I.J. Wireless and Microwave Technologies, 2026, 4, 152-177
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.04.10

Multiclass Cyber Attack Classification in Smart Home IoT Networks Using Ensemble Machine Learning with the ML-EdgeIIoT Dataset

Abhay Kumar Ray*
Department of Computer Applications. SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar – 201204, Ghaziabad, Uttar Pradesh, India
E-mail: ar2587@srmist.edu.in
ORCID iD: https://orcid.org/0009-0005-6990-7238
*Corresponding Author

Rupak Sharma
Department of Computer Applications. SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar – 201204, Ghaziabad, Uttar Pradesh, India
E-mail: rupaks@srmist.edu.in
ORCID iD: https://orcid.org/0000-0003-3266-4656

Sunil Kumar Pandey
Department of IT, Institute of Technology & Science, Mohan Nagar, Ghaziabad, 201007, India
E-mail: Sunilpandey@its.edu.in
ORCID iD: https://orcid.org/0000-0002-8085-2275

Received: June 6, 2026; Revised: June 24, 2026; Accepted: July 8, 2026; Published: August 8, 2026

Abstract: With the rapid adoption of smart home solutions and related technologies, edge computing has emerged as a key enabler by offering low-latency data processing, increased efficiency and improved scalability. However, this integration in IoT systems introduces complex security challenges in smart home edge environments, increasingly susceptible to cyber threats such as denial-of-service (DoS), malware injection, passive surveillance, and unauthorized access. This paper investigates intelligent intrusion detection and attack classification strategies specifically designed for smart home edge systems. Using the comprehensive ML-EdgeIIoT dataset, this study designs and evaluates a machine learning-based intrusion detection framework for multiclass classification of eight categories of IoT network attacks, namely Backdoor, MITM, DDoS, Ransomware, Password Attack, SQL Injection, Prob-attacks, and Normal traffic while minimizing false positives and false negatives. The framework incorporates data cleaning, correlation- and feature importance-based feature selection, hyperparameter optimization using gridsearchCV, model training, and ensemble learning. A set of machine learning models comprising Artificial Neural Network, Balanced Random Forest, K-Nearest Neighbours, Random Forest, and Logistic Regression was implemented and comparatively evaluated. Two ensemble techniques were subsequently developed using the three best-performing classifiers: (1) a stacking ensemble with Logistic Regression as the meta-learner and (2) a Top-3 majority voting ensemble. Model performance was evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. Robustness and generalization of the individual machine learning models were assessed through stratified 10-fold cross-validation for the three best-performing classifiers. The Top-3 voting ensemble subsequently achieved the highest performance on the independent test set, with accuracy of 99.24%, average precision of 98.75%, recall of 99.00%, and an F1-score of 99.00% for all attack classes, while reducing misclassification compared with individual classifiers. The findings of this study significantly enhance the understanding of smart home edge computing security, which will pave the way for more robust and intelligent threat detection frameworks.

Index Terms: Smart Home Security, Edge Computing, Intrusion Detection System (IDS), IoT/IIoT Security, Machine Learning, Multiclass Classification, Cyberattack Detection, Ensemble Learning

1. Introduction

A smart home refers to a modern house that can be remotely monitored and controlled by the homeowner and is connected with various devices such as a heating system, lighting, entertainment units and security system. The central device or an application on a smartphone makes it easy to automate and personalize the technology for more comfort, security as well as energy savings. Popular smart home technologies are IoT based, web enabled and voice-controlled devices like Alexa or Google Home, smart locks, smart refrigerators and ovens, smart thermostats, and other appliances. Homeowners can effectively manage their living conditions by skillfully merging these technologies to create a customized atmosphere.

Integration of intelligent technologies into household appliances has improved living standards through the adoption of smart home technologies [1]. Smart thermostats, security cameras voice-assisted devices and IoT enabled appliances are some of the examples that help improve comfort, protection and energy efficiency. Smart homes use IoT technology to permit devices integration, thereby enabling real-time personalization and adaptation of the systems to create a better living condition. Cloud computing remains essential for smart homes; however, issues of latency, privacy, and bandwidth usage have pushed forward the need for a more efficient alternative known as edge computing.

Edge computing processes data or carries out computation at the periphery of a network in contrast to the traditional method that leverages one or more cloud servers. This paradigm shift augments instantaneous communication responsiveness, guarantees privacy and reduces demand placed on bandwidth with the ever-increasing smart devices. For example, an edge AI enabled smart surveillance camera can identify, capture, and process potential threats at real-time without waiting for a signal from the cloud. With complex smart home architectures, edge computing profoundly takes personal smart homes one step closer towards greater privacy and security [2].

1.1. Importance of security in smart home edge environments [3]

Smart home not just as a collection of gadgets, but as a responsive, almost intuitive extension of living space. That instant adjustment of the thermostat when someone walks in, the lights fading as home owner settle on the couch, the camera alerting the security system in a fraction of a second when a package arrives – that seamless magic often happens due to edge computing, where data is processed there in local switch or network device, instead of a far-off cloud server. This speed and local awareness are fantastic, but they place a massive responsibility squarely on the security system. The edge layer serves as the primary security boundary protecting interconnected smart home devices. Without robust security guarding that local processing hub, the very features designed to make the homeowner’s life easier and safer-controlling door locks, monitoring cameras, managing climate – become potential backdoors. A breach here isn't just about stolen passwords; it could mean someone silently watching your private moments through a compromised camera, unlocking your front door remotely, or even tampering with critical systems like medical monitors or fire alarms. In an edge-powered smart home, strong security isn't an optional add-on; it's the essential foundation that lets you truly relax and trust the technology woven into your sanctuary.

The unique nature of edge computing introduces distinct risks that demand tailored security vigilance. Because data is processed locally across various devices (smart switches, smart speaker, security cameras), the traditional "castle walls" approach of just securing the internet gateway isn't enough anymore. It's required to secure not just the front gate or entry points of communication network, but every window, back doors, and every individual entity, because each section might be a potential weak spot for attacks. Hackers might target a less secure smart home spot or network device to hop across the network to the hub controlling the locks. It is also possible that a simple malware could infect the local processing unit on a camera, manipulating its feed before it even sends anything out, making theft or intrusion invisible. In a home using edge computing, a compromised device isn't just offline, it can become an active threat within home's private space, potentially disrupting critical local functions or spying undetected. A sophisticated layered security system requires strong encryption for local device communication (but it is not feasible in IoT system due to limited computation power), regular firmware updates for all edge devices, strong authentication and authorization, and intrusion detection systems. also It is not just about protecting data but safeguarding the physical security of the smart home, the privacy of family within its walls, and the fundamental peace of mind that makes the home truly feel like a safe haven.

1.2. Overview of intrusion detection and attack classification

The application of machine learning (ML) for intrusion detection and attack classification [4] is crucial in improving cybersecurity in smart home's edge computing system. Widely adopted IoT devices for smart homes bring with them relatively low computational power and poor security features, increasing the potential attack surface greatly in comparison to older generations. Traditional rule-based intrusion detection systems face difficulties in dealing with the ever-changing and diverse environment of smart home networks. ML based approaches analyze device behavior and network traffic patterns to identify and classify cyber threats in a driven and adaptive manner, providing the much needed flexibility through data-based methods. Models located at the edge can also perform real-time detection with less latency, reduced bandwidth, and greater privacy due to lower data transfer to centralized servers. Supervised learning algorithms make it possible to classify pre-defined attacks, while novel and zero-day threats are identified using unsupervised and deep learning algorithms. While facing problems such as limited computing resources, data imbalance, and vulnerability

to adversarial inputs, ML-based intrusion detection systems still offer scalable solutions.

1.3. Research objectives and contributions

This research aims to develop and evaluate a robust machine learning framework using multiple machine learning algorithms for attack detection and multi-class classification of cyber-attacks within IoT/IIoT edge computing environments, utilizing the comprehensive ML-EdgeIIoT dataset. Moving beyond simple binary detection (attack vs. normal), the objective is to accurately identify and distinguish between specific types of malicious network activities present in the dataset, including Backdoor attacks, Man-in-the-Middle (MITM) attacks, Distributed Denial-of-Service (DDoS), Ransomware, Password attacks, SQL injection/code attack, and Prob-attacks [5]. The core challenge is to create a robust model capable of precisely categorizing network traffic records into one of these distinct attack classes or recognizing them as benign ("Normal") activity.

The primary objectives of this research are:

a) To preprocess the ML-EdgeIIoT dataset through data cleaning, normalization, and feature engineering for effective multiclass learning.
b) To implement and compare multiple supervised machine learning classifiers (multi-class variants of classifiers like KNN, Random forest model, Logistic regression, Balanced RFM, SVM [6], deep learning like ANN [7]) for multiclass cyberattack detection.
c) To develop heterogeneous ensemble learning models using stacking and Top-3 majority voting to improve attack classification performance.
d) To evaluate model performance using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix, and stratified 10-fold cross-validation.
e) An edge-oriented inference optimization strategy is proposed that combines correlation and global feature importance-based feature reduction, model optimization, and Top-3 ensemble classifier to reduce inference latency and computational overhead while maintaining high multiclass detection performance.

The major contributions of this work are summarized as follows:

a) A hybrid feature engineering strategy combining correlation analysis with global feature importance is applied to reduce feature redundancy.
b) Two ensemble learning mechanisms—stacking and Top-3 majority voting—are developed and evaluated.
c) Multiple machine learning classifiers are assessed under identical experimental settings to provide a comprehensive comparison.
d) Extensive multiclass performance evaluation is conducted using multiple metrics and cross-validation to assess robustness and generalization of the proposed ensemble learning mechanisms.

2. Smart Home Edge Computing: Architecture and Security Challenges

The edge computing system [8] offers a paradigm shift in context of home automation by decentralizing data processing and moving it closer to the source (like different sensor nodes and smart devices within the home environment itself) rather than uploading the data and relying exclusively on remote cloud based servers for processing. This architectural approach enables the smart home environment for real-time data analysis and decision-making at the edge, which reducing latency, improving system responsiveness and reducing the bandwidth use for applications such as environmental control, security monitoring, and automation.

Additionally, processing data locally minimizes bandwidth consumption and energy use, which are critical considerations in the resource-constrained context of smart homes. The integration of edge computing hardware in smart homes, including local processors and accelerators or co-processors, which facilitate efficient handling of the large amounts of interconnected smart devices.

In smart homes architecture [9], the distributed, heterogeneous nature of edge devices makes the system vulnerable to cyber-attacks. This is in addition to posing unique safety concerns. Many edge devices have low computational power which makes it impossible to put in place strong security safeguards such as complex encryption and intrusion detection systems. Moreover, inadequate uniform policies and regulatory authentication standards are caused by the wide range of devices and manufacturers. These problems pose the need to set up strong security measures using trusted execution environments, secure boot systems, and frequent software updates in order to protect sensitive information and keep the system safe. Smart home solutions relying on edge computing are still majorly focused on these two factors; privacy and anti-intrusive systems. The reliability of these solutions is greatly reduced by failing to install or add smart security frameworks on a logical group of smart devices.

2.1. Role of edge devices in smart home ecosystems

Edge devices make your smart home feel much quicker and more responsive by handling data and making decisions

right where things happen—in your home. Instead of sending every command or piece of information out to the cloud and waiting for a response, these devices can process things locally. For an instance, if the smart camera of main or front gate spots movement, it can instantly trigger an alert or turn on the spot light on the moving object without any delay from internet lag. This means actions like adjusting the thermostat's temperature, responding to voice commands, or activating security features happen almost without any delay. By keeping most of the processing locally, edge devices also cut down the amount of data that needs to travel over the internet connection and reduce bandwidth usage, which helps avoid slowdowns and keeps communication network running smoothly. and also sensitive information doesn't always have to leave smart home network or local servers, so the data privacy gets an extra layer of protection. All in all, edge devices help the smart home react faster, work more reliably, keep personal data safer and overall communication cost.

2.2. Examples of real-world security breaches in smart home environments

Security breaches in smart home environments have shown just how important it is to take device protection seriously. Take, for example, the incidents of Amazon's Ring cameras in year 2019 [10], where hackers were able to break into people's live video feeds simply because users hadn't changed default passwords or old passwords or were using weak ones. In some cases, strangers not only watched family's videos but also listen their conversation's through the camera's microphones, causing fear and distress. These situations really drove home the need for strong, unique passwords and extra security steps for strong authentication. Another real-life example happened in year 2019, Wyze [11] , a smart home company whose database was accidentally left exposed online for approximately 23 days due to some human error. This incident revealed email addresses and Wi-Fi details of millions of customers using the services of Wyze, putting their privacy and data security at risk and potentially making it easier for hackers to target consumer's smart home networks. An another incidence of which primarily seems like an attack with same company, in early 2024, Wyze [12], faced a troubling security incident when a technical glitch during a system recovery caused some users to see images and video clips from other customers' cameras in their own app feeds. This mix-up of video and audio streams happened due to internal error rather than an outside attack, which affected approximately 13,000 users and exposed private moments. Wyze responded by notifying those impacted, strong access controls, and promised for further improvements to its security processes. These incidents served as a reminder for consumers that even trusted smart home companies can make such mistakes that it put user privacy at risk. so such IT services required to focus on rigorous AI enabled safeguards and transparent communication in the world of connected home technology. These architectural characteristics and security limitations motivate the need to understand the major categories of cyberattacks affecting smart-home edge environments, which are discussed in the next section.

3. Major classification of Attacks in Smart Home Edge Computing

Network-based attacks [13]: Network-based attacks pose a risk to smart home edge computing because they disrupt the communication or interfacing of devices with each other and with external systems. For example, during a Denial-of-Service (DoS) attack, the attacker could overwhelm the network with such high volumes of traffic that smart devices (for instance, security cameras or smart locks that are part of the Internet of Things) become sluggish or non-responsive. This could interfere with daily routines works or endanger safety and security related work. Another common threat is Man-in-the-Middle (MitM) attacks, where a hacker intercepts data streams between devices without the parties knowing, allowing them to eavesdrop or tamper with private information and commands. Such smart home devices, everyday appliances that enhance our standard of living, may have weaker security mechanisms due to limited processing power, which makes them more susceptible to the mentioned attacks. As home automation continues to grow, defending these networks and ensuring seamless operation of smart devices is becoming more critical.

Device-based attacks [14]**:** Smart home edge computing security risks related to firmware changes and malware injections, which threaten the devices from malfunctioning, therefore this type of attack posing a risk to abnormal use of smart home devices. By changing their firmware or injecting malicious code, cyber attackers may seize control of smart locks, security cameras, or even kitchen appliances. Once these devices are compromised, they may be able to silently grant access to rooms, spy on the residents, or devices become part of a botnet that launches distributed attacks on the Internet. The rapid exploitation of vulnerabilities in devices such as those infected by the Mirai Malware demonstrates the speed at which these threats are executed—especially when devices are set with factory-default usernames and passwords, infrequent password changes, lack of routine security checks, and no routine security maintenance. These threats underscore the importance of robust security measures to defend against the vulnerabilities posed by smart home devices and require diligent use of sound security practices, strong passwords, regular software updates, and technology shields on devices.

Data-related attacks [15]: Data-related attacks in smart home edge computing can have a strong impact on user's privacy and data security, this type of attack enable hacker for unauthorized access to sensitive information or large-scale breaches. Such incidents frequently arise when security settings are neglected or passwords are not strong enough, giving chances to attackers or an opportunity to access personal data or even manipulate devices remotely. Examples include unauthorized viewing of camera feeds or access to network credentials. These challenges highlight the critical need for

strong encryption, consistent security updates, and careful management of authentication to protect the vast amounts of private data generated by smart home technologies.

User-centric attacks [15]: User-centric attacks in smart home edge computing, such as phishing and social engineering take advantage of homeowner's innocence rather than exploiting technical vulnerabilities. In cases of phishing scams target user receives emails or messages from legitimate manufacturers and service providers with the intent of harvesting sensitive information are commonplace. Through these unknowing traps, users may end up sharing vital passwords or enabling unauthorized access to their smart homes where attackers can remotely operate devices or private information can be accessed. Because these attacks rely on deceiving people rather than technical exploits, they are likely to be much harder to defend against with technology. This highlights the need for good regular user training, strong authentication mechanisms, and proactive primary instructions by the manufacturers on how to identify and mitigate exposure to such threats. Table 1 describes the crisp details about the different aspects of attack types.

Table 1. Attack Summary in Smart Home Edge Computing

Attack Type	Attack's Working	Typical Examples	Effect on Smart Home Systems	Potential Outcomes
Network-based	Targets the data/command under transmission between devices and networks	DoS attacks, MitM	Interrupts device communication, access or alter or misuse of sensitive data	Devices may go offline, privacy may be compromised
Device-based	Exploits the flaws or default settings in the device's hardware or software	Changing device's firmware, installation of malicious software/ code/ backdoor attacks	Allows the attackers to take over or disable the devices under attack, or use them in larger attacks using bonnets	Loss of device control, risk of persistent threats
Data-related	Focuses on accessing , leaking or capturing personal and operational data	Unauthorized access, data leaks, credentials theft, Ransomware attack	Jeopardizes user privacy and the integrity of stored data	Exposure of personal information or manipulation on stored data
User-centric	Relies on tricking users rather than breaking technology	Phishing emails, social engineering scams	Users may unknowingly give the access of devices or transfers sensitive information	Unauthorized access, misuse of devices, privacy loss

The Table 1 highlights each broad category of attacks in smart home edge computing comes with, its own risks and consequences. Network-based attacks can disrupt how devices communicate to each other using some protocols, while device-based attacks may give hackers direct control over home gadgets or smart devices. Data-related attacks put personal information at risk, and user-centric attacks exploit human behavior to bypass technical defenses. Therefore, identification and classification of these attacks are essential for developing effective security strategies in modern smart home environments. Understanding these attack categories establishes the basis for selecting appropriate intrusion detection mechanisms, which are reviewed in the following section.

4. Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) [16] play a crucial role in keeping smart homes safe, especially as more of smart home's devices—from lights and locks to cameras and thermostats—become connected to the internet. These systems work right at the edge of the home network, which means they can spot and react to suspicious activity almost instantly, without having to send all the data to a distant cloud server. smart algorithms that learn the normal behavior and communication pattern of each device, IDS can quickly take decision for anything unusual or usual, for instance hacker is trying to break in or a device acting strangely in smart home network. This local, real-time protection not only helps stop threats before they cause harm but also keeps your personal data more private. As smart homes become more common, having IDS in place is becoming essential for giving homeowners peace of mind and keeping their digital lives secure. Different types of IDS are as discussed in next sections.

Traditional IDS Approaches: The traditional Intrusion Detection Systems (IDS) [17] have always been considered essential for the protection of a communication network as these systems are based on two very basic methods: signature based and anomaly detection. Signature-based IDS functions similarly to an anti-virus application by checking the network traffic for signals which correspond to specific threats in the form of patterns. This type of detection is useful in pointing out attacks that are common, however, unfamiliar, new, or evolving threats are often missed unless the attack database is constantly modified. Anomaly based IDS works in a different fashion, it monitors network activity to understand what normal activity constitutes and subsequently flags all activities exceeding that benchmark as suspicious. Although this method may help in identifying new attack methods, it often mistakes harmless changes for security threats leading to unnecessary alarms (false positive activity). Today, many systems combine both approaches in an attempt to capture a greater variety of attacks. These older techniques, along with emerging systems, are under immense pressure from modern methods of cybercrime.

Modern IDS Techniques: Modern intrusion detection systems (IDS)[18] are now smarter and more adaptable to the newer requirements of attached devices. Rather than relying on fixed rules or known attack signatures, such systems for IoT use more sophisticated approaches such as machine learning with SVM, Random Forest Model, Neural Networks, etc. for self-training on network activity to uncover unusual behaviours which could suggest a threat. Many IDS solutions

now integrate both traditional and intelligent methods to not only identify familiar attacks but also catch new and unexpected ones. Additionally, many IoT smart homes utilize edge computing, allowing nearby servers to handle intrusive computations while preserving resources on the actual devices. All these advancements increase the efficiency of IDS in safeguarding the ever-increasing smart devices in homes and workplaces, providing more prompt and precise threat detection. The following section presents the three primary types of modern IDS.

Behavior-based anomaly detection: It takes a more personalized approach to security by getting to know the usual habits and routines of users and devices on a network. Once the system understands what's typical activities performed by smart home owner, it keeps watch for anything out of the ordinary—like family member's movement from an unusual location or a smart home device suddenly sending large amounts of data. This method is especially important for spotting new or sophisticated attacks that do not match any known patterns. it advances in behavioural analytics are making these systems smarter and more accurate, helping security system to catch threats in early stages without overwhelming them with false positives.

Machine learning and AI-based intrusion detection: The implementation of AI and Machine learning algorithms in attack detection in ids become the game changer of cybersecurity. These systems can monitor network activities and recognize what are the usual activities and what might be an indicator of a problem or attack. This Modern detection system is very adaptive rather than static as with older or traditional system. These systems are able to train the models from previous data to identify previously encountered attacks types and newer one, and it maintains a greater overall system security. This adaptive capability of AI reduces not just false alarms, but it also enhances an organization's efficiency because real problems can be investigated instead of harmless anomalies. AI and Machine Learning are effective in proactively defending networks and devices from attacks.

Federated learning and decentralized security models: Federated learning-based intrusion detection systems (IDS) [18] offer a fresh and privacy-friendly way to protect networks of smart devices. Instead of sending all your data to one place for analysis, each device learns from its own activity and only shares the results or insights, not the actual raw data, with a central system. It means that personal data is stored on local devices, but the complete security system gets smarter because it combines insights and patterns from many different sources at different sites of deployment. As a result, federated learning-based IDS can identify new and unusual threats more quickly and accurately, and increase degree of user's data privacy and reducing the risk of data leaks. This approach is especially useful for the growing world of IoT environment, where keeping data secure and data privacy is just as important as staying protected from cyberattacks. Table 2 details the comparison between traditional and modern IDS applications

Table 2. Comparison of IDS Approaches and Their Effectiveness

Feature/Aspect	Traditional IDS	Modern IDS	Effectiveness Overview
Threats detection process	Relies on fixed rules and known attack signatures	Uses smart algorithms and learns from previous and newer data	Modern IDS can spot both old and new threats, while traditional IDS mostly catches what it already knows or meeting the criteria of fixed rules
Types of Attacks Detected	Mostly identify familiar, well-documented attacks	Can identify unknown, evolving, and the complex attacks	Modern IDS offers better protection, especially against new or tricky threats.
False Alarms	Usually low if using signature databases, but can be high with basic anomaly detection	Improved accuracy due smarter anomaly analysis	Modern IDS generally generates fewer false alarms, and making life easier for security teams members.
How They Respond	Typically just send alerts to the users or administrator	Can automate mitigation plan and provide more details to security teams	Modern IDS can take action quickly and provides more helpful insights.
Ability to Adapt	Needs regular manually updates or provide settings for automatic updates to stay current	Learns automatically as threats change. So highly adaptability.	Modern IDS keeps up with new threats without regular manual work.
Resource Needs	Works well on basic hardware and low resource demand	Needs more computing power, but edge/cloud helps	Modern IDS may need more resources, but can handle bigger and busier network.
Scalability	Limited scalability with large or fast-growing networks	Designed to handle large number of devices and fast growing network	Modern IDS is better suited for connected systems like IoT and IIOT systems
Context Awareness	Limited to basic data like logs or network packets	Pulls in communication it's related data from different sources for better analysis	Modern IDS are more context aware and making it easier to understand alerts.
Where can this IDS be used	Mostly installed on local networks or single devices	Can run on the cloud systems, at the edge, or across the hybrid setups	Modern IDS offers more flexible deployment options for different needs.

Modern IDS solutions are quite different from traditional systems in many aspects. These IDS use artificial intelligence and machine learning for attack identification, classification and mitigation, a wider range of threats, adapt quickly to new attack methods, and provide more meaningful alerts. While these security systems may need more computing power, their ability to scale and their smarter, faster responses make them a better fit for today's complex and connected digital world. Based on the capabilities and limitations of existing IDS approaches, the following section reviews recent literature to identify current research gaps that motivate the proposed multiclass machine learning

framework.

5. Related Work

The rapid growth of smart home environments, powered by edge computing, has spurred significant research into intrusion detection and attack classification. Recent studies (2022–2025) have explored a variety of machine learning (ML), deep learning (DL), and ensemble-based approaches, each with unique strengths and limitations in terms of accuracy, real-time performance, and adaptability to resource-constrained devices.

M. A. Ferrag et al. [19] Introduced the Edge-IIoT dataset, a complete and realistic cybersecurity dataset that collected from the IoT and Industrial IoT (IIoT) systems, with the aim to solve the problem of both centralized and federated learning environments for attack identification and classification. The dataset can be seen as a sensor of all kinds of cyber threats that attack the network, including Denial of Service (DoS), Man-in-the-Middle (MitM), and reconnaissance attacks, script attack, ransomware attack etc , thus, providing a machine learning-based intrusion detection system with a rich foundation of development and testing. The authors of the work conducted several machine learning experiments that consisted of choosing appropriate models for the task of attack detection and classification. The Decision Tree (DT) approach exhibited the lowest result of 67.11% accuracy, while the RF, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) classifiers reached 80.83%, 77.61%, and 79.18% accuracy respectively. The Artificial Neural Network (ANN) classifier led to high score of 96.01% when performing multi-class classification of six different attack types, thus, still better than RF (82.90%), SVM (85.62%), KNN (83.39%), and DT (77.90%). These numbers mark this dataset as an excellent resource for developing intrusion detection systems and attack classification, especially if the user leverages advanced models like ANN in an edge computing scenario.

Gueriani et al. [20] explored the application of attention-enhanced LSTM-CNN architecture for multiclass intrusion detection in Industrial Internet of Things (IIoT) networks. Using the Edge-IIoTset dataset, the authors proposed a hybrid deep learning model capable of learning diverse patterns from network traffic. Their experimental evaluation reported 99.32% accuracy, 99.31% precision, 99.30% recall, and 99.30% F1-score, together with a remarkably low 0.002% false positive rate. The study showed that incorporating an attention mechanism enabled the model to emphasize the most informative traffic features, resulting in more reliable multiclass attack identification than conventional CNN or LSTM architectures.

Sohail et al. [21] proposed and implemented an optimally configured Artificial Neural Network (ANN) for identifying attacks in the smart home domain. Their work produced exceptional results in the range of 99.9% accuracy for the binary classification problem, 99.7% accuracy for the category-level detection task, and 97.7% accuracy for the subcategory-level classification challenge. They have all the time focused on explainability and have fine-tuned hyperparameters which lead high to precision as well as turn the system into a transparent one. Thus, they have solved one of the crucial problems of ML-based smart home security systems.

Aljabri et al. [22] have developed an IoT intrusion detection system referring to only the use of supervised machine learning techniques for the detection of Flood and Brute-Force attacks detection. They have executed their Gradient Boosting classifier with 95.9% of accuracy using six features and 95.3% of accuracy with only three features. This study is especially significant for smart home devices at the edge where computational resources and memory are limited because it has been proved to be capable of still ensuring high detection accuracy when the feature input is minimal.

Farea and Kucuk[23] proposed IDPS (Hybrid Intrusion Detection and Prevention System) machine learning-based intrusion detection system for edge-based Industrial IoT (IIoT) applications, which is highly applicable to smart home environments. Using KoU-6LoWPAN-IoT and Edge IIoT datasets, their system achieved binary classification accuracy up to 99.9% and multi-class classification accuracy of 95.65% accuracy using ANN classifier. This paper uses the classifiers like Decision Trees highlights the system's feasibility for edge deployment, which offers a favorable trade-off between computational efficiency and classification performance.

Aldaej et al. [24] introduced a deep learning-based intrusion detection system for IoT deployed in edge–cloud environments. The approach begins by partitioning the large BoT-IoT time-series dataset according to attack types and then applying feature selection to aggressively reduce dimensionality—cutting the number of attributes by around 85%—without sacrificing model performance. This paper trained hybrid models like Random Forest, SVM, RNN and Bi-LSTM layers on this cleaned dataset. Models of this study delivered outstanding results up to: 99.56% accuracy, 99.45% precision, 98.25% recall, and a 99.12% F1-score for RNN classifier. This demonstrates that intelligent feature selection not only helps meet edge-device resource constraints but also maintains, or even improves, high detection performance in real-time IoT security applications.

Ismail et al. [25] examined the effectiveness of lightweight machine learning algorithms across three widely used intrusion detection benchmarks, namely Edge-IIoTset, TON_IoT, and WUSTL-IIoT-2021. To reduce feature redundancy, the authors first applied Mutual Information (MI)-based feature selection before evaluating classifiers such as Decision Tree, Random Forest, Bagging, Stacking, and LightGBM. Instead of limiting the analysis to classification accuracy, the study considered precision, recall, Micro-F1 score, model complexity, training time, inference time, and cross-dataset generalization. The results revealed that Stacking and Bagging delivered consistently stronger multiclass detection performance than individual classifiers, while LightGBM provided an attractive compromise between prediction

capability and computational efficiency, making it well suited for resource-constrained IIoT deployments.

Rieger et al. [26] introduced ARGUS, an unsupervised DNN model that is contextually aware which detects stealthy infiltration by monitoring the states of IoT devices and the environmental context. The ARGUS achieved a 99.64% F1-score and identified contextual anomalies (e.g., misuse of smart-lock) besides it.

Tanveer et al. [27] proposed LightEnsemble-Guard, a lightweight ensemble learning framework for intrusion detection in resource-constrained IoT environments. Their approach combines LightGBM, XGBoost, and Extra Trees classifiers through a majority voting mechanism, together with embedded feature selection to lower computational overhead without sacrificing detection capability. The framework was evaluated using accuracy, precision, recall, F1-score, ROC-AUC, five-fold cross-validation, and execution cost. Experimental findings demonstrated that the ensemble strategy achieved robust detection performance while keeping computational complexity and execution time low, indicating that the framework is well suited for practical deployment on edge-enabled IoT devices where computing resources are inherently limited. Table 3 compares recent studies.

Table 3. State-of-the-Art Comparison of Recent Smart Home and Edge-IIoT Intrusion Detection Studies

Ref. (Author, Year)	Dataset	Objective / Purpose	Accuracy / Precision / Recall / F1-score	Key Findings	Limitations
[19] Ferrag et al. (2022)	Edge-IIoTset	Introduce a realistic benchmark dataset for centralized and federated IoT/IIoT intrusion detection.	Accuracy: ANN 96.01% (multiclass); RF 82.90%; SVM 85.62%; KNN 83.39%; DT 77.90%.	Developed the Edge-IIoTset benchmark containing diverse cyberattacks and established baseline ML performance for multiclass intrusion detection.	Evaluated only baseline ML models; no advanced feature engineering or ensemble learning.
[20] Gueriani et al. (2025)	Edge-IIoTset	Improve multiclass intrusion detection using an attention-guided hybrid deep learning architecture.	Accuracy: 99.32% Precision: 99.31% Recall: 99.30% F1-score: 99.30%.	Attention-enhanced LSTM-CNN effectively captured temporal and spatial traffic characteristics, improving multiclass attack classification	Hybrid deep learning architecture is computationally expensive and memory intensive for resource-constrained edge devices.
[21] Sohail et al. (2022)	Smart Home IDS Dataset	Develop an explainable and optimally configured ANN for smart-home intrusion detection.	Accuracy: 99.9% (Binary), 99.7% (Category), 97.7% (Subcategory).	Hyperparameter optimization significantly improved detection performance while increasing model explainability.	Focused only on ANN optimization without feature engineering or ensemble learning.
[23] Farea & Kucuk (2024)	KoU-6LoWPAN-IoT + Edge-IIoT	Develop a hybrid Intrusion Detection and Prevention System (IDPS) for edge-based IIoT environments.	Accuracy: 99.9% (Binary), 95.65% (Multiclass).	Demonstrated that ANN-based IDS can be effectively deployed on edge devices with acceptable computational overhead.	Limited multiclass performance and no heterogeneous ensemble strategy.
[25] Ismail et al. (2025)	Edge-IIoTset, TON_IoT, WUSTL-IIoT-2021	Compare lightweight ML algorithms for binary and multiclass intrusion detection across multiple IIoT datasets.	The study performs a comparative evaluation across multiple IIoT datasets and therefore does not report a single unified performance metric for the Edge-IIoTset dataset	Demonstrated that lightweight ensemble learning combined with Mutual Information feature selection improves predictive performance while reducing computational cost.	Comparative benchmarking study; does not propose a unified heterogeneous ensemble framework dedicated to smart-home multiclass IDS.
[27] Tanveer et al. (2025)	RT-IoT-2022	Develop a lightweight ensemble learning framework for real-time IoT intrusion detection.	Performance evaluated using: Accuracy:99.55% Precision:99.51% Recall: 99.49% F1-score: 99.50%	Combined LightGBM, XGBoost and Extra Trees using majority voting to improve detection performance while reducing execution time and computational overhead.	Evaluated on IoT benchmark datasets rather than exclusively on the Edge-IIoTset benchmark.
Proposed Framework	ML-EdgeIIoT	Develop a ensemble multiclass IDS using hybrid feature engineering and heterogeneous ensemble learning for smart-home edge computing.	Accuracy: 99.24% Precision: 98.75% Recall: 99.00% F1-score: 99.00% cross validation score : 98.98 ± 0.13%	Integrates Correlation Analysis + Global Feature Importance with Stacking and Top-3 Majority Voting ensembles, followed by comprehensive evaluation using ROC-AUC, confusion matrix, and stratified 10-fold cross-validation.	—

Although recent studies have substantially improved intrusion detection performance for smart home and IIoT environments, several research gaps remain. Many investigations emphasize either deep learning architectures or lightweight classifiers, while comparatively fewer studies jointly exploit hybrid feature engineering and heterogeneous ensemble learning within a unified multiclass framework. In addition, several studies report excellent classification accuracy but provide limited analysis of model robustness through comprehensive evaluation metrics such as ROC-AUC, confusion matrix analysis, and stratified cross-validation. Motivated by these limitations, the proposed framework

combines correlation analysis and global feature importance for feature engineering with stacking and Top-3 majority voting ensembles, followed by extensive multiclass evaluation on the ML-EdgeIIoT dataset.

5.1. Key Observations and Research Gaps

- **a)** Model Strategies: Research predominantly explores DL and ensemble-based mechanisms. Collaborative and federated frameworks (e.g., CADeSH, SVM-based federated IDS) show excellent potential for balancing performance, accuracy and other metrics.
- **b)** Datasets and Evaluation: Studies make use of standard datasets such as CIC-IDS-2017, NSL-KDD, Edge-IIoT dataset, and KDDTest etc, along with real-world deployments. However, only fewer datasets belong to realistic smart-home multi-attack datasets. [29]
- **c)** Attack identification and Classification: While anomaly detection is mature, explicitly classifying multiple types of attacks (e.g., DDoS, ransomware, MITM, contextual misuse) is still underdeveloped and researchers must develop more sophisticated approaches for robust, ensemble and optimum multiclass classification.

Few studies investigate realistic smart home datasets containing multiple attack categories under smart home edge computing environment. Although anomaly detection (Binary classification of attack/Normal) has achieved considerable success, But the multiclass attack classification capable of accurately distinguishing diverse cyber threats such as DDoS, ransomware, MITM, password attacks, and reconnaissance attacks remains a challenging research problem., In addition, comparative evaluation of multiple machine learning algorithms and heterogeneous ensemble learning strategies using a common experimental framework is relatively limited. Table-3 is summarizing the Comparison of Recent Smart Home and Edge-IIoT Intrusion Detection Studies. Therefore, this study addresses these research gaps by developing and evaluating a comprehensive multiclass intrusion detection framework based on multiple machine learning models and ensemble techniques using the ML-EdgeIIoT dataset.

6. Proposed Machine Learning Framework for Multiclass Attack Detection

Based on the architectural challenges discussed in Sections 2–4 and the research gaps identified in Section 5, this study proposes an integrated machine-learning framework for multiclass cyber-threat detection in smart-home edge environments. The proposed framework directly addresses the identified limitations through optimized feature selection, heterogeneous model's comparison, and ensemble learning using stacking and majority voting approaches using Top-3 best performing models.

6.1. Objective of the Study

The primary goal of this study is to proposed and implement a ML based algorithm which load clean dataset and train , test and evaluate a set of machine learning models that can effectively distinguish between normal network traffic operation and malicious attacks and do the multi-classification using stacking technique and also experimenting with another method to choose best three models out of six model(applied in this study like KNN ,Logistic regression, Random forest model, BRFM, ANN with multiple hidden layers etc) [28] and then collect their classification and combine their decision two methods first using stacking ensemble method and second using voting method using top-3 performer models, to find out final classification of attack for appropriate action. The multi-classification task will use ML-Edge IIoT attack dataset [19] containing labelled network activity record in 15 different classes like (Normal, DDoS_UDP attack, DDoS_ICMP attack, Ransomware, DDoS_HTTP, SQL_injection, Uploading , DDoS_TCP, Backdoor, Vulnerability_scanner, Port_Scanning, XSS, Password, Fingerprinting, MITM) and these 15 classes further grouped into only 8 classes to reduce complexity on the basis of their nature of attacks and operations like (Backdoor attack, Code attack, Dosattack, MiMT, Normal, Password attack, Prob attack, Ransomeware Attack) with final labelling (0 to 7) for final multi-classification with minimum false negative rate. In other words, this study is focusing on the development of a machine learning-based intrusion detection system that can provide more precise attack identification and multi-class attack classification with high accuracy, and efficient resource utilization, suitable for real-time deployment on smart home edge computing platforms. This research aims to address this gap by designing, implementing, and evaluating a smart home-specific IDS which integrate the decisions of multiple ML algorithms to predict more accurate attack identification and multiclass classification of attacks that meets the objectives of security leak protection in edge environments.

6.2. Dataset Overview

The ML-Edge IIoT dataset [19] encompasses thorough network traffic collection data streams containing communications logs labelled with types of attack or normal from an intelligent Internet of Things (IoT) system which was set up in four rooms, making use of DHT, flame sensors, Infrared (IR) sensors, and servo motors connected to Arduino Uno boards. The edge computing environment was done on a Raspberry Pi 4 Model B. Cyber-attacks were performed on the wireless network router from a set of host computer that had Kali Linux installed to mimic real-world attack scenarios [19]. The original dataset has 157,800 labelled records which contains 63 different features (explained in 6.3.a) related to

IoT and IIoT edge network communications. After the removal of duplicate records, a total of 156,986 samples were retained for further operations. In which 24,301 samples were belongs to normal network traffic while the remaining (132,685) were belongs to different attack traffic.

Coverage of Dataset:

- Multi-Protocol Coverage: IoT and IIoT environments are captured by traffic data for TCP, UDP, ICMP, HTTP, MQTT [32], Modbus and DNS protocols.
- Detailed Flow Information: Each network session contains source and destination IP's which ensure meticulous monitoring.
- MQTT-Specific Features: Protocol versions, message types and topics, message lengths with acknowledgment flags alongside some other important metrics mark MQTT attributes as detailed thus valuable for analysis purposes.
- Temporal and Flow Characteristics: Documenting time-stamped events such as capturing request methods as well as content lengths provides for better flow analysis.
- Comprehensive Scope: The total size of this set came out to be 78.3 MB as it contains 63 features spread over 157800 records perfectly labelled within 15 classes.

The dataset is structured in a way to allow application of supervised machine learning techniques while deep learning frameworks could be employed for the purposes of attack detection and classification.

6.3. Edge-Aware Computational Optimization

The proposed framework is designed for deployment on smart-home edge gateways rather than resource-constrained sensing nodes. To improve computational efficiency, the framework incorporates four optimization mechanisms:

(i) Combination of correlation and global feature importance based feature selection to eliminate redundant or less impactful attributes before training,
(ii) Offline model training with online inference only,
(iii) Hyperparameter optimization to obtain compact models without sacrificing predictive performance.
(iv) An optimized Top-3 ensemble strategy that restricts inference to the three highest-performing classifiers instead of executing all candidate models.

These design choices reduce computational overhead, memory utilization, and inference latency while maintaining high multiclass classification accuracy.

6.4. Methodology for attack identification and multi -classification

Detecting attacks and intrusions in the networks of IoT and IIoT (Industrial Internet of Things) is essential for securing vital assets and information. With more devices being connected such as smart thermostats or security cameras, there is a significantly large area that can be attacked. There should be efficient mechanisms for identifying suspicious behavior in cyberattacks or unauthorized access to restricted areas which are breaches of essential security protocols. Such threat identification allows the security administrator to take actions that would mitigate risks, thus avoiding damages, data breaches, or crashing into private spaces uninvited. In addition, this automated threat recognition increases security sustainability in IIOT cyber systems as well as improves user trust towards these technologies Overall scalable safety measures offer extensive protection against losses for owners of smart homes. Following is the suggested diagram for attack identification with hybrid machine learning models that includes steps data preprocessing, model training and model evaluation.

As smart homes and Industrial Internet of Things (IIoT) networks become more common, ensuring their security is increasingly important for both property protection and personal privacy. Everyday devices such as smart thermostats, lights, and security cameras are now connected to home networks, making life more convenient but also expanding the potential targets for cyberattacks. Identification of attacks and intrusions quickly and in very early stage is essential. When any unusual activity or suspicious patterns are detected early, the homeowners and security professionals can respond before any major harm is done by that event. This proactive approach helps the smart home system to prevent unauthorized access, data breaches, dos attacks, and even physical intrusions, all of which can have serious consequences.

Building trust in smart home technology relies on these robust security measures. When users know that their devices are protected by advanced detection systems, they are more likely to embrace and rely on smart home solutions. To address these challenges, this research paper proposes a multi-step machine learning framework for detecting and classifying cyberattacks in smart home networks. The major operations involved in the proposed framework are described below.

a) Data Preprocessing: Cleaning and organizing the data collected from Smart home and Edge-IIoT dataset to ensure it is suitable for analysis, attack identification and classification.
b) Model Training and Testing: Developing several machine learning models to recognize and classifying different types of attacks, evaluating their performance and cross validation done for every machine learning model.
c) Ensemble Learning [30]: This experiment applied two ensemble technique first is stacked model by selecting the top three models that perform best accuracy and low false negative values as base model and used one meta machine learning model for final prediction and in second approach, selecting the top three models that perform best and combining their predictions by applying simple mode for voting and after that both approach evaluated for better predication. These approaches leverage the strengths for the attack identification and classification approach, which leads to more accurate detection and classification.
d) Model Evaluation: Assessing the overall effectiveness of the system using established metrics to ensure reliability.

By implementing this framework, this study aims to improve the accuracy with minimum misclassification and robustness of attack detection and classification in smart home environments. This not only helps prevent losses and privacy breaches for homeowners but also contributes to the broader adoption and trust in smart home technologies. Fig. 1. illustrates the overall workflow of the proposed approach.

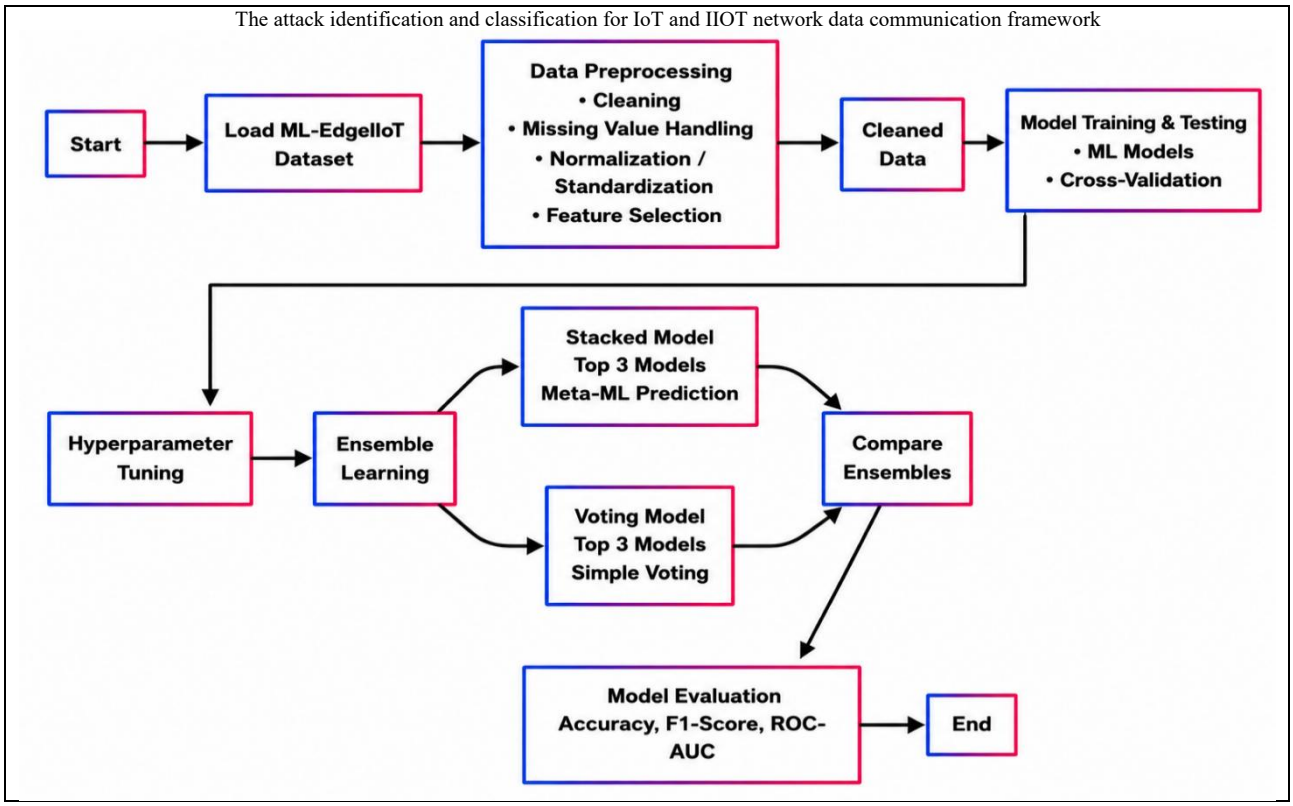

Fig. 1. The attack identification and classification for IoT and IIOT network data communication framework

The detailed algorithm for the proposed ensemble-based multiclass cyberattack classification framework is presented below.

Input
Let D ← ML-EdgeIIoT Dataset
Output
Trained intrusion detection model M* with optimal performance

Step 1: Load Dataset
D ← LoadDataset("ML-EdgeIIoT.csv")

Step 2: Data Preprocessing
2.1 Clean the Dataset:
D_c ← RemoveDuplicates(D)

$D\_c \leftarrow RemoveNoise(D)$

2.2 Handle Missing Values:

$\forall x \in D\_c$, if $x\_i = NULL \Rightarrow x\_i \leftarrow Impute(x_i, mean/mode)$ or $Remove(x)$

2.3 Encode Nominal Features:

$D\_e \leftarrow LabelEncoder(D\_c)$

2.4 Normalize Features:

$D\_e \leftarrow StandardScaler(D\_e)$ where $\mu = 0, \sigma = 1$

2.5 Feature Selection:

Compute the Pearson correlation matrix among predictor variables:

$\rho_{ij} = Corr(f_i, f_j)$ where: ρ_{ij} Pearson correlation coefficient between Predictor features in the dataset f_i, f_j

Construct an initial feature subset by retaining predictor variables satisfying: $0.3 < |\rho_{ij}| \leq 0.9$

$F_{corr} = \{ f_i \mid 0.3 < |\rho_{ij}| \leq 0.9 \}$ Where: F_{cor} is set of selected features for next step

Rank the retained features using Random Forest feature importance:

$I(F_i) = RFImportance(F_i)$ Where $F_i \in F_{corr}$ and $I(f_i)$: Global importance score of feature f_i .

$F_{selected} = F_i$ if $\{F_i \in F_{corr}$ and $I(F_i) > 0\}$ where $F_{selected}$: Final feature subset obtained by selecting the highest-ranked features with importance scores greater than zero.

$D \leftarrow De[F_{selected}]$

Step 3: Preprocessed Data

D_train, D_test ← D (send to model training with split of 70:30)

y_test=D_test[target]

D_test= drop(D_test[target]) // Remove the target feature form test dataset

Step 4: Model Training & Testing

Let $M = \{M_1, M_2, ..., M_n\}$, where:

M_1 = Logistic Regression,

M_2 = Random Forest,

M_3 = KNN,

M_4 = BRFM,

M_5 = SVM,

M_6 = ANN

For each $M_i \in M$:

$M_i \leftarrow Train(M_i, D\_train)$ and Evaluate M_i using {Accuracy, Precision, Recall, F_1, AUC}

Step 5: Hyperparameter Tuning

For each $M_i \in M$:

$M_i^* \leftarrow Optimize(M_i,$ GridSearch or RandomSearch or Empirical evaluation)

Step 6: Model testing

For each $M_i \in M$:

$\hat{y}_i = M_i (D\_test)$

Print (all performance metrics)

Step 7: Ensemble Learning

$M_{top3} = \{M\_a, M\_b, M\_c\} \in M$, top 3 based on:

Maximize Accuracy, Minimize False Negative Rate

7.1 Stacked Model:

$M_{stacked} = StackingClassifier(M\_a, M\_b, M\_c$,meta-learner = M_meta , fusion = Decision-level (Late Fusion))

$Z = [\hat{y}a, \hat{y}b, \hat{y}c]$ where: $\hat{y}a, \hat{y}b, \hat{y}c$ are predictions of M_a, M_b, M_c

$\hat{y} = Mmeta(Z)$

where: Z = Meta-feature vector formed by the predicted class labels of the three base classifiers and Mmeta = Logistic Regression meta-classifier.

7.2 Voting Model:

$M_{voting} = ModeVoting(M\_a, M\_b, M\_c,$ strategy = Mode of classification result)

$\hat{y} = Mode(\hat{y}a, \hat{y}b, \hat{y}c)$ where Mode() returns the attack class predicted by the highest number of models.

Step 8: Compare Ensemble Approaches
Evaluate $M_{\_stacked}$ and $M_{\_voting}$ on:
Metrics = {Accuracy, Precision, Recall, F_1, AUC}
$M^* \leftarrow \max (m \in \{M_{\_stacked}, M_{\_voting} M\}$ Performance(m))

Step 9: Final Model Evaluation
Evaluate M* on D_test using {Accuracy, Precision, Recall, F1-score, ROC-AUC, Confusion Matrix}
Validate $m \in M^*$ using Stratified 10-Fold Cross Validation to ensure robustness

Step 10: Finalize and Deploy
Step 11: End
Return M*

The mathematical formulations and computational complexity analysis of the proposed algorithm across different stages are summarized in Table 4.

Table 4. Mathematical formulas and over all time complexity of ensemble approach

Stage	Equation	Description / Variable Details
Dataset Representation	$\mathbf{D} = \{(\mathbf{x\_i}, \mathbf{y\_i})\}\_(\mathbf{i} = \mathbf{1})\text{^}\mathbf{N}$	D: dataset; x_i: feature vector; y_i: class label; N: number of samples
Data Cleaning	$\mathbf{D\_c} = \mathbf{D} - \{\mathbf{Duplicate}, \mathbf{Noise}\}$	D_C is dataset with removed duplicate and noisy records
Missing Value Handling	$\mathbf{x\_i} = \{\mathbf{Impute}(\mathbf{x\_i}), \mathbf{if\ x\_i} = \mathbf{NULL};\ \mathbf{otherwise}, \mathbf{x\_i}\}$	Mean/mode imputation or removal
Feature Encoding	$\mathbf{D\_e} = \mathbf{LabelEncoder}(\mathbf{D\_c})$	Encodes categorical attributes
Feature Normalization	$\mathbf{z} = (\mathbf{x} - \boldsymbol{\mu})/\boldsymbol{\sigma}$	μ: mean, σ: standard deviation
Feature Selection	$\mathbf{F_{selected}} = \{\mathbf{f_i} \mid \mathbf{0.3} \le \lvert\mathbf{Corr}(\mathbf{f\_i}, \mathbf{y})\rvert < \mathbf{0.9}\}$	Pearson correlation-based feature selection
Classifier Prediction	$\hat{y}\_\mathbf{i} = \mathbf{Mi}(\mathbf{x})$	Prediction by i-th classifier
Model Optimization	$Mi *= Optimize(Mi, \theta)$	M_i*=optimized model θ: optimized hyperparameters
Stacking Ensemble	$M\_stacked = StackingClassifier(M_a, M\_b, M\_c, meta - learner = M\_meta)$	M_meta: Logistic Regression meta-classifier ŷ_RF: Random Forest Classifier ŷ_KNN: KNN classifier ŷ_ANN: ANN classifier
Majority Voting	$\hat{y} = Mode(\hat{y}\_RF, \hat{y}\_KNN, \hat{y}\_ANN)$	Majority voting among top-3 models
Final Model Selection	M*=argmax_(m∈{M_stacked,M_voting}) Performance(m)	M_stacked: Stacked Model classifier M_voting: Voting classifier M*:Selected best ensemble
Voting Ensemble Complexity	$O(N.F + T \log N + W)$	N: samples, F: features, T: Number of RF trees, W: ANN weights
Overall Framework Complexity	$O(P \times C\_train + NF + T \log N + W)$	P: parameter combinations; C_train: training cost

The computational complexity of the proposed Top-3 Majority Voting Ensemble is determined by the inference cost of its constituent classifiers, namely K-Nearest Neighbors (KNN), Random Forest (RF), and Artificial Neural Network (ANN). The prediction complexity of KNN is O(NF), where N denotes the number of training samples and F represents the number of selected features, since each test instance is compared with all stored training samples. The Random Forest classifier requires O(TD) operations, where T is the number of decision trees and D is the average depth of each tree. Assuming balanced trees, the average tree depth is approximated as D≈logN, resulting in a complexity of O(TlogN). The ANN performs inference through a forward propagation process with a computational complexity of O(W), where W denotes the total number of trainable weights in the network. The majority voting operation combines the predictions of the three classifiers and incurs a constant computational cost of O(3), which is equivalent to O(1).

Consequently, the overall inference complexity of the proposed voting ensemble is expressed as O(NF+TD+W), which simplifies to O(NF+TlogN+W) for balanced Random Forests. Since the voting operation contributes only constant overhead, the computational cost is primarily dominated by the prediction complexities of the three constituent classifiers. Furthermore, the proposed framework performs computationally intensive tasks, including model training and hyperparameter optimization, in an offline environment, while only the optimized inference stage is executed on the edge gateway.

This strategy reduces online computational overhead and latency, making the framework suitable for smart-home edge gateway deployment. Therefore, the final time complexity of the proposed Top-3 Majority Voting Ensemble is O(NF+TlogN+W), where the KNN prediction stage constitutes the dominant computational cost, while the majority voting operation itself introduces only constant-time overhead.

The described diagram and algorithm were implemented and executed offline on a laptop featuring 8 GB of RAM,

an Intel Pentium Silver N5000 processor running at 1.10 GHz (with 2 cores), and a 119 GB SSD for internal storage. The system operated on Windows 11 and utilized Python version 3.2. The objective of this platform was model development and comparative evaluation rather than deployment. In practical smart-home environments, only the trained inference model is expected to execute on edge gateways such as Raspberry Pi, NVIDIA Jetson Nano, or similar embedded edge devices. Offline training on a conventional workstation followed by lightweight edge inference is a common deployment strategy for machine learning-based intrusion detection systems. Data analysis and predictive tasks were carried out using popular Python libraries, including pandas, numpy, sklearn, datetime, scikitplot, matplotlib, seaborn etc[31] . The detailed of major steps are as discussed in next subsections.

6.4.1. Data Pre-processing

At this stage of this study, the first step was to bring in the dataset into data frames for pre-processing and analysis. The original dataset contained 157,800 labelled records which were defined by 63 different attributes pertaining to IoT and IIoT network communications. After removing duplicate records, a total of 156,986 records were retained. Out of these, 24,301 were classified as normal network traffic while the remaining were varieties of attack traffic amounting to 132,685. This distribution is summarized in Table 5 and plots (Fig. 2). For ease of analysis as well as reduction of classification complexity, the initial 15 attack categories had been further reduced into eight more generic classes detailed in Table 6.

Table 5. Distribution of Original Dataset Samples

Labels of samples	No. of Samples
Normal	24301
DDoS_UDP	14498
DDoS_ICMP	14090
Ransomware	10925
DDoS_HTTP	10561
SQL_injection	10311
Uploading	10269
DDoS_TCP	10247
Backdoor	10195
Vulnerability_scanner	10076
Port_Scanning	10071
XSS	10052
Password	9989
Fingerprinting	1001
MITM	400

Table 6. Mapping of Original Attack Categories to Proposed Study's Labels

Labels of samples	New Mapped Labels	Numeric Labels
Backdoor	Backdoor	0
SQL_injection	Code_attack	
Uploading		
XSS		1
DDoS_UDP	DosAttack	
DDoS_ICMP		
DDoS_HTTP		2
DDoS_TCP		
MITM	MIMT	3
Normal	Normal	4
Password	Password	5
Vulnerability_scanner	Probattack	
Port_Scanning		6
Fingerprinting		
Ransomware	Ransomware	7

An initial inspection found no missing values in the dataset. There were some nominal features such as mqtt.message , ip.src_host, mqtt.topic , mqtt.protoname, http.request.full_uri, http.request.version, tcp.options etc that were converted into numeric forms using LabelEncoder from sklearn.preprocessing for effective normalization and model training during later stages. Selection of features was done through Pearson method of correlation analysis combined with global feature importance ranking. An individual feature with minimum standard correlation coefficient with other features having absolute value was 0.3 and also features with global feature importance value greater than 0.0 were selected. Thus out of

the initially defined 63 features with target feature included, a subset of 21 features including target feature was selected to be used for further studies. Fig. 2 shows the attack class distribution of the dataset. Fig. 3 presents the global feature importance plot, highlighting the contribution of each selected feature to the classification model. Fig. 4 illustrates the correlation heat map of the selected dataset features, showing the relationships among the input variables.

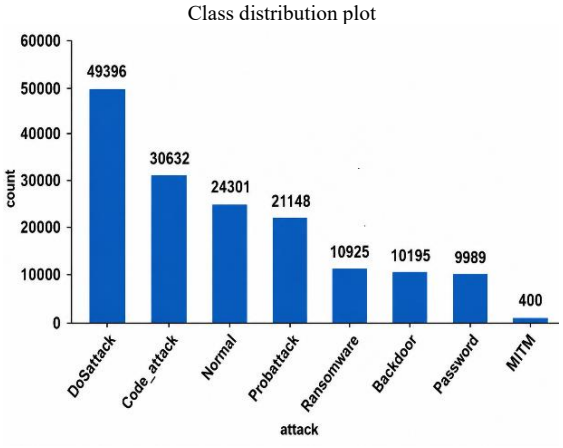

Fig. 2. Class distribution plot

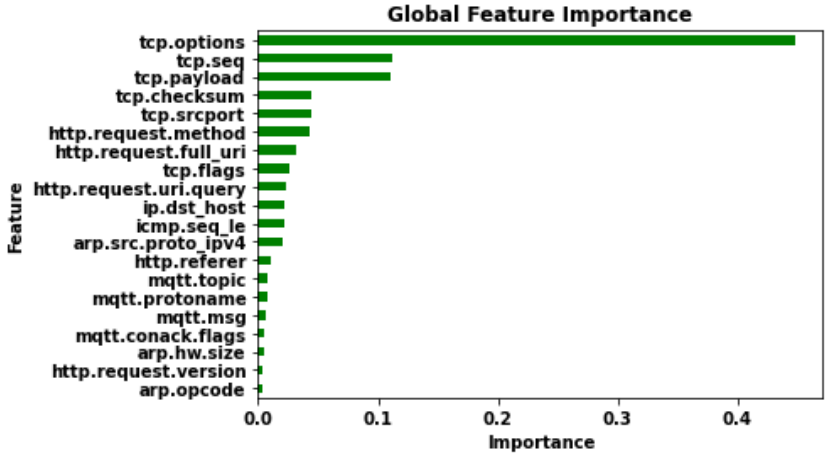

Fig. 3. Global features importance plot

Correlation Heat Map of selected features of dataset

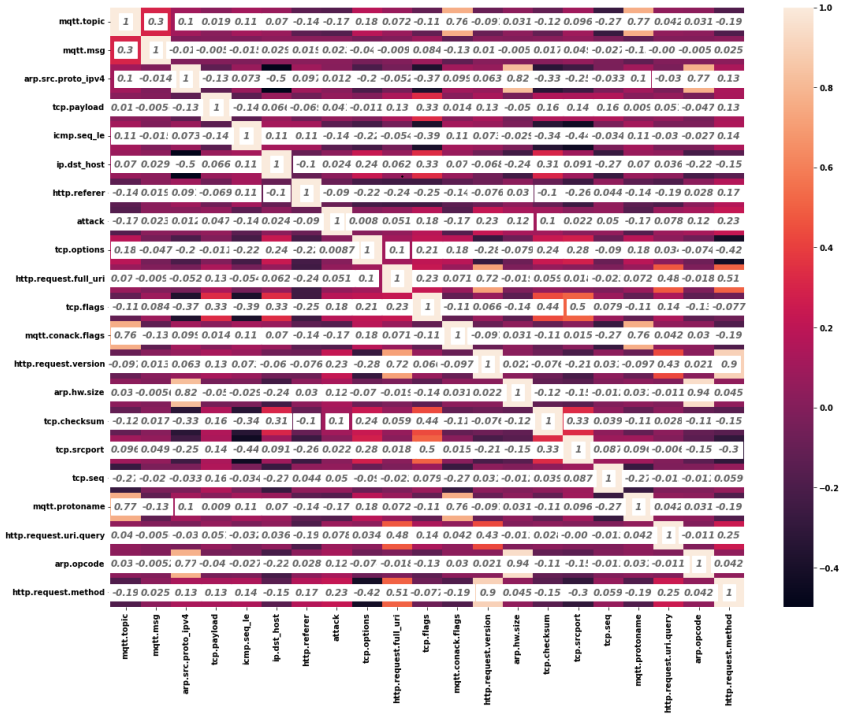

Fig. 4. Correlation Heat Map of selected features of dataset

Major operations performed in this phase on the dataset are as discussed below:

- Data Summary and Initial Exploration: It provides the basic and statistical information related to complete data set and it's features. it includes basic information as given below.

➢ Inspecting dataset structure (number of rows, columns, data types).
➢ Identifying null/missing values and outliers.
➢ Generating basic statistical metrics (mean, median, standard deviation, etc.).
➢ Checking the distribution of data across features.

- Handling Missing and Duplicate Data: Removing duplicate records to maintain data integrity and also checking missing values and Imputing values using mean/median for numerical data or mode for categorical data (if required).
- Feature Engineering and Transformation: Normalizing or standardizing numerical of data done for uniform scale using StandardScaler() and encoding categorical features done using LabelEncoder to ensure the feature's uniformity.
- Correlation Matrix and Feature Importance Plot: Visualizing Pearson correlation between numerical features to detect multicollinearity. it helps in identifying redundant features that can be dropped or merged. Also identifying important features using Tree-based models like Random Forest Regressor to calculate feature importance.
- Class Label Mapping and Analysis: Mapping class labels to the numerical values (e.g., 0 → "Backdoor Attack", 1 → "Code attack", etc.). and a bar plot showing sample count per class and also checking for class imbalance, which can affect model performance.
- Data Splitting: Dividing the cleaned and balanced dataset into training set 70% and Test set 30% using Stratified splitting ensures class distribution is preserved in all subsets . The dataset was partitioned into training and testing subsets using random_state = 100, while machine learning models with stochastic initialization, including Random Forest and Balanced Random Forest, were configured using random_state = 42.
- Visualization Summary: Feature Correlation Heatmap (Fig. 4) shows the inter-feature relationships and feature's multicollinearity, Global Feature Importance Plot(Fig. 3), which identifies features contribute most to classification, based on a Random Forest Regressor and Class Distribution Bar Chart, which visualizes the sample count for each class and helping detect imbalance.

6.4.2. ML Approaches in This Study

Six machine learning algorithms are evaluated in this study under the training and testing phase for attack identification and 3 top performer selected on the basis of their performance matrix. A brief detail of ML models, which used in this implementation are as discussed in next subsections.

Logistic Regression: In multi-class classification problems, logistic regression is modified to predict outcomes in which the target variable has more than two categories. This modification is often applied by using techniques like One-vs-Rest (OvR) where each binary classifier is learned for each class versus all other classes, or by applying multinomial logistic regression that uses the softmax function to predict the probability of every feasible class. The advantage of multi-class logistic regression is its interpretability and efficiency in computations, and hence it is used as a common baseline approach in multiple research and applications, e.g., placing students in various academic departments based on entrance exam scores. LR model's performance can be low or not as per expectation in those cases where the boundaries between two classes are unclear or the dependence between features and classes is very nonlinear.

Random Forest : Random Forest is a type of ensemble learning that aggregates the results of many decision trees to make predictions more accurate and more robust. Every decision tree in the forest is trained on a random subset of the dataset, using random feature selection at every node, which encourages diversity among the models and prevents over-fitting. At inference, predictions are made based on a majority voting scheme across all trees. This method both performs well in generalization and is efficient in handling high-dimensional and noisy datasets; therefore, it stands as the best method for solving complex classification problems.

Balanced Random Forest: Balanced Random Forest is an extension of the standard Random Forest algorithm, specifically designed to address the challenges posed by class-imbalanced datasets in binary classification. In this variant, each decision tree is trained on a bootstrap sample that is balanced by under-sampling the majority class and over-sampling the minority class. This technique ensures that both classes are equally represented during the training phase, which improves the model's sensitivity to the minority class. The final prediction is obtained by aggregating the outputs of all balanced trees, thereby maintaining the strengths of the random forest framework while mitigating bias toward the dominant class.

K-Nearest Neighbors (KNN) [6]: This algorithm is an intuitive instance-based classifier which determines the class label of data points based on the closest 'K' instances in feature space. In this case, KNN was applied to intrusion detection dataset to take advantage of the spatio-temporal proximity of network traffic features. Because KNNs do not assume a parametric model, it required minimum or no training phase, all training data is stored and classification occurs during prediction. Although-performing on-the-fly computations may be resource intensive with large scale data like IoT and IIoT's attack classification, the ability to clearly differentiate, expose and learn attack patterns allows it serve as an effective baseline benchmark model in this study.

Support Vector Machine (SVM)[6] : As a supervised learning method, SVM constructs an optimal separating hyperplane between two or more classes in high-dimensional feature space. For our purpose, This study used SVM to classify normal and attack traffic within the IoT smart home ecosystem framework. Its capability for maximizing the margin between decision boundary and data reinforces its strengths for capturing subtle as well as intricate complex attacks when combined with kernel functions that account for non-linear separability. SVM is resistance to overfitting makes it particularly useful even in scenarios where real time data is noisy in such a way that data points having overlapping classes or outliers common in real-world IoT datasets.

Artificial Neural Network (ANN) [7] : ANN is inspired with the structure and functioning of the human brain, which consists of layers of interconnected processing units known as neurons. To enhance performance, a three hidden layer feed forward neural network (150, 100, 50 numbers of neurons, ReLu activation function, maximum iteration =1000 and solver parameter with adam) was applied in this study. The network optimizes the cross-entropy loss, and the default L2 regularization mechanism available in the implementation (using MLPClassifier) was used to improve generalization The ANN model can learn hierarchical representations of features or attributes because of its multi-layer perceptron architecture, which basically enable ANN for better in recognizing complex patterns. For this reason, it performs better on large datasets with high-dimensional and non-linear interdependencies, like the Edge-IIoT traffic dataset and the IoT traffic dataset.

Stacking Ensemble [30]: To further increase the classification accuracy and reliability, this study used a stacking ensemble approach. In this method, three base classifiers chosen (top -3 performer of the current study please refer the result section) Random Forest, KNN, and ANN chosen based on their relative performance on the Edge-IIoT dataset individually. The output (decision-level late fusion) of base learners were employed as to provide input to the Logistic Regression model, which was the meta-classifier. This hierarchical learning approach enables themeta-learner to extract second-order patterns through the fusion of decision boundaries that are learned by the base models. The mathematical expressions for stacking ensemble are mentioned in section 6.4. The ensemble actually benefits from model diversity to enhance generalization performance and learns relationships among classifiers and having higher risk of overfitting if the meta-classifier is not trained carefully., that is essential when dealing with high-dimensional and imbalanced intrusion detection datasets.

Voting Ensemble with top-3 Voting Strategy: To further enhance classification reliability, a simple voting ensemble approach was implemented using the same top-performing heterogeneous models—Random Forest, K-Nearest Neighbors (KNN), and Artificial Neural Network (ANN) as participant's model. Each of these models was independently trained

and evaluated on the Edge-IIoT dataset for attack detection and multi classification. In the ensemble framework, predictions from Top- three models were combined using a majority voting strategy (Mode of final decision of individual models), This method utilize the diverse learning mechanisms of each model like tree-based, instance-based, and neural architectures, to reduce the risk of individual model biasness and also improve robustness and lightweight than stacking ensemble. So this approach leverages complementary decision boundaries, leading to a more balanced and accurate classification for all classes of malicious traffic patterns in the IoT and IIoT environment for attack identification and classification. The mathematical expressions for stacking ensemble are mentioned in section 6.4.

6.4.3. Model Training, Testing and Validation

The cleaned dataset was partitioned into training and testing subsets using a 70:30 ratio split. In machine learning, this type of division is done for model training , testing and evaluation purposes. It helps assess the model built objectively without any bias through mitigated overfitting as it enables building models that reflect real-world usage scenarios. Such splits also help in building stronger and reliable predictive models which aids in objective comparison among different algorithms along with efficient hyperparameter tuning. Multiple machine learning algorithms were applied in this case study including Logistic Regression (LR), Random Forest (RF), Balanced Random Forest (BRF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Artificial Neural Network (ANN). In addition to these, further ensemble methods known as Top-3 Stacking with Meta model and Majority Voting were used. All models were trained, tested and cross validated on the same datasets which had been standardized using StandardScaler() function resulting in zero mean with a unit variance. This step is crucial because during model training, standardized feature ranges aid in preventing biases from disproportionate scales due to uneven feature influence.

Accuracy, precision, recall, F1-score and other performance matrix were also used as the model comparison benchmarks. To improve generalizability and reliability of estimates and variance in performance across different runs of the model, 10 stratified cross validations were done. It ensured that the original class ratios were maintained for every fold minimizing chances for model overfitting. To achieve optimal model performance, the following key approaches were taken:

- The overall dataset was divided into training and testing portions with a split ratio of 70% to 30%.
- Cross validation using Stratified 10-fold Cross-validation was enforced to confirm generalizability and stability of the models.
- Hyperparameter optimization by bootstrap sampling and GridSearchCV with 5-fold internal cross validation on parameters and estimators was performed for Random Forest Models
- For KNN classifier, the best performing configuration determined using different K value range from 1 to 20 with minimum misclassification count.
- ANN classifier empirically evaluated with different numbers of hidden layers, numbers of neurons, activation functions, solver parameter etc. to attained optimum performance. Table 7 provides a comprehensive details of Hyper-parameter Optimization Strategy for different models used in this study.

Table 7. Approaches for Hyperparameter Optimization

Model	Hyperparameter Optimization Strategy	Final Configuration
Random Forest	Grid Search with 5-fold Cross Validation	n_estimators = 200, max_depth = 12, min_samples_split = 5, bootstrap = True, class_weight = balanced
K-Nearest Neighbors	Evaluated (k = 1 to 20) and selected the value with the minimum validation misclassification error	k = 5
Artificial Neural Network	Empirical evaluation of multiple network architectures and training parameters	hidden_layer_sizes = (150,100,50), ReLU, Adam, max_iter = 1000
Logistic Regression	Default configuration with balanced class weights	class_weight = balanced
Stacking Ensemble	Logistic Regression meta-learner with optimized base classifiers	RF + KNN + ANN
Voting Ensemble	Majority voting among the three highest-performing classifiers	RF + KNN + ANN

6.4.4. Performance Metrics used for model Evaluation of this study

Accuracy: It measures the overall effectiveness of the model in correctly predicting both positive and negative instances and It is calculated using the formula: Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision and Recall: These metrics use to measure the model's ability to correctly detect attack instances, especially in the presence of imbalance classes dataset. Precision refers to the proportion of correctly predicted positive observations to the total predicted positives: Precision = TP / (TP + FP) and Recall (also known as sensitivity) measures the proportion of actual positives correctly identified: Recall = TP / (TP + FN)

F1-Score: The F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both. It is especially helpful when models deals with uneven class distributions or dataset with minor classes . The formula for F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

ROC Curve and AUC: The Receiver Operating Characteristic (ROC) curve illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate. The Area Under the Curve (AUC) summarizes the overall ability of the model to distinguish between classes. A higher AUC indicates better model performance.

Note: Details TP, TN, FP, FN described in confusion matrix section

6.4.5. Results and discussion:

Through extensive experimentation and cross validation done for all models for multi-classification for 8 classes as discussed above. Every model achieved good performance and The model had highest accuracy identified as the best-performing model, for this final attack classification. Key performance metrics including performance summary for all models is reported in Table 8, Attack class-wise F1-Score, Precision and Recall metrics for all Models is reported in Table 9 ,Table 10 and Table 11.

Table 8. Performance metrics for all model for testing phase

Model	Accuracy	AUC value
Logistic Regression model	0.833	0.98
Random Forest Model	0.990	0.9998
Balanced RF Model	0.965	0.999
KNN	0.990	0.998
SVM	0.842	0.982
ANN	0.986	0.999
Stacking approach	0.9832	N/A
Top-3 Ensemble Voting Model	0.9924	N/A

To ensure robustness and generalization of the proposed methodology, Stratified 10-fold cross-validation was performed on the three most effective classifiers: Random Forest (RF), K-Nearest Neighbour (KNN), and Artificial Neural Network (ANN). The Random Forest classifier obtained a mean cross-validation accuracy of 98.98 ± 0.09%, and the K-Nearest Neighbour classifier obtained 98.98 ± 0.13%. The Artificial Neural Network obtained an accuracy of 98.36 ± 0.11%. The mean of the mean cross-validation accuracy of these three classifiers is 98.77%, which shows that there is stability and consistency in the prediction performance of these three classifiers. The low standard deviations of these three classifiers prove their reliability in varied training and validation sets, thus ensuring the robustness and generalization ability of the proposed intrusion detection system.

Table 9. Attack class-wise F1-Score metrics for all Models

Classes	LR Model	RFM	BRM	KNN	SVM	ANN	Stacked	Top-3 E-Voting
Backdoor	0.90	0.97	0.96	0.96	0.90	0.97	0.93	0.97
Code_attack	0.67	0.99	0.97	0.99	0.63	0.99	0.99	0.99
DoSattack	0.89	1.0	0.96	1.0	0.92	1.0	1.0	1.0
MITM	1.0	1.0	1.0	1.0	1.0	1.0	0.97	1.0
Normal	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Password	0.61	0.96	0.82	0.98	0.59	0.99	0.99	0.99
Probattack	0.86	1.0	1.0	1.0	0.89	0.98	0.98	1.0
Ransomware	0.84	0.97	0.97	0.97	0.84	0.94	0.93	0.97
Average F1	0.864625	0.98625	0.96	0.9875	0.84625	0.98375	0.97375	0.99

Table 9 presents the class-wise F1-score of all evaluated machine learning models for all eight attack categories. The results indicate that several classifiers achieved an F1-score of 1.00 for the Normal class, except stacked classifier, which demonstrating perfect identification of benign traffic in the independent test dataset. This result, however, should not be interpreted as perfect multiclass classification. The reported overall multiclass accuracy of 99.24% is computed across all eight classes and includes the classification errors that occur among different attack categories such as DDoS, MITM, Password Attack, and Probe Attack. Therefore, perfect performance on the Normal class is statistically consistent with an overall accuracy below 100% for multi classification of attack, because the remaining prediction errors arise from confusion among attack classes rather than between normal and malicious traffic.

Table 10. Attack class-wise Precision metrics for all models

Classes	LR Model	RFM	BRM	KNN	SVM	ANN	Stacked	Top-3 E-Voting
Backdoor	0.85	1.00	1.00	0.98	0.85	1.00	0.91	1.00

Code_attack	0.77	1.00	0.99	0.99	0.79	0.99	1.00	0.99
DoSattack	0.98	1.00	1.00	1.00	0.99	1.00	1.00	1.00
MITM	1.00	1.00	1.00	1.00	1.00	1.00	0.95	1.00
Normal	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
Password	0.45	0.93	0.70	0.99	0.42	0.97	0.99	0.98
Probattack	0.78	1.00	1.00	0.99	0.85	0.99	1.00	0.99
Ransomware	0.87	0.94	0.94	0.95	0.87	0.88	0.90	0.94
Average Precision	0.8375	0.98375	0.95375	0.9875	0.84625	0.97875	0.9675	0.9875

Table 11. Attack class-wise Recall metrics for all models

Classes	**LR Model**	**RFM**	**BRM**	**KNN**	**SVM**	**ANN**	**Stacked**	**Top-3** E-Voting
Backdoor	0.95	0.94	0.93	0.95	0.96	1.00	0.95	0.94
Code_attack	0.59	0.98	0.96	0.99	0.53	0.99	0.98	0.99
DoSattack	0.81	0.99	0.93	1.00	0.87	1.00	0.99	1.00
MITM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Normal	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Password	0.94	1.00	1.00	0.97	0.99	0.97	0.99	0.99
Probattack	0.94	1.00	1.00	1.00	0.94	0.99	0.96	1.00
Ransomware	0.81	1.00	1.00	0.98	0.82	0.88	0.96	1.00
Average recall	0.88	0.98875	0.9775	0.98625	0.88875	0.97875	0.97875	0.99

These metrics indicate that top -3 performer models are Random Forest model, KNN, ANN with highest value of accuracy, F1-score, Precision and Recall values among all classifiers used in this study as base model, after that these three models used in a stacked ensemble models as base learner/classifier with the integration of Logistic regression model as meta classifier. In another way top-3 performer classifiers participated in voting by taking mode of their decisions to infer the final decision of multi-classification of attack. The Top-3 voting approach is leading in all metrics in compare to stacking approach all aspects. The accuracy and other metrics of stacked model and Top-3 simple voting model are very competitive but the voting model metrics are better than stacked model in all aspects.(like Highest accuracy 0.9924, Average F-1 score 0.99 , Average precision 0.9875 and Average recall score 0.99 for all attack classes) The Area under curve value (AUC) and cross validation average accuracy score for stacked and voting model is not available because , both models are depend upon their top-3 performer models (RF,KNN and ANN).

6.4.6. Confusion Matrix and important graphs of Edge-IIoT attack dataset Multi-Classification

In machine learning, a confusion matrix visualizes the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) counts, thus evaluating the performance of a classification model. In a multiclass classification problem, such as identifying normal activities and seven types of attacks within an Internet of Things (IIoT) network, the confusion matrix gives a clearer picture of model accuracy as well as the misclassification patterns. For the Random Forest, K-Nearest Neighbors, Balanced Artificial Neural Network, and stacked ensemble models as well as the majority voting ensemble model which were tested in this work, confusion metrics were created at testing time. These metrics aid in transforming multiclass predictions into binary classification which makes understanding the separation ability of the models for normal operational activities and potential threat indicators much easier. The confusion matrix plots and miss-classification graph is showing Top-3 voting model is classifying the attack data set more accurately with minimum misclassification than Stacking, ANN, KNN and RF models.

Confusion metrics and Miss-classification graphs of models

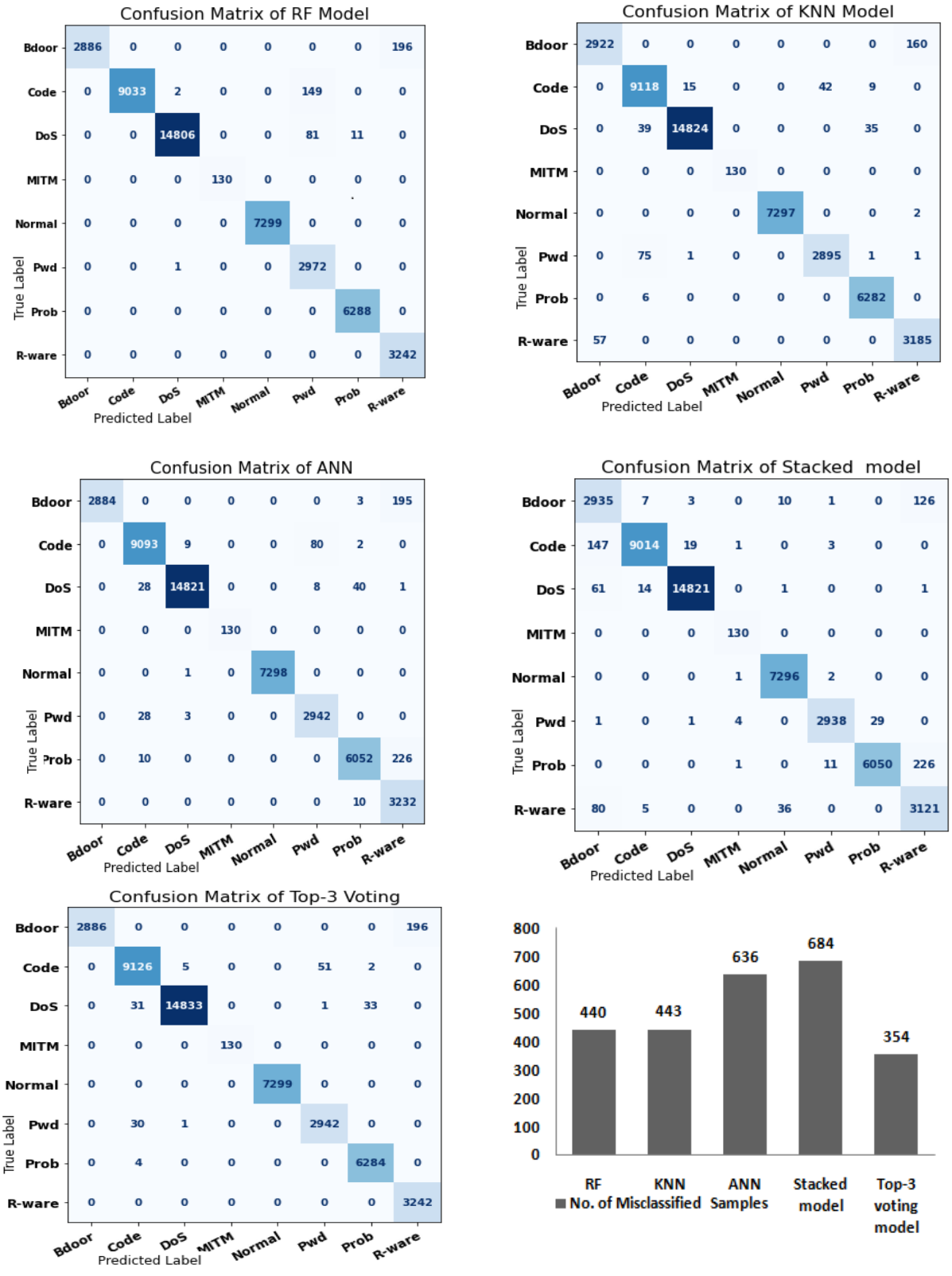

Fig. 5 Confusion metrics and Miss-classification graphs of models

Confusion matrix reported in Fig. 5 for all models reported that except Random forest model, KNN and Top-3 voting model was able to clearly identify the normal communication activity with Zero misclassification, it means system alert will trigger alert when attack happened. When mapping of multiclass confusion matrix done in binary classification for (attack and normal) then value of FN (false Negative) is Zero which is sum of all values in corresponding row (except TP which value is 7299) and value of FP (False Positive) is also Zero which is sum of values in corresponding column and rest all are value of TN :39797 (True Negative) in case of normal communication TN represents the attacks. The three

highest-performing classifiers (Random Forest, Balanced Random Forest, and ANN) achieved an F1-score of 1.00 for the Normal class on the independent test dataset, which means attack identification has been done with 100% accuracy for binary classification (attack/normal) as derived from the multiclass confusion matrix. and multi classification done with 99.24% overall accuracy and average of F1 -score for every class is 0.99, which shows an optimum case of classification on Edge IIoT dataset with improved scores in every aspect.

Some of key graphs of are shown below, all models achieved a high level of predictive accuracy with is cross validated and measured in terms of ROC-AUC curve with highest value of AUC is 0.999 (≈1 or RF and ANN) The results from cross-validation using a stratified K-fold approach with K=10 show that all top three performer models performed consistently well with average highest value of 98.9% for Random forest model and ANN.

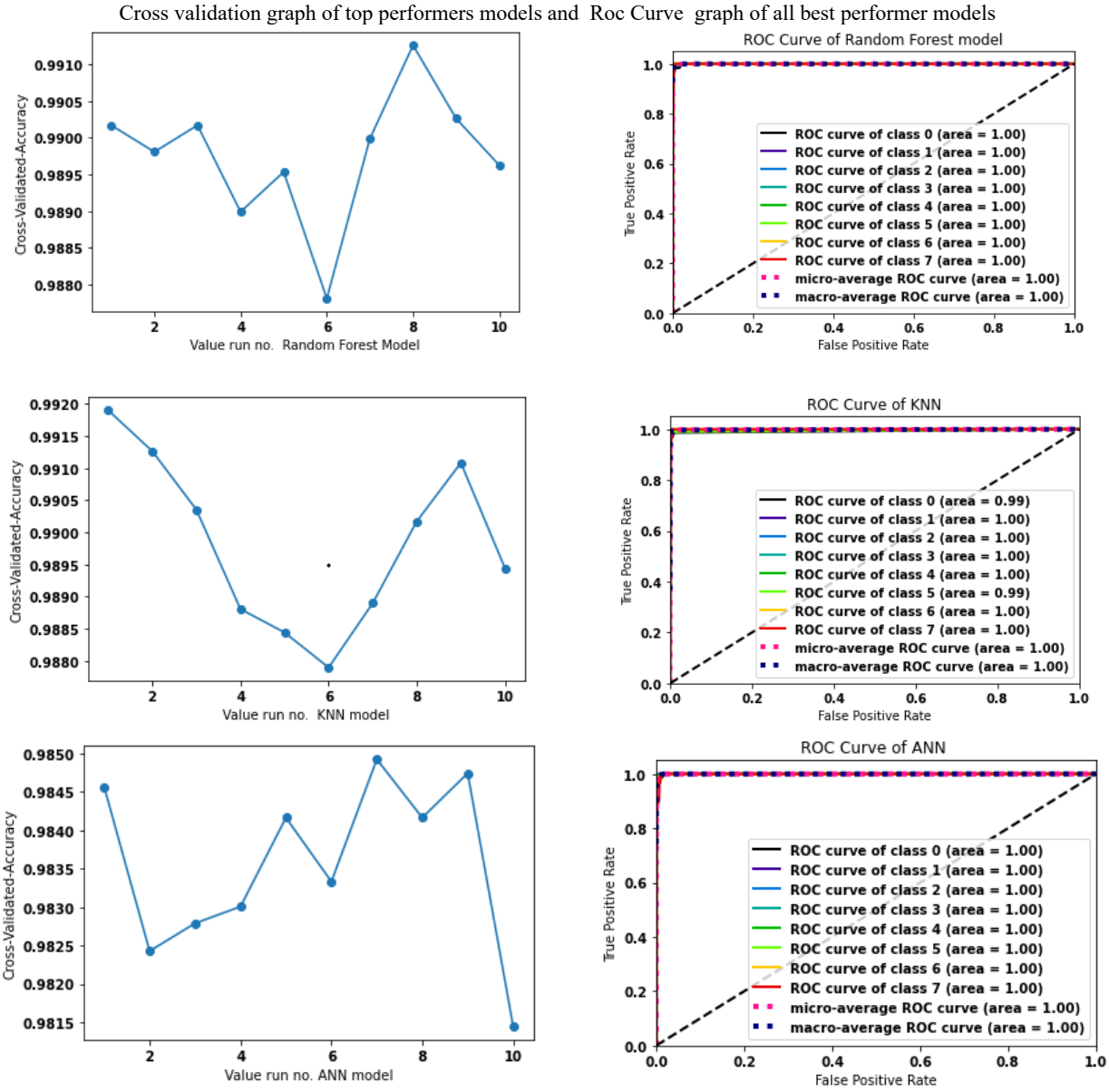

Fig. 6. Cross validation graph of top performers models and Roc Curve graph of best performers models

The Stratified K-10 cross-validation applied to ensure the robustness of the models and also ensure the generalization of models for unseen or real time data. Moreover, the ROC curves for the multi-class classification show outstanding discrimination power of classification models across multiple threshold values by plotting the graph for TPR (True Positive Rate) against the FPR (False Positive Rate). These visualizations confirm that, these models work efficiently in classifying multiple classes in the given dataset. Fig. 6 showing the plots of Cross validation graph of top performers models and Roc Curve graph of top performers models.

The proposed framework Model training, testing, hyperparameter optimization, and ensemble construction are performed offline on a high-performance workstation. During deployment, only the optimized ensemble model is executed on the smart-home edge gateway for real-time inference. This strategy significantly reduces computation,

communication overhead, and energy consumption while preserving user privacy because network traffic remains within the local environment.

The computational cost of the proposed framework is dominated by the optimized ensemble model performs inference. Correlation-based feature selection reduces the input dimensionality, while the Top-3 ensemble strategy eliminates unnecessary classifier execution. Although KNN contributes the highest prediction cost, the overall inference complexity remains practical for gateway-class edge devices because Random Forest and ANN predictions are computationally efficient and the majority voting operation incurs only constant overhead.

7. Conclusion and Future Work

This study proposed and implemented a comprehensive machine learning framework which aimed to enhance the security and privacy of smart home edge environments by accurately detecting and classifying multiple types of cyberattacks. By taking the rich and diverse ML-EdgeIIoT dataset in this experiment, the proposed work extends beyond traditional binary classification to support detailed multiclass identification of eight distinct attack categories, including Backdoor attack, MITM attack, DDoS attacks, Ransomware, Password attack, SQL-Injection, Prob-attacks, and Normal traffic. The study implemented a complete methodology ranging from data pre-processing to model hyperparameter tuning and performance evaluation. Furthermore, the Artificial Neural Networks algorithm with multiple hidden layers along with the Balanced Random Forest, K-Nearest Neighbors, Random Forest model, and Logistic Regression were incorporated to train the model and evaluate its performance. In an attempt to improve detection accuracy and multi-classification process, two ensemble approaches were also developed using RFM, KNN, ANN (as these models are top-3 performers), a Top-3 Stacking ensemble with Logistic Regression as the meta-classifier, and second Top-3 Majority Voting ensemble. These two ensemble models achieved impressive performance, with accuracy of 98.0% and 99.24%, and average F1-scores of 0.97 and 0.99, respectively. The Top-3 Voting model exhibited the lowest number of misclassifications in the confusion matrix, which indicates better reliability for real-world deployment. The confusion matrix analysis further confirmed that the Top-3 Majority Voting model correctly identified all Normal traffic instances without any misclassification. This observation should not be interpreted as perfect multiclass classification because the reported overall accuracy of 99.24% was computed across all eight traffic classes. The remaining prediction errors occurred only among a small number of attack categories with similar network traffic characteristics, while no Normal traffic instance was classified as malicious. All top-3 models were cross-validated using Stratified K-fold Cross Validation (K = 10) for robustness and generalization, and their classification capability was further verified using ROC-AUC analysis across different attack classes. The obtained results demonstrate that combining machine learning models through heterogeneous ensemble learning improves multiclass intrusion detection performance while maintaining reliable classification of both benign and malicious network traffic, making the proposed framework suitable for smart-home edge computing environments.

In view of the growing threat landscape in smart home systems and the limitations of existing intrusion detection systems, this research contributes a scalable and effective solution suitable for deployment on edge computing devices. Future work will focus on validating the proposed framework on resource-constrained edge devices by analyzing inference latency, memory utilization, computational overhead, and energy consumption under realistic deployment conditions. The framework will also be extended through federated learning and self-supervised learning to support privacy-preserving model adaptation across distributed smart-home environments. In addition, Explainable Artificial Intelligence (XAI) techniques such as SHAP and LIME will be incorporated to improve the interpretability of attack predictions and assist security analysts in understanding model decisions. The proposed framework will also be integrated with the mitigation framework to enable automated response against identified attacks. Finally, the proposed framework will be evaluated across heterogeneous IoT environments using diverse communication protocols to further assessment.

All the Declarations and Statements

Author Contribution Statement

Abhay Kumar Ray: Conceptualization, Methodology, Data exploration, Software, Investigation, Formal Analysis, Validation, Visualization, Writing – Original Draft Preparation.
Rupak Sharma: Conceptualization, Methodology, Validation, Writing – Review & Editing, Project Administration.
Sunil Kumar Pandey: Methodology revision, Validation, Resources, Writing – Review & Editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement

The original Edge-IIoTset Cyber Security Dataset of IoT and IIoT Applications used in this study was obtained from the publicly available dataset developed by Mohamed Amine Ferrag et al. [19]. The dataset is available through the TechRxiv publication (DOI: 10.36227/techrxiv.18857336.v1) and the associated Kaggle repository. To ensure the reproducibility and transparency of this research, the processed dataset, including the selected features, preprocessed data, experimental results, confusion metrics, ROC curves, feature-importance plots, and other supplementary materials generated during this study, have been made publicly available through the following Google Drive repository:
https://drive.google.com/drive/folders/1KcDsXzWT0xM7tTOzoIwT7IwPnjFFUrYR?usp=sharing
These materials are freely accessible for academic research, verification, and reproducibility of the experimental results reported in this paper accessed on 30-06-2026

Ethical Declarations

This study does not involve human participants or animals.

Acknowledgements

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment, reporting and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

The authors declare that generative AI tools were used solely to assist with language refinement, grammar correction, and improvement of the manuscript's readability during the preparation of this work. The use of these tools did not influence the research design, methodology, model development, hyperparameter optimization, data preprocessing, experimental setup, data analysis, result interpretation, or the conclusions of the study.
All scientific and technical contributions, including the development of the proposed Multiclass Cyber Attack Classification using the Ensemble Top-3 Voting Model framework, experimental evaluation, performance analysis, and interpretation of results, were conceived, implemented, and validated entirely by the authors.

Abbreviations

The following abbreviations are used in this manuscript:

ANN – Artificial Neural Network
AUC – Area Under the Curve
BRF – Balanced Random Forest
CNN – Convolutional Neural Network
DDoS – Distributed Denial-of-Service
DL – Deep Learning
DoS – Denial-of-Service
DT – Decision Tree
Top-3 E-Voting- Top-3 Ensemble Voting Model
Edge-IIoT – Edge Industrial Internet of Things
IDPS – Intrusion Detection and Prevention System
IDS – Intrusion Detection System
IIoT – Industrial Internet of Things
IoT – Internet of Things
KNN – K-Nearest Neighbors
LightGBM – Light Gradient Boosting Machine
LSTM – Long Short-Term Memory
MITM – Man-in-the-Middle
ML – Machine Learning
ML-EdgeIIoT – Machine Learning Edge Industrial Internet of Things Dataset
MI – Mutual Information
RF – Random Forest
RNN – Recurrent Neural Network
ROC-AUC – Receiver Operating Characteristic–Area Under the Curve
SQL – Structured Query Language
SVM – Support Vector Machine
XGBoost – Extreme Gradient Boosting

Appendix A/B/C…, with appendix title

None.

References

[1] K. Maswadi, N. B. A. Ghani and S. B. Hamid, "Systematic Literature Review of Smart Home Monitoring Technologies Based on IoT for the Elderly," in IEEE Access, vol. 8, pp. 92244-92261, 2020, doi: 10.1109/ACCESS.2020.2992727.

[2] Yar, H.; Imran, A.S. Khan, Z.A. Sajjad, M. Kastrati, Z. " Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm". Sensors 2021, 21, 4932. https://doi.org/10.3390/s21144932

[3] Vardakis, G. Hatzivasilis, G. Koutsaki, E. Papadakis, N. "Review of Smart-Home Security Using the Internet of Things" . Electronics 2024, 13, 3343. https://doi.org/10.3390/electronics13163343

[4] S. Shaukat "Intrusion Detection and Attack Classification Leveraging Machine Learning Technique," 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 2020, pp. 198-202, doi: 10.1109/IIT50501.2020.9299093.

[5] Alani, M. M., Alshaer, J., Alrashdi, I., et al. (2023). XRecon: An explainable IoT reconnaissance attack detection framework using ensemble machine learning. Sensors, 23(11), 5298. https://doi.org/10.3390/s23115298

[6] R. A. Nugrahaeni and K. Mutijarsa, "Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification," 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), Semarang, Indonesia, 2016, pp. 163-168, doi: 10.1109/ISEMANTIC.2016.7873831.

[7] K. Zhang, K. Yang, S. Li, D. Jing and H. -B. Chen, "ANN-Based Outlier Detection for Wireless Sensor Networks in Smart Buildings," in IEEE Access, vol. 7, pp. 95987-95997, 2019, doi: 10.1109/ACCESS.2019.2929550.

[8] G. Sharkov, W. Asif and I. Rehman, "Securing Smart Home Environment Using Edge Computing," 2022 IEEE International Smart Cities Conference (ISC2), Pafos, Cyprus, 2022, pp. 1-7, doi: 10.1109/ISC255366.2022.9921912.

[9] A. K. Ray and A. Bagwari, "IoT based Smart home: Security Aspects and security architecture," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 2020, pp. 218-222, doi: 10.1109/CSNT48778.2020.9115737

[10] Rani Molla, "Amazon's Ring blamed hacks on consumers reusing their passwords. A lawsuit says that's not true." Reported on: Jan 18, 2020, URL: https://www.vox.com/recode/2020/1/17/21068703/amazon-ring-hacks-lawsuit-passwords last seen May ,2025

[11] Sandra E. Garcia, "Data Breach at Wyze Labs Exposes Information of 2.4 Million Customers" URL: https://www.nytimes.com/2019/12/30/business/wyze-security-camera-breach.html last seen may-2025

[12] Eray Eliaçık, "Wyze camera breach impacts 13,000 users' security" Published on: February 20, 2024 URL:https://www.dataconomy.com/2024/02/20/wyze-camera-breach-security/ last seen May-2025

[13] Rupanetti, D., Kaabouch, N.," Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities." Appl. Sci. 2024, 14, 7104. https://doi.org/10.3390/app14167104

[14] Ansari, M.S., Alsamhi, S.H., Qiao, Y., Ye, Y., Lee, B. (2020). "Security of Distributed Intelligence in Edge Computing: Threats and Countermeasures." In: Lynn, T., Mooney, J., Lee, B., Endo, P. (eds) The Cloud-to-Thing Continuum. Palgrave Studies in Digital Business & Enabling Technologies. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-41110-7_6

[15] Mutleg, M.L., Mahmood, A.M., Al-Nayar, M.M.J. (2024). "A comprehensive review of cyber-attacks targeting IoT systems and their security measures", International Journal of Safety and Security Engineering, Vol. 14, No. 4, pp. 1073-1086. https://doi.org/10.18280/ijsse.140406

[16] Hassen Sallay, "Designing an Adaptive Effective Intrusion Detection System for Smart Home IoT A Device-Specific Approach with SDN Deployment " (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 15, No. 1, 2024 URL: https://thesai.org/Downloads/Volume15No1/Paper_94-Designing_An_Adaptive_Effective_Intrusion_Detection_System.pdf

[17] Khraisat, A., Gondal, I., Vamplew, P. et al. "Survey of intrusion detection systems: techniques, datasets and challenges", Cybersecur 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7

[18] Jose Luis Hernandez-Ramos, Georgios Karopoulos, Efstratios Chatzoglou, Vasileios Kouliaridis, Enrique Marmol, Aurora Gonzalez-Vidal, and Georgios Kambourakis. 2025. "Intrusion Detection Based on Federated Learning: A Systematic Review", ACM Comput. Surv. 57, 12, Article 309 (December 2025), 65 pages. https://doi.org/10.1145/3731596

[19] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning." IEEE Access, 10, 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809

[20] Gueriani, A., Kheddar, H., & Mazari, A. C. (2025) "Adaptive cyber-attack detection in IIoT using attention-based LSTM-CNN models" arXiv. Doi:https://doi.org/10.48550/arXiv.2501.13962 .

[21] S. Sohail, Z. Fan, X. Gu, and F. Sabrina, "Explainable and optimally configured artificial neural networks for attack detection in smart homes," arXiv, May 2022.

[22] Malak Aljabri , Afrah Shaahid et. al. "IoT Attacks Detection Using Supervised Machine Learning Techniques," HighTech Innovation Journal, vol. 5, no. 3, Sept-2024 . DOI: https://doi.org/10.28991/HIJ-2024-05-03-01

[23] Ali. H. Farea and Kerem Kucuk "Machine learning-based intrusion detection technique for IoT: Simulation with Cooja," International Journal of Computer Network and Information Security, vol. 16, no. 1, pp. 1–23, 2024.

[24] A. Aldaej, T. A. Ahanger, and I. Ullah, "Deep Learning-Inspired IoT-IDS Mechanism for Edge Computing Environments," Sensors, vol. 23, no. 24, Art. no. 9869, Dec. 2023.

[25] Ismail, S., Dandan, S., & Qushou, A. (2025). "Intrusion detection in IoT and IIoT: Comparing lightweight machine learning

techniques using TON_IoT, WUSTL-IIoT-2021, and Edge-IIoTset datasets" IEEE Access. Advance online publication. Doi: https://doi.org/10.1109/ACCESS.2025.3554083

[26] Rieger, P., Chilese, M., Mohamed, R., Miettinen, M., Fereidooni, H., & Sadeghi, A.-R. (2023). ARGUS: Context-based detection of stealthy IoT infiltration attacks. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security '23) (pp. 4301–4318). USENIX Association. https://dl.acm.org/doi/10.5555/3620237.3620478

[27] Tanveer, M. U., Munir, K., Amjad, M., Alyamani, H. J., Bermak, A., & Ur Rehman, A. (2025). "LightEnsemble-Guard: An optimized ensemble learning framework for securing resource-constrained IoT systems" IEEE Access, 13, 101764–101781. Doi: https://doi.org/10.1109/ACCESS.2025.3576218

[28] Thanh Noi, P.; Kappas, M. "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery." Sensors 2018, 18, 18. https://doi.org/10.3390/s18010018

[29] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment." Sensors, 23(13), 5941. https://doi.org/10.3390/s23135941

[30] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," in IEEE Access, vol. 10, pp. 99129-99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[31] Ebo Jackson "Top 10 Python Libraries for AI and Machine Learning: A Comprehensive Guide with Code Examples" URL: https://medium.com/@ebojacky/top-10-python-libraries-for-ai-and-machine-learning-a-comprehensive-guide-with-code-examples-e89bfa1ab7df last seen July 2025

[32] A. K. Ray, R. Sharma and S. K. Pandey, "Integrated Machine Learning Approach for Attack Detection in MQTT-Enabled Smart Home Systems," 2024 International Conference on Emerging Technologies and Innovation for Sustainability (EmergIN), Greater Noida, India, 2024, pp. 351-356, doi: 10.1109/EmergIN63207.2024.10960978.

Authors' Profiles

Abhay Kumar Ray is research scholar in Department of Computer applications, SRM- Institute of Science & Technology, Delhi –NCR Campus, Modi Nagar, Ghaziabad, India He holds degrees of MCA (Master of computer Applications). He has published several papers in national and international journals and conferences and conducted 50+ workshop on cutting edge technologies in different institutes of India. Mr. Ray has experience of 16 years in both industry and academia, his area of interest is Internet of Things (IoT), Web Programming, Security System and Artificial Intelligence and Machine Learning.

Rupak Sharma is an Associate Professor and Head of the Department of Computer Applications at SRM Institute of Science & Technology (SRMIST), Delhi-NCR Campus. He completed his PhD in Computer Science & Engineering from Singhania University (Rajasthan) in 2012 and holds an MCA from Meerut Institute of Engineering & Technology. His research interests include artificial intelligence, machine learning, IoT, cloud computing, and mobile / ad-hoc networks. Dr. Sharma has published numerous peer-reviewed conference and journal papers, including in IEEE and Elsevier venues. Some of his recent works include AI-driven systems for physiological forecasting, IoT-based monitoring of industrial machines, and robotics for disinfection applications. He is passionate about mentoring students in emerging areas of computing and has guided both undergraduate and postgraduate research projects.

Sunil Kumar Pandey is currently working as Professor in Institute of Technology & Science with an experience of over 24+ Years in Industry and Academia and having interest in Cloud, Blockchain, Database Technologies & Soft Computing. He has been credited with 65+ Research papers (including SCI/ Scopus Indexed), 03 Book Chapters and 3 Books with reputed publishers including Springer, IGI, IEEE Xplore, River Press – Denmark, Wiley, Hindawi, Journals/ Conferences. He has been a regular author of Articles in different Print and Online Platforms including Interviews, Views and has published 11 edited volumes on different relevant themes of Information Technologies. He has been providing & coordinating training and consultancy to various reputed organizations including Indian Air Force and has conducted 25+ National/ International Conferences/ Summits/ Conclaves in association with AICTE, CSI, DST and other leading organizations. He has also conducted large number of FDP/Entrepreneurship Programs supported by AICTE/ DST/ UGC/ EDI etc.

How to cite this paper

Abhay Kumar Ray, Rupak Sharma, Sunil Kumar Pandey, "Multiclass Cyber Attack Classification in Smart Home IoT Networks Using Ensemble Machine Learning with the ML-EdgeIIoT Dataset", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.4, pp. 152-177, 2026. DOI:10.5815/ijwmt.2026.04.10