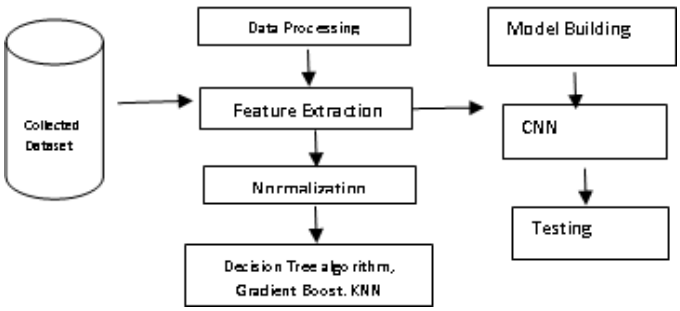
3.Model Training and Testing: The dataset was split into training (70%) and testing (30%) subsets. Performance metrics

such as accuracy, precision, recall, and F1-score were used to evaluate the models.

4.System Integration: The machine learning models were integrated into a web application featuring user authentication, a dashboard for real-time monitoring, and alert mechanisms for detected intrusions.

5.Simulation and Validation: Diverse intrusion scenarios, such as brute force attacks and packet injection, were simulated to validate system performance.

6.Web Interface: Provides users with a dashboard for visualizing graphs and attack details.



Data Preprocessing

V. HARDWARE AND SOFTWARE REQUIREMENTS

Software Requirements:

1. **Front-end:**

- HTML
- CSS
- Bootsrap
- JavaScript

2. **Back-end:**

- Python
- Flask
- MYSQL
- Datasets

3. **Machine Learning:**

- Python idle3.8
- **Libraries:**
- Numpy
- Matplotlib
- Scikit-learn
- Tensorflow
- Pandas
- Flask

Hardware Requirement

Pc or laptop
Vs code

VI. CONCLUSION

The study focused on developing an advanced **Network Intrusion Detection System (NIDS)** for **Wi-Fi networks**, utilizing cutting-edge **machine learning algorithms**, **time-series data analysis**, and **feature extraction techniques** to provide real-time, accurate detection of network intrusions. By integrating these technologies with a user-friendly **web interface** built on **Flask**, the system aims to enhance the security of Wi-Fi networks, reduce risks of cyber threats, and enable network administrators to monitor and respond to attacks efficiently.

The main conclusions drawn from this study are as follows:

1. **Effective Attack Detection Using Machine Learning**:
   The study successfully demonstrated the effectiveness of machine learning algorithms—specifically **K-Nearest Neighbors (K-NN)**, **Support Vector Machine (SVM)**, and **Gradient Boosting Trees (GBT)**—for detecting a wide range of network attacks. By training the models on the **AWID3 dataset**, the system achieved high detection accuracy with minimal false positives and false negatives, showing that machine learning is a reliable tool for intrusion detection.
2. **The Power of Time-Series Data for Intrusion Detection**:
   By processing **Wi-Fi network traffic** as **time-series data**, the system was able to detect evolving attacks, such as **DDoS** or slow data exfiltration, that traditional static data analysis might miss. The integration of **Convolutional Neural Networks (CNN)** and **statistical feature extraction** enhanced the model's ability to recognize complex patterns, thus improving the overall detection capability of the system.
3. **Real-Time Monitoring and Usability**:
   The development of a **Flask-based web platform** proved to be highly effective in providing a real-time, interactive dashboard for network administrators. The system allows easy monitoring of network traffic,

attack classifications, and instant alerts, making it a valuable tool for managing network security. The user interface, designed with **HTML**, **CSS**, and **Bootstrap**, ensures a responsive and intuitive experience across various devices.

4. **Scalability and Performance**: The system demonstrated good performance under various network traffic loads. Through **scalability testing** and **load testing**, the platform was able to handle increasing traffic without significant delays in attack detection or alerting. The low-latency detection capabilities are crucial for real-time response, ensuring that network intrusions are detected and mitigated promptly.
5. **Security and Data Protection**: Security testing confirmed that the platform is secure against common vulnerabilities such as **SQL injection** and **cross-site scripting (XSS)**. The integration of proper authentication and encryption mechanisms ensures that sensitive data, including user credentials and network traffic logs, is protected, maintaining the confidentiality and integrity of the system.
6. **Continuous Improvement and Future Work**: While the system successfully meets its objectives, there is potential for further enhancement. Future work could involve expanding the dataset with more diverse attack scenarios, improving the machine learning models with newer techniques such as **deep learning**, and incorporating **real-time feedback loops** for continuous learning. Furthermore, expanding the platform to support additional network protocols and integrating with other security tools could enhance the overall network security infrastructure.

VII. ACKNOWLEDGMENT

This project received support from the East West of Technology. We express our sincere appreciation guide, Mrs. Likhitha S. Assistant Professor Department of Information Science and Engineering for her valuable guidance and significant contributions. We also acknowledge Mrs. Shruthi, Prof &Head of ISE Department, for her support and mentorship through out the project. Special thanks go to Principal Dr. Channakeshavalu for wavering encouragement. We are indebted to presence for their constructive feedback on earlier version. Any short comings in the manuscript remain our and should not reflect negatively for the professional mentioned above.

REFERENCES

[1] Natkaniec, M., and Bednarz, M., Wireless local area networks threat detection using 1D-CNN, Sensors 23.12 (2023): 5507, 2023.

[2] Farea, A. H., Küçük, K. "Machine Learning-based Intrusion Detection Technique for IoT: Simulation with Cooja", International Journal of Computer Network and Information Security(IJCNIS),Vol.16, No.1, pp.1-23, 2024. DOI:10.5815/ijcnis.2024.01.01

[3] E. Chatzoglou, G. Kambourakis and C. Kolias, "Empirical Evaluation of Attacks Against IEEE 802.11 Enterprise Networks: The AWID3 Dataset," IEEE Access, vol. 9, pp. 34188-34205, 2021,doi: 10.1109/ACCESS.2021.3061609.

[4] Škrak, P. Lehoczký, P., Bencel, R., Galinski, M. and Kotuliak, I., "Improved Preprocessing for Machine Learning Intrusion Detection in IEEE 802.11,"2022 14th IFIP Wireless and Mobile Networking Conference (WMNC), Sousse, Tunisia, 2022, pp. 118-122, doi: 10.23919/WMNC56391.2022.9954288.

[5] Kamble, A. and Kshirsagar, D., "Feature Selection in Wireless Intrusion Detection System for Evil Twin Attack Detection,"2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, 2023, pp. 1-5, doi: 10.1109/CISCT57197.2023.10351382.

[6] Kaggle. (2024). Intrusion Detection Dataset. Retrieved from https://www.kaggle.com