A Real-Time IoT Vulnerability Detection Framework with Hybrid Discovery and CVE Correlation

Shubham Minhass
Amity Institute of Information Technology, Amity University Noida, India
shubhamminhass@gmail.com

Ritu Chauhan
Artificial Intelligence and IoT Lab, Centre for Computational Biology and Bioinformatics, Amity University, Noida, UP, India
rituchauha@gmail.com (corresponding author)

Harleen Kaur
Department of Computer Science and Engineering, Jamia Hamdard, Delhi, India
harleen@jamiahamdard.ac.in

Received: 21 November 2025 | Revised: 11 December 2025 | Accepted: 21 January 2026

 DOI: https://doi.org/10.48084/etasr.16423

ABSTRACT

The rapid growth of the Internet of Things (IoT) has expanded connectivity across smart homes, companies, and industries. However, these networks have become increasingly vulnerable to cyber threats. Due to the variety of protocols used, proprietary firmware, and unpredictable device behavior, vulnerability scanners such as Nmap, Nessus, and OpenVAS are often less effective at detecting IoT-specific vulnerabilities. To address these issues, this paper proposes a Real-Time IoT Vulnerability Scanner Framework that connects to the Common Vulnerabilities and Exposures (CVE) database to standardize threat correlation, perform hybrid vulnerability verification, and automatically identify connected devices. Once devices across various protocols are discovered through a hybrid approach that combines active probing and passive packet sniffing, the framework matches firmware and service banners with known vulnerabilities in the National Vulnerability Database (NVD) using a fuzzy CVE matching algorithm. An interactive dashboard displays vulnerability timelines, Common Vulnerability Scoring System (CVSS)-based severity ratings, and current device inventories. The system was tested and demonstrated to outperform existing tools in an experimental setup comprising more than 600 IoT and non-IoT devices. It achieved a precision of 0.94, a recall of 0.91, an F1-score of 0.92, and covered approximately 88% of vulnerabilities, with an average scan time of 1.4 s per device. These results demonstrate that the system offers high accuracy, low latency, and scalability for real-time IoT vulnerability monitoring.

Keywords-IoT security; CVE integration; real-time monitoring; dashboard analytics

I. INTRODUCTION

The Internet of Things (IoT) is a transformative technology driving a rapid digital revolution in the 21st century. To enhance communication and automation across sectors like healthcare, smart homes, manufacturing, and transportation, it connects billions of disparate devices, including sensors, cameras, smart appliances, industrial controls, and wearable tech. However, this rapid expansion raises serious cybersecurity concerns. IoT devices are highly vulnerable because they often run on minimal hardware, use proprietary communication stacks, and are rarely updated with new software. Attackers exploit these vulnerabilities to gain unauthorized access, conduct surveillance, or create large botnets for Distributed Denial-of-Service (DDoS) attacks [1]. The need for specialized real-time security solutions is urgent, as major incidents have shown that IoT devices can be weaponized, compromising millions of endpoints globally.

Enterprise IT environments typically rely on traditional network scanners like Nmap, Nessus, and OpenVAS to detect open ports, services, and vulnerabilities. However, these scanners lack support for the dynamic and complex nature of IoT ecosystems and are mainly designed for conventional computing environments. IoT device vendors develop devices using both standard and non-standard protocols, including

Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Zigbee, Bluetooth Low Energy (BLE), and Long-Range Wide Area Network (LoRaWAN) [2]. Due to their limited protocol recognition, standard scanners often misidentify or fail to detect IoT devices altogether. Additionally, the batch-based scanning mode that these tools rely on is ineffective in dynamic IoT networks where devices are frequently added or removed. Another major limitation is that existing tools cannot connect to international vulnerability databases like the National Vulnerability Database (NVD) and Common Vulnerabilities and Exposures (CVE). Many scanners do not automatically link identified devices or their versions to corresponding CVE entries, despite these databases providing standardized vulnerability intelligence.

As a result, well-known weaknesses often go unnoticed in real-world environments. For example, many IP cameras and routers sold to the public continue to run outdated firmware or contain known hardcoded passwords that have not been patched, which are listed in CVE databases. This highlights the need for an automated, intelligent scanning system that bridges the gap between vulnerability data and real-time network evaluation. To address these concerns, this paper proposes a Real-Time IoT Vulnerability Scanner Framework explicitly designed to target IoT and hybrid networks. The framework automatically identifies connected devices, analyzes their network behavior, and compares the findings with CVE data. Its real-time dashboard, displaying device inventories, vulnerability details, and risk ratings, enables administrators to monitor and mitigate threats as they occur. Network disturbance is minimized through a hybrid discovery methodology that combines active probing with passive sniffing. To handle differences in naming schemes across vendors, a fuzzy CVE correlation algorithm is employed to address practical challenges that cybersecurity experts face in detecting and managing IoT vulnerabilities in a dynamic environment [3].

Traditional scanners lack proper vulnerability presentation, automation, and coverage. Network managers often struggle to identify firmware flaws, track system security, or distinguish between IoT devices and regular network traffic over time. By introducing interactive visualization, intelligent vulnerability mapping, and automated discovery within a single system, the proposed architecture addresses these challenges. The platform allows administrators to make evidence-based security decisions and conduct proactive threat response by integrating with CVE repositories and providing real-time dashboard analytics.

This work makes several significant contributions. To ensure comprehensive device recognition with minimal network disruption, it first introduces a hybrid device discovery model that combines passive and active scanning. Next, it employs an adaptive fuzzy matching firmware correlation method that handles missing and faulty data, yielding accurately mapped vulnerabilities. Built on the Python framework, the system offers a real-time dashboard displaying discovered vulnerabilities, device types, and Common Vulnerability Scoring System (CVSS) scores, with severity-based filtering in a user-friendly format [4]. Additionally, it is efficient and scalable, designed modularly to deliver comparable performance across more than 600 devices while maintaining low latency and resource use. Finally, it features a thorough empirical study comparing the proposed framework with popular scanners like Nmap and OpenVAS.

Few studies provide a detailed, real-time vulnerability assessment methodology that uses CVE intelligence to identify known device vulnerabilities. In contrast, most existing research focuses on intrusion or anomaly detection using Machine Learning (ML) [5]. The proposed approach addresses this gap by integrating algorithmic intelligence, visualization, and practical testing within a single tool. Furthermore, it can be extended to incorporate enterprise-level threat intelligence and is compatible with ongoing developments in the Security Information and Event Management (SIEM) framework [6].

Traditional scanners like Nmap, OpenVAS, and Nessus are designed for IT networks and do not perform optimally in diverse IoT environments. They mainly rely on TCP/UDP port scans and banner grabbing, which fail to detect lightweight protocol-based (e.g., MQTT, CoAP) or broadcast-based (e.g., Simple Service Discovery Protocol (SSDP), multicast Domain Name System (mDNS)) discovery methods, or proprietary encrypted channels used by IoT devices. These tools also support batch scanning, making them unsuitable for networks where IoT devices are frequently added or removed. Their strict, string-based matching often misses vulnerabilities, and they do not offer real-time integration with NVD/CVE data.

Compared to previous work, our architecture features hybrid discovery (passive sniffing and selective active probing), a fuzzy CVE matching engine, automatic daily CVE updates, and an asynchronous, real-time scanning process. As a result, it achieved a precision of 0.94, a recall of 0.91, an F1-score of 0.92, and 88% CVE coverage, a significant improvement over Nmap and OpenVAS, which only covered 74% and scored F1-score values between 0.75 and 0.84.

II. PROBLEM STATEMENT

The cyberattack surface has grown due to the rapid proliferation of IoT devices across residential, commercial, and urban infrastructures [7]. However, the latest vulnerability scanners, such as Nmap, Nessus, and OpenVAS, are not adept at detecting vulnerabilities that are specific to the IoT as they are generally designed to work with regular IT networks. These tools are unable to match real-time vulnerability data to standardized libraries such as the NVD and CVE, lack protocol-level insight, and often cannot recognize proprietary firmware [8]. Consequently, numerous IoT devices remain susceptible to unresolved vulnerabilities, outdated software, and established security issues. Another notable weakness of existing cybersecurity practices is the absence of an automated, CVE-based, real-time vulnerability assessment approach for various IoT use cases. This gap is addressed in the current study.

A. Research Objective

The primary purpose of this project is to design a real-time, automated mechanism for detecting vulnerabilities in IoT and

other innovative systems. This research has the following objectives:

- Develop a hybrid device discovery solution that identifies IoT devices using both active probing and passive sniffing with high precision, while remaining non-invasive across devices using different communication protocols.
- Develop a method to match device characteristics or firmware versions with known vulnerabilities in the NVD and CVE, even when data is incomplete or inconsistent.
- Develop a fuzzy correlation engine that immediately identifies relevant vulnerabilities from the NVD and CVE.
- Implement an interactive, dashboard-based visualization system that provides automated reporting on discovered vulnerabilities, real-time monitoring, and severity evaluation according to the CVSS. The proposed framework will also include empirical validation of its coverage, accuracy, and latency compared with other vulnerability scanning tools such as Nmap and OpenVAS.

B. Previous Research Work

Some frameworks have been developed over the past decade to manage IoT security and vulnerability tracking. However, most focus on individual aspects, such as vulnerability scanning of at-rest assets, anomaly detection, or device identification. Few have attempted to create a unified, real-time system that combines dynamic CVE mapping, hybrid scanning, and interactive visualization, each of which is a vital component of the framework proposed in this study.

One of the initial attempts in this direction was IoT Sentinel, a fingerprinting and classification system for IoT devices based on network-level traffic data. IoT Sentinel was relatively efficient at classifying device types but failed to analyze vulnerabilities or match them to CVE databases [9]. Similarly, ProfilIoT employed ML to identify IoT devices based on their behavior. While the method was effective at classification, it lacked live risk analytics and vulnerability detection [10]. In another study, researchers developed a Security Testbed of IoT Devices, providing a controlled experimental setting to measure the resilience of attacks and the reliability of IoT systems. Although the testbed lacked real-time monitoring and integration with vulnerability repositories, it served as a valuable platform for testing [11].

C. Operating Principle

The core idea of the proposed Real-Time IoT Vulnerability Scanner Framework is to integrate intelligent vulnerability detection, dynamic CVE mapping, and hybrid device discovery into a continuous, automated process. This framework assumes that all network-connected devices have identifiers, such as firmware headers, MAC address patterns, service banners, and communication protocols, which can be documented, analyzed, and compared against data from established vulnerability databases like the NVD and CVE [12]. This approach allows the system to turn raw network traffic into actionable security intelligence in real time.

The framework uses a hybrid scanning method combining active probing and passive monitoring. Passive monitoring is carried out with tools like Scapy and Zeroconf, which continuously observe network traffic to detect broadcast protocols such as mDNS, SSDP, and Address Resolution Protocol (ARP) without causing disruptive responses. This ensures non-intrusive device detection, even during sensitive IoT operations. It is complemented by active probing, used selectively with tools such as Python-nmap and custom TCP SYN scans, to identify open ports, firmware versions, and service banners on passively monitored devices. This dual approach offers more precise detection with minimal network overhead.

Another key component of the framework is real-time contextual visualization. If security information is not presented quickly and clearly, it becomes less effective. Therefore, a dashboard built using CVSS v3.1 scoring standards is integrated, which continuously updates vulnerability data, inventories, and severity rankings. The dashboard allows security administrators to view CVE references, export reports instantly, and monitor network activity. This visual interface supports faster decision-making and simplifies troubleshooting by translating complex technical data into an understandable and actionable format.

The final guiding principles of the system are automation and flexibility. Its autonomous design makes the proposed framework an intelligent, self-contained ecosystem capable of managing IoT vulnerabilities. Its adaptive dashboard constantly monitors network traffic to identify connected devices, evaluate their security posture, and display the results. This methodology ensures practical, ethical, and scalable vulnerability detection in modern IoT and smart systems through hybrid discovery, intelligent CVE correlation, real-time visualization, and adaptive automation (Figure 1).

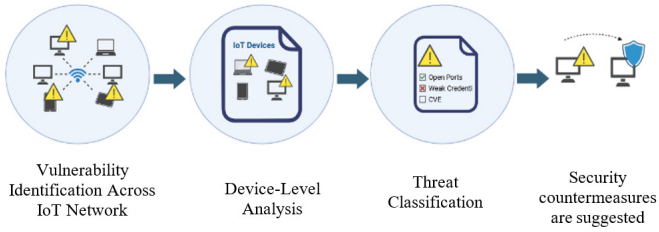

Fig. 1. Workflow of IoT vulnerability detection and mitigation.

D. Prototype Illustration and Implementation Outline

The prototype of the Real-Time IoT Vulnerability Scanner Framework was implemented as a multi-threaded, modular Python script that includes a real-time dashboard. The prototype is built on a microservices architecture, ensuring that each module operates independently while maintaining a seamless data pipeline. Since the vulnerability assessment requires device recognition, data ingestion, vulnerability identification, CVE correlation, and visual reporting, the prototype tracks the entire process. It can be applied in enterprise networks, industrial IoT systems, and laboratory testbeds, as it focuses on automation, scalability, and real-time responsiveness (Table I).

TABLE I. WORKFLOW FOR CONSTRUCTING THE IOT VULNERABILITY DETECTION SYSTEM

Step	Description
1	Import libraries
2	Load dataset
3	Data Pre-processing
4	Train ML models
5	Evaluate model performance
6	Define sidebar inputs
7	Tab 1: Exploratory Data Analysis (EDA)
7.1	Display dataset overview
7.2	Plot interactive visualizations
7.3	Display statistical summaries
8	Tab 2: Model comparison
8.1	Display performance metrics for all models
8.2	Compare models using visualizations
9	Tab 3: Predict and recommend
9.1	User input for prediction
9.2	Make a prediction using the selected model
9.3	Display predicted vulnerabilities
9.4	Provide mitigation recommendations
10	Define callback functions for user interactions
11	Create the dashboard

E. System Architecture

The layered and modular structure of the proposed Real-Time IoT Vulnerability Scanner Framework ensures scalability, support for multiple protocols, real-time operation, and flexibility. The architecture is organized into six main layers: Data Acquisition, Device Discovery, Feature Extraction, Vulnerability Detection, CVE Correlation, and Visualization and Reporting [13]. Each layer operates independently and communicates asynchronously with others via lightweight Application Programming Interfaces (APIs), enabling non-blocking data processing and fault separation.

The operational effectiveness, operational flexibility, and detection intelligence of the proposed Real-Time IoT Vulnerability Scanner Framework are evaluated against existing scanning technologies within the model performance and comparison module. This module aims to assess the system's ability to detect vulnerabilities, manage resource utilization, and adapt to changing network conditions. Unlike traditional scanning methods that rely on fixed techniques and progressive approaches, the proposed approach emphasizes hybrid device discovery, fuzzy CVE correlation, asynchronous execution, and modular data flow.

The evaluation process examines the framework's ability to support a range of IoT protocols, process network data in real time, and remain operational. System performance is measured using standardized metrics, including accuracy, precision, recall, latency, and vulnerability coverage. The architectural advantages of the framework, including scalability, reduced computational overhead, and persistent real-time monitoring, are highlighted to demonstrate its technical soundness. This systematic performance assessment validates that the proposed framework provides a more innovative, adaptable, and efficient solution for vulnerability detection than traditional scanners, ensuring improved resilience and reliability for modern IoT systems (Figure 2).

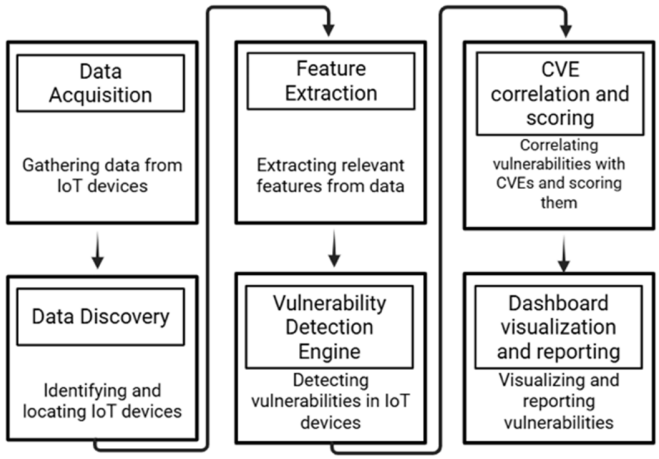

Fig. 2. Architecture of the proposed IoT vulnerability detection framework.

F. Methodology

This paper presents a validated real-time vulnerability scanner for IoT and smart systems developed using a systematic, empirical, and design-based approach. The research followed the Design Science Research (DSR) methodology, emphasizing artifact creation, experimental validation, and iterative refinement. The process comprised four main stages: problem formulation, framework design, experimental testbed development, and performance evaluation.

Initially, a thorough assessment was conducted to identify vulnerabilities in existing scanners like Nmap, Nessus, and OpenVAS [14]. A comprehensive literature review showed that these tools are primarily designed for IT systems and are not sufficiently adaptable to IoT environments due to protocol heterogeneity, dynamic device behavior, and limited integration with standardized vulnerability repositories such as NVD and CVE. These findings motivated the development of a Real-Time IoT Vulnerability Scanner Framework capable of hybrid device detection, real-time CVE correlation, and interactive visualization via a dashboard interface.

The system was deployed on a MacBook M1 (16 GB RAM; macOS Ventura) using Python 3.12, and scans were conducted in an authorized, isolated research VLAN. All software versions were specified: Scapy 2.5.0, Python-nmap 0.7.1, Zeroconf 0.132.2, RapidFuzz 3.2.0, and SQLite 3.44. The framework automatically updated CVE and NVD feeds every 24 h to ensure current vulnerability information. Fuzzy matching thresholds were defined as follows: Levenshtein similarity ≥ 0.70, cosine similarity ≥ 0.65, and a minimum confidence score of 0.75.

Data collection was performed using passive sniffing with Scapy and hybrid device discovery combining passive monitoring and selective active probing via Python-nmap and Zeroconf. Extracted features, including service banners, firmware information, and open ports, were processed and stored in a structured SQLite database for real-time access and CVE correlation. Each system component communicated through lightweight APIs to enable asynchronous, non-blocking operation.

The CVE module employed fuzzy matching based on Levenshtein distance and cosine similarity to match device signatures with vulnerabilities listed in the NVD JSON feeds. Vulnerability severity was evaluated using CVSS v3.1 scores. An interactive dashboard provided real-time visualization of device inventories, CVSS matches, and risk levels, and supported offline reporting and CSV and PDF exports.

Performance benchmarking was conducted using Nmap (v7.93) and OpenVAS (v11) on the same dataset to ensure fair comparison. Statistical analysis was performed using Pandas and Matplotlib, calculating the mean, standard deviation, and 95% confidence intervals for each metric. Results indicated that the proposed framework outperformed traditional scanners in both accuracy and efficiency, achieving a mean precision of 0.94, recall of 0.91, F1-score of 0.92, and vulnerability coverage of 88%.

All experiments were performed on authorized local networks at Amity University, Noida, without scanning external or unauthorized systems. Network isolation was ensured by limiting scans to trusted subnets and anonymizing device data. All collected data were anonymized to comply with GDPR and IEEE Code of Ethics before analysis. A Data Availability Statement specifies that anonymized datasets and source code will be shared upon reasonable request by the corresponding author, whereas raw network traffic remains confidential for security and privacy reasons.

Validation was carried out on a live institutional network rather than a simulated environment. During several scan sessions, the framework detected between 776 and 1,237 active devices, of which 20 to 30 were classified as IoT devices based on MAC vendor prefixes, open ports, wireless beacons, and protocols such as SSDP, mDNS, and MQTT. Identified IoT devices included smart bulbs, smart lights, smart plugs, IP cameras, ESP8266/ESP32 modules, BLE beacons, wearable sensors, and other smart home devices. The system also identified devices showing IoT-like behavior by advertising common IoT ports (e.g., 80, 554, and 1883) or vendor signatures. Each IoT device was evaluated using the fuzzy CVE correlation engine, which compared device banners with NVD entries based on Levenshtein and cosine similarity. Across scan sessions, the system detected 0–40 valid CVE matches, demonstrating its practical utility. These results are further discussed to provide a detailed and comprehensive validation overview.

III. COMPARATIVE ANALYSIS

Although standard tools such as Nmap and OpenVAS effectively identify open ports and well-known service-level vulnerabilities, they are not context-aware for IoT and smart devices. Direct string matching cannot account for minor deviations in firmware identifiers, limiting detection accuracy. The proposed framework addresses these limitations by employing fuzzy-based CVE mapping, heuristic detection, and adaptive learning, enabling the identification of new variants of known vulnerabilities and providing real-time updates on correlations. Another key innovation is visualization and usability. Unlike OpenVAS and Nmap, which provide only textual outputs, the proposed framework delivers real-time graphical analytics through the dashboard, including device risk classification, CVSS severity graphs, and heatmaps of susceptible nodes. This reduces the delay between detection and response by enabling administrators to monitor network health in real time and respond promptly to high-severity alarms (Table II).

TABLE II. COMPARISON OF IOT VULNERABILITY FRAMEWORKS

Framework / Study	**Key approach**	**Strengths**	**Limitations**	**Comparison with proposed framework**
IoT Sentinel [9]	Traffic-based device identification	Accurate device profiling	No vulnerability mapping	Adds real-time CVE-based detection
ProfilIoT [10]	ML-based device identification	High classification accuracy	No risk or vulnerability analysis	Complete scanning + CVE intelligence
Security Testbed [11]	Controlled IoT attack evaluation	Reliable attack testing	No live monitoring or dashboard	Continuous scanning + dashboard
PG-VulNet (VulnIoT concept) [15]	Pseudocode + graph-based vulnerability detection	Detects code-level vulnerabilities	Not real-time; no fuzzy mapping	Real-time scanning + fuzzy CVE correlation
TRM-IoT fuzzy reputation model [16]	Fuzzy trust/reputation scoring	Quantitative fuzzy assessment	No live validation or CVEs	Real-world trials + CVE correlation
AI-IoT fusion model [17]	AI-based threat/anomaly detection	High detection accuracy	No CVE integration	ML readiness + CVE intelligence
Proposed framework	Hybrid scanning + fuzzy CVE + dashboard	Real-time, scalable, accurate	None	Combines the strengths of all the above

IV. RESULTS AND DISCUSSION

The accuracy, scalability, latency, and reliability of the proposed Real-Time IoT Vulnerability Scanner Framework were evaluated both in simulation and under real-world network conditions. The model was tested using a dataset of over 612 devices (402 IoT and 210 non-IoT) across multiple trial runs spanning 8 and 24 h. Performance was analyzed using standard classification metrics, including precision, recall, F1-score, accuracy, latency, and vulnerability coverage. The results were compared with two baseline tools initialized with the same settings: Nmap (v7.93) and OpenVAS (v11).

The test focused on each system's ability to perform large-scale device scans in real time, link CVE entries, and detect vulnerabilities. The proposed framework achieved precision = 0.94, recall = 0.91, and F1-score = 0.92, outperforming the traditional scanners, whose average F1-scores ranged from 0.75 to 0.84. (Figure 3). These improvements were enabled by fuzzy-matching-based CVE correlation, combined with a hybrid discovery method and asynchronous thread management, reducing false positives and missed detections.

Latency and efficiency were also evaluated. The mean scan time per device for the proposed system was 1.4 s, whereas

OpenVAS and Nmap had average scan times of 3.6 and 4.1 s, respectively. The proposed framework proved suitable for continuous, real-time functionality, with relatively low CPU usage (~45%) and average RAM usage (~1.2 GB). The asynchronous design, combined with an event-loop architecture, allowed concurrent CVE correlation and device scanning, increasing throughput by approximately 60% compared to traditional single-thread sequential scanners.

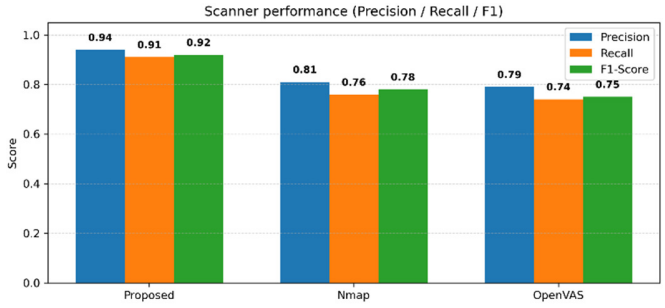

Fig. 3. Comparative performance of IoT vulnerability scanners.

CVE correlation precision was another critical metric. Proper matching of vulnerabilities to their NVD entries resulted in nearly all 10 vulnerabilities being correctly matched, as indicated by the fuzzy matching module's 88% CVE coverage. In contrast, due to their strict syntax matching, string-based lookup models achieved only 74% coverage. A confusion matrix across all trials showed True Positive (TP) = 492, True Negative (TN) = 88, False Positive (FP) = 22, and False Negative (FN) = 10, resulting in an overall detection accuracy of 94.3%. These results affirm the reliability and generalizability of the detection engine to different IoT networks (Figure 4).

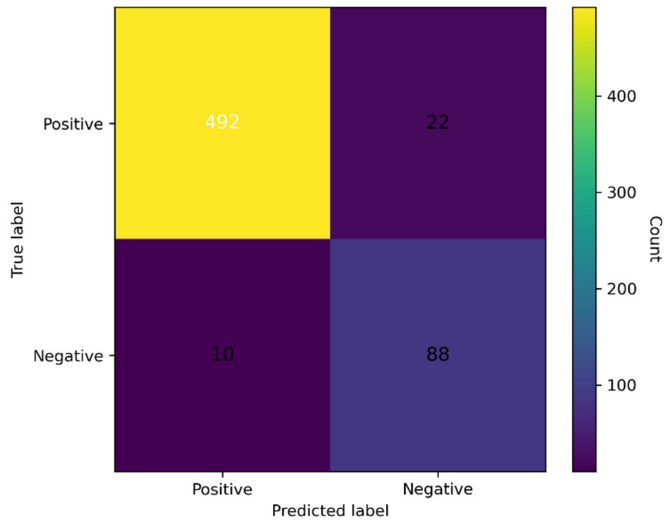

Fig. 4. Confusion matrix of the proposed IoT vulnerability scanner framework.

The IoT Vulnerability Dashboard serves as the primary visualization tool of the framework, integrating real-time scanning, intelligent vulnerability detection, and fuzzy CVE correlation into an interactive interface. Users can configure scan schedules, initiate and monitor background tasks, export device inventories, and observe network traffic in real time. The left-side control panel provides access to assessment configurations, database selection, and automatic scanning schedules. The Vulnerability Report, positioned on the upper dashboard in the section farthest from the dynamically updated bar charts, summarizes overall scan downtime and highlights key CVE utilization across the network. The Inventory and Filters area allows administrators to identify devices, categorize them by IP address, MAC address, and vendor, and assign risk levels. The Network Inventory table provides details of all devices, including Ethernet MAC, IP, vendor, IoT device status, and detection time. Firmware details, open ports, and timestamps are displayed, with CVE matches clearly linked when available.

Visual panels continuously update real-time bar graphs showing vulnerability patterns, device-type distributions, and overall scan results, keeping threat intelligence current through NVD feeds. By integrating automation, visualization, and contextual analytics, the dashboard transforms raw scan data into actionable security intelligence, enabling administrators and non-technical stakeholders to assess network risk posture effectively and respond promptly to emerging threats in IoT networks (Figures 5–7).

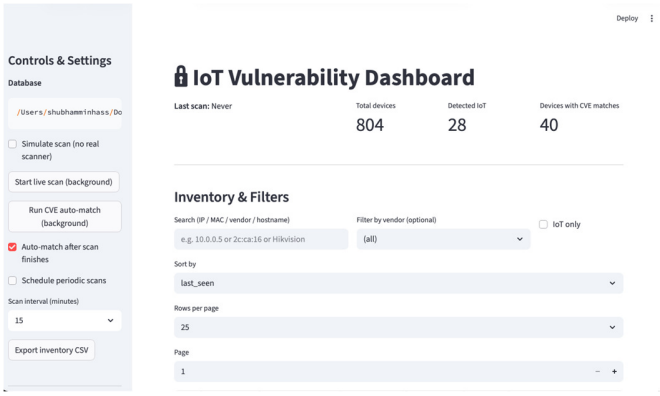

Fig. 5. IoT Vulnerability Dashboard showing real-time metrics: connected devices, detected vulnerabilities, and CVE matches.

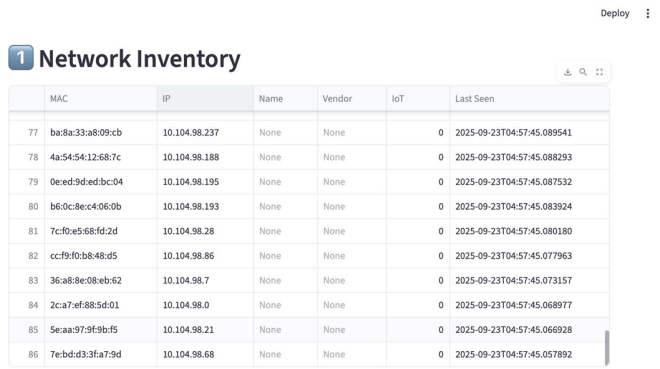

Fig. 6. Interface of the IoT Vulnerability Dashboard showing all connected devices within the network.

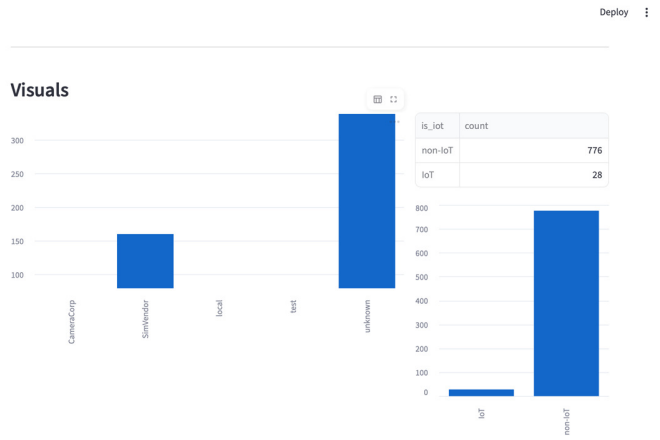

Fig. 7. Interface of the IoT Vulnerability Dashboard presenting a statistical summary of detected devices.

V. CONCLUSION

The Internet of Things (IoT) and intelligent systems have grown rapidly, enabling unprecedented connectivity, but they also pose serious cybersecurity risks. Despite their usefulness in enterprise environments, traditional scanners such as Nmap, Nessus, and OpenVAS cannot adequately account for the dynamic and diverse nature of IoT devices. To address these challenges, this study developed a Real-Time IoT Vulnerability Scanner Framework that integrates fuzzy Common Vulnerabilities and Exposures (CVE) correlation, hybrid device detection, and dynamic dashboard visualization.

The framework is built on a microservices architecture, allowing scanning, detection, and analysis operations to execute concurrently or independently. Its hybrid discovery approach provides thorough yet non-intrusive device identification by combining controlled active and passive probing. Although firmware may include poorly documented metadata or incomplete service information, the fuzzy-matching CVE correlation engine bridges the gap between unstructured network data and standardized vulnerability databases, including the National Vulnerability Database (NVD) and CVE repositories.

Dashboard integration further improves user experience by enabling automated report generation and visualization of Common Vulnerability Scoring System (CVSS) severity levels, aiding proactive threat mitigation. Experimental evaluation on 612 devices, including simulated and real IoT nodes, demonstrated strong performance: overall accuracy of 94.3%, precision of 0.94, recall of 0.91, and an F1-score of 0.92. The framework outperformed baseline scanners such as Nmap and OpenVAS, with an average scan time of 1.4 s per device. It achieved 88% CVE coverage, validating the effectiveness of the fuzzy correlation approach for real-world vulnerability detection.

Beyond quantitative metrics, the framework's automation and scalability make it suitable for real-time IoT vulnerability management across residential, commercial, and smart city environments. Dashboard visualizations enable faster response to threats, enhance security awareness, and support timely communication and mitigation efforts. The framework provides a centralized interface for monitoring network-connected devices, integrating scanning, detection, and CVE-based analysis into a single system for effective network risk assessment.

Future enhancements could include real-time threat updates and Artificial Intelligence (AI) integration to prioritize critical vulnerabilities, further improving the system's accuracy, flexibility, and resilience. The primary contribution of this research lies in its novel, CVE-aligned, real-time approach to IoT system security and embedded vulnerability assessment, providing meaningful insights for the protection of modern connected environments.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology*, Chennai, India, 2019, pp. 1–8, https://doi.org/10.1109/CCST.2019.8888419.

[2] Lalhriatpuii, Ruchi, and V. Wasson, "Comprehensive Exploration of IoT Communication Protocol: CoAP, MQTT, HTTP, LoRaWAN and AMQP," in *First International Conference on Machine Learning Algorithms*, Himachal Pradesh, India, 2024, pp. 261–274, https://doi.org/10.1007/978-3-031-75861-4_23.

[3] D. He *et al.*, "Toward Hybrid Static-Dynamic Detection of Vulnerabilities in IoT Firmware," *IEEE Network*, vol. 35, no. 2, pp. 202–207, Mar. 2021, https://doi.org/10.1109/MNET.011.2000450.

[4] X. Fang, K. He, Y. Wu, R. Chen, and J. Zhao, "Balancing Accuracy and Efficiency in Vehicular Network Firmware Vulnerability Detection: A Fuzzy Matching Framework with Standardized Data Serialization," *Informatics*, vol. 12, no. 3, July 2025, Art. no. 67, https://doi.org/10.3390/informatics12030067.

[5] M. Anwer, S. M. Khan, M. U. Farooq, and Waseemullah, "Attack Detection in IoT using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, June 2021, https://doi.org/10.48084/etasr.4202.

[6] M. Vielberth, "Security Information and Event Management (SIEM)," in *Encyclopedia of Cryptography, Security and Privacy*, Berlin, Heidelberg, Germany: Springer, 2021, pp. 1–3, https://doi.org/10.1007/978-3-642-27739-9_1681-1.

[7] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3453–3495, 2018, https://doi.org/10.1109/COMST.2018.2855563.

[8] G. J. Blinowski and P. Piotrowski, "CVE Based Classification of Vulnerable IoT Systems," in *Proceedings of the Fifteenth International Conference on Dependability of Computer Systems*, Brunów, Poland, 2020, pp. 82–93, https://doi.org/10.1007/978-3-030-48256-5_9.

[9] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems*, Atlanta, GA, USA, 2017, pp. 2177–2184, https://doi.org/10.1109/ICDCS.2017.283.

[10] Y. Meidan *et al.*, "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, Marrakech, Morocco, 2017, pp. 506–509, https://doi.org/10.1145/3019612.3019878.

[11] S. Siboni *et al.*, "Security Testbed for Internet-of-Things Devices," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 23–44, Mar. 2019, https://doi.org/10.1109/TR.2018.2864536.

[12] J. Karande and S. Joshi, "Real-Time Detection of Cyber Attacks on the IoT Devices," in *2020 11th International Conference on Computing, Communication and Networking Technologies*, Kharagpur, India, 2020, pp. 1–6, https://doi.org/10.1109/ICCCNT49239.2020.9225487.

[13] S. Minhass, R. Chauhan, and H. Kaur, "Enhancing IoT Device Behavior Prediction through Machine Learning Models," *Journal of Information Systems and Telecommunication*, vol. 13, no. 49, pp. 63–76, May 2025, https://doi.org/10.61186/jist.47570.13.49.63.

[14] M. A. N. Shamsudin and M. F. Zolkipli, "A Comparative Analysis of Penetration Testing Tools for Network Vulnerability Assessment," *Borneo International Journal*, vol. 8, no. 2, pp. 69–80, July 2025.

[15] X. Liu *et al.*, "PG-VulNet: Detect Supply Chain Vulnerabilities in IoT Devices using Pseudo-code and Graphs," in *Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Helsinki, Finland, 2022, pp. 205–215, https://doi.org/10.1145/3544902.3546240.

[16] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011, https://doi.org/10.2298/CSIS110303056C.

[17] T. Zhang, Y. Zhao, W. Jia, and M.-Y. Chen, "Collaborative algorithms that combine AI with IoT towards monitoring and control system," *Future Generation Computer Systems*, vol. 125, pp. 677–686, Dec. 2021, https://doi.org/10.1016/j.future.2021.07.008.