Behavior-Aware Access Control for IoT Networks Using Lightweight Machine Learning at the Gateway Level

Yaseen Alduwayl, Abdullah Alessa, Mounir Frikha
Department of Computer Networks and Communications,
College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa, 31982 Saudi Arabia

***Abstract*—The growing amount of heterogeneous devices with scarce resources is compromising the security of the Internet of Things (IoT), as they are more likely to adapt to a fixed and identity-based access control. Conventional security systems tend to assume that once a device is authenticated, the network may be exposed to credential theft, firmware, and insider abuse. In this study a behavior-sensitive access control solution is presented, which integrates lightweight supervised Machine Learning (ML) on the IoT gateway to provide dynamic authorization. Unlike the traditional passive intrusion detection models, the proposed framework uses a Supervised Random Forest model to process real-time statistical feature summaries in terms of mean, standard deviation, and sparsity of the IoT telemetry data. The method converts the output of anomaly detection directly into access (full, restricted or blocked) levels. The system was implemented on a Flask-based gateway and tested with ToN-IoT benchmark dataset. The results of the experiments show an anomaly-class recall of 0.9986 (99.86%) with 91,169 correctly detected attack and 125 false negatives among the 91,294 attack instances, for a security-oriented Zero Trust profile. As an example, when rounded to two decimal places, this value is 1.00, but the unrounded value is reported so as not to suggest 100% detection. The enforcement layer focuses on reducing risk and removes or filters out requests that were determined to be malicious or unauthorized in the scenarios. The architecture is designed to provide low latency through feature extraction and inference on the edge, which provides data privacy because telemetry processing is locally done without relying on the cloud.**

***Keywords*—Behavioral-aware access control; edge intelligence; gateway-based security; IoT security; intrusion detection; lightweight machine learning; random forest**

I. Introduction

The rationale behind the need of Secure Access Control in IoT is as follows.

Internet of Things (IoT) devices have greatly contributed to an increase in the attack surface of contemporary networks because of their rapid proliferation in the fields of smart homes, health care, industrial automation, and intelligent transportation. IoT systems are usually heterogeneous and resource-restricted gadgets that ceaselessly interact with gateways and cloud computing, usually devoid of human oversight. These systems allow automating and making decisions in real-time, but they also cause severe security threats in terms of unauthorized access, compromised devices, and insider abuse [1], [2].

Access control is a very important aspect of IoT security as it defines what devices can access network resources and services. Nevertheless, most current IoT implementations use static access control mechanisms that are based on rules and believe that a device will be trusted once it has been authenticated. This perception is progressively false in real-life situations where attackers are able to utilize stolen credentials, imitate legitimate devices, or utilize authenticated nodes and execute malicious activity [3]. Consequently, identity-based access control can not be effectively implemented by pure mechanisms alone to defend against the threats that are constantly changing IoT networks.

Recent research points to the importance of adaptive and situation-aware security measures that would continuously evaluate the behaviors of the devices, instead of using only initial authentication [4]. Specifically, dynamic access control that is applied to observed behavior can go a long way in minimizing the consequences of insider attack, compromised nodes, or privileged abuse. This impulse has led to the investigations on the behavior-sensitive access control systems which have incorporated intelligence directly at the IoT gateways to facilitate timely and effective enforcement of security.

A. Identity-Based Authentication Weaknesses

Sharing secrets, certificates, or tokens Identity based authentication mechanisms are common in IoT systems because they are very simple and do not require much computation. After an authenticated device is usually given some preset permissions, which do not change during its lifetime or session. Although this paradigm is effective, it implicitly presupposes that authenticated machines act in a benign manner, which is not always so in practice [5].

There are a number of security breaches which have shown that attackers could compromise identity-based controls through insecure credential management, device cloning, firmware vulnerabilities, or insider access [6]. In the context of that, a rogue device might be allowed to execute a full range of its permitted identity, yet still manifest malicious intent, e.g., stealing data, denial-of-service, and unauthorized access to resources. The traditional authentication systems find the detection of such behaviors as absolutely impossible since they do not observe the behavior of a device after authentication.

Also, statical authorization policies are not flexible enough to react to dynamic changes in the behavior of a device or network state. When permissions are received, they are not often removed without some form of manual intervention, so responses may be delayed and the attack period is very long

[7]. These restrictions have underscored the need to go beyond identity-based security models to adaptive mechanisms that keep reassessing the notion of trust on the basis of real-time behavioral observations.

B. Machine Learning and IoT Security

ML and IoT Security ML has become an important instrument to improve the security of the IoT, especially intrusion detection and traffic classification. Although deep learning models (e.g. CNNs and RNNs) can achieve high detection rates, they can be high in both latency and computational requirements, so that they are infeasible in resource constrained edge gateways. On the other hand, unsupervised schemes tend to confuse between complicated attack patterns and non-malicious anomalies. In order to overcome these drawbacks, this study capitalizes on lightweight supervised learning through a Random Forest classifier. The system is trained on the ToN IoT data and thus learns to differentiate between normal and attack patterns with high fidelity based upon statistical summaries of data streams. This method has a unique edge over deep learning in that it has low inference times that are conducive to real-time gateway enforcement, where detection coverage is also significantly better than that of basic threshold-based or unsupervised methods [8], [9], [10].

C. Research Scope and Organization of the Study

In this study, the design and test of a behavior-sensitive access control architecture of IoT networks is considered in which the authorization decisions are dynamically calculated according to real-time behavior of the devices. In contrast to the traditional methods which decouple access control and intrusion detection, the suggested system closely integrates lightweight ML with the authorization enforcement in the level of the IoT gateway. The low-latency and computational-overhead goals and deployability in the real-world IoT environment are the objectives of this design.

This study is devoted to the design, implementation, and evaluation of the behavior-aware access control architecture of the IoT networks. The major contributions of this work are:

- Benchmark-Driven Evaluation: Compared to the past literature, which uses synthetic data, the system is trained and assessed using the ToN-IoT dataset, which has solid performance in the presence of realistic heterogeneous sensor telemetry and network traffic.
- Statistical Feature Extraction: The addition of 5-dimensional feature extraction module, which is small in size, summarizes complex telemetry, facilitating the sparse feature processing in the edge.
- Zero Trust Enforcement: Zero Trust Enforcement: A Supervised Random Forest model is used to highlight the importance of a high anomaly recall and directly link the detection result to access-control enforcement. The robot model had a recall of 0.9986 when tested on the provided set of anomalies and the gateway only allowed restricted or blocked access when it detected high-level risk or unauthorized behavior.

The rest of this study will be structured in the following manner. Section II provides background on the IoT architectures, authentication models and gateway-centric security models. Section III will specify the problem, research goals, threat model, and contributions. Section IV explains how the selection of literature was performed and what shortcomings were found in the existing research. In Section V, the proposed system architecture and methodology are described. Section VI describes the experimental procedure, data and the results of the evaluation. Section VII and VIII address workflow issues, proposed workflows, and a comparative analysis. In Section IX, security is more widely discussed and more practically implications are explained, and the future research directions are discussed in Section X. Lastly, Section XI is the conclusion of the study.

II. Background and Context

A. IoT Architecture and Security Problems

The architecture of the Internet of Things (IoT) is most commonly divided into three basic layers, namely, the perception layer (sensing devices), the network layer (communication and data transfer), and the application layer (data processing and service delivery). In the real-world implementation, a second layer of gateway is used as an intermediary between the limited IoT devices and external networks or cloud services. The functions of this gateway include protocol translation, aggregation of data, authentication, and policy implementation [11].

As shown in Fig. 1 the layers of IoT architecture and outstanding the gateway between each layer.

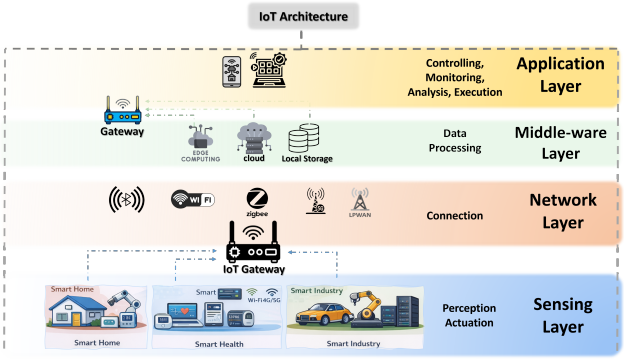


Fig. 1. IoT architecture layers (Sensing, Network, Middleware, and Application).

Although IoT has great architectural advantages, it presents serious security issues because it is heterogeneous in devices, has fewer computing capabilities, and it has many connections. Numerous IoT devices do not have the memory and processing power required to implement robust cryptographic algorithms or sustained security supervision and thus are targets when it comes to attacks [2]. Moreover, IoT networks are frequently installed in unmanaged or malevolent spaces and this raises the risk of physical interference, devices intrusion, and unauthorized access.

The other significant problem is the dynamic nature of IoT devices. The patterns of communication between the device

can vary with time and based on the circumstances of its operations, the frequency of requests and the services accessed. Conventional security systems cannot handle these types of behavioral differences, resulting in too much false warnings or unnoticed malicious breaches [3]. These issues indicate a necessity of a responsive, behavior-sensitive security system that functions near the network border [12].

B. IoT Authentication and Authorization Models

IoT security has some basic elements in authentication and authorization. Authentication is used to check the identity of a device and authorization is used to check what a device, which has been authenticated can do. In the IoT, the most commonly used authentication tools are pre-shared keys, digital certificates, and lightweight token-based authentication [5]. Such methods are also normally optimized around low overhead but exclusively identity verification.

In IoT settings, the process of authorization is typically realized with the help of such a static access control model as Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC). Although these models are flexible in terms of the definition of permissions, they are generally based on predefined policies that do not change once authenticated [7]. Consequently, after a device has been authenticated and authorized then it is implicitly trusted until the end of its access session.

This assumption of static trust opens a big gap of security. In case a device is hacked after authentication, the traditional methods of authorization can not identify and react to the malicious activity in real-time. It has been observed that insider attacks and misuse of credential are some of the hardest threats to be detected by identity-based mechanisms alone [4]. Thus, continuous authorization models with dynamically re-evaluated trust according to dynamic behavior and not based on an identity are gaining interest.

C. IoT Network Intrusion Detection Systems

The topic of the Intrusion Detection Systems (IDS) has gained widespread attention as the method that helps strengthen the level of IoT security by detecting any malicious activities and detecting any unusual patterns of traffic. IoT IDS solutions can be broadly classified as signature-based, anomaly-based and hybrid solutions. These are anomaly-based IDS based on ML, which have become widely known thanks to their capability to identify unknown and zero-day attacks [8].

Recent studies on the application of deep learning models like convolutional neural network (CNNs), recurrent neural network (RNNs), and autoencoders to IoT intrusion detection have been examined. Although such models can be highly accurate in detecting objects, they can be expensive to implement in the IoT gateway because they often require heavy computational resources and GPU acceleration, making them unsuitable for deployment on resource-constrained IoT gateways [9]. In addition, a cloud-based IDS architecture creates extra latency and privacy issues since information is continuously being sent.

The other weakness of majority of IDS solutions is that they do not integrate with access control systems. Authorizations are normally made in real time without paying direct attention to detected anomalies which are usually logged or reported. Such a division between detection and enforcement slows down mitigation efforts and makes IDS as a whole less effective at attack prevention [10]. These drawbacks drive the adoption of lightweight anomaly detection being directly embedded into access control processes at the gateway level.

D. Gateway-Based Security Architectures

A new architecture of security Gateway based security has been developed as a viable way of implementing security policies in IoT networks. Centralization of security capabilities at the gateway allows devices with limited resources to avoid any complicated computations but receive extra protection [1]. On behalf of networked devices, gateways are able to monitor traffic, impose authentication, apply access control policy, as well as carry out anomaly detection.

Recent works support the concept of edge intelligence, in which ML models are executed on the gateway to examine device behavior on the device side. This technology lowers the latency and enhances scaling and maintains privacy of data over cloud-centric solutions [13]. Small ML models, including tree-based classifiers, unsupervised anomaly detection algorithms, etc., are especially applicable in this scenario because of their low computational footprint [14].

In spite of these, most of the current gateway-based security solutions continue to view intrusion detection and access plate as two distinct elements. Anomalies can be identified but authorization is not dynamically reconfigured according to behaviour identified. This vulnerability restricts the capacity of the gateways to be responsive towards the threats. Thus, the exploration of behavior-aware ML as a part of the authorization process at the gateway is an important milestone to more resilient and flexible IoT security systems.

III. Problem Statement, Objectives and Contributions

A. Problem Definition

Although authentication and access control mechanisms have been widely adopted in the IoT networks, the available solutions are still mostly stagnant and identity-based. After a device authenticates successfully with such credentials as pre-shared keys or tokens, it is normally issued with fixed permissions, which are not affected by any future actions. The given security model does not consider the possible cases of authenticated devices turning malicious because of credential theft, firmware attack, or insider abuse [5], [6], [15].

Fig. 2 illustate the procedure of risk that occured on IoT environment. Recent studies in the field of IoT security have paid much attention to the development of intrusion detection based on ML methods. Although such systems may be used to detect abnormal traffic patterns or malicious activities, they usually act as passive surveillance devices which do not have the option of enforcing these activities. Anomalies are captured and logged or reported, but do not happen instantaneously to influence authorization measures, leading to slow or inefficient mitigation [10], [16].

Moreover, a vast majority of AI-based IoT security algorithms are based on computationally intensive deep learning

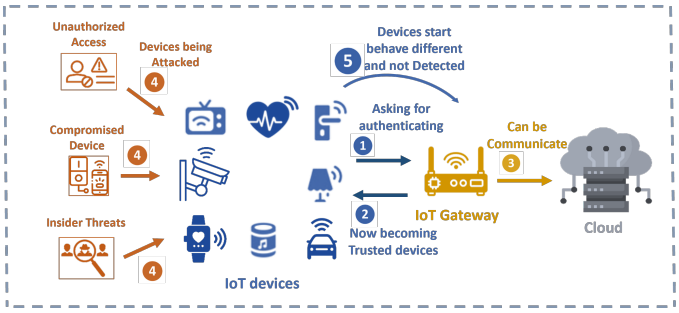


Fig. 2. Security challenges and behavioral anomalies in IoT environments.

architecture implemented on clouds. These methods have high latency, high bandwidth, and privacy risks, which complicate their practical use in the constraints of the IoT gateway resources [9]. Consequently, practitioner-oriented IoT deployment is facing a critical variation between behavioral threat identification and enforcement of dynamic access control [17].

B. Research Objectives

The main goal of this study is to develop and test a behavior-aware access control architecture of IoT network, which can dynamically modify the authorization decisions influenced by the online device behavior. However, in contrast to the traditional identity-related frameworks, the presented system keeps track of post-authentication actions and denies or blocks access in case of anchoring suspicious activity.

In order to accomplish this objective, the study aims at the following specific objectives:

- To examine the drawbacks of the in situ authentication and authorization systems in IoT-based systems and illustrate the fact that they fail to counteract insider and post-authentication attacks.
- To develop a lightweight ML-based behavioral analysis gateway-centric security architecture, not based upon cloud-based processing.
- To derive effective behavioural characteristics in the interaction of IoT devices that may be used to discriminate between normal and malicious activity successfully.
- To deploy a lightweight anomaly detection model to be used in real time at the level of an IoT gateway.
- To closely bind the results of anomaly detection to access control policies, so as to allow a response to be taken as soon as possible, that is, restricting or revoking access.
- To assess the proposed framework regarding the detection accuracy, the response time and the computational overhead.

All these are objectives to help close the divide between detection and enforcement but still be practical to the real-world IoT systems.

C. Assumptions and Threat Model

The study has its specific assumptions and realistic threat framework as per real-life IoT applications. The assumption is that the initial IoT device authentication with authentic credentials is never unsuccessful, and the gateway is not compromised. The communication between the devices and the gateway will be assumed to be secured with the help of the standard secure channels, including TLS [18].

The threat model takes into account the attackers who can interfere with the IoT devices once they have been authenticated. These enemies can use stolen credentials, duplicate legitimate nodes, or even inject malicious code on authorized nodes. The attacker can target to cause excessive traffic, misuse authorized services, gain unauthorized access to endpoints, or destabilize the normal processes of the gateway [3], [19].

The behavior that can be referred to as denial-of-service, abrupt fluctuations in the frequency of requests, atypical access patterns, and departures of learned behavioral patterns are all viewed as malicious activity indicators. Unlike rigid signature-based systems, the proposed framework utilizes a Supervised Random Forest model trained on the ToN-IoT dataset. By analyzing statistical feature summaries (such as mean, standard deviation, and range) rather than specific attack payloads, the system can generalize from the training data to detect malicious behaviors that statistically resemble known attack profiles, even if the specific execution varies.

This study is out of scope of physical attacks on the gateway, complex adversarial ML attacks, and large scale distributed botnets. The emphasis is still on the efficient behavior-aware authorization within the computational limits of one IoT gateway as shown in Fig. 3.

D. Summary of Contributions

This study brings the following main contributions to the study of the IoT security:

- Behavior-based Authorization Framework. It presents a new approach to access control whereby the authorization decision is continuously updated depending on how the device has acted as opposed to just based on identity alone.
- Lightweight ML on the Gateway. The system combines supervised, lightweight ML model, which can be applied to the IoT gateway, without depending on deep learning-based solutions on the cloud.
- Close Interconnection of Detection and Enforcement. In comparison to the traditional IDS methods, identified anomalies directly cause access control measures, allowing to restrict or revoke device privileges immediately.
- Practical and Low-Latency Design. The suggested scheme will be capable of working with low computational load and, therefore, suitable within a real-time IoT setting.
- Experimental Validation. An emulated IoT network is executed to test the framework using the ToN-IoT benchmark dataset to ensure fidelity. This allows the

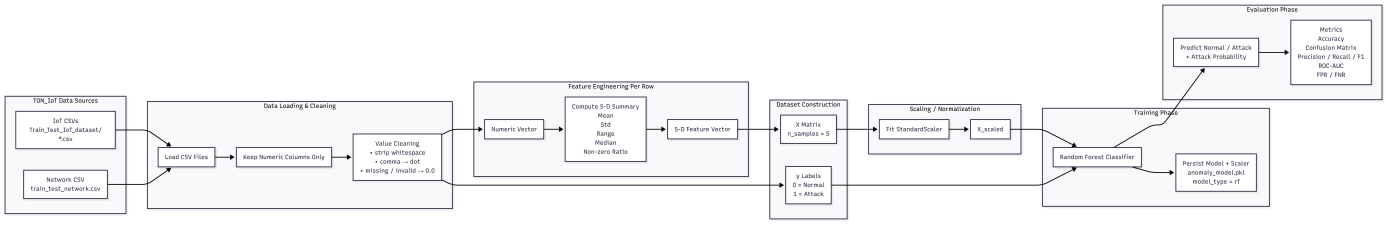


Fig. 3. Overview of the proposed Random Forest classification pipeline on the TON-IoT dataset.

system to be evaluated against realistic, heterogeneous sensor telemetry and actual network traffic patterns, covering diverse attack cases and performance parameters.

All these contributions cover the most important shortcomings in current IoT security offerings and show a realistic way forward to behavioral and adaptable access control.

IV. Literature Review Methodology

A. Literature Search Strategy

A literature search was carried out, which was systematic, to determine the current and topical studies in the field of IoT security, access control, intrusion detection, and ML-based anomaly detection. The main sources of literature searched were peer-reviewed journal articles and high-quality conference proceedings from 2020 or later to keep the review up to date with the latest research in IoT security. The earlier studies have only been added if they have been used as a basis for IoT architecture, historical developments in intrusion detection, or gateway-level security concepts.

Earlier publications like [11] and [20] and [21] are therefore not included as part of the current evidence base, but are kept as background or reference material. This distinction will not exclude seminal works, but will keep the focus on recent literature of IoT security.

A total of major academic databases were utilized such as IEEE Xplore, Elsevier ScienceDirect, SpringerLink, ACM Digital Library, and MDPI. The search queries were built based on the combination of terms, including the following: IoT security, access control, behavior-based authorization, intrusion detection, ML, anomaly detection, gateway security and edge intelligence. Refining took place with the help of the use of the Boolean operators to exclude irrelevant studies [22], [20].

Title and abstract screening were used to filter the first search results in order to make them relevant. A shortlisting was followed by full-text reviews on shortlisted papers to determine its congruency to the research interest of behavior-aware access control and lightweight ML in IoT settings.

B. Inclusion and Exclusion Criteria

There were clear inclusion and exclusion criteria that ensured that the literature review process was consistent and of good quality. The papers were to be included in case they met the following criteria:

- Published in 2020 or later, or published earlier but retained as a foundational work directly relevant to IoT architecture, intrusion-detection development, or gateway-level security.
- Specialized in IoT security, admission control, intrusion detection, or ML-based behavioral analysis.
- Security measures to be provided at the IoT, edge or gateway levels suggested or considered.
- Offered experimental assessment based on data sets, simulations or testbeds.

Articles were filtered out in case they fell into these categories:

- Published before 2020 and not required as a foundational or contextual reference for the architecture, threat model, or comparative baseline.
- Only concerned with cloud-based security and not with edge or gateway limitation.
- Only cryptographic authentication, without behavior analysis after authentication.
- Absence of experimental assessment or based entirely on theoretical discourse.
- Specific non-IoT networks like conventional enterprise or data center networks.

The filtering process ensured that the studies chosen were recent as well as relevant to the problem under investigation in this research.

1) Classification of selected studies: The chosen papers were divided into four broad groups in relation to the focus of research and the methodology applied:

- Models of Identity-Based Authentication and Authorization. These research works pay attention to the conventional authentication procedures like certificates, tokens, or attribute-based access control. Although they are useful in identity verification, they typically presuppose immobile trust following authentication and lack a concern about post-authentication conduct [7].
- Intrusion Detection Systems based on ML. The works in this category involve the application of supervised or unsupervised ML to identify malicious traffic or anomalous activity in an IoT network. Most of the solutions highlight the accuracy of detection but lack enforcement mechanisms [8], [9].

- Security Architecture based on Edge and Gateway. These researches suggest the use of security intelligence at the edge layer or IoT gateway to minimize latency and computation load. Nevertheless, the vast majority of practices consider intrusion detection and access control as independent entities [13].
- Behavior-Aware and Context-Aware Security Models. A more limited group of papers examines behavioral-based trust or adaptive security measures. Although these methods advance towards persistent authorization, they are usually based on complicated models or not include practical implementation specifications that can fit on lightweight gateways [4] [23].

This classification offers a systematic overview of available literature and the manner in which various methods deal with partial security concerns of IoT.

C. Limitations Revealed in Previous Studies

The discussion of chosen studies shows that there are a number of limitations, which are common and trigger the proposed research. To begin with, majority of identity based access control systems do not identify malicious activities that occur following authentication, which makes the IoT networks susceptible to insider and credential related attacks. Second, most ML-based IDS systems are detection-only and lack access controls which leads to slow or manual mitigation.

Moreover, much of the current literature is utilizing deep learning models that are implemented in the cloud, which adds latency, privacy implications and very high computational costs that cannot be implemented on the gateway. Even the gateway-centric solutions do not take into consideration the tight coupling between behavioral analysis and authorization enforcement.

Lastly, most of the proposed systems are evaluated solely through offline analysis of static datasets, decoupling detection from actual enforcement. This restricts their applicability to real-time IoT systems, where authorization decisions must be made instantaneously. The combination of these limitations shows that it is necessary to have a lightweight and behavior-aware access control system that is integrated directly into the IoT gateway to combine detection with immediate authorization decisions.

D. Related Work

The most recent studies on IoT network protection have discussed ML, edge computing, and access control models. Although such studies have enhanced the level of detection and efficiency of the system, there is still a fundamental gap of combining real-time behavioral monitoring with adaptive access control per gateway. This field has 20 recent contributions, the discussion of which is critically compared below.

E. Machine Learning-Based Intrusion Detection in IoT

In [24] the authors have reviewed and proposed some of the ML techniques to optimize the IoT in regard to security, authentication, access control, and malware detection. They also discussed spoofing, jamming and eavesdropping and the goal was to prove that ML may be utilized instead of conventional security in dynamic settings. The study is about generic IoT applications and the examples of smart grids and vehicular networks. Regarding methodology, they used game-theoretic modeling and simulation to evaluate different strategies in ML such as Q-learning and supervised learning. They used performance measures as detection accuracy, false alarm, utility of a system and demonstrated that Q-learning could reduce authentication error by approximately 10-15 percent compared to traditional schemes in their simulation. The key findings are that, though ML can be utilized to improve security, it introduces overhead, which can be challenging to handle using low-power devices. This is not a fully appropriate solution to your provided problem as, though it deals with ML to provide access control, it relies on reinforcement learning (Q-learning) and game theory models, which are computationally expensive and can require long convergence times, which is not ideal in resource-bounded gateway processing.

In this study [12], the authors suggested a framework to identify cyberattacks of smart city apps with the help of different ML algorithms. They have tackled the problem of security of smart cities, i.e., detecting anomalies of sensor and communication systems. It is an intelligent urban application. Their approach was to train and test classifiers such as Decision Trees, Random Forest, and k-NN on a dataset they created. They used accuracy, precision, recall, and F1-score and noted that Random Forest achieved 99.9% accuracy with 0.1% false positive. The findings show that Random Forest is powerful with smart city traffic. They used a dataset created by a simulated smart city testbed with 362,377 samples. This solution uses Random Forest but is not suitable to your problem because it is not an active IDS and does not tie detections to real-time enforcement actions at the gateway level.

The author in the article of [9] proposed a Deep Learning-based hybrid Intrusion Detection System (IDS). Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) were employed by them to identify sophisticated attacks. They worked on improving the ability to detect attacks like DDoS and ransomware. Particle Swarm Optimization (PSO) was used to optimize the model. They evaluated their model with accuracy, precision, recall and F1-score with 99.64 Bot-IoT and 99.45 TON_IoT dataset. The hybrid CNN-LSTM model was more effective compared to the single models. This version is inappropriate as it relies on a computationally-intensive Deep Learning architecture (CNN + LSTM) which uses processing power and cannot be implemented on an IoT gateway with limited resources. Furthermore, it is a detection system (IDS) only, which lacks an access control mechanism that provides an automatic implementation of blocking policies.

In [17], the authors introduced an ensemble Intrusion Detection System (IDS) that uses flow statistical features to detect malicious traffic. They concentrated on identifying sophisticated attacks in the IoT networks by combining classifiers (Naive Bayes, Decision Tree, and Artificial Neural Network). They produced statistical features and employed a voting - based ensemble classifier. They measured the performance with Accuracy, False Alarm Rate, and Detection Rate reaching 99.5% detection rate with low false alarms. The findings indicated that ensemble method minimized errors when compared

to single classifiers. This is also inappropriate as it employs an ensemble, which is more computationally expensive than one Random Forest model and is still a passive IDS, which does not enforce anything in real-time.

In [10], the authors applied an ensemble ML approach to detect DDoS attacks in the IoT network. They trained algorithms (Random Forest, Extra Trees, XGBoost) and voted on them with a soft voting classifier. They compared the performance based on Accuracy, Precision, Recall, and F1-score with a 99.7 accuracy and 0.99 F1-score. The findings revealed that ensemble approaches are more consistent than individual models. This proposal is not applicable because it concentrates only on DDoS and has no behavior-aware mechanism of access control to be used in real-time.

In the study by [3], the authors performed the review of Intrusion Detection Systems (IDS) in IoT with the focus on Multi-Access Edge Computing (MEC) and ML. They stressed the idea of moving security functions to the edge in order to minimize latency and bandwidth. They used a method of a systematized literature review that categorized IDS based on the deployment and detection method. Some of the performance measures they reported included performance in terms of detection and processing time. The results indicate that MEC-based IDS can decrease the response time but might not be scaled to heterogeneous IoT data. The article is inappropriate as it is a review of architectures and does not give an actual system or a behavior-conscious access control implementation.

The authors of the article in [15] proposed an Intrusion Detection System (IDS) using a supervised ML to detect malicious behavior in IoT networks. They trained classifiers (Random Forest, Decision Tree, k-NN and Naive Bayes) with the UNSW-NB15 dataset. They assessed the performance by Accuracy, Precision, Recall, and F1-score with the accuracy of the performance in the case of Random Forest and Decision Tree being 98.9% and 99.5%. The findings revealed that tree-based models are correct and less expensive to compute. This is not a solution to you because it is a passive IDS which lacks automated access control measures at the gateway.

The authors of the article in [4] evaluated Intrusion Detection Systems (IDS) based on the usage of ML in Ioot. They compared the classifiers: Random Forest, SVM and Ensemble methods between the datasets. They employed Detection Rate, False Positive Rate and Accuracy with ensemble techniques showing detection rates greater than 99%. The findings indicated that the performance is dependent on the dataset. They were used in NSL-KDD, CIDDS-001 and UNSW-NB15. This solution is not applicable to your problem as it is benchmarking research and lacks the integration of detection and real-time authorisation.

In [25] the authors enhanced the Isolation Forest algorithm by incorporating it with the k-means clustering algorithm to identify anomalies. They followed a two-step process, in which k-means splits data into clusters and Isolation Forest identifies outliers. They tested the performance in terms of AUC, accuracy and processing time with better AUC than the traditional iForest. The findings indicated a better detection of local anomalies at the expense of increase in computational cost. They were synthetic datasets and KDD Cup 99. This is inappropriate because it is an unsupervised anomaly detector and it does not offer dynamic access control.

F. Deep Learning and Network Traffic Analysis in IoT

The authors in [26] developed a Deep Learning model to identify botnet attacks on healthcare IoT networks. A Deep Belief Network (DBN) that was trained on the Bot-IoT dataset was used. They assessed the performance based on accuracy, precision, recall, and F1-score, obtaining an accuracy of 99.25% and a recall of 98.8%. These findings indicated that Deep Learning has an application in monitoring botnet activity. This does not fit well as it involves Deep Belief Network, which is computationally intensive and cannot fit a resource-constrained IoT gateway.

The authors of the article in [7] studied the detection techniques of the IoT botnets. They contrasted signature-based, anomaly-based and DNS-based techniques. It was a comparison literature review procedure. They used detection rates greater than 99 percent and false positive rates to evaluate methods. The results indicated that ML methods are prospective but they are very expensive to process. They mentioned datasets like N-BaIoT. This is not a solution to your problem because it is a survey and lacks an integrated mechanism of access control to automatically block at the gateway.

These authors in [6] provided a case study of the Mirai botnet that led to DDoS attacks. They evaluated the Mirai scan, infect and organization of IoT devices to attack with more than 1 Tbps. They analyzed the source code and pattern of attacks of Mirai. They announced the size of attacks (e.g., 620 Gbps) and compromised devices (more than 600,000). The results revealed the use of weak default credentials. This solution is inappropriate as it is an analysis of an attack and not a defense ML or gateway-based solution.

G. Surveys and Emerging Technologies in IoT Security

The authors in [27] surveyed the problem of security and forensics in the IoT ecosystem. They categorized threats in each of the layers of IoT and addressed the problem of data privacy, authentication, and standardization. They looked at the available literature and offered security requirements in form of a taxonomy. The results revealed the gaps in the IoT forensics and difficulties associated with the resources limitations of devices. The study is inappropriate as it is a theoretical survey and lacks practical access control solution based on ML or gateway.

The researchers in [28] have carried out a survey to categorize security challenges and solutions in the IoT. They discussed the vulnerabilities, data privacy, and authentication in various areas of IoT. They were able to review over 100 articles which lacked experimental analysis or performance measurements. These results highlighted the importance of light-weight security as a result of limited resources. The study is not appropriate as it is theoretical and lacks a real-world architecture and behavior-based access control mechanism.

In this study [29], the authors found out the use of Federated Learning (FL) along with IoT to address centralized ML constraints. They solved the problem of data privacy, latency and bandwidth because they allowed distributed training on

edge devices. The areas included in the study were IIoT, IoV, healthcare, and smart cities. The research design was literature review that categorized the approaches into resource management, privacy, and security. The results indicated that FL enhances privacy but lacks in data heterogeneity and device limitations. This solution cannot be used as it has to be decentralized in terms of coordination and is not compatible with a centralized lightweight gateway-based solution.

The authors of the study in [30] reviewed the application of Federated Learning (FL) to enhance cybersecurity in IoT. They tackled data privacy and centralized processing through collaborative detection without transfer of raw data. The research design was a theoretical analysis of FL architectures. They talked of accuracy, communication overhead, and convergence time, with comparable accuracy to centralized models with less bandwidth. The results indicated weaknesses like susceptibility to poisoning attacks and expensive communication. Your problem is not well addressed using this solution as the solution is based on a distributed training process as opposed to a centralized lightweight gateway-based solution.

In the study by Li and et al. [13] they made a survey of Edge Computing within the Internet of Everything (IoE) was conducted. They also resolved the problem of latency, bandwidth, and privacy of cloud-based processing, demonstrating that edge computing decreases latency. The research design was a literature review with no experimentation. The results indicated that edge computing lowers the latency but presents a set of problems including resource limitations and security control. This is not helpful to your problem as it is a theory overview and does not offer a practical access control algorithm or behavior-based security.

The researchers in [1] performed an IoT security, privacy and trust review. They talked about scalable middleware gaps, secure data management, and authentication gaps. A literature review that categorizes security challenges was used as the methodology. The results revealed the absence of lightweight and single security models. This does not apply as it is a descriptive survey and does not give a practical implementation to be applied to real-time blocking at the gateway.

H. Access Control and Authentication Mechanisms in IoT

Within the framework of this general survey [31], The researchers analyzed the use of blockchain with IoT to enhance access control. Among the problems that they solved were centralized control and single point of failure, which sought to have a decentralized structure. The research design was a survey that reviewed and classified blockchain-based solutions. The results of the study was revealed that blockchain enhances security but bring with latency and computational overhead. Your problem does not fit this solution, as it is both a heavy and decentralized scheme with delays, and is not suitable to a lightweight real-time gateway design.

The researchers in [32] delivered a lightweight authentication system to smart home devices with resource-constrained capabilities. They dealt with big computational cost and attacks like replay attacks and man-in-the-middle to facilitate secure communication at low energy consumption. The security analysis and performance comparison methods were used. They compared the cost of computation, the cost of communication and the energy consumed, which demonstrated a lower cost at the expense of security. It is not a valid solution as it remains an authentication protocol and does not deal with post-authentication behavior or compromise devices identification.

The researchers in [33] have developed an access control model of Electronic Healthcare Systems based on trust. They dealt with the dynamic control of access by calculating the trust score using user behavior. The technique involved user behaviors to modify trust and access privileges. They tested the success rate of access, time to compute trust and penalties to behavior. The findings demonstrated that the model is able to adjust to changes in behavior. This does not fit as it is human user-based and complicated computation of trust, rather than device behavior at a real-time IoT gateway.

I. Lightweight and Edge-Based Security Solutions for IoT

As the researchers state [34], the multimedia traffic of IoT is suggested to be provided by the so-called media-aware security architecture. To solve this trade-off between security and QoS they adjusted security according to delay and jitter. The testbed methodology involved a delay, jitter, packet loss and throughput simulation testbed. The findings demonstrated better adaptive security QoS. This does not apply as it prioritizes multimedia traffic and sacrifices security in favor of QoS rather than providing a strict security enforcement of the general IoT devices.

The researchers in [21] suggested an inexpensive flow-based security architecture of smart home IoT devices. They modeled the behavior of devices based on network traffic without deep packet inspection. The method involved actual traffic of the IoT devices to create typical behavior patterns. They tested the accuracy of classification and the accuracy of anomaly detection. The findings indicated that IoT devices are easy to profile and their traffic is predictable. This is not a solution as it is a passive monitoring tool and does not give automated access control to prevent malicious devices (Table I).

J. Summary of Gaps Identified

The current literature is characterized by the following limitations:

- The majority of solutions revolve around detection and not prevention.
- ML-based IDS are not always coupled with access control.
- IoT gateways cannot support heavy deep learning models.
- Behavior-based authorization that is real-time is poorly explored.

These loopholes are the direct inspiring factors of the proposed behavior-aware access control framework that brings together the lightweight ML with real-time enforcement of authorization at the IoT gateway tier.

TABLE I. ANALYSIS OF RELATED STUDIES

Ref	Year	ML/DL Models	Dataset	Results	Key Findings	Limitations
[4]	2021	Decision Tree, Random Forest	Smart Home / IoT Traffic	High accuracy; offline analysis	Effective for smart home and city profiling	Offline analysis only; no real-time integration
[3]	2021	LSTM (Deep Learning)	IIoT Traffic	High detection accuracy	Suitable for large-scale IIoT	Cloud dependency; high latency
[10]	2021	Isolation Forest (Unsupervised)	IoT Traffic	Effective anomaly detection	Works well for unseen DDoS attacks	Independent IDS; cannot prevent attacks
[9]	2021	CNN + RNN (Hybrid DL)	Healthcare / Industrial IoT	High detection rates	Good performance in heterogeneous IoT	High complexity; no adaptive revocation
[13]	2022	Lightweight ML + Edge Intelligence	IoT Edge Traffic	Reduced response time	Scalable edge-based security	Detection and access control are separate
[23]	2022	Statistical Trust Scoring	Healthcare IoT	Improved authorization decisions	Trust-based decisions enhance insider resistance	Historical data only
[15]	2021	Clustering-based ML	IoT Traffic	High flexibility	Effective for unknown anomalies	High false positives; no ACL feedback
[30]	2021	Edge Traffic Filtering	IoT Gateway Traffic	Reduced backend load	Efficient local filtering	Signature-based; not adaptive behavior
[36]	2022	Distributed Edge Authorization	IoT Edge Networks	Lower latency	Effective low-latency access control	Static policies; no behavioral scoring
[40]	2020	Lightweight Authentication	IoT Devices	Preserved battery life	Efficient initial handshake	No post-authentication anomaly handling
[17]	2021	Ensemble ML Models	IoT Traffic	Increased detection accuracy	Reduced classification errors	Computationally heavy
[16]	2020	Distributed Deep Learning	IoT / Botnet Traffic	Improved botnet detection	Localized edge learning benefits detection	Complex models; high compute requirement
[1]	2020	Survey / Theoretical Analysis	N/A	Identified security challenges	Highlights trust and privacy gaps	No implementation/testing
[2]	2020	Deep Learning + Big Data	IoT Traffic	Improved detection metrics	Useful for large-scale IoT traffic	High resource usage; cloud reliance
[5]	2021	ML-based Botnet Detection	IoT / Mirai Traffic	Accurate botnet detection	Recommends future research directions	Reactive approach; no adaptive access control
[6]	2021	ML + MEC Framework	IoT Traffic	Efficient anomaly detection	Edge-level anomaly detection	Limited dynamic access control integration
[7]	2021	Botnet Pattern Analysis	IoT Networks	Detection of Mirai variants	Highlights IoT network threats	Detection only; no adaptive enforcement
[8]	2021	CNN + RNN Supervised Learning	IoT Traffic	Improved detection	Good for heterogeneous IoT	Centralized; heavy computation
[11]	2020	Cluster-based Isolation Forest	IoT Traffic	Efficient anomaly detection	Works without labeled data	No access control integration
[12]	2022	Lightweight ML at Edge	IoT Edge Traffic	Reduced latency; privacy preserved	Edge-level enforcement feasible	Detection and access control disconnected

V. METHODOLOGY

A. System Architecture Overview

The suggested system is based on the gateway-centric security architecture where all the interactions with the IoT devices are facilitated by a centralized gateway. The gateway can be used as the point of enforcement of authentication, behavior monitoring, anomaly detection, and access control. In this design, the computational load on IoT devices with limited resources is reduced, and real-time security decisions are made near the network edge.

It is composed of four closely coupled modules: 1) authentication, 2) behavioral monitoring, 3) lightweight ML-based anomaly detection and 4) access control enforcement. Devices are constantly checked after successful authentication and their behavior is assessed with the help of the lightweight anomaly detection model. ML model output has a direct impact on making decisions on authorization, which enables the system to enable and limit or deny access dynamically as shown in Fig. 4.

Designing these elements at the gateway can help the system to avoid the latency and privacy problems that come with cloud-based intrusion detection but at the same time, the detected anomalies are immediately converted to enforcement measures.

B. IoT Network and Gateway Model

The IoT network model presupposes the communication between a number of heterogeneous devices and a single gateway through the REST-based communication. Devices can initially register and identify to the gateway through a unique device identifier and a shared secret. When a device has been successfully authenticated, it is first assigned full access permissions and behavior monitoring is then initiated [28].

The gateway stores state data of each device, such as authentication status, access level, and history of recent interaction. The gateway becomes the only access control decision maker since all device requests are funnelled through it. This model of a centralized gateway is representative of a real-life IoT implementation that is usually used in smart home, industrial, and healthcare settings [1], [24].

The gateway also determines the various access levels, which include full, restricted, and blocked and are related to a predetermined list of authorized endpoints. This allows the capabilities of devices to be easily controlled to fine grains and also enables the gradual decrease of capabilities instead of abrupt disconnection to enhance system robustness and usability.

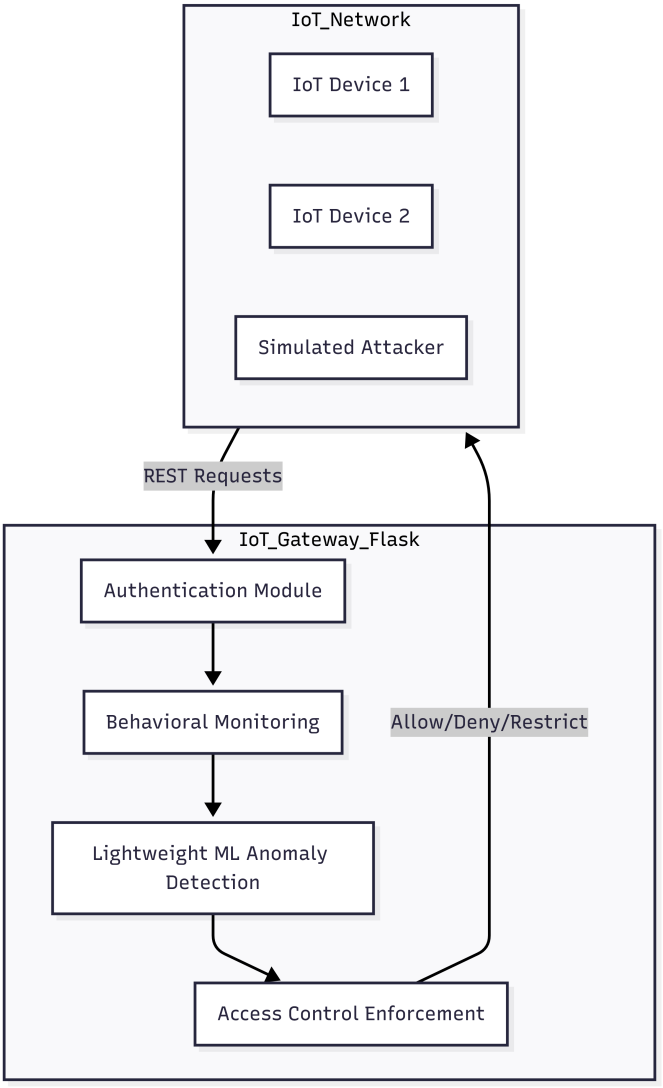


Fig. 4. Gateway-centric behavior-aware access control architecture.

C. Extracting Behavioral Features

In order to deal with the heterogeneity of IoT traffic, in which different sensors (e.g., thermostats, GPS trackers, Modbus devices) produce radically different data payloads, the system no longer inspects raw payloads, instead promoting statistical summarization. A fixed dimension 5-dimensional feature vector is obtained, such that each row or data window processed by the gateway is described by the current behavioral state of the device. [35], [36].

These features are: As can be seen in the analysis of the ToN-IoT dataset, they are:

- Mean: The arithmetic mean of the numeric values in the telemetry row which gives a baseline of the signal magnitude.
- Standard deviation (std): Indicates the degree of volatility or stability in data stream. Large deviations can be associated with unpredictable behavior as found in flooding attacks or malfunctioning sensors.
- Range: The maximum value less the minimum value and is used to measure the spread of the data.
- Median: The center of interest of the data used to reduce the skewing effect of the outliers.
- Non-zero Ratio: Fraction of the values in the row that are non-zero. This sparsity measure is important in detecting attacks that either flood the fields with zeros or, alternatively, flood the otherwise empty fields with data.

It is a 5-D representation scale of the input to the ML model, so that the computational cost of extracting the features is low irrespective of the type of device.

D. Lightweight Supervised Model Selection

The detection engine is a supervised random forest classifier at its core. It was chosen instead of unsupervised Isolation Forests because there are labeled data (Normal vs. Attack) in the ToN-IoT dataset which enables the system to learn particular limits of deviant behavior, as opposed to mere outliers. [25]. The pipeline of training is as follows:

- Data Loading: The system loads the ToN-IoT training set, a non-homogeneous set of datasets that consists of IoT and Network datasets.
- Preprocessing: The training data is then preprocessed using a StandardScaler to normalize the 5-D feature vectors, so that those with larger values (such as the one labelled Range) do not dominate the learning mechanism.
- Training: The Random Forest classifier is trained by the entire labeled data (28706 normal samples and attack samples). The model is trained to estimate the non-linear association among the 5 statistical features and the binary class labels (0: Normal, 1:Attack).
- Serialization: The trained model and scaler are pickled (serialized) to be deployed on the Flask gateway, which allows inference times of sub-milliseconds in live mode.

E. Data Loading and Preprocessing

The system relies on the ToN-IoT benchmark dataset that includes the files of IoT telemetry and network traffic in CSV. In order to deal with heterogeneity of these sources, there is a strict filtering procedure based on numbers only.

- Column Filtering: Non-numeric values, e.g., timestamps, IP addresses, and protocol strings, are filtered out so that the model is lightweight and geared to statistical behavior.
- Data Cleaning: Raw data is purged by removing the whitespace and standardizing the decimal separators.
- Null Handling: Null values or invalid placeholders are represented as 0.0 to have a continuous numeric vector per row.

F. Statistical Feature Engineering

Each cleaned telemetry row or data window is converted to a fixed 5 dimensional feature vector containing the mean, standard deviation, range, median and ratio- where a data row is not all zeros. See Section V-C for details. The goal of this subsection is to avoid duplication, so it only discusses the impact of the previously defined representation on the post-pre-processing. The cleaned numeric values are converted to a standard 5-D feature space, and no matter how many numeric features the data source of each IoT contains, the dimensions of the resulting matrix are all the same, which can be used for scaling and classification using Random Forest.

G. Model Training and Scaling

The pipeline of learning under supervision transforms the processed 5-D vectors to a labeled training dataset on Fig. 5 showing all steps of implementation including the 5-D vectors as part of feature engineering.

- Normalization: A `StandardScaler` object is applied to the feature matrix X, so that all five statistical features are in a similar range. This ensures that features with larger ranges do not dominate the learning process.
- Random Forest Training: A supervised Random Forest classifier is trained on 28,706 normal and attack samples to learn the non-linear boundaries between benign and malicious behavior.
- Model Serialization: The trained model and scaler are pickled (`anomaly_model.pkl`) to be deployed in real-time into the Flask-based gateway.

H. ML and Access Control Mechanism Integration

The major contribution of this study is that there is a close link between ML-based anomaly detection and access control enforcement. In contrast to the conventional IDS models that are put in place separately, the anomaly score generated by the ML model directly changes the level of authorization of the device [21].

There are two threshold values, which are used to categorize behavior as normal, suspicious, and malicious. Normal devices are not restricted, whereas suspicious devices get access but with limited permissions and rate limits. A gateway is accessed to block unscrupulous actions and revoke access immediately.

This integration will provide the security responses to be proactive and automated that reduces the time frame of attack and removes the aspect of manual intervention. The system directly incorporates intelligence into the authorization pipeline to provide behavior-sensitive access control which can be used in real-time IoT contexts.

VI. Experiment Preparation and Findings

A. Simulation Environment and Tools

The proposed behavior-aware access control framework was experimentally tested with the help of the simulated IoT setting in Python. The simulation was created in a way that it was modeled after real-life gateway-based IoT deployments, whereas the behavior of each device and attack situations could be fully controlled.

TABLE II. ToN-IoT Dataset Specifications: Rows, Raw Columns, and Feature Extraction

Dataset Component	Max Rows	Raw Cols	Numeric Cols	Final Features
IoT Fridge	15,000	6	2	5
IoT Garage Door	15,000	6	2	5
IoT GPS Tracker	15,000	6	2	5
IoT Modbus	15,000	8	4	5
IoT Motion Light	15,000	6	2	5
IoT Thermostat	15,000	6	2	5
IoT Weather	15,000	7	3	5
Network Traffic	15,000	44	∼23	5
Combined Dataset	**120,000**	**Varies**	**Varies**	**5 (Fixed)**
Normal Samples (Train)	28,706	–	–	5
Total Samples (Eval)	120,000	–	–	5

The IoT gateway was deployed on Flask framework in which the framework presented RESTful APIs to allow level of registration of devices, authentication, processing of requests and implementing access control. NumPy and scikit-learn libraries were used to provide lightweight ML and data processing. Supervised Random Forest algorithm and normal feature scaling methods were used in detecting anomalies. Other tools like matplotlib were employed in visualising the results and in analyzing performances.

The experiments were performed on a normal computing platform without any hardware accelerations, and the fact that it was possible to run the suggested system on devices with limited resources indicates the presence of IoT gateways that can be used to implement the system. This toolchain represents the software elements that are found in the real-world IoT edge settings [13].

B. Dataset Specifications

The framework utilizes the ToN-IoT benchmark dataset, which contains heterogeneous telemetry from diverse IoT sources and network traffic. To ensure a balanced and representative training environment, the system processes seven distinct IoT device logs and one comprehensive network traffic dataset.

As the IoT devices produce varied data payloads (e.g., temperature vs. Modbus registers), a strict numerical extraction policy is applied. Categorical fields such as IP addresses, protocol strings, and timestamps are excluded to focus the model on statistical behavior. Table II provides the detailed specifications of the rows and columns processed from each source.

1) Data processing and dimensionality reduction: Regardless of the original number of numeric columns (ranging from 2 in simple sensors to 23 in network traffic), every raw row is mapped to a standard five-dimensional (5-D) feature space. This ensures interoperability across the gateway. The mapping process involves cleaning missing values (replacing them with 0.0) and calculating the five summary statistics: mean, standard deviation, range, median, and non-zero ratio.

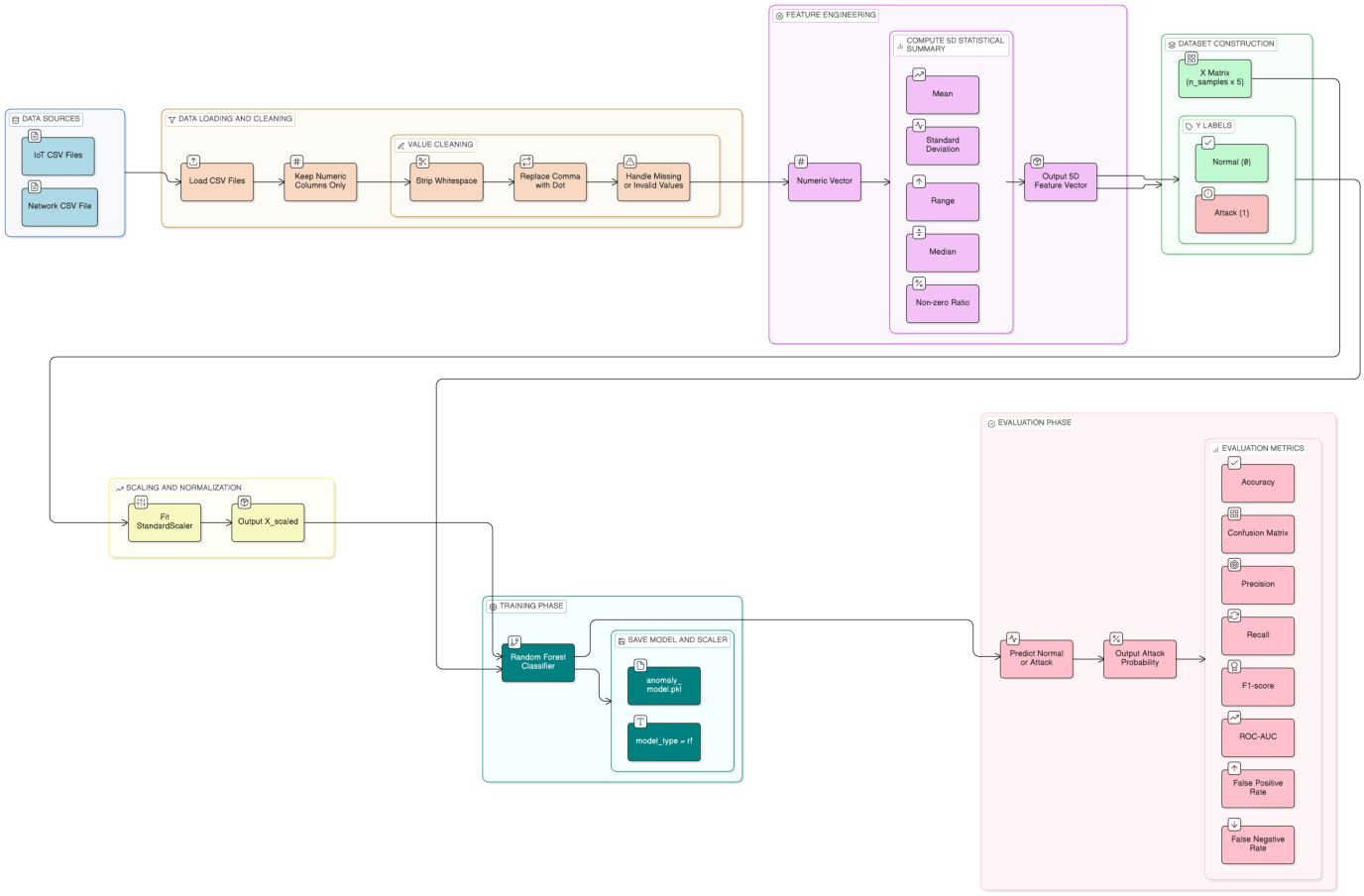


Fig. 5. End-to-end supervised random forest training pipeline: From raw ToN-IoT data loading and 5-D statistical feature engineering to model serialization and evaluation.

C. Dataset and Experimental Setup

The use of the ToN-IoT dataset as the ground truth to prove the performance of the system in a realistic threat scenario was applied. This dataset has been widely known in determining attacks in heterogeneous IoT networks [37].

- Training Phase: It was trained on a smaller set of samples where it involved 28,706 normal samples and attack samples where the Random Forest was able to develop a full spectrum of a decision boundary.
- Evaluation Phase: The system was assessed with regard to a separate assessment set of 120,000 total samples. Such a large-scale test was rich in diverse attack types (e.g., DoS, DDoS, scanning, and ransomware) and non-attack traffic, which was a stringent stress test of the classifier.

D. Performance Metrics

The proposed system was tested in terms of several performance measurements to measure the performance in detection and the feasibility in practice. To identify the general correctness of the anomaly classification, detection accuracy was measured. The trade-off between false positives and false negatives, especially with regard to detecting malicious behavior, was determined by computing precision, recall and F1-score.

Furthermore, receiver operating characteristic (ROC) analysis and area under the curve (ROC-AUC) to measure the capability of the model to differentiate between normal and abnormal behavior at various threshold values were also applied. The confusion matrix analysis was used in order to obtain the true positive, false positive, true negative and false negative rates.

To gauge the level of practical suitability, the response time and the computational overhead were also monitored qualitatively, with respect to the capability of the gateway to handle requests and make access decisions in real time. Such metrics can be considered an overall assessment of the effectiveness of security and efficiency of the systems [8].

E. Results and Analysis of the Experiment

Findings and Discussion The system performance was evaluated with the help of the standard classification metrics, and it was found that the system has a peculiar zero-trust security profile.

- Accuracy and Overall Performance: The model has a total Accuracy of 81.17% over the 120, 000 samples

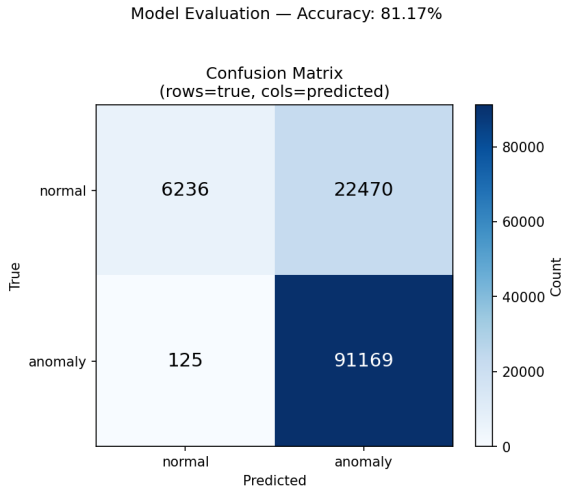
of evaluation. Although this is a competitive accuracy, the distribution of errors (False Positives vs. False Negatives) is the most important observation that is necessary in a security application.

- Recall: Zero-one anomaly-class recall of 99.86% was obtained by dividing the number of correctly detected attack samples by the total number of attack instances, which were 91,169 and 125, respectively. It is reported with two decimal places throughout the manuscript, but is rounded to 1.00 for the purpose of discussing the value of this determination. This result confirms that the framework is able to identify almost all the attack samples evaluated and that any false negative results should not be interpreted as a result of complete or perfect attack detection.
- Precision and False Positives: The precision in the anomaly classes was 0.80. The confusion matrix is reporting 22,470 false positives, which means that there were many more legitimate samples that were incorrectly identified as anomalous. This is a situation with a security conservative operating point, a situation in which the model is security-oriented, meaning that it minimizes false negatives while allowing for more false alarms. In real-world IoT implementations, these false positives can temporarily deny access to legitimate devices, delay telemetry reporting, burden the operator's workload, and decrease the trust in the access control system by the user. To alleviate this effect, anomalous cases with low confidence should not be blocked outright but instead should be placed in a limited-access mode; device-specific thresholds, periodic retraining, whitelisting of safety-critical telemetry, second-stage verification can be employed to minimize unnecessary disruptions.
- ROC-AUC: The model has a ROC-AUC score of 0.82, which represents a profound capacity to arrange randomly selected positive examples higher than negative one.

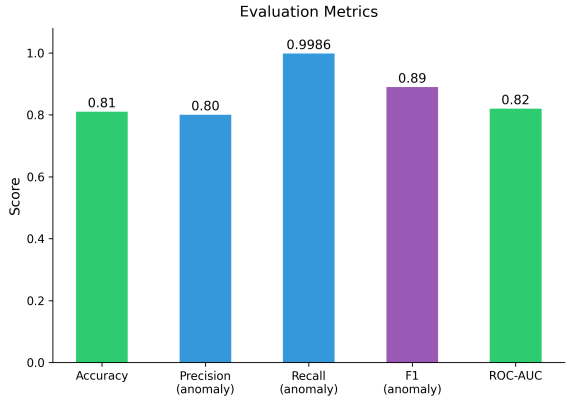
On Fig. 6 illustrates the confusion matrix and the main performance metrics, including accuracy, F1-score, and ROC-AUC.

Operationally, the enforcement of access controls was implemented as soon as an anomaly was detected to prevent the rapid restriction or revocation of device permissions. The model was lightweight and, therefore, provided low computational cost and insignificant effects on the response time of gateways. These findings confirm the feasibility of deploying lightweight ML at the gateway level with the access control and makes sure that the offered framework meets the security-efficiency balance [34].

The evaluation was conducted through a controlled execution script designed to stress the gateway across four distinct phases. First, baseline latency was established using 25 sequential requests from a single device with a 0.1s delay to capture the minimum, maximum, and average processing times reported by the gateway. Second, "Normal Operation" metrics were gathered by simulating 5 devices under a round-robin load for 10 seconds, alternating between standard requests and



(a) Confusion matrix of the random forest model.



(b) Performance metrics (Accuracy, F1-Score, ROC-AUC).

Fig. 6. Evaluation results of the random forest model.

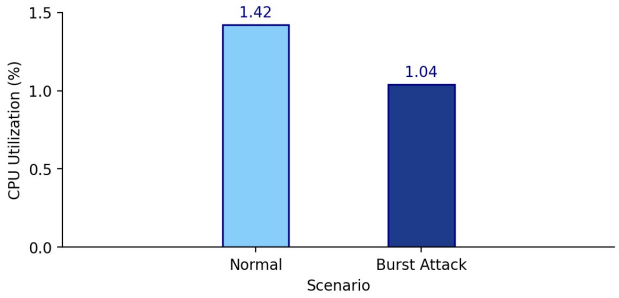


Fig. 7. CPU Utilization under normal vs. burst attack scenario.

administrative status checks. Third, a "Burst Attack" scenario was emulated by a single device issuing 150 requests at high frequency (0.02s intervals), with CPU and RAM samples recorded every 10 requests. Finally, scalability was tested by registering varying device counts (n=1,5,10,20,50) and calculating the average gateway-reported latency across 10 requests per device. These metrics were then combined into a JSON object, which was then processed to create the CPU-utilization, memory-usage and average-processing-latency visualizations

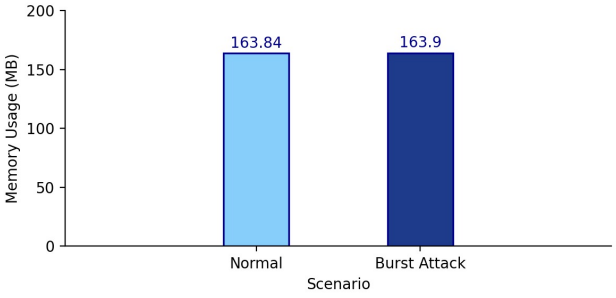


Fig. 8. Memory usage under normal vs. burst attack scenario.

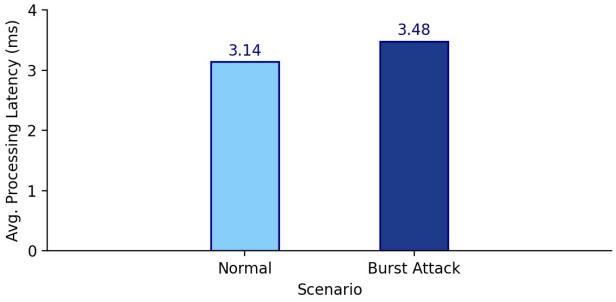


Fig. 9. Average processing latency under normal vs. burst attack scenario.

TABLE III. GATEWAY COMPUTATIONAL OVERHEAD: NORMAL VS. BURST ATTACK

Metric	Normal Operation	Attack Scenario
CPU Utilization (%)	1.42%	1.04%
Memory Usage (MB)	163.84 MB	163.90 MB
Avg. Processing Latency	3.14 ms	3.48 ms (est.)

displayed in, respectively, Fig. 7, 8, and 9. The computation overhead is summarized in Table III both for normal and burst-attack conditions.

F. Advanced Scenarios and Scalability Evaluation

To further assess the robustness and real-world deployability of the proposed behavior-aware access control framework, additional experiments were conducted focusing on large-scale deployments and adversarial attack strategies. These scenarios address the ability of the gateway to 1) scale to hundreds of concurrently connected IoT devices and 2) defend against "smart" attackers that attempt to evade anomaly detection by mimicking normal request rates.

1) Simulation scaling with 500+ virtual devices: A large-scale simulation was executed to evaluate whether the gateway can sustain high device counts while maintaining low latency and stable performance. In this experiment, 520 virtual IoT devices were registered and authenticated at the gateway. Each device issued two legitimate requests to an authorized endpoint, resulting in a total of 1,040 requests processed during the run.

The objective of this scenario was to measure scalability-related indicators, including successful request handling, access denials, processing latency, and gateway resource usage.

TABLE IV. SCALING RESULTS FOR 500+ VIRTUAL DEVICES

Metric	Value
Virtual devices	520
Requests per device	2
Total requests	1,040
Successful responses (200)	1,040
Denied responses (403)	0
Other errors	0
Average latency (ms)	1.68
Latency samples	1,040
Gateway CPU usage (after run)	0%
Gateway memory usage (MB)	164.44

TABLE V. DETECTION RATES FOR MIMICRY ATTACKS (NORMAL RATE, UNAUTHORIZED ENDPOINTS)

Attack Scenario	Endpoint Targeted	Requests Sent	Detected (403)	Rate (%)
Mimicry: normal rate	/admin/train	25	25	100
Mimicry: normal rate	/admin/delete	25	25	100
Mimicry: normal rate	/internal	25	25	100
Mimicry: normal rate	/system	25	25	100
Mimicry: normal rate	/root	25	25	100

All requests in this scenario corresponded to normal behavior and authorized access patterns.

Table IV illustrating the results demonstrate that the gateway successfully handled over 500 concurrent devices and more than 1,000 requests without any access denials or errors. The average processing latency remained below 2 ms, indicating that the integration of behavioral monitoring and lightweight ML does not introduce significant overhead even at scale. These findings confirm the suitability of the proposed framework for deployment in medium- to large-scale IoT environments.

2) Adversarial evaluation mimicry attacks: In addition to the scalability, the framework was tested in mimicry attacks listed in Table V. This is a more sophisticated attack where an adversary mimics the attributes of normal traffic patterns (including frequency and timing), and establishes a connection with an untrusted endpoint.

In this scenario, attackers issued requests at a normal rate (one request every 0.5 seconds) but targeted forbidden administrative and internal endpoints. The detection objective was to verify whether the gateway could prevent unauthorized access even when behavioral patterns appeared benign. Importantly, these attacks are expected to be blocked primarily by access control policy enforcement, rather than anomaly score thresholds alone.

In all the evaluated mimicry scenarios, all the unauthorised requests made against the endpoints selected for the Gateway were denied by the Gateway with a HTTP 403 response. It's important to keep in mind that this result is based on the endpoints tested, the number of requests, and the access-

control policies applied. It demonstrates that in such cases, the attackers were not able to avoid the system simply by mimicking the legitimate request timing for access to explicitly forbidden resources.

3) Discussion: The advanced evaluation confirms that the proposed system scales effectively to hundreds of devices while maintaining low latency and minimal resource consumption. Furthermore, the mimicry attack results demonstrate resilience against sophisticated attackers who attempt to evade anomaly detection by imitating legitimate behavior. Together, these findings reinforce the practicality of deploying the framework in real-world IoT environments, where both scalability and adversarial robustness are essential requirements.

VII. Challenges and Techniques

A. Resource Constraints at the Gateway

IoT gateways have limited computational and memory resources in comparison to cloud servers. Their tasks include authentication, protocol translation, traffic forwarding, and enforcement of security policies, often under strict real-time constraints. Deploying ML-based security mechanisms at this level is therefore challenging, particularly when low latency is required [2].

To address these constraints, the proposed framework employs lightweight feature extraction and a computationally efficient anomaly detection model. The behavioral features are derived from simple statistical properties of request activity (e.g., request frequency and endpoint access patterns) rather than deep packet inspection or payload analysis. These features are consistent with those available in widely used IoT security datasets, such as the ToN_IoT dataset, which emphasizes telemetry and flow-level characteristics suitable for resource-constrained environments.

Furthermore, the system avoids frequent retraining and reliance on large labeled datasets. The Random Forest model operates in a supervised manner and is trained primarily on normal behavior derived from the ToN_IoT dataset, resulting in lower memory usage and reduced computational overhead. This design ensures that continuous security monitoring does not interfere with the gateway's primary networking functions.

B. Model Accuracy vs. Computational Overhead

One trade-off inherent in the IoT security systems is the trade-off between the accuracy of detection and the cost of computation. As high-accuracy models can be trained with the aid of complex deep learning models, this is usually not feasible in resource constrained gateways because the training and inference is expensive [9].

The proposed system is focused on feasible deployability, whereby it is chosen to select a lightweight anomaly detection model that is both accurate enough with an acceptable overhead. More so, feature scaling and model inference are carried out efficiently and real-time assessment of each device request is possible. Even though lightweight models might not be able to measure very sophisticated attack patterns, the experimental findings suggest that they could be used to identify common post-authentication anomalies with low latency. This trade-off is reasonable in the case of enforcement of the gateway level, where the response timeliness is of great importance and the marginal accuracy gains are not always so important.

C. Dealing with Concept Drift in Behavior of Devices

The IoT device behavior can change with the passage of time because of the change in application requirement, update of firmware or the condition of use. Changes like a concept drift may make the behavior occurred earlier in normalcy to seem drastic hence producing false positives [3].

To overcome the above issue, the proposed framework will accommodate the re-training process periodically with verified benign behavior gathered at the gateway. The system can adapt to the gradual changes in behavior, while still recognizing the sudden or extreme changes. But retraining needs to be carefully done so that the animals do not become poisoned or learn bad habits. Therefore, on the borderline cases, it is recommended to manage them temporarily by giving restricted access instead of blocking them immediately, minimizing the operational impact of false alarms, while maintaining the security.

D. Security and Robustness Issues

Although the suggested framework improves the security of IoT by enforcing behavior-based access control, various issues of robustness still exist. The enemies can strive to imitate normal behavior so as to escape detection of anomaly or they could progressively adjust their behavior so as not to cause thresholds. Also, there are such attacks as poisoning against the training data, which may adversely affect the performance of the models [6].

To overcome such issues, the system restricts training of models to the times when they are confirmed to be operating benignly and limits administrative access to training functionality. The selection of the behavioral features is aimed at capturing many dimensions of interaction with the device and thus increasing the challenge to the attacker to avoid detection across all dimensions at the same time. Even though improved adversarial attacks are a research problem, the suggested design offers a viable and robust base to the defense of security by the gateway [38] [26].

E. The Security vs. Usability Trade-off

The results show that there is a clear compromise between recall (to avoid missing attacks) and precision (to affect operational usability). The proposed Random Forest implementation focuses on the anomaly-class recall and obtained unrounded anomaly-class recall at 0.9986. A critical infrastructure or industrial Internet of Things application may be able to tolerate this preference due to the potential cost of a missed attack being greater than the cost of a false alarm. However, the design decision should be carefully interpreted as high false positive rate may have an impact on the legitimate IoT operation.

Many samples were considered to be anomalous given the operating threshold and are reflected in the number of false positives, which was 22,470. In reality, this can result in a delay to legitimate device communication, alert volume or temporary denial of access for devices with abnormal but benign activity, for example sensors that do not read smoothly, but read bursts during a change in environmental conditions. Therefore, a

production deployment must have a staged response: low-confidence anomalies should be blocked in restricted mode and only high-confidence malicious behavior should be blocked immediately. Other mitigation measures are device specific baselines, calibration of thresholds, review by an administrator for 'borderline' cases, whitelisting of safety-critical endpoints, and retraining periodically with verified benign data.

VIII. Proposed Behavior-Aware Access Control Framework And Evaluation

A. Authorization Workflow Behavior-Aware

The suggested behavior-sensitive authorization protocol is a continuation of conventional IoT access control implemented with the evaluation of continuous behavior analysis following authentication. After a successful authentication at the gateway, a device receives an initial access level, and it is put under the close supervision. A request by the device made after a certain behavior is utilized to create a new behavioral profile indicative of real-time operational properties [39].

Behavioral characteristics are derived out of recent interactions with the device and sent to the lightweight anomaly detection model. A scoring on the resulting anomaly is the extent of deviation of the learned normal behavior. This score is then instantly translated to the authorization decision, which allows the system to dynamically change access privileges. The proposed workflow will establish a continuous reassessment of trust in the authorization pipeline instead of presumed forever by embedding behavior analysis within it [4].

The interaction between the IoT device, gateway and the ML model are illustrated in Fig. 10 for real-time authorization. A REST request is received by the gateway and behavioral features are extracted from the request and sent to the trained ML model. The anomaly score is then sent back to the gateway and correlated with an authorization decision, such as allow, restrict, or block.

B. Enforcement of Real-time Access Control

One of the distinguishing characteristics of the proposed system is the possibility to impose decisions in access control in real time. The proposed framework is also able to implement changes in authorization immediately an authorization change is detected unlike conventional intrusion detection systems that only send out alerts without implementing changes in authorization.

Devices with normal behavior are allowed to access all the authorized resources whereas suspicious devices are redirected into restricted access mode with restricted endpoint accessibility and reduced request rates. Gates ban the access of devices that are classified as malicious at the gateway. Such a graduated response mechanism reduces the impact of legitimate devices as well as facilitates the quick containment of the potentially dangerous devices.

The enforcement logic is directly applied at the gateway whereby every request is assessed and governed without involving any services. This design also contributes greatly to speed of response and allows to mitigate threat proactively in IoT environments that have latency constraints [13].

C. Comparative Study to Traditional Approaches

The proposed framework has significantly better resilience to post-authentication attacks compared to the conventional identity-based access control systems. The existing authorization models cannot react to the corrupted machines that are authorized after a credential is proven, but the proposed system can constantly modify the access privileges in accordance with observed behavior [31].

Unlike cloud-based intrusion detection systems, a lightweight ML implementation in the gateway level minimizes the communication overhead as well as conserves the privacy of the data and yields quicker enforcement measures. Even though deep learning-based methods can be used to perform better when detection rates are high and in a controlled environment, their computational costs and lack of connection with access control make them impractical in enforcing security in real-time IoT settings [29].

In general, the suggested framework balances security effectiveness, computational efficiency, and deployability, which is why it is more applicable to a real-world Internet of Things than most of the existing solutions.

D. Comparative Performance Analysis and Baseline Comparison

For test purposes, the framework is expanded to include the traditional identity-based access control and cloud-based deep learning IDS systems. The proposed framework is envisioned as a gateway level ML solution, hence a representative lightweight ML IDS baseline is included. This baseline is edge, gateway IDS approaches that are lightweight, do local ML based anomaly detection but not directly link detection results to dynamic access control enforcement.

The proposed gateway-based framework using random forest algorithm is summarized in Table VI with the traditional RBAC/ABAC, the lightweight ML IDS baselines deployed by the gateway and the deep learning IDS approaches deployed in the cloud.

1) Analysis of improvements: The proposed system has achieved 99.86% or 0.9986 Recall (Anomaly Class) instead of 100%. This number gets rounded only to 1.00 if rounding to 2 decimal places. The framework is more advanced than the traditional RBAC/ABAC systems by introducing post-authentication monitoring and dynamic enforcement. Its primary benefit, over its gateway-deployed lightweight ML IDS counterpart, is not only local detection, but the direct mapping of anomaly results to access-control decisions. The framework offers local enforcement with lower latency, which is not possible with cloud-based deep learning IDS based approaches, as it takes an average of 3.14 ms to process the data in the evaluated environment. But, the comparison needs to be taken with a pinch of salt, as datasets and metrics vary between studies, as do deployment assumptions.

IX. Discussion

A. Security Implications

Behavior-conscious ML combined with access control can greatly improve the security of the IoT through post-authentication threats. The proposed system mitigates the

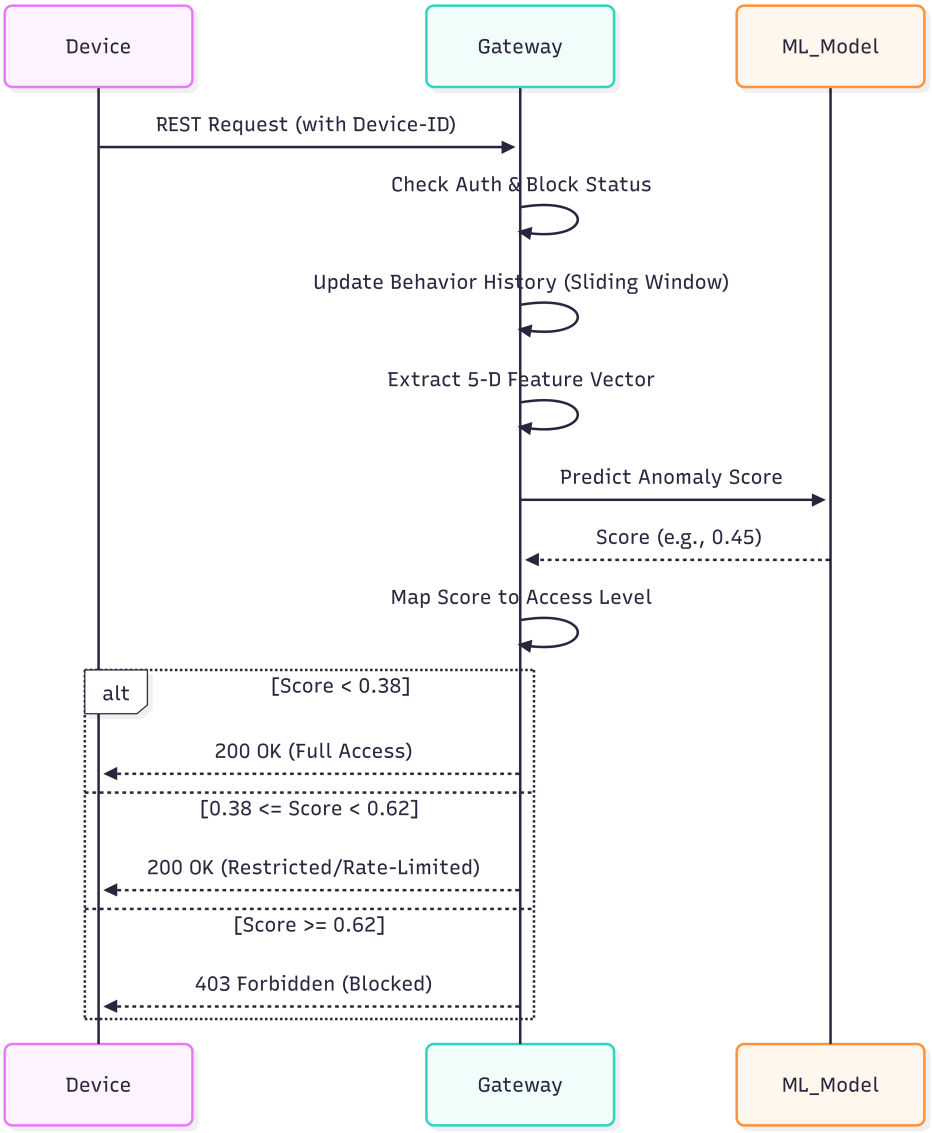


Fig. 10. Real-time request processing and dynamic authorization mapping.

TABLE VI. BASELINE COMPARISON: PROPOSED FRAMEWORK VS. TRADITIONAL, LIGHTWEIGHT ML, AND CLOUD-BASED IDS MODELS

Feature/Metric	**Traditional RBAC/ABAC**	**Gateway-Deployed Lightweight ML IDS Baseline**	**Cloud-Based DL IDS**	**Proposed Framework**
Post-authentication monitoring	None	Passive anomaly monitoring	Passive anomaly monitoring	**Active monitoring and enforcement**
Deployment level	Gateway/local policy	Gateway/edge	Cloud or edge-cloud	**Gateway/edge**
Model type	No ML	Lightweight classifiers such as Decision Tree, Random Forest, or k-NN	CNN, LSTM, DBN, or hybrid DL	**Random Forest plus access-control policy engine**
Detection performance	N/A	High but dataset-dependent; usually evaluated as IDS accuracy/recall only	98.8%–99.5% reported in related work	**99.86% anomaly recall; rounded to 1.00 only at two decimal places**
Processing latency	Low, but no anomaly detection	Low to moderate depending on gateway hardware	High due to cloud transmission and heavier inference	**Very low average processing latency of 3.14 ms in the evaluated setting**
Access-control link	Static	Usually decoupled from access control	Decoupled from gateway authorization	**Tightly coupled with allow, restrict, and block decisions**
Main limitation	Cannot detect post-authentication compromise	Detects anomalies but usually requires separate/manual enforcement	High computational cost, bandwidth use, and privacy concerns	**Higher false positives; dataset-specific validation; requires retraining for concept drift**

threat of insider attacks, credential theft, and compromised nodes because it constantly assesses the conduct of devices [30].

Supervised anomaly detection can help identify malicious behavior that statistically resembles learned attack profiles; however, its ability to detect entirely unseen attacks depends on the representativeness of the training data and the selected behavioral features.

B. Practical Deployability

In a practical sense, the suggested framework is highly applicable to the implementation in the real-world IoT gateways. The ML model and feature extraction are lightweight, which guarantees low computational overhead, and the gateway-centric design makes it easy to integrate the new system with the existing IoT system.

The framework is not dependent on modifying individual IoT devices, which means that it can be used with heterogeneous and legacy deployments. Moreover, gateway-based local processing promotes the concept of privacy by not relaying sensitive behavioral information to other servers [27].

C. Comparison to Cloud-Based IDS Solutions

The cloud-based intrusion detection systems are scalable and have a great computational capability but are affected by the latency, bandwidth, and privacy issues. Conversely, the solution suggested at the gateway level is capable of detecting and enforcing on-the-fly without any connection to the cloud.

Although cloud-based systems can be the right choice in the large scale analytics and long term threat intelligence, the suggested structure fits better the real-time protection and access control implementation. The combination of the gateway-level enforcement and a cloud-based analysis might also be further improved in the future to enhance the security of the deployments [9] [32].

D. Limitations and Threats to Validity

The envisioned framework illustrates a promising behavior-aware enforcement at gateway level, but some limitations must be pointed out. The evaluation is given based on the ToN-IoT benchmark dataset, which is the first one. ToN-IoT also offers heterogeneous IoT telemetry and attack traffic, but does not necessarily contain all real-world deployments, device types, device firmware behaviors or attack strategies. Thus, results reported should be viewed as findings of effectiveness for the evaluated data set and scenarios, and not as general assurances of performance.

Second, the system is based on a fixed five dimensional statistical feature representation (mean, stddev, range, median, non-zero ratio). This representation facilitates lightweight gateway deployment but might lack some protocol-level, temporal, or semantic aspects that are necessary to identify some advanced attacks from atypical device activity. The number of false positives seen in the experiment suggests further refinement of the features or calibration of the devices for production deployment.

Thirdly, the attack concept and patterns change in real IoT environments. Firmware updates, environment changes, new applications or changes in user behavior may alter normal device behaviour. If not retrained at regular intervals, and if the threshold is not adjusted, the model can make false-positive or false-negative errors. Lastly, the experimental gateway was tested in a lab setting, and the framework should be further tested on physical IoT testbeds, multiple deployment of gateways, and extended observation periods.

X. Conclusion and Future Work

This study presented a behavior-aware access control framework by embedding Supervised Random Forest learning into the access control logic for an IoT gateway. The study also demonstrated that the lightweight five-dimensional statistical summaries of the ToN-IoT benchmark dataset can benefit the low-latency edge anomaly detection.The study further confirmed that the five-dimensional statistical summaries of the ToN-IoT benchmark dataset, including mean, standard deviation, range, median, and non-zero ratio, can help with low latency anomaly detection at the edge. The experimental evaluation gave a Recall of 0.9986 (99.86%) on a test set of 120,000 samples and 125 false negatives in 91,294 attack instances, which is an anomaly-class recall. The results should thus be reported as “near-complete detection” for the scenarios considered instead of “perfect” or “universal” attack blocking. The framework illustrates the potential for Zero Trust inspired enforcement on resource limited gateways, without the need for cloud-based deep learning services, however, the false positive results, dependency on a training set, fixed feature representation, and concept drift problems need to be further validated before being deployed in real-world production systems.

The next study will focus on evaluating the proposed framework in a larger scale of IoT, where hundreds or thousands of devices are involved. Scalability issues, including not only higher behavioral diversity but also heavier load on the gateways will be explored to guarantee the stability of performance. Moreover, such approaches as federated learning or online learning might enable multiple gateways to effectively improve their models without needing to share raw data. These could improve flexibility to new threats and privacy as well as minimize centralization of dependencies [40]. Also, the suggested behavior-sensitive authorization model is consistent with the concepts of Zero Trust security. The future could also further integrate with the Zero Trust architectures, where ongoing authentication and least-privilege access are used throughout all the layers of the IoT ecosystem.

Author Contribution

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

Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency fo Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia under the [GRANT No. KFU263248].

Data Availability Statement

The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments

The authors wish to express their gratitude to the Deanship of Scientific Research, Vice Presidency fo Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia. We would like to acknowledge the anonymous reviewers who made great contributions with their brilliant scholarly intuitive comments and sagacious recommendations to improve the quality and clarity of this study.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, 2025.

[2] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. M. Nainar, and M. Imran, "Deep learning and big data technologies for iot security," *Computer Communications*, 2020.

[3] E. Gyamfi and A. Jurcut, "Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets," *Sensors*, vol. 22, no. 10, 2022.

[4] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for iot applications," *arXiv preprint arXiv:2302.12452*, 2023.

[5] "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, 2025.

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2025.

[7] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of things botnet detection approaches: Analysis and recommendations for future research," *Applied Sciences*, vol. 11, no. 12, 2021.

[8] A. L. Buczak and E. Guven, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2025.

[9] A. Awajan, "A novel deep learning-based intrusion detection system for iot networks," *Computers*, vol. 12, no. 2, 2023.

[10] N. Pandey and P. K. Mishra, "Detection of ddos attack in iot traffic using ensemble machine learning techniques," *Networks and Heterogeneous Media*, vol. 18, no. 4, pp. 1393–1409, 2023.

[11] A. Zanella, N. Bui, A. Castellani, and L. Vangelista, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[12] M. M. Rashid, J. Kamruzzaman, M. M. Hassan, T. Imam, and S. Gordon, "Cyberattacks detection in iot-based smart city applications using machine learning techniques," *International Journal of Environmental Research and Public Health*, vol. 17, 2020.

[13] X. Kong and W. Yuhan, "Edge computing for internet of everything: A survey," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2022.

[14] Noshina et al, "Securing the internet of things in artificial intelligence era: A comprehensive survey," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2024.

[15] A. Bilal, Q. Riaz, M. Zeeshan, H. Tahir *et al.*, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using unsw-nb15 data-set," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, 2021.

[16] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *RMIT Repository*, 2024, posted 2024-11-02.

[17] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, pp. 4815–4830, 2024.

[18] S. Alhasan, G. Abdul-Salaam, and Y. M. Missah, "Hybrid network intrusion detection systems: A systematic review," *Preprint*, 2024, january 2024.

[19] P. Cardoso, "Internet of things: Security challenges and solutions," *Preprint*, 2022, june 2022.

[20] E. Hodo, X. Bellekens, A. W. Hamilton, and P.-L. Dubouilh, "Threat analysis of iot networks using artificial neural network intrusion detection system," 2016.

[21] A. Sivanathan, D. Sherratt, H. H. Gharakheili, and V. Sivaraman, "Low-cost flow-based security solutions for smart-home iot devices," 2016.

[22] B. Kitchenham, O. P. Brereton, D. Budgen, and M. Turner, "Systematic literature reviews in software engineering–a systematic literature review," *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2025.

[23] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based iot-botnet attack detection with sequential architecture," *Sensors*, vol. 20, 2020.

[24] L. Xiao, X. Wan, and D. Wu, "Iot security techniques based on machine learning: How do iot devices use ai to enhance security?" *IEEE Signal Processing Magazine*, January 2020, published 19 January 2020.

[25] C. Shao, X. Du, J. Yu, and J. Chen, "Cluster-based improved isolation forest," *Entropy*, vol. 24, no. 5, 2022.

[26] G. Kalleshappa and B. Savadatti, "Network traffic analysis through deep learning for detection of an army of bots in health iot network," *International Journal of Pervasive Computing and Communications*, 2022, ahead-of-print.

[27] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, 2025.

[28] N. B. Karankar and A. Seth, "A comprehensive survey on internet of things security: Challenges and solutions," pp. 711–728, 2023.

[29] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2021, 37 pages; also available as arXiv:2104.07914 [eess.SP].

[30] M. Alazab, S. Priya, P. M, and P. K. Reddy, "Federated learning for cybersecurity: Concepts, challenges and future directions," *IEEE Transactions on Industrial Informatics*, October 2021.

[31] E. Abdulrahman, S. Alshehri, and A. Cherif, "Blockchain-based access control for the internet of things: A survey," 2021.

[32] J. Oh, S. Yu, J. Lee, S. Son, M. Kim, and Y. Park, "A secure and lightweight authentication protocol for iot-based smart homes," *Sensors*, vol. 21, 2021.

[33] A. Singh and K. Chatterjee, "Trust based access control model for securing electronic healthcare system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 783–794, 2025.

[34] L. Zhou and H.-C. Chao, "Multimedia traffic security architecture for the internet of things," *IEEE Network*, vol. 25, no. 3, pp. 35–40, 2025.

[35] S. Suganth and D. Usha, "A survey of intrusion detection system in iot devices," *International Journal of Advanced Research*, vol. 6, no. 6, pp. 23–30, 2025.

[36] B. R. Kikissagbe and M. Adda, "Machine learning-based intrusion detection methods in iot systems: A comprehensive review," *Electronics*, vol. 13, no. 18, 2024.

[37] S. B. Sharma and A. K. Bairwa, "Leveraging ai for intrusion detection in iot ecosystems: A comprehensive study," *IEEE Access*, vol. 13, pp. 66 290–66 317, 2025.

[38] S. Krishnapriya and S. Singh, "A comprehensive survey on advanced persistent threat (apt) detection techniques," *Computers, Materials & Continua*, vol. 80, no. 2, pp. 1–10, 2024.

[39] A. Diro, S. Kaisar, A. V. Vasilakos, A. Anwar, A. Nasirian, and G. Olani, "Anomaly detection for space information networks: A survey of challenges, techniques, and future directions," *Computers & Security*, 2024, open Access.

[40] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2025.