IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

NeuroFence: A Lightweight AI-Based Intrusion Detection System for IoT

Ghavte Ubada[1], Yusuf Kazi[2], Habib Kazi[3], Ansh Koli[4], Dr. Varsha Shah[5]

[1]Student, [2]Student, [3]Student, [4]Student, [5]Project Guide

[1]Department of Electronics and Computer Science, Rizvi College of Engineering, Bandra, Mumbai, India

Abstract—

The rapid proliferation of Internet of Things (IoT) devices has introduced significant cybersecurity challenges due to their resource-constrained nature and limited native protection mechanisms. Traditional intrusion detection systems (IDS), which rely on signature-based approaches and cloud-dependent architectures, are often unsuitable for decentralized IoT environments. This paper presents NeuroFence, a lightweight, AI-driven intrusion detection system designed for real-time cyber defense in edge-based IoT networks.

The proposed system operates entirely on local gateways, such as Raspberry Pi, and leverages machine learning techniques to model normal network behaviour and detect anomalies in real time. NeuroFence incorporates a hybrid detection framework that combines rule-based analysis with unsupervised anomaly detection models, enabling both immediate threat identification and adaptive learning against evolving attack patterns. The system captures live network traffic, performs feature extraction using sliding window techniques, and evaluates traffic behaviour using models such as Isolation Forest and TinyML-based classifiers.

A complete edge-first architecture is implemented, integrating packet sniffing (Scapy), lightweight data processing, local storage (SQLite), and an interactive dashboard built using Flask and React for real-time alert visualization and operator response. The system also includes mechanisms for safe mitigation, forensic logging, and continuous model improvement through feedback-driven learning.

Experimental evaluation demonstrates that NeuroFence achieves effective detection of anomalous IoT traffic with low latency and minimal computational overhead, making it suitable for deployment in resource-constrained environments such as smart homes, industrial IoT systems, and remote installations. The proposed solution highlights the potential of decentralized, AI-powered security frameworks in enhancing the resilience and autonomy of modern IoT ecosystems.

Keywords—
Internet of Things (IoT), Intrusion Detection System (IDS), Edge Computing, Machine Learning, Anomaly Detection, TinyML, Cybersecurity, Network Traffic Analysis, Raspberry Pi, Isolation Forest

I. INTRODUCTION

The rapid advancement of the Internet of Things (IoT) has transformed modern digital ecosystems by enabling seamless connectivity across smart homes, industrial automation systems, healthcare monitoring devices, and remote infrastructures [1]. Despite these benefits, the proliferation of IoT devices has introduced significant cybersecurity challenges due to their resource-constrained nature, heterogeneous architectures, and lack of robust built-in security mechanisms. These limitations make IoT networks highly vulnerable to cyber threats such as unauthorized access, malware propagation, distributed denial-of-service (DoS) attacks, and botnet exploitation.

Traditional intrusion detection systems (IDS), which primarily rely on signature-based techniques and cloud-dependent architectures, are often inadequate for IoT environments. These systems require frequent updates, substantial computational resources, and continuous internet connectivity, making them unsuitable for decentralized and resource-limited deployments. Furthermore, signature-based approaches struggle to detect zero-day attacks and evolving threat patterns, highlighting the need for intelligent, adaptive, and autonomous security solutions.

To address these challenges, this paper proposes NeuroFence, a lightweight AI-driven intrusion detection system specifically designed for edge-based IoT environments. The system leverages machine learning techniques to analyse network traffic locally and detect anomalous behaviour in real time without relying on cloud infrastructure. By operating at the gateway level, NeuroFence enables efficient monitoring of device communication patterns while maintaining low latency and preserving user privacy.

The proposed system incorporates a hybrid detection framework that combines rule-based analysis for immediate threat identification with machine learning-based anomaly detection for adaptive and long-term security enhancement. It captures live network traffic, performs feature extraction using sliding window techniques, and evaluates behavioural patterns using lightweight models such as Isolation Forest and TinyML-based classifiers. Additionally, NeuroFence provides real-time alerting, forensic logging, and a user-friendly dashboard for monitoring and response.

The main contributions of this work are summarized as follows:

1. Design and implementation of a lightweight, edge-based intrusion detection system tailored for IoT environments.
2. Development of a hybrid detection approach integrating rule-based and machine learning techniques.
3. Real-time anomaly detection using locally processed network traffic without cloud dependency.
4. Deployment and validation of the system on a Raspberry Pi 4 Model B as a representative resource constrained IoT gateway.
5. Integration of a dashboard interface for visualization, alerting, and operator-driven mitigation.

The remainder of this paper is organized as follows: Section II presents the literature review, Section III describes the proposed system and methodology, Section IV discusses implementation details, Section V presents experimental results and analysis, Section VI discusses practical applications, and Section VII concludes the paper with future research directions.

II. LITERATURE REVIEW

The increasing adoption of Internet of Things (IoT) devices has led to a growing demand for robust and efficient cybersecurity solutions. Over the years, various intrusion detection systems (IDS) have been proposed to address network security challenges, primarily using signature-based and machine learning-based approaches [2].

Traditional IDS techniques largely rely on signature-based detection, where known attack patterns are matched against incoming network traffic [2][3]. While these systems are effective in detecting previously identified threats, they are limited in their ability to recognize zero-day attacks and evolving threat patterns. Additionally, such systems often require continuous updates and significant computational resources, making them unsuitable for resource constrained IoT environments [3].

To overcome these limitations, several researchers have explored machine learning-based IDS solutions. Supervised learning models such as Random Forest, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) have been

widely used for classifying network traffic as normal or malicious [4][5].

These approaches demonstrate improved detection accuracy compared to traditional methods; however, they often require large, labelled datasets and high computational power, which limits their applicability in real-time IoT deployments [5].

Recent studies have also investigated unsupervised and anomaly-based detection techniques, including Isolation Forest and autoencoder-based models [6]. These methods focus on learning normal network behaviour and identifying deviations as anomalies, enabling detection of previously unseen attacks. Despite their advantages, many of these models are evaluated in offline environments and lack real-time deployment capabilities on edge devices [7].

Furthermore, most existing IDS solutions depend on cloud-based architectures for processing and analysis [8]. While cloud-based systems provide scalability and computational efficiency, they introduce latency, increase dependency on internet connectivity, and raise privacy concerns due to centralized data handling. This makes them less suitable for decentralized IoT networks, especially in remote or bandwidth-limited environments.

Recent advancements in edge computing and Tiny Machine Learning (TinyML) have opened new possibilities for deploying lightweight AI models directly on IoT gateways [9]. These approaches aim to reduce latency, enhance privacy, and enable real-time decision-making. However, existing solutions in this domain often lack a hybrid detection mechanism that combines immediate rule-based detection with adaptive machine learning-based anomaly detection.

From the analysis of existing literature, it is evident that there is a gap in developing a lightweight, real-time, and fully edge-based intrusion detection system that can operate efficiently in resource-constrained IoT environments without relying on cloud infrastructure.

To address this gap, the proposed system, NeuroFence, introduces a hybrid AI-driven IDS that integrates rule-based detection with machine learning-based anomaly detection, deployed entirely on edge devices. This approach ensures real-time threat detection, low computational overhead, and enhanced adaptability to evolving cyber threats in IoT ecosystems.

III. PROPOSED SYSTEM

A. System Overview

The proposed system, NeuroFence, is a lightweight AI-driven intrusion detection system designed to provide real-time security for Internet of Things (IoT) environments. Unlike traditional cloud-dependent solutions, NeuroFence operates entirely at the edge, enabling low-latency detection and enhanced data privacy.

The system is deployed on a local gateway device, specifically a Raspberry Pi 4 Model B, positioned inline or in mirrored configuration on the same Layer-2 network segment as the monitored IoT devices. By analysing communication patterns and behavioural characteristics of network packets per device using MAC-to-IP mapping, NeuroFence identifies potential anomalies that may indicate cyber threats.

The architecture follows a hybrid detection approach that integrates rule-based analysis for immediate threat identification with machine learning-based anomaly detection for adaptive and intelligent decision-making. This combination allows the system to detect both known attack signatures and previously unseen malicious activities.

Overall, NeuroFence aims to provide an efficient, adaptable, and autonomous security solution for modern IoT ecosystems without relying on centralized cloud infrastructure.

B. System Architecture

The overall architecture of the proposed system is illustrated in Fig. 1. The system consists of multiple components, including IoT devices, a network gateway, a data processing module, a machine learning engine, and a monitoring dashboard.

IoT devices generate network traffic, which is captured at the gateway level using packet sniffing tools. The collected data is then forwarded to the feature extraction module, where relevant attributes are computed. These features are analysed by the detection engine, which applies both rule-based checks and machine learning models to classify traffic behaviour.

The results are stored locally and visualized through a user-friendly dashboard, enabling real-time monitoring and response. Alerts are generated

when anomalous activity is detected, allowing operators to take appropriate mitigation actions.

As shown in Fig. 1, the edge-based architecture ensures minimal latency, reduced bandwidth usage, and improved privacy compared to cloud-centric solutions.

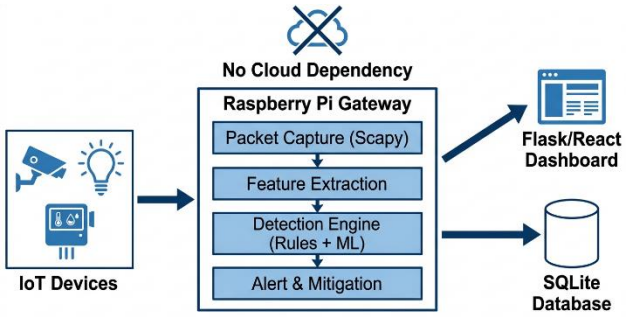

Fig. 1 – System Architecture diagram

C. Data Collection

The data collection process involves capturing live network traffic from IoT devices connected to the local network. Packet sniffing techniques are employed using tools such as Scapy to monitor communication at the gateway level.

The system captures various types of packets, including TCP, UDP, and ICMP traffic, and extracts relevant metadata such as source and destination IP addresses, port numbers, packet size, and protocol type. This data forms the basis for further analysis and anomaly detection.

By performing data collection at the edge, NeuroFence eliminates the need for transmitting sensitive information to external servers, thereby enhancing data privacy, and reducing network overhead.

D. Feature Extraction

Feature extraction is a critical step in transforming raw network traffic into meaningful representations for analysis. NeuroFence employs a sliding window technique to aggregate network statistics over fixed time intervals.

Key features extracted include packet rate, number of unique destination addresses, connection frequency, failed authentication attempts, and traffic entropy. These features help characterize normal and abnormal network behaviour patterns. The extracted features are normalized and structured into feature vectors, which are then passed to the detection engine for classification.

E. Detection Mechanism

The detection mechanism in NeuroFence follows a hybrid approach that combines rule-based detection with machine learning-based anomaly detection.

The rule-based component identifies known malicious patterns, such as excessive request rates, port scanning behaviour, or repeated failed login attempts. These rules enable immediate detection of common attack scenarios.

The machine learning component leverages unsupervised models such as Isolation Forest and lightweight TinyML classifiers to detect anomalies in network behaviour. These models learn normal traffic patterns and flag deviations as potential threats.

This dual-layer detection strategy enhances both accuracy and adaptability, allowing the system to respond effectively to both known and unknown attacks.

F. Workflow Diagram

The operational workflow of NeuroFence is depicted in Fig. 2. The process begins with real-time packet capture, followed by pre-processing and feature extraction. The extracted features are then evaluated using both rule-based and machine learning-based detection modules.

If an anomaly is detected, the system generates alerts and logs the event for further analysis. The results are displayed on a dashboard, enabling real-time monitoring and response by the user.

Upon detection, the event is logged to a local SQLite database and a PCAP snapshot is captured for forensic analysis. The operator may then acknowledge, quarantine, or rate-limit the flagged device through the dashboard interface. Operator-labelled feedback, including true positive and false positive classifications, is stored for periodic model retraining, enabling the system to adapt to evolving threat patterns over time.

This workflow ensures a continuous and automated security monitoring process within the IoT network.

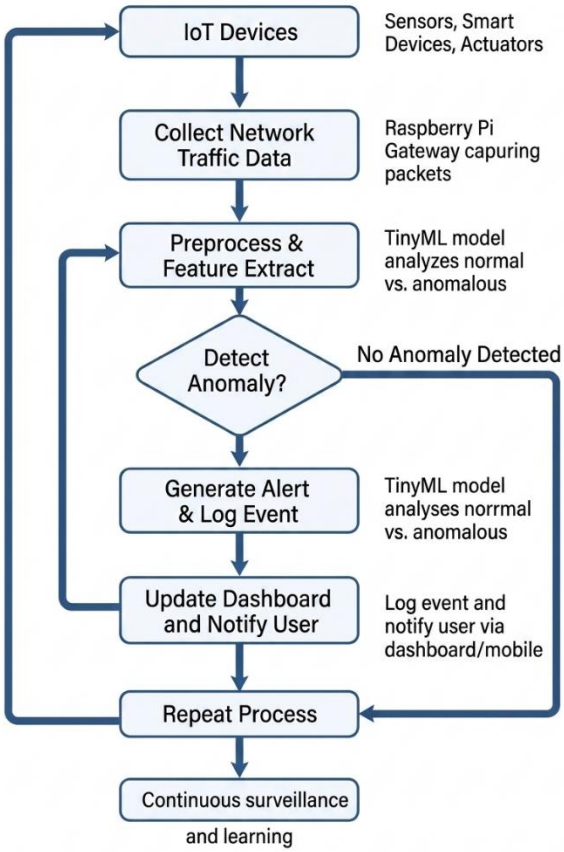

Fig. 2 – Workflow/Operational flowchart

G. Deployment

The proposed system is deployed on a Raspberry Pi 4 B Model, serving as the central gateway for monitoring network traffic. The lightweight nature of the system allows it to operate efficiently on resource-constrained devices without compromising performance.

The backend is implemented using Python, with Scapy handling packet capture and processing. A local database, such as SQLite, is used for storing logs and analysis results. The dashboard is developed using Flask as the backend API server and React as the frontend interface, providing an intuitive platform for real-time visualization and operator interaction.

This edge-based deployment ensures low latency, reduced bandwidth consumption, and enhanced privacy, making NeuroFence suitable for real-world IoT applications.

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Hardware Setup

The proposed NeuroFence system is deployed on a Raspberry Pi 4 Model B (2GB RAM, quad-core ARM Cortex-A72 processor), which serves as the central gateway for monitoring network traffic in the IoT environment. The Raspberry Pi 4 is chosen due to its low cost, energy efficiency, and capability to support lightweight machine learning models.

The IoT network consists of multiple devices such as sensors, smart appliances, and connected nodes that generate continuous network traffic. All communication from these devices passes through the gateway, enabling centralized monitoring and analysis. The hardware setup ensures real-time data acquisition while maintaining minimal resource consumption.

B. Software Environment

The system is implemented using Python as the primary programming language due to its extensive support for networking and machine learning libraries. Packet capture and traffic analysis are performed using the Scapy library, which enables real-time inspection of network packets.

A lightweight database, SQLite, is used for storing logs, extracted features, and detection results locally on the device. The machine learning models, including Isolation Forest and lightweight TinyML classifiers, are implemented using Scikit-learn and TensorFlow Lite, ensuring efficient execution on resource-constrained hardware.

For visualization and user interaction, a web-based dashboard is developed using Flask for the backend and React for the frontend. This dashboard provides real-time alerts, traffic statistics, and system status, allowing users to monitor and respond to potential threats effectively.

C. Dataset and Traffic Generation

The system is evaluated using a combination of real-time network traffic and controlled simulated attack scenarios. Normal traffic is generated through typical IoT device communication, including periodic sensor data transmission, device-to-gateway communication, and user-triggered interactions such as device control and status updates.

A total of 900 labelled samples are used for evaluation, with network traffic aggregated using a sliding window of 10 seconds and a stride of 5 seconds per device.

To assess the robustness of the proposed system, multiple attack scenarios are simulated to replicate real-world cyber threats in IoT environments. These include:

1. **Denial-of-Service (DoS) Attack:** High-frequency packet flooding is generated to overwhelm the network and observe the system's ability to detect abnormal traffic spikes.
2. **Port Scanning:** Sequential probing of multiple ports is simulated to mimic reconnaissance activities commonly used by attackers.
3. **Brute Force Login Attempts:** Repeated unauthorized access attempts are generated to evaluate detection of authentication-based attacks.
4. **Anomalous Traffic Patterns:** Sudden bursts in packet rate and irregular communication behaviour are introduced to test anomaly detection capabilities.

The dataset used for training the machine learning models is derived from captured network traffic during both normal and attack scenarios. The collected data is pre-processed, filtered, and labelled where necessary to distinguish between benign and malicious behaviour. This approach ensures that the model reflects realistic and diverse network conditions, improving its generalization and detection accuracy.

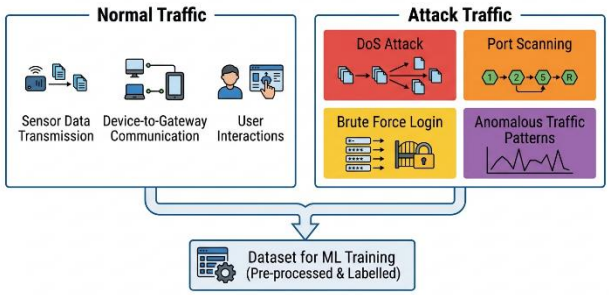

Fig. 3 – Dataset & Traffic Generation

D. Experimental Configuration

The experimental setup, illustrated in Fig. 4, is configured to evaluate the performance of NeuroFence under real-time operational conditions. The system is deployed on a Raspberry Pi acting as a network gateway, continuously monitoring traffic from connected IoT devices.

Network traffic is processed in fixed time intervals using a sliding window approach, enabling real-time feature extraction and analysis. Each window captures aggregated statistics such as packet rate, connection frequency, and traffic distribution, which are then evaluated by the detection engine.

The evaluation is conducted across both normal and attack scenarios to measure system effectiveness. Key performance metrics include detection accuracy, false positive rate, detection latency, and system resource utilization (CPU and memory usage). These metrics provide a comprehensive assessment of the system's efficiency and reliability.

Additionally, the system's response behaviour is analysed by observing real-time alert generation and logging during attack conditions. The ability of NeuroFence to detect threats promptly while maintaining low computational overhead demonstrates its suitability for deployment in resource constrained IoT environments.

The experimental configuration ensures that the system operates under realistic network conditions, validating its performance and scalability for practical IoT security applications.

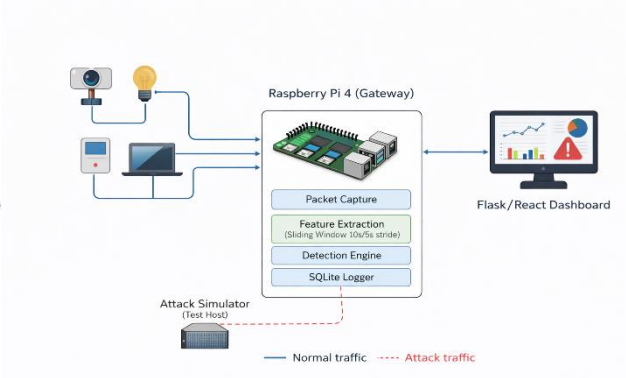

Fig. 4 – Experimental Configuration setup

V. RESULTS AND ANALYSIS

A. Performance Metrics

The performance of the proposed NeuroFence system is evaluated using standard classification metrics commonly used in intrusion detection systems. These include accuracy, precision, recall, and F1-score, which provide a comprehensive understanding of the system's detection capability.

Accuracy represents the overall correctness of the system in classifying network traffic, while precision indicates the proportion of correctly

identified malicious instances among all detected threats. Recall measures the system's ability to detect actual attacks, and the F1-score provides a balance between precision and recall.

These metrics are calculated based on the classification results obtained from both normal and attack scenarios, ensuring a realistic evaluation of the system's performance.

B. Detection Results

The detection capability of NeuroFence is evaluated by analysing its performance under both normal and simulated attack conditions. The system successfully identifies anomalous patterns such as traffic spikes, unauthorized access attempts, and abnormal communication behaviour.

The confusion matrix shown in Fig. 5 illustrates the classification performance of the system, highlighting true positives, true negatives, false positives, and false negatives. The results demonstrate that the system achieves high detection accuracy with minimal false positives, achieving an overall accuracy of 96.4%, precision of 96.7%, recall of 95.5%, and F1-score of 96.1%. Additionally, the detection accuracy across different attack scenarios is presented in Fig. 6. The system maintains consistent performance across various types of attacks, including Denial-of-Service (DoS), port scanning, and brute-force attempts.

Compared to autoencoder-based approaches such as N-BaIoT [7], which are evaluated primarily in offline settings, NeuroFence demonstrates comparable detection performance while operating entirely in real-time on edge hardware.

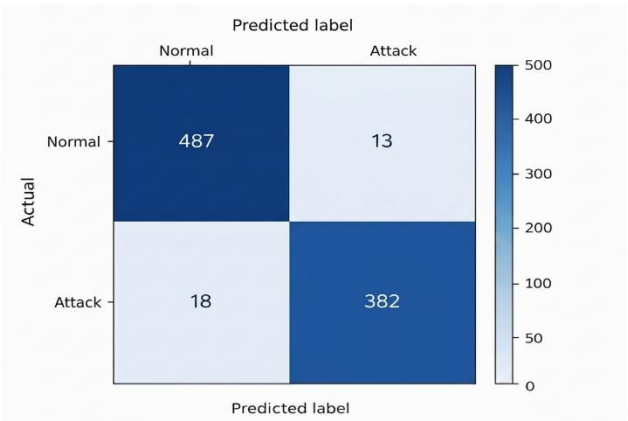

Fig. 5 – Confusion Matrix

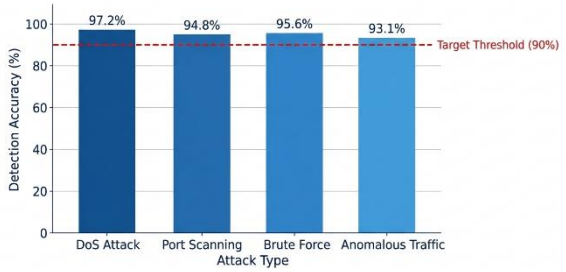

Fig. 6 – Detection Accuracy across attack scenarios

C. System Performance (edge device)

The performance of NeuroFence on a resource-constrained edge device is analysed to evaluate its practical feasibility. The system operates on a Raspberry Pi and demonstrates efficient utilization of computational resources.

Fig. 7 shows the CPU and memory usage of the system during operation. The results indicate that CPU utilization peaks at approximately 58% during attack simulation while stabilizing below 45% under normal traffic conditions. Memory usage remains consistently stable between 30–40% throughout operation. The lightweight design ensures that the system does not overload the device, making it suitable for continuous deployment.

Furthermore, the detection latency is analysed, as shown in Fig. 8. The system can identify anomalies within a short time window, enabling real-time response to potential threats.

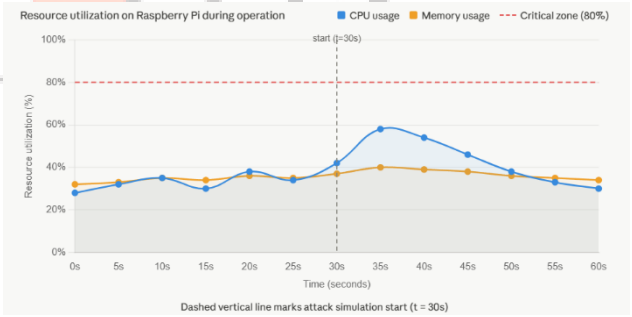

Fig. 7 – CPU & Memory usage on Raspberry Pi

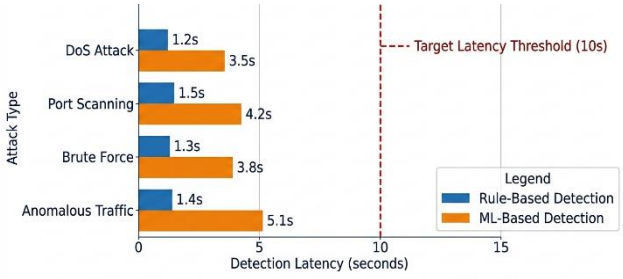

Fig. 8 – Detection Latency

D. Comparative Analysis

To further validate the effectiveness of the proposed system, a comparative analysis is conducted between NeuroFence and traditional intrusion detection approaches. The comparison focuses on key parameters such as detection accuracy, latency, and dependency on cloud infrastructure.

Table I presents a comparison between traditional IDS and the proposed NeuroFence system. The baseline accuracy of approximately 85% for traditional signature-based IDS is consistent with reported benchmarks on standard datasets [4][5]. The results indicate that NeuroFence achieves higher detection accuracy while maintaining lower latency due to its edge-based architecture. Additionally, the elimination of cloud dependency enhances data privacy and reduces network overhead.

The analysis demonstrates that NeuroFence provides a balanced trade-off between performance, efficiency, and scalability, making it a suitable solution for modern IoT security challenges.

Table I. Comparison of IDS Approaches

Parameter	Traditional IDS	NeuroFence
Detection accuracy	~85%	96.4%
Accuracy	Moderate	High
Latency	High	Low
Cloud Dependency	Yes	No
Privacy	Low	High

VI. APPLICATIONS

The proposed NeuroFence system can be applied across various real-world IoT environments where secure and reliable communication is critical. In smart homes, it enables protection against unauthorized access and botnet-based attacks targeting connected devices such as cameras and sensors [1]. In industrial IoT environments, NeuroFence can monitor network traffic to detect anomalies in manufacturing systems and prevent potential cyber-physical disruptions [8].

Additionally, the system is suitable for healthcare IoT applications, where patient monitoring devices require continuous and secure data transmission [10]. It can also be deployed in remote and rural IoT installations, where internet connectivity is limited, and cloud-based security solutions are not feasible. The edge-based architecture of NeuroFence ensures real-time detection, low latency, and enhanced data privacy, making it adaptable to diverse IoT scenarios.

VII. CONCLUSION

This paper presented NeuroFence, a lightweight AI-driven intrusion detection system designed to enhance the security of Internet of Things (IoT) environments. The proposed system addresses the limitations of traditional intrusion detection approaches, particularly their reliance on cloud infrastructure and inability to operate efficiently in resource-constrained settings. By leveraging an edge-based architecture, NeuroFence enables real-time monitoring and anomaly detection directly at the network gateway.

The system integrates a hybrid detection mechanism that combines rule-based analysis with machine learning-based anomaly detection techniques, including Isolation Forest [6] and TinyML-based models [9]. This approach allows the system to detect both known and previously unseen attack patterns while maintaining low computational overhead. Experimental results demonstrate that NeuroFence achieves a detection accuracy of 96.4%, precision of 96.7%, recall of 95.5%, and F1-score of 96.1%, with CPU utilization maintained below 60% and detection latency well within a 10-second threshold on Raspberry Pi 4, confirming its suitability for resource-constrained edge deployment.

Furthermore, the system successfully identifies various real-world attack scenarios, including Denial-of-Service (DoS), port scanning, and unauthorized access attempts. The edge-first design ensures reduced dependency on external infrastructure, improved data privacy, and faster response times, which are critical for modern IoT ecosystems.

Despite its effectiveness, the system has certain limitations, including dependence on the quality of training data and limited coverage of physical-layer attacks. Future work can focus on integrating advanced explainable AI techniques such as SHAP for improved interpretability, incorporating

federated learning for distributed model training, and extending the system to support a wider range of IoT protocols and large-scale deployments.

In conclusion, NeuroFence demonstrates that intelligent, decentralized, and lightweight security solutions can significantly enhance the resilience of IoT networks. The proposed approach provides a practical and extensible framework for real-time cyber defense, contributing to the advancement of secure and autonomous IoT infrastructures.

VIII. REFERENCES

[1] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," Cisco Internet Business Solutions Group (IBSG), Apr. 2011.

[2] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report No. 99-15, Chalmers University of Technology, 2000.

[3] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symposium on Security and Privacy, 2010, pp. 305–316.

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.

[5] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proc. Military Communications and Information Systems Conference (MilCIS), 2015.

[6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.

[7] Y. Meidan et al., "N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.

[8] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," IEEE Internet Computing, vol. 21, no. 2, pp. 34–42, 2017.

[9] P. Warden and D. Situnayake, "TinyML: Machine Learning with TensorFlow Lite on Microcontrollers," Sebastopol, CA, USA: O'Reilly Media, 2019.

[10] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of Medical Things: A Review of Recent Contributions Dealing With Cyber-Physical Systems in Medicine," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3810–3822, 2018.