An Improved Intrusion Detection Scheme in a Smart Home Environment

Samuel Mtswenem Vanen
University of Mkar, Mkar, Nigeria

Sever Kwaghbee
Federal University of Technology, Owerri, Nigeria

Vershima Iyorter
University of Mkar, Mkar, Nigeria

Tivlumun Ge
Benue State University, Makurdi, Nigeria

Adekunle Adedotun Adeyelu
Benue State University, Makurdi, Nigeria

ABSTRACT
The fast increase in the number of Internet of Things (IoT) devices in smart homes makes them more exposed to cybersecurity threats. In turn, this creates an urgent need for robust intrusion detection systems. This study proposes an IoT Smart Home Multi User Access Control Intrusion Detection System (SHMUACIDS), with a view to improving the security by more efficiently detecting anomalies. It was designed based on a multi-layer architecture that consists of a Packet Capture Layer, a Feature Extraction Layer, the Machine Learning Model, and the Alerting System, all knitted together to work in tandem for proactively meeting the security challenges in IoT smart home environments. Intrusion Detection Data were obtained from kaggle website containing list of simulated TCP/IP connections were employed in training different machine learning models. The methodology also embeds digital signatures and proper key management, data integrity countermeasures together with an alert system which immediately notifies administrators on the detected anomalies. Results indicated that SHMUACIDS considerably outperformed the detection of anomalous activities in smart home IoT environments compared to some classical methods and previous studies. This holistic approach makes SHMUACIDS competitive in the smart home cybersecurity landscape.

Keywords
Internet of Things, Intrusion Detection, Machine Learning, Attacks, Smart Home, key management, Access control.

1. INTRODUCTION
The Internet of Things (IoT), also known as the Internet of Objects, is an extensive network of heterogeneous devices offering a wide range of innovative applications and services [1][2]. [1] suggest that IoT networks can integrate data and services to seamlessly interconnect cyber and physical systems, such as home and industry appliances and vehicles, embedded with software, sensors, and connectivity which enables them exchange data. Similarly, [3] describe the IoT as an ecosystem comprising smart objects equipped with sensors, actuators, networking, and processing technologies. These components integrate and collaborate to provide an environment where smart services are delivered to end users. Furthermore, IoT brings numerous benefits to human life by providing smart services that enhance daily activities anytime and anywhere, delivered through various applications within the IoT environment. IoT can also be described as embedding intelligence in individual objects, enabling them to detect changes in their physical state [4]. It is a system where computers, sensors, and objects interact, process data, and share them among devices, that emerges as a new technology system composed of multiple information technologies. Internet of Things (IoT) was designed to enhance the functionality of the original Internet, by enabling users to share information provided by both humans in databases and things in the physical world. Essentially, it connects physical objects to the Internet and to each other through various intelligent technologies, creating a smart ecosystem of pervasive computing. These systems are used in numerous application domains, including healthcare, smart homes, smart industries, environmental monitoring, smart agriculture, and smart cities just to mention a few. The benefits of using IoT systems include improved efficiency and cost-effectiveness, connectivity and automation, data collection and analysis, enhanced safety and security, environmental sustainability, and advancements in healthcare [5]. However, despite these benefits, IoT systems are vulnerable to various security attacks such as code injection, denial-of-service (DoS), distributed denial-of-service (DDoS), and man-in-the-middle (MITM) attacks. These attacks can significantly damage IoT services and smart environment applications. Consequently, securing IoT systems has become a major concern, as these attacks can affect IoT devices, websites, and online services and applications [6]. The concept of an Intrusion Detection System (IDS) aims to detect threats or intrusions into the network by actively monitoring network traffic, detecting potential malicious events, loging information about them, and stoping incidents. An IDS is a software tool specifically designed to discover irregularities in network traffic. Unwarranted or unexplained network changes could indicate malicious activity at any stage, from the beginning of an attack to a full-blown breach [7] [8]. To address these problems, IDS plays a crucial role in the IoT security framework, detecting both known and unknown attacks [9] [10]. This research focuses on developing an improved Smart Home Multi User Access Control Intrusion Detection scheme (SHMUACIDS) in an IoT Smart Home Environment, aiming to create a multi-user access control system that quickly identifies some malicious attacks, determines the level of threat, and alerts system administrators. The solution proposes use of access control mechanisms and digital anomaly techniques to secure the integrity of the IoT smart home environment at the application level and the alerting system to alert the administrators. This study is motivated by the need to ensure the integrity of system and data in smart home IoT environment which is crucial for maintain the safety and security of the smart home and its users. Attacks on the integrity and confidentiality of smart home IoT systems can lead to the manipulation, modification, or destruction, resulting in inaccurate or misleading information, which can result in severe consequences for the safety and security of the smart

home and its users. The proposed Smart Home User Access Control Intrusion Detection system (SHUACIDS) was able to provide a multi user access control mechanism and real-time intrusion detection system with intrusion alerting system. This system mitigates the burden posed by the previous system thereby improving intrusion in the IoT smart home system.

The contributions of this study are as follows:

i. Develop real-time packet sniffing with Scapy for anomaly detection, using a Decision Tree Classifier to trigger alerts on network anomalies.
ii. Integrate cryptographic functions for key management and data integrity, ensuring secure communications within the smart home system network.
iii. Create multifactor authentication and role-based access control, with detailed logging to monitor and prevent unauthorized access to IoT devices.
iv. Evaluate system performance and effectiveness through continues testing, monitoring and comparing the enhanced IDS with existing solutions in terms of detection rate, false positive, and response time.

2. LITERATURE REVIEW

The Internet of Things (IoT) are internet enabled devices embedded with wireless sensor networks which form a network of devices that provide advance and intelligent services [11]. They communicate and interact over the internet with the connected devices through a standard communication protocol and can be monitored and controlled remotely to perform a desired functionality. This has eventually transformed human-to-human communication and human-to-machine communication to machine-to-machine communication [12] [13]. The integration of IoT systems and smart environments makes smart objects more effective. How-ever, the impacts of IoT security vulnerabilities are very dangerous in critical smart environments such as homes. IoT-based smart homes face security and privacy challenges that span all layers of the IoT architecture. The creation of smart environments in the real world faces two notable barriers: the security of IoT systems and the complexity and compatibility of IoT smart home environments. One of these is single User Access Control using machine learning algorithms. Series of attempts have been made to address this challenge [14]. In the study by Azizjon Ikromjon O ‘G ‘Li (2024) on an IoT Network Intrusion Detection System (IDS) utilizing machine learning techniques, key findings reveal the performance of various models tested on the CICDDoS 2019 dataset. The Support Vector Machine (SVM) model achieved the highest accuracy at 0.996, effectively detecting attack traffic with 99% accuracy but exhibiting a false positive rate of 0.012. This suggests a need for further tuning to reduce misclassifications. The Random Forest model recorded an accuracy of 0.943, but its confusion matrix indicated significant misclassification of normal traffic as attacks (FP = 0.96). Despite this, it maintained a high true positive rate (TP = 0.91) and low false negative rate (FN = 0.087), with consistent performance across training epochs. Feature selection was crucial in enhancing model performance, focusing on key attributes such as packet size and protocol type. The study concludes that SVM is particularly effective for improving IoT network security but highlights ongoing challenges, including minimizing false positives and addressing dataset imbalance. Future work is recommended to explore real-time detection optimizations and advanced techniques like deep learning to further enhance system performance. According to the findings in [15], proposed an Anomaly-based intrusion detection system in IoT using deep learning techniques. These techniques have been adequately adopted by researchers in various fields as a solution in securing the IoT environment. In moving forward [16] proposed the use of deep learning architectures to develop an adaptive and resilient network intrusion detection system (IDS) to detect and classify network attacks. Their emphasis was on how deep learning or deep neural networks (DNNs) can facilitate flexible IDS with learning capability to detect recognized and new or zero-day network behavioural features, consequently ejecting the systems intruder and reducing the risk of compromise. [17] also proposed a comprehensive Intrusion Detection System (IDS) for Internet of Things (IoT) applications, named IP-IDS, capable of detecting intrusions across multiple application layer protocols, including MQTT, HTTP, and CoAP. The accuracy of the models was assessed through various metrics including precision, recall, and F1-score. Notably, while LSTM performed remarkably well on larger datasets, the DT model exhibited superior resource efficiency and shorter training times, making it more suitable for resource-constrained IoT environments. The study highlighted that the application of data balancing techniques and feature selection significantly enhanced the models' detection capabilities. Overall, IP-IDS outperformed several existing machine learning and deep learning models, setting a benchmark for future research in intrusion detection for IoT networks. In a report finding by [18] proposed a novel intrusion detection system that uses machine learning algorithms to detect security anomalies in IoT networks. This detection platform provides security as a service and facilitates interoperability between various network communication protocols used in IoT. They provided a framework of the proposed system and discuss the intrusion detection process in detail. The proposed intrusion detection system was evaluated using both, real network traces for providing a proof-of-concept, and on simulation for providing evidence of its scalability. Their results confirm that the proposed intrusion detection system is capable of detecting real-world intrusions effectively. The research in [19] focused on improving intrusion detection in Internet of Things (IoT) networks using a machine learning-based feature selection and ensemble modeling approach. They developed a heterogeneous stack classifier model that selected the top 15 features using the K-Best algorithm and combined traditional classifiers: Naive Bayes (NB), Random Forest (RF), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM). The findings revealed that the proposed stack classifier achieved remarkable performance metrics, including an accuracy of 99.99%, precision of 99.98%, recall of 99.99%, and an F1-score of 99.98%. In contrast, the individual classifiers demonstrated varying effectiveness; KNN attained 99.90% accuracy with perfect precision, RF achieved 98.20% accuracy, and SVM recorded 98.10% accuracy, showcasing their strong classification capabilities. However, NB exhibited significantly lower performance, with an accuracy of 80.46%. Findings from [20], proposed IoT-based intrusion detection system using convolution neural networks. They proposed the Temporal Convolution Neural Network (TCNN), an intelligent model for IoT-IDS that aggregates convolution neural network (CNN) and generic convolution, based on these concepts. To handle unbalanced datasets, TCNN is accumulated with synthetic minority oversampling technique with nominal continuity. It is also used in conjunction with effective feature engineering techniques like attribute transformation and reduction. The presented model is compared to two traditional machine learning algorithms, random forest (RF) and logistic regression (LR), as well as LSTM and CNN deep learning techniques,

using the Bot-IoT data repository. The outcomes of the experiments depict that TCNN maintains a strong balance of efficacy and performance. The work conducted in [21] highlighted that smart home systems often involve multiple users accessing multiple devices, typically through a dedicated app on a mobile device. Traditional access control mechanisms are designed for a single trusted user who manages device access, but multi-user, multi-device environments in smart homes introduce entirely new challenges. In such environments, users often have conflicting, complex, and dynamically changing demands for multiple devices, which traditional access control techniques cannot accommodate. Additionally, the integration of smart devices from various platforms or vendors within the same home environment renders existing access control methods ineffective. To address these challenges, the authors introduce KRATOS+, an innovative access control mechanism designed for multi-user and multi-device smart home settings. [22] explored the concept of access control in IoT systems, highlighting the numerous benefits IoT offers to individuals, industries, and society. However, the use of resource-constrained devices combined with the adoption of various technologies increases security threats and introduces new vulnerabilities. One of the most critical vulnerabilities in IoT applications is insecure access to smart home systems, web interfaces, backend APIs, cloud services, and mobile applications. Typically, smart devices are configured and managed through vendor applications, which may provide both smartphone-based and web-based interfaces via cloud-hosted services. Remote users can exploit these vulnerabilities, and performing unauthorized actions such as modifying, Destroying and even altering permissions to remotely control the systems. To mitigate these risks, a fine-grained authorization system should be implemented to restrict access to IoT device and smart home interfaces and data strictly to authorized users [22]. [23] reported that smart homes, as a key application of the Internet of Things (IoT), are significantly enhancing the living environment. To ensure data integrity, device security, and confidentiality, various cryptography-based protection schemes have been proposed. Among these, access control is considered a promising method for safeguarding generated data and devices from unauthorized access, making it an essential security measure for smart homes. However, [23] noted that existing centralized or single-user access control schemes are inadequate in terms of security and performance. These schemes are still susceptible to issues such as a single point of failure, low reliability, and poor scalability [24]. To address these challenges, their study proposed a decentralized or multi-user and reliable access control scheme for smart homes, known as DAC4SH, which leverages smart contracts. Specifically, the framework comprises four key components: An Access Policy Management Contract (APMC), a Data Attribute Management Contract (DAMC), a Subject Attribute Management Contract (SAMC), and a Data Access Control Contract (DACC). Together, these components enable fine-grained data access control, enhancing security and efficiency in smart home environments. [25] highlighted that with the vast amounts of IoT data, IoT search technology plays a crucial role in quickly retrieving accurate information to meet users' real-time search demands. However, this process often requires access to sensitive user data, including personal health records, location details, and social relationships, to deliver personalized services. Without a robust access control mechanism, the use of such private information poses significant security risks. Implementing an effective access control system ensures that resource access is properly monitored, allowing only authorized users to retrieve information under legitimate conditions. [26] identified data security as one of the major challenges in cloud computing. Since data is distributed across multiple machines and storage devices, including smart homes, servers, computers, and mobile devices, uncontrolled access to these resources poses significant security risks for end users. To maintain cloud reliability and build user trust, it is essential for cloud service providers (CSPs) to implement effective access control measures and ensure the protection and security of data and resources.

3 METHODOLOGY

The conceptual model for an improved intrusion detection system in smart homes within IoT environments comprises four primary layers: digital anomaly, user access control, intrusion detection and alerting system. Digital anomaly ensure the authenticity and integrity of messages exchanged between smart home devices and the central monitoring system. Access control restricts access to smart home devices and services based on predefined policies. The intrusion detection layer monitors smart home devices for abnormal or malicious activities indicative of intrusion attempts and the alerting system send an email alerts to the administrators for proper actions and decisions. These layers integrate into a central monitoring system that receives messages from smart home devices and applies digital anomaly verification, access control, and intrusion detection techniques to identify potential threats. Upon detecting a potential threat, the system alerts the user or administrator and takes appropriate actions, such as blocking access to the affected device or service, logging the incident, and generating a report.

3.1 Model of The Proposed System

The architectural model of the proposed system integrates several layers and components designed to enhance the functionality and security of the Smart Home User Access Control Intrusion Detection System (SHUACIDS). Below is a detailed breakdown of how these components interact within the system:

i. Packet Capture Layer

The Packet Capture Layer is the first component in the system architecture, responsible for capturing and processing network packets in real time. This layer utilizes the Scapy library, which provides powerful tools for network packet manipulation and analysis. The primary functions of this layer include:

Packet Capture: Scapy continuously monitors the network interface to capture packets. This real-time capture ensures that all network traffic in smart home IoT networks were analyzed without delay.

Packet Filtering: Filtering mechanisms are implemented to focus on specific types of traffic, reducing the processing load and enhancing the system's efficiency.

Pre-processing: Basic pre-processing of packets, such as handling different protocol layers and extracting initial raw features, is performed to prepare the data for further analysis.

ii. Feature Extraction Layer

The Feature Extraction component is critical for converting raw packet data into a structured format suitable for machine learning analysis. This layer involves:

***Data Parsing*:** Extracting relevant fields from the captured packets, including protocol type, source and destination IP addresses, and port numbers.

***Packet Length*:** This feature represents the size of the packet in bytes. Packet length can provide insights into the type

of traffic and potential anomalies, such as unusually large or small packets.

***Protocol Type*:** Identifying the protocol type (e.g., TCP, UDP, ICMP) helps classify the nature of the traffic. Different protocols have different behaviours, and anomalies may be protocol-specific.

***Service*:** The type of service being accessed (e.g., HTTP, FTP, SMTP) is crucial for understanding the context of the traffic. The service type is inferred based on the destination port and other packet attributes.

Flag: TCP flags indicate the state of the connection (e.g., SYN, ACK, FIN). These flags are essential for detecting unusual connection patterns, such as port scans or unexpected terminations.

***Source Bytes*:** The number of bytes sent from the source to the destination. This metric helps in identifying data exfiltration or large uploads from compromised hosts.

***Destination Bytes*:** The number of bytes sent from the destination to the source. Analysing this feature helps in detecting large downloads or abnormal data reception patterns.

***Source IP*:** The IP address of the packet's origin. Monitoring source IPs can help detect unauthorized access attempts or traffic from suspicious locations.

***Destination IP*:** The IP address of the packet's destination. This feature is useful for identifying targeted attacks or unusual communication patterns with external entities.

iii. Machine Learning Model

At the heart of the system lies the Machine Learning Model, which is responsible for detecting anomalies in the network traffic. The chosen model for this system is a Decision Tree Classifier, selected for its interpretability and effectiveness in handling complex decision boundaries. The integration of this model involves:

Dataset Description

The dataset used for training the Machine Learning Model in this study is the "Network Intrusion Detection" dataset, available on the Kaggle website (https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection). This dataset was originally created by simulating a military network environment, specifically a typical US Air Force LAN, which was subjected to various network attacks. The dataset contains raw TCP/IP dump data, where each connection represents a sequence of TCP packets exchanged between a source IP address and a target IP address under a specific protocol. Each connection is labelled as either "normal" or "anomalous," with anomalies corresponding to different types of simulated attacks. The dataset consists of 41 features extracted from each TCP/IP connection, including 3 qualitative and 38 quantitative features. These features provide a comprehensive view of network activity, which is crucial for detecting deviations from normal behaviour. The class variable in the dataset has two categories:

1. **Normal**
2. **Anomalous**

This labelled dataset serves as the foundation for training the Decision Tree Classifier, which is central to the anomaly detection capabilities of the Smart Home User Access Control Intrusion Detection System (SHUACIDS).

Model Loading: The pre-trained Decision Tree Classifier model is loaded. This model has been trained on a labelled dataset containing both normal and anomalous traffic, enabling it to learn the patterns and characteristics of different types of network activities.

Training: The model is trained on a labelled dataset containing both normal and anomalous network traffic. This training process allows the model to learn patterns and distinguish between benign and malicious activity.

Prediction: Once trained, the model is used to predict the likelihood of incoming network traffic being anomalous. Each captured packet's features are fed into the model, which outputs a classification result.

Alerting System

The alerting system is a critical component of the overall architecture, responsible for notifying administrators when anomalies are detected. This section describes the alerting mechanisms and processes implemented to ensure timely and effective notifications.

Alert Types

The alerting system categorizes alerts based on the severity and type of anomaly detected. This categorization helps administrators prioritize their responses and take appropriate actions. Key aspects include:

Severity Levels: Alerts are classified into different severity levels (e.g., low, medium, high) based on the potential impact of the detected anomaly. For example, a high volume of suspicious traffic might trigger a high-severity alert, whereas a minor deviation might be classified as low severity.

Anomaly Types: Alerts are also categorized by the type of anomaly detected (e.g., True or False). This categorization provides context to the alerts, helping administrators understand the nature of the threat.

Email Notification: To ensure prompt notification, the alerting system utilizes Flask-Mail to send email alerts to designated recipients. The email notifications provide detailed information about the detected anomalies, enabling administrators to respond quickly and effectively. Key steps in the email notification process include:

Alert Generation: When an anomaly is detected, the system generates an alert containing detailed information about the anomaly. This includes the type of anomaly, severity level, and relevant packet features.

***Email Composition*:** The alert information is formatted into a structured email message. The email includes a subject line indicating the alert type and severity, and a body containing detailed information about the detected anomaly.

Recipient Management: The system maintains a list of designated recipients who should receive the alerts. This list can include network administrators, security personnel, and other stakeholders responsible for network security.

Email Sending: Using Flask-Mail, the system sends the composed email alerts to the designated recipients. This ensures that relevant personnel are promptly informed of the detected anomalies and can take appropriate actions

v. Data Signing

Data signing is a fundamental technique used to guarantee that the data has not been altered or tampered with after it has been captured. This is accomplished through the use of cryptographic algorithms that generate a unique digital signature for each piece of data. The specific steps involved in data signing include:

Cryptographic Algorithms: Advanced cryptographic algorithms, such as RSA (Rivest-Shamir-Adleman) or ECDSA (Elliptic Curve Digital Signature Algorithm), are employed to create digital signatures. These algorithms provide strong security assurances and are widely accepted standards in the industry.

Signature Generation: For each captured packet, a hash function is applied to generate a digest of the data. This digest is then encrypted with a private key to create the digital signature. This process ensures that even a small change in the data will result in a completely different signature, making tampering easily detectable.

Integration with Packet Capture: The digital signature is appended to the packet data at the time of capture, ensuring that any modifications to the data can be detected later during the verification process.

Vi Verification

The verification process is essential to confirm the authenticity and integrity of the captured data. It involves validating the digital signatures against the original data to ensure that no alterations have occurred. The steps involved in verification include:

Signature Verification Algorithms: Public key cryptography is used to verify the digital signatures. The public key corresponding to the private key used in the signing process is employed to decrypt the signature and compare it.

Data Integrity Check*:* By comparing the decrypted signature with the hash of the captured data, the system can detect any discrepancies that indicate tampering. If the hash values match, the data is considered authentic; otherwise, it is flagged as potentially compromised.

Automated Verification**:** The verification process is automated to ensure real-time validation of captured packets. This automation enables the system to continuously monitor data integrity without human intervention, providing immediate alerts in case of any issues.

Vii Key Management

Effective key management is crucial for maintaining the security of the cryptographic operations. This involves the secure generation, storage, distribution, and rotation of cryptographic keys to prevent unauthorized access and ensure the ongoing security of the system. Key management practices include:

Key Generation: Secure cryptographic keys are generated using strong random number generators to ensure their unpredictability. These keys form the foundation of the cryptographic operations and must be protected against unauthorized access.

Key Storage: Cryptographic keys are stored in secure hardware modules, such as Hardware Security Modules (HSMs) or encrypted key vaults, to prevent unauthorized access. These storage solutions provide tamper-resistant environments for key storage, ensuring that keys are only accessible to authorized components of the system.

Key Distribution: Secure channels are used to distribute cryptographic keys to authorized entities within the system. This distribution process includes measures such as encryption and authentication to protect the keys during transit.

Key Rotation and Revocation: Regular key rotation policies are implemented to minimize the risk of key compromise. In the event of a suspected key breach, immediate key revocation and replacement procedures are in place to maintain the integrity and security of the system.

viii. Multi Access User Control and Management

Effective user control and management are critical components of the system's overall security framework. By implementing robust access control mechanisms, the system ensures that only authorized users can access specific functionalities, thereby protecting sensitive information and maintaining system integrity. The primary focus of the multi-access user control and management includes both authentication and authorization processes, which are detailed below:

ix. Authentication

Authentication is the first line of defence in the system's security architecture. It verifies the identity of users attempting to access the system, ensuring that only legitimate users can proceed. The authentication process incorporates several key elements:

User Credentials**:** Users are required to provide valid credentials, such as a username and password, to gain access. Passwords are stored securely using hashing algorithms to prevent unauthorized retrieval.

Multi-Factor Authentication (MFA): To enhance security, the system supports multi-factor authentication, which requires users to provide additional verification, such as a one-time password (OTP) sent to their mobile device or email, or using biometric verification.

Session Management: Once authenticated, the system maintains secure user sessions to track user activity. Sessions are managed using secure tokens that expire after a predetermined period of inactivity, ensuring that unauthorized users cannot hijack sessions.

x. Authorization

Authorization determines what authenticated users are allowed to do within the system. It involves assigning roles and permissions to users based on their responsibilities and the principle of least privilege. The authorization process includes the following components:

Role-Based Access Control (RBAC): Users are assigned specific roles, such as administrator, operator, or views. Each role has a predefined set of permissions that dictate what actions users can perform and what resources they can access.

Granular Permissions: Permissions are defined at a granular level, allowing precise control over user actions. For example, administrators may have full control over system configuration and user management, while operators may be limited to monitoring network traffic and responding to alerts.

4 RESULTS AND DISCUSSION

4.1 Model Evaluation Results

The results of selected models were evaluated and presented below.

4.1.1 *Naive Bayes Classifier*

The Naive Bayes classifier demonstrated a cross-validation mean score of 0.907 and an overall accuracy of 0.907 on the test data. While it performed reasonably well, its accuracy and precision were lower compared to other models, particularly in detecting anomalies as presented in table 1.

4.1.2 *Decision Tree Classifier*

The Decision Tree classifier achieved a cross-validation mean score of 0.996 and an accuracy of 1.0, making it the best-performing model in this evaluation as presented in table 1.

4.1.3 *K-Nearest Neighbours (KNN)*

The KNN classifier achieved a cross-validation mean score of 0.991 and an accuracy of 0.994. It performed well but fell slightly short of the Decision Tree model in terms of precision and recall as presented in table 1.

4.1.4 *Logistic Regression*

The Logistic Regression produced a cross-validation mean score of 0.954 and an accuracy of 0.955. Despite its strong performance, it was outperformed by the Decision Tree and KNN models in this context as seen in table 1.

4.2 System Implementation

The Smart Home Multi-User Access Control Intrusion Detection SHMUACID was implanted and the system's user interface was designed for ease of use and accessibility, offering a comprehensive overview of intrusion detection integration within a smart home environment. The following pages demonstrated the system's capabilities, including real-time monitoring, user management, and alert handling, ensuring that users can effectively safeguard their smart home against potential intrusions.

4.2.1 *Main Page*

The main page of the SHMUACIDS as seen on figure 1 provides a dashboard overview of the system's status, including real-time monitoring of network traffic, detected anomalies, and system alerts. The interface is designed for ease of use, allowing users to quickly assess the security of their smart home environment.

4.2.2 *Registration Page*

The registration page as depicted on figure 2 allows new users to create an account, ensuring that only authorized individuals can access the system. It supports secure password creation and stores credentials in a securely encrypted format.

4.2.3 *Login Page*

The login page as depicted on figure 3 provides secure access to the system, incorporating multi-factor authentication (MFA) for enhanced security. This page is the gateway to all system functionalities.

4.2.4 *Alert Page*

The alert page as seen on figure 4 displays a list of all detected anomalies, including details such as the time of detection, the nature of the anomaly, and the actions taken by the system. Users can review these alerts and take further action if necessary.

4.2.5 *Devices Page*

The devices page as seen on figure 5 lists all connected smart home devices, providing information on their status, network activity, and assigned users. This page also allows users to manage their devices effectively.

4.2.6 *Assign and Remove Device Page*

The Assign and Remove Device page as seen on figure 6 allows users to assign devices to specific users or remove them from the system. This page simplifies device management by ensuring that only the necessary devices are active and associated with the correct users, improving the overall security and efficiency of the system.

4.2.7 *Email Alert Page*

The Email Alert page depicted on figure 7 enables users to configure their email notifications for various system events, such as security alerts or device status changes. Users can select which types of alerts they wish to receive, ensuring that they stay informed without being overwhelmed by unnecessary notifications.

4.3 Discussion

From the results presented in table 1, the Naive Bayes classifier shows a moderate performance with some misclassifications. The relatively high number of false positives (1245) indicates that the model tends to overestimate the presence of anomalies, which could lead to unnecessary alerts in a real-world system. Despite this, the model has a reasonable balance between detecting true anomalies and maintaining a low false negative rate. Again, from the results presented above, the Decision Tree classifier demonstrates perfect performance with no misclassifications. This indicates a high level of precision and recall, making it the most reliable model for detecting both anomalies and normal instances in the dataset. The absence of false positives and false negatives is crucial for a security system, as it ensures that legitimate traffic is not misclassified as malicious and that all potential threats are correctly identified. The KNN classifier performs very well with a minimal number of misclassifications. The small number of false positives (77) and false negatives (33) indicates that the model is effective in distinguishing between normal and anomalous network activities. However, it slightly underperforms compared to the Decision Tree classifier, which had no misclassifications. Lastly, from the results presented above, the logistic Regression demonstrates solid performance, though it has a higher number of false positives (483) and false negatives (313) compared to the KNN and Decision Tree models. This indicates a slight tendency towards both under-detections of anomalies and over-detection of normal instances as anomalies. While still effective, it is less reliable than the

Decision Tree model for this application. While the Naive Bayes classifier is reasonably effective, its higher rates of false positives and false negatives suggest it might not be the best fit for real-time anomaly detection in a smart home system setting, where precision is paramount. Similarly, the KNN and Logistic Regression models, though robust, show some limitations that could lead to potential vulnerabilities or unnecessary alerts. These results underscore the importance of selecting a model that can not only achieves high accuracy but also maintains low rates of both false positives and false negatives, ensuring that the improved intrusion detection system is both secure and user-friendly. The Decision Tree classifier's ability to meet these criteria makes it the ideal choice for the Smart Home Intrusion Detection System. Its perfect accuracy and high cross-validation score underscore its reliability in detecting network anomalies.

Comparing the results of the implemented SHUACIDS to the previously provided results, with a focus on key performance metrics, specifically accuracy, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The breakdown of the results based on the classifiers we discussed is presented in table 2, while the comparison between the integrated machine learning model for the proposed smart home intrusion detection system against other previous works is presented in figure 3.

Table 1: Performance Metrics Comparison for SHUACIDS

Model	Cross-Validation Mean Score	Accuracy	True Positive	True Negative	False Positive	False Negative
Naive Bayes	0.907	0.907	8997	7000	1245	392
Decision Tree	0.996	1.00	9389	8245	0	0
K-Nearest Neighbors	0.991	0.994	9356	8168	77	33
Logistic Regression	0.954	0.955	9076	7762	483	313

Table 2: Accuracy Comparison Table

Model/Study	Accuracy (%)
The SHUACIDS (Decision Tree)	100.00
Naive Bayes	90.70
K-Nearest Neighbors	99. 40
Logistic Regression	95.50
[27]	99.99
[28]	99.60
[29]	94.94
[30]	99.90
[17]	99.90
[19]	99.99

From table 2, the Smart Home User Access Control Intrusion Detection System (SHMUACIDS) using the Decision Tree Classifier achieved a perfect accuracy of 1.0, which is the highest among all listed models. This performance surpasses that of traditional classifiers like Naive Bayes, K-Nearest Neighbours, and Logistic Regression, all of which showed lower accuracy rates. Compared to recent studies, the proposed integrated machine learning model remains competitive. While studies such as Xu et al. (2023) and Almotairi et al. (2024) also reported very high accuracy (99.9970% and 99.99%, respectively), the proposed model's performance suggests a potential for practical application without the complexity of ensemble methods. The system's interface is designed to provide a user-friendly experience, making it accessible to both tech-savvy and non-technical users. The inclusion of robust security features, such as multi-factor authentication and secure device management, further enhances the system's effectiveness in protecting smart home environments from potential intrusions there by protecting the integrity of data and devices in the entire Smart Home System.

4.4 System Testing and Result

System testing involves executing a program to identify errors. An effective test is one that has a high likelihood of uncovering errors. The application system was thoroughly tested, and the results were obtained during the process where a new user signs up using the signup page of the Intrusion Detection System. The table 3 illustrates the test cases and results for the Smart Home Access User Control Improved Intrusion Detection System (SHAUCIIDS).

5. CONCLUSION AND FUTURE WORKS

The findings of this study highlight the importance of a multi-layered approach to securing smart home IoT environments. The research demonstrated that integrating machine learning algorithms with advanced cryptographic techniques significantly improves the accuracy and efficiency of intrusion detection systems. The developed IDS not only detects intrusions in real time but also ensures the integrity and confidentiality of the data being transmitted within the network. A future work of fine-tuning the machine learning algorithms with more data to further improve the detection capability and finding solutions for dynamically changing threats driven by ever-growing IoT device adoption is suggested.

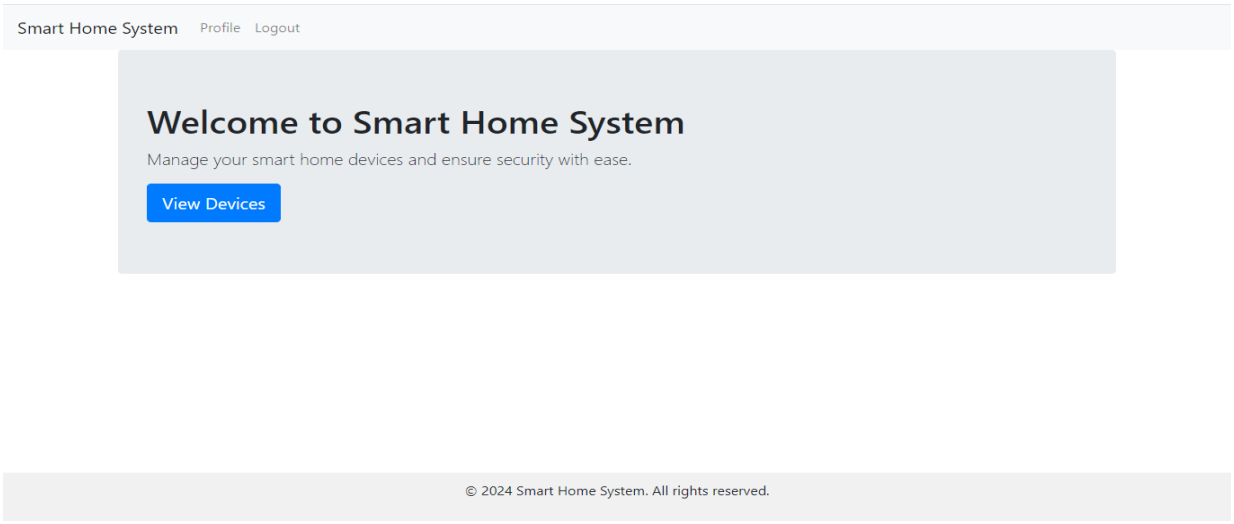

Figure 1: Main Page of the Smart Home IoT Intrusion Detection System.

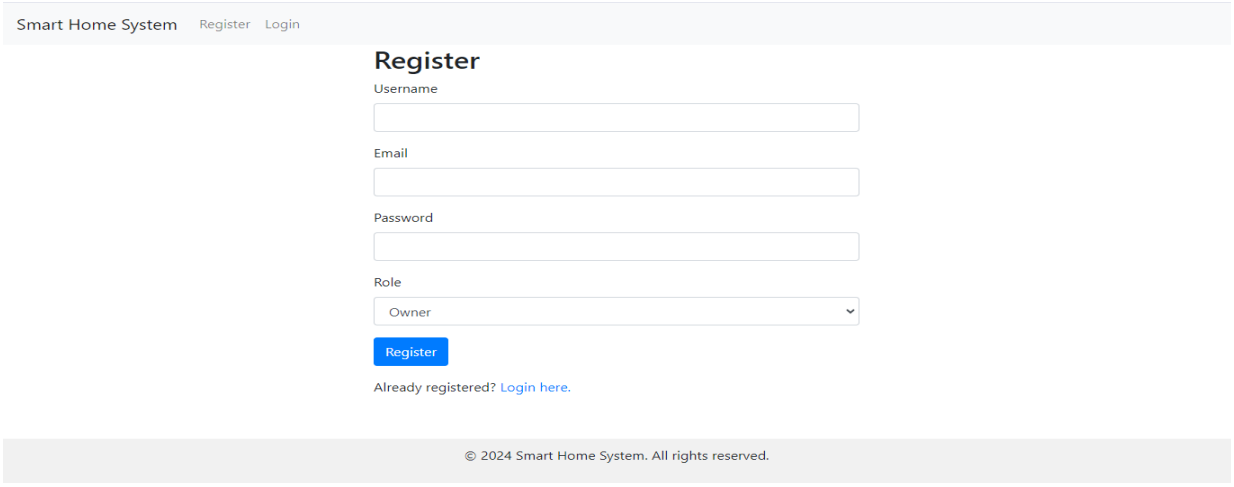

Figure 2: Registration Page

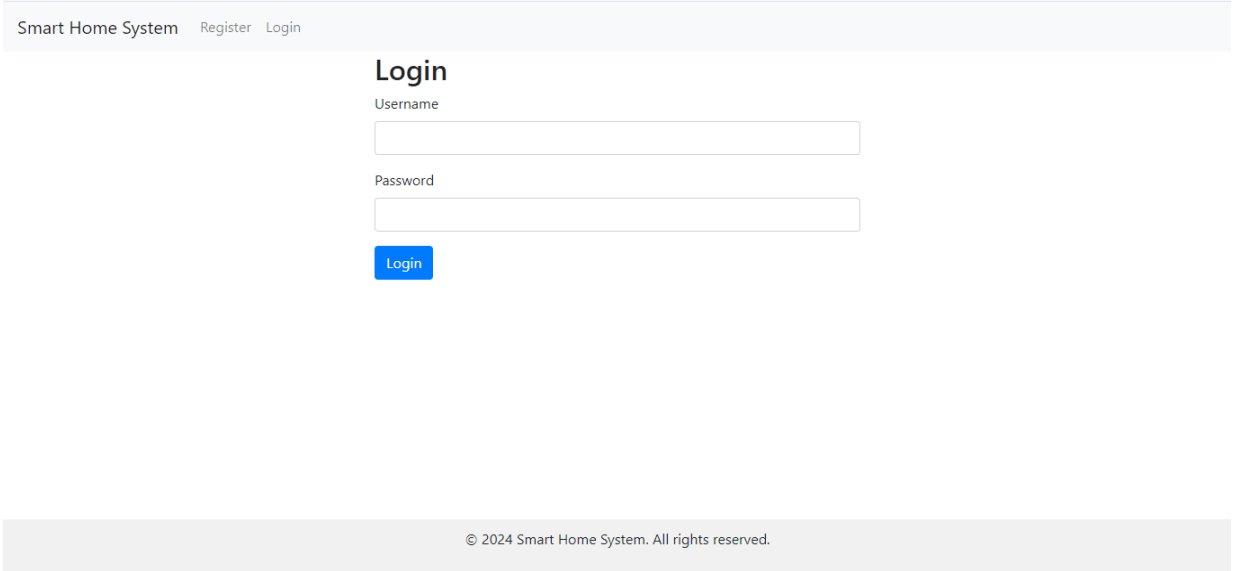

Figure 3: Login Page

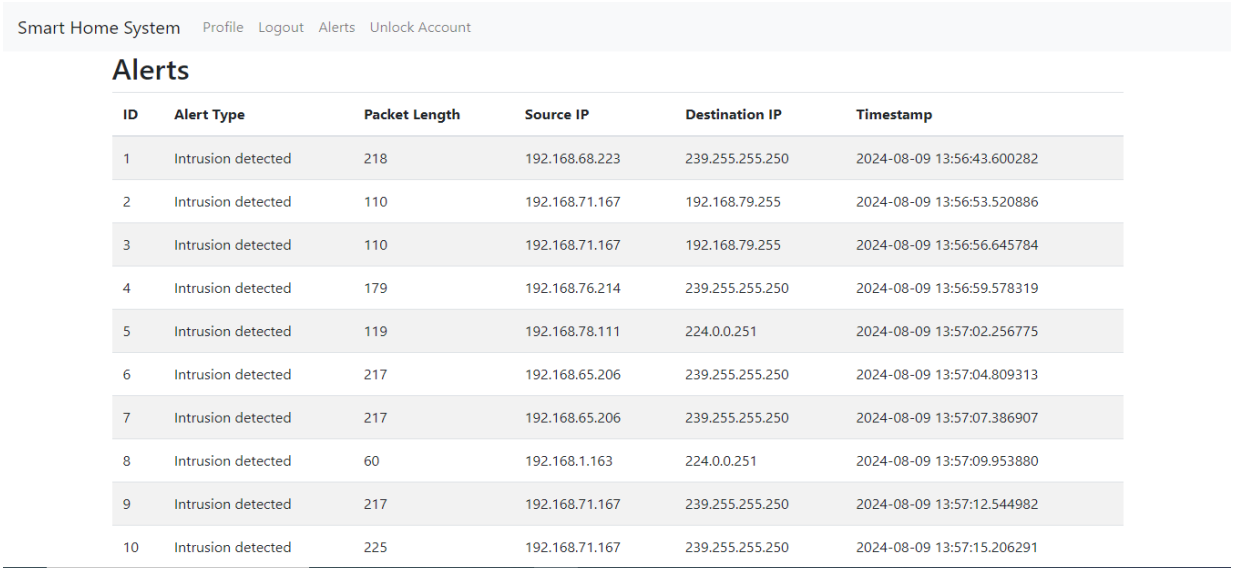

Figure 4: Alert Page

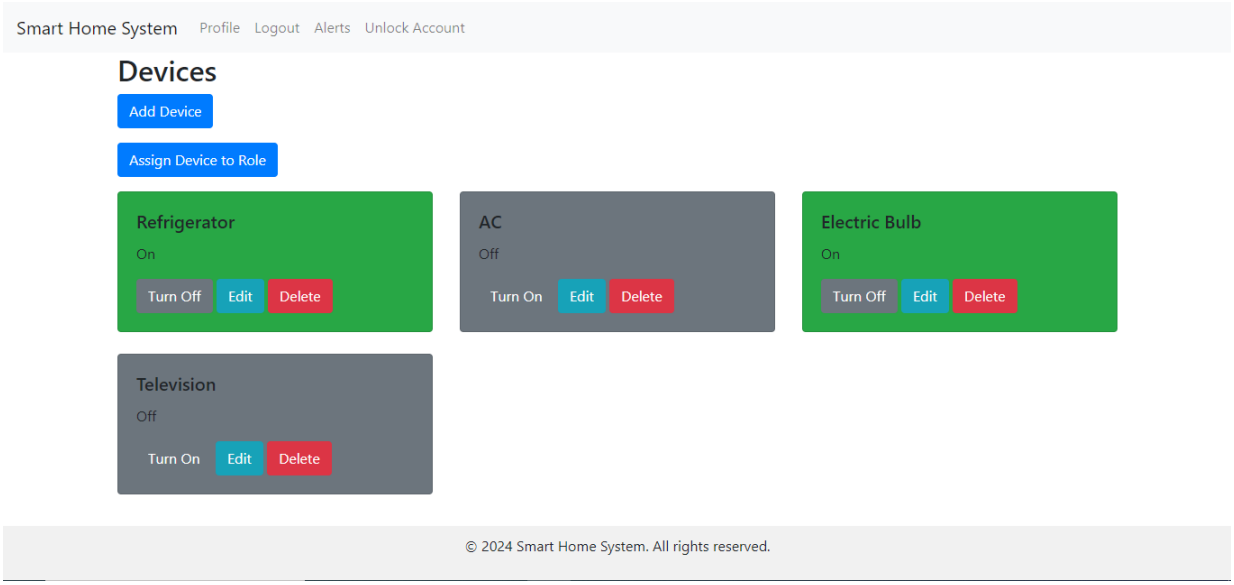

Figure 5: Devices Page

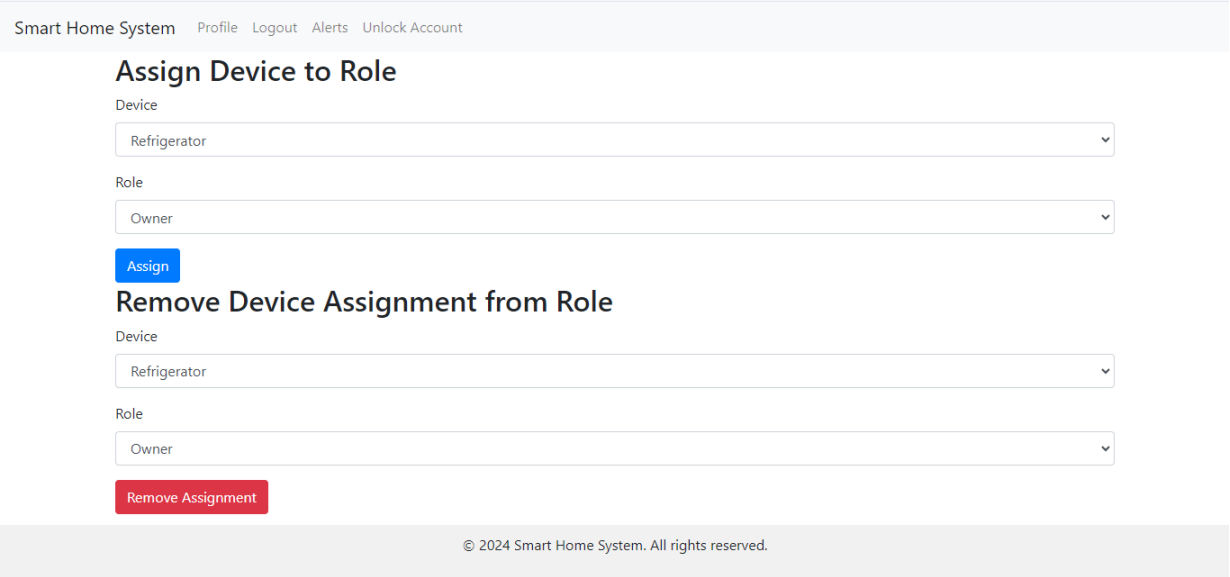

Figure 6: Assign and Remove Device Page

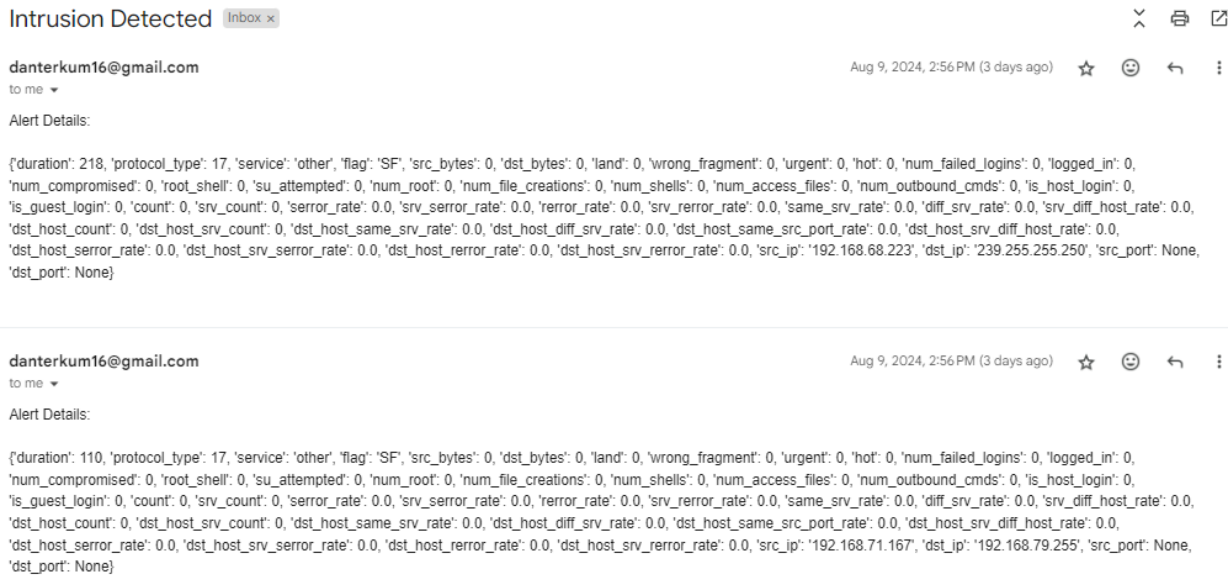

Figure 7: Email Alert Page

Table 3: Test cases and result for the Improved Intrusion Detection System

Test Case	Input	Expected Result	Test Result
User Registration	Valid user details (tor, tor@gmail.com, testing1234)	New user account created successfully and redirected to the login page	Pass
User Login	Valid username (tor) and password (testing1234)	User is authenticated and granted access to the system's dashboard	Pass
Device Registration	Valid device information (device ID, name, type)	Device is successfully registered and appears on the Devices Page	Pass
Edit Device	Modify device name and save changes	Device name is updated and displayed correctly on the Devices Page	Pass
Email Alert Configuration	Select specific alert types (Intrusion Detection).	Email alerts are configured successfully, and user receives the selected alerts	Pass
Password Unlock	Enter correct password and confirm	Device or system section is unlocked successfully	Pass
Assign Permissions to Device	Assign a permission to a registered device.	Permission is assigned to the selected device and appears under their profile	Pass
Remove Device from System	Select and remove a device	Device is removed from the system and no longer appears on the Devices Page	Pass
View Alerts	Access the Alerts Page	All recent system alerts are displayed with accurate details	Pass
Logout	User clicks the logout button	User is logged out and redirected to the login page	Pass
Invalid Login Attempt	Incorrect username or password	Error message is displayed, and login is denied	Pass
Reset Password	Request password reset with registered email	Password reset link is sent to the email, and user can reset their password	Pass
Unauthorized Device Access	Attempt to access a restricted device	Access is denied, and an error message is displayed	Pass
System Alert Generation	Trigger a known anomaly	Anomaly is detected, and an alert is generated and displayed on the Alerts Page	Pass

6. REFERENCES

[1] Ali, O., Ishak, M. K., Bhatti, M. K. L., Khan, I., & Kim, K. I. (2022). A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface. Sensors, 22(3), 995.

[2] Salem, R., Aidaros, B., & Belqasmi, F. (2024, April). Sustainable Industrial Agriculture in the UAE: Leveraging IoT and AI Technologies. In 2024 15th Annual Undergraduate Research Conference on Applied Computing (URC) (pp. 1-6). IEEE.

[3] Asghari, P., Rahmani, A. M., & Javadi, H. H. S. (2019). Internet of Things applications: A systematic review. Computer Networks, 148, 241-261.

[4] Mohiuddin, M., Hosseini, E., Tajpour, M., & Bahman-Zangi, B. (2024). Internet of Thing (IOT) Based Sensor Technologies and Smart Irrigation System: An Analysis of Critical Success Factors in Emerging Markets. The Journal of Developing Areas, 58(4), 167-188.

[5] Nag, A., Hassan, M. M., Das, A., Sinha, A., Chand, N., Kar, A., ... & Alkhayyat, A. (2024). Exploring the applications and security threats of Internet of Thing in the cloud computing paradigm: A comprehensive study on the cloud of things. Transactions on Emerging Telecommunications Technologies, 35(4), e4897.

[6] Elrawy, M. F., Awad, A. I., & Hamed, H. F. (2018). Intrusion detection systems for IoT-based smart environments: a survey. *Journal of Cloud Computing*, 7(1), 1-20.

[7] Jose, S. C. S., Singh, S. K., Rathore, S., & Park, J. H. (2020). A comprehensive analyses of intrusion detection system for IoT environment. *Journal of Information Processing Systems*, *16*(4), 975-990.

[8] Abdulganiyu, O. H., Tchakoucht, T. A., & Saheed, Y. K. (2024). Towards an efficient model for network intrusion detection system (IDS): systematic literature review. Wireless Networks, 30(1), 453-482.

[9] Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2018). Fog computing for Healthcare 4.0 environment: Opportunities and challenges. Computers & Electrical Engineering, 72, 1-13.

[10] Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. Cluster Computing, 26(6), 3753-3780.

[11] Restuccia, F., D'Oro, S., & Melodia, T. (2018). Securing the internet of things in the age of machine learning and software-defined networking. IEEE Internet of Things Journal, 5(6), 4829-4842.

[12] Giri, A., Dutta, S., Neogy, S., Dahal, K., & Pervez, Z. (2017, October). Internet of Things (IoT) a survey on architecture, enabling technologies, applications and challenges. In Proceedings of the 1st International Conference on Internet of Things and Machine Learning (pp. 1-12).

[13] Rejeb, A., Rejeb, K., Treiblmaier, H., Appolloni, A., Alghamdi, S., Alhasawi, Y., & Iranmanesh, M. (2023). The Internet of Things (IoT) in healthcare: Taking stock and moving forward. Internet of Things, 22, 100721.

[14] Ali, B., & Awad, A. I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. *sensors*, *18*(3), 817.

[15] Muaadh S. A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. Applied sciences, 11(18), 8383.

[16] Lirim, M., & Cihan D. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.

[17] Alsulami, R., Alqarni, B., Alshomrani, R., Mashat, F., & Gazdar, T. (2023). IoT Protocol-Enabled IDS based on

Machine Learning. Engineering Technology & Applied Science Research, 13(6), 12373–12380. https://doi.org/10.48084/etasr.6421

[18] Janardhana, D. R., Kumar, V. P., Lavanya, S. R., & Manu, A. P. (2021, November). Detecting security and privacy attacks in iot network using deep learning algorithms. In 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (pp. 35-40). IEEE.

[19] Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 12(1).

[20] Abdullah, S., Rehman, A., Qureshi, M. A., Ali, T., Irfan, M., Yasin, S., ... & Węgrzyn, M. (2021). Smart Fire Detection and Deterrent System for Human Savior by Using Internet of Things (IoT). Energies, 14(17), 5500.

[21] Sikder, A. K., Babun, L., Celik, Z. B., Acar, A., Aksu, H., McDaniel, P., ... & Uluagac, A. S. (2020, July). Kratos: Multi-user multi-device-aware access control system for the smart home. In Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (pp. 1-12).

[22] Ravidas, S., Lekidis, A., Paci, F., & Zannone, N. (2019). Access control in Internet-of-Things: A survey. *Journal of Network and Computer Applications*, *144*, 79-101.

[23] Li, H., Han, D., & Chang, C. C. (2023). DAC4SH: A Novel Data Access Control Scheme for Smart Home Using Smart Contracts. IEEE Sensors Journal, 23(6), 6178-6191.

[24] Mohammad, Z. N., Farha, F., Abuassba, A. O., Yang, S., & Zhou, F. (2021). Access control and authorization in smart homes: A survey. Tsinghua Science and Technology, 26(6), 906-917.

[25] Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S., & Fang, B. (2020). A survey on access control in the age of internet of things. *IEEE Internet of Things Journal*, *7*(6), 4682-4696.

[26] El Sibai, R., Gemayel, N., Bou Abdo, J., & Demerjian, J. (2020). A survey on access control mechanisms for cloud computing. Transactions on Emerging Telecommunications Technologies, 31(2), e3720.

[27] Xu, B., Sun, L., Mao, X., Ding, R., & Liu, C. (2023). IoT Intrusion Detection System based on Machine learning. Electronics, 12(20), 4289. https://doi.org/10.3390 /electronics12204289.

[28] Azizjon Ikromjon O ‘G‘Li, M. (2024). Iot Network Intrusion Detection System Using Machine Learning Techniques. Kokand University Herald. https://doi.org/10.54613/ku.v11i11.972

[29] Verma, A., & Ranga, V. (2019). Machine learning based intrusion detection systems for IoT applications. Wireless Personal Communications, 111(4), 2287–2310. https://doi.org/10.1007/s11277-019-06986-8

[30] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, *61*(12), 9395–9409. https://doi.org/10.1016/j.aej.2022.02.063