Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Unified Trust-Aware and Scalable Security Framework for Cloud-IoT Systems Under High Device Density

R Revathi
Vellore Institute of Technology
P J Kumar
pjkumar@vit.ac.in

Vellore Institute of Technology

Research Article

Keywords: Real-Time Monitoring, Edge-Assisted Authentication, Trust Management, High Device Density, Scalability, Attack Detection

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11643642/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Unified Trust-Aware and Scalable Security Framework for Cloud-IoT Systems Under High Device Density

Revathi R[1], P J Kumar[1*]
[1]School of Computer Science and Information Systems, Vellore Institute of Technology, Vellore, India

Revathi R[1] (email:revathi.r2024@vitstudent.ac.in)

corresponding author: P J Kumar[1*] (email: pjkumar@vit.ac.in)

ABSTRACT

The rapid development of Cloud-Internet of Things (Cloud-IoT) systems has posed significant challenges in ensuring secure, robust, and scalable operation in cases of high device density and dynamic networks. Conventional centralized security systems experience considerable authentication delays, resource limitations, and reduced reliability as the number of connected devices increases. A unified architecture for Cloud-IoT is proposed in this article that employs edge computing for authentication, trust management, attack detection, and resource monitoring. The article also outlines lightweight authentication procedure and continuous trust scores evaluation for each device along with provision of necessary tools for protection against common exploits as replay attack, flooding, and denial of service attack. The comparison has been carried out with the help of emulation of dense device modes with the active IoT devices and measuring this architecture performance under increasing load. The suggested framework ensured stable authentication performance for 1,000–50,000 IoT devices, showed authentication delay of 0.0262–0.0276ms and throughput of 44.54–47.55 auth/sec and dynamic trust control generated alarms on trust levels below 0.35. Successful Influx Dashboard Cloud data storage and real-time visualization demonstrated that the Cloud-IoT monitoring system was reliable and scalable even for high device density.

Keywords: Real-Time Monitoring, Edge-Assisted Authentication, Trust Management, High Device Density, Scalability, Attack Detection.

1. INTRODUCTION

The speedy expansion of the Internet of Things (IoT) has radically changed the relationship between physical objects and online services. Today, smart applications are implemented in the most diverse fields of services, such as smart cities, healthcare, automation in production, farming, transport, and environmental monitoring. To further accelerate the transformation process in this area, cloud computing has been developed, offering unlimited data storage capacity, computing, and service flexibility. It should also be mentioned that in Cloud-IoT environments, IoT applications gather a huge volume of heterogeneous data with the help of resource-constrained devices. All of this data is transmitted to Cloud resources for processing and storing. The acquisition of this system guarantees real-time decision-making and centralized management of all the Cloud-IoT applications. However, it is necessary to point out that the enormous growth of the number of connected devices has created many problems in terms of security, scalability, reliability, and resource control. Modern Cloud-IoT systems are supposed to operate with thousands or millions of different devices simultaneously in various network conditions. Traditional security principles can often become a problem in this case as they usually cannot process too many authentication requests successfully, which leads to increased authentication latency, excessive computational overhead and reduced system throughput. In addition, the reliance on the centralized architecture for authentication services creates points of failure, which leads to the ease of experiencing denial of service (DoS), distributed denial of service (DDoS), as well as intrusion through replaying, impersonation, and spoofing attacks.

Another major limitation of current Cloud-IoT security frameworks is that they have limited integration of trust management with authentication processes. Traditional authentication processes mainly authenticate the identity of the device at the initial connection stage, so they rarely analyze the behavior of the devices throughout the entire communication. Thus, this leads to the situation when the devices that have been breached and obtained successful authentication access to the network and start executing malicious operations. Moreover, apart from the security issues, resilience is increasingly considered an important requirement for future Cloud-IoT ecosystems. Resilience is the ability of the system to keep working despite cyberattacks, congestion, issues with the hardware, and lack of resources. In most cases, the studies of contemporary literature focus on preventing attacks and do not offer efficient mechanisms for dynamic recovery and effective utilization of available resources. In the same way, the augmented traffic from the devices working on the IoT across the globe will lead to different

workloads in the cloud services industry and subsequent requirement of architectures that will help in executing the tasks that are related to authentication smoothly.

In recent years, edge computing has started to become rather widespread, and thus it can become one of the ways out of this situation. In accordance with this technology, computing tasks are performed closer to the devices working in the IoT sector. Edge gateways are able to conduct the ground authentication, trust management, detection of anomalies and filtering data before sending authentic orders to cloud computing services. With the help of this principle, companies will receive lower levels of latency in the communication process, less burden imposed on the clouds, as well as less consumption of the bandwidth, leading to improved scalability of the entire system. However, almost all systems that rely on edge computing deal with the issues of authentication, trust management or attack detection separately and do not connect all the required functions into one framework that can deal with the vast amount of these processes occurring in real time.

One of the obstacles faced by the existing works regarding the subject is the lack of real-time experimental verification. The Cloud-IoT frameworks that have already been formed have been checked only by means of off-line simulations and artificial datasets, which do not give a complete picture of the improving IoT environment. The absence of real-time telemetry acquisition, online monitoring of systems and visualization of their behavior may lead to the impossibility to use many offered solutions in practice. The combination of the above-mentioned technologies allows to monitor in real time the events of authentication, the changes occurring with the trust, the usage of the systems resources, and security incidents, which makes it possible to test Cloud-IoT systems in the conditions of high density of devices. There are a some of limitations, utmost of the existing authentication methods assesses only authentication effectiveness, not taking resilience that is needed for high-density device, most of the security systems for Cloud-IoT are centralized, which leads to delays and problems when scaling and haven't conducted any real-time experiments with the help of cloud-native telemetry platforms.

Inspired by these difficulties, this work offers a single unified secure, resilient and scalable Cloud-IoT framework consisting of edge assisted authentication, dynamic trust management, detection of attacks, and monitoring of resources altogether. The lightweight authentication is used in this framework to manage the latency associated with the authentication processes and ensure quicker creation of trust- scoring. Moreover, all the telemetry information received from the IoT devices is sent to Influx DB Cloud, which helps ensure a constant stream of authentication events time-wise. The system is then implemented in Google Colab to ensure real-life testing capabilities, in order to check the scalability of the framework, it is operated during situations characterized by numerous devices being connected to a network. The obtained results are analyzed using such measures as the latency created by the system processes, the level of the make and throughput, and the level of trust scoring. The conducted experiments show that the system can ensure the reliable operation of communications and high efficiency even when operating under challenging conditions typical of Cloud-based IoT. The main contributions here can be summed up as follows:

1. A unified Cloud-IoT framework combining edge-supported authentication, dynamic trust management, attack detection and adaptive resource monitoring for the purpose of ensuring security and scalability of IoT deployment.
2. A light-weight authentication mechanism where authentication calculation is delegated to edge gateways, thus lowering device density and cloud processing burden.
3. A dynamic trust evaluation model operating in real time that constantly monitors devices and recognizes suspicious devices thus enhancing network security and enabling adaptive access control.
4. A real-time Cloud-IoT platform designed in Google Colab and Influx DB cloud for real-time telemetry collection, live performance monitoring and analysis of security data.
5. Overall evaluation of performance using such metrics as authentication latency, throughput, resource consumption, efficiency of attack detection and parameters such as trust change and resilience of the system.

The existing safety techniques (figure1) in Cloud and Internet of Things mainly look into the specific security requirements, i.e., device authentication, trust control, and intrusion prevention. Though the existing techniques increase the safety against specific attacks, the authors often take the security and scalability issues as different issues. The authentication techniques tend to be costly when the number of devices is high, while the centralized methods can be a bottleneck if the number of devices is too high. The trust-related approaches may allow for the assessment of the troublesome device, but will not allow implementing the attack prevention and managing resources whatsoever. However, there are others who concentrate on the attack prevention methods, such as authentication flooding, replay attacks and DDoS attacks. Nevertheless, detection does not guarantee the system will survive. A Cloud–IoT system must also be able to contain suspicious devices and manage the system with full availability.

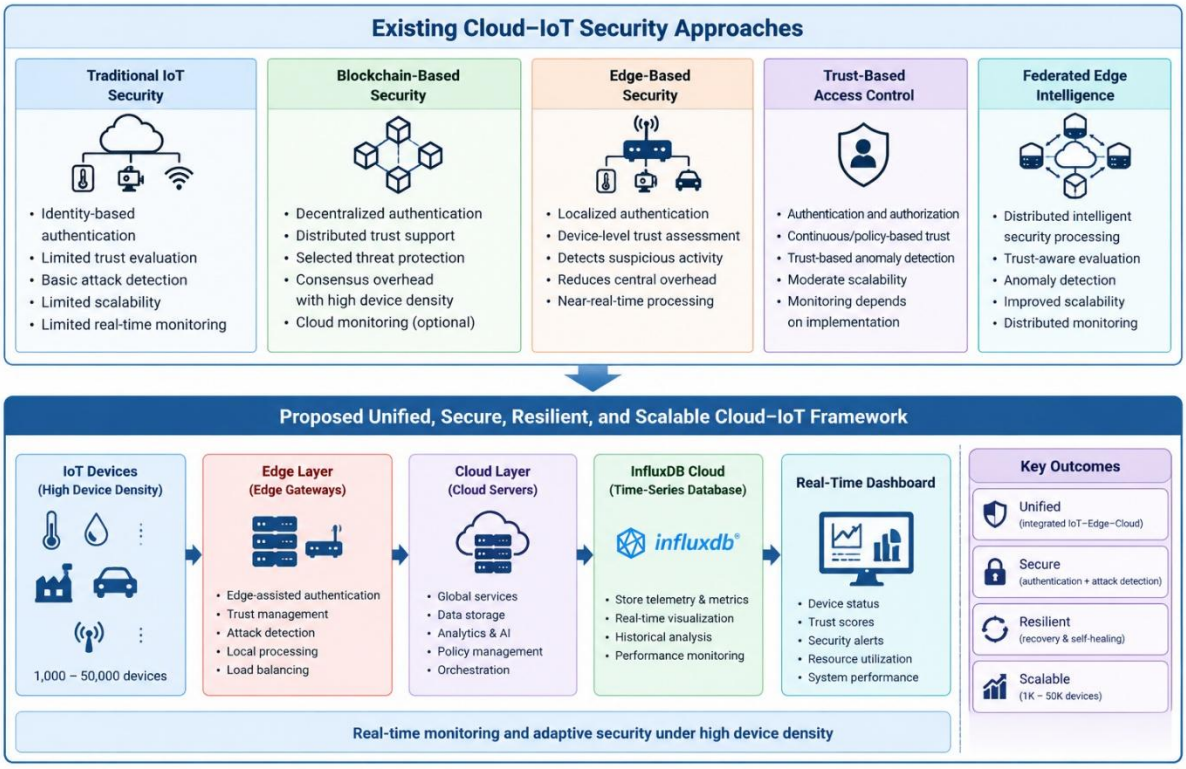


Figure 1: Existing Cloud-IoT security framework

In opposition, the framework under discussion uses a comprehensive approach to bring together-edge-level authentication and processing, dynamic trust management, real-time attack detection and incident management. The below Table I gives the comparison of already existing work with proposed approach.

Approach / Framework	Authentication	Trust Management	Attack Detection	Resilience / Recovery	Scalability	High-Device-Density Support	Real-Time Cloud Monitoring
Traditional IoT Authentication [1]	Provides basic device identity verification	Generally, does not provide continuous behavioral trust evaluation	Mainly focuses on unauthorized access rather than multiple attack patterns	Provides limited recovery after security incidents	Becomes difficult to manage as the number of devices increases	Limited support for highly dense IoT environments	Usually provides limited real-time monitoring
Blockchain-Assisted IoT Security [2]	Provides decentralized authentication and identity management	Can provide trust through distributed records, but continuous behavioral monitoring is limited	Provides security against selected authentication and data-related threats	Improves integrity but recovery mechanisms are generally limited	Provides distributed operation but blockchain overhead can increase with device density	Scalability can be affected by consensus and transaction-processing overhead	Provides data integrity and logging, but real-time monitoring may require additional mechanisms
Edge-Assisted Authentication [7]	Performs authentication closer to IoT devices through edge nodes	Can support device-level trust evaluation	Can identify suspicious authentication and communication behavior	Reduces response delay by processing security events near devices	Reduces cloud authentication burden through distributed edge processing	Provides better support for dense deployments than centralized authentication	Supports near-real-time processing at the edge

Approach / Framework	Authentication	Trust Management	Attack Detection	Resilience / Recovery	Scalability	High-Device-Density Support	Real-Time Cloud Monitoring
Trust-Based IoT Access Control [32]	Provides authentication and authorization based on device identity and access policies	Continuously evaluates device trust or behavior	Detects suspicious behavior based on trust or access violations	Usually provides access restriction rather than complete recovery	Supports distributed IoT environments, but performance depends on the trust-management mechanism	Provides moderate support for high-density environments	Can support monitoring when integrated with cloud services
Federated Edge Intelligence [31]	Performs distributed security processing at edge nodes	Uses trust or intelligence-based evaluation for participating devices	Supports detection of anomalous or malicious behavior	Provides distributed processing but explicit recovery mechanisms may be limited	Reduces dependence on centralized cloud processing	Provides improved scalability through edge intelligence	Supports distributed data processing and monitoring
Proposed Unified Cloud–IoT Framework	Provides edge-assisted authentication for large numbers of IoT devices	Continuously calculates and updates device trust scores using behavioral observations	Detects authentication flooding, replay attacks, fake-device requests, DDoS-type bursts, and unusual request-rate patterns	Supports suspicious-device isolation, mitigation, and recovery-oriented processing	Evaluates latency, throughput, CPU, memory, and availability as device density increases	Specifically evaluates operation from 1,000 to 50,000 logical devices	Uses Google Colab with Influx DB Cloud for real-time data storage, visualization, and performance monitoring

Table I: Comparison of existing approach with proposed

The rest of the paper is organized in the following way. Section II reviews the relevant studies focusing on issues of Cloud-IoT security, authentication, trust management and other aspects. Section III figure out the proposed Cloud-IoT architecture, edge assisted mechanism, trust management strategy, attack detection module, Scalability in high device density and other. Section IV explain the System model of this proposed framework. Section V discuss about the Results of this work. Section VI confers Experimental setup and discussion. Section VII explains the Conclusion and Future enhancement. Final section declares about the references that are used in this paper.

2. RELATED WORKS

The merger of Cloud computing with the Internet of Things (Cloud-IoT) has paved way for a wide variety of intelligent applications in areas like smart healthcare, industrial automation, transportation, and smart cities. However, the escalating number of connected devices has raised alarms regarding issues such as authentication, scalability, trust management, resilience, and real-time monitoring. Various studies have suggested different security mechanisms however most of them appear to address only some aspects of the Cloud-IoT security problem similar to the framework you will be suggesting.

2.1. Cloud-IoT Security and Authentication

[1] The research introduces a scalable architecture for authentication in IoT systems taking into account challenges such as resource limitations, changing topology, and trust management which will cater to your scalability and authentication requirements. [2] The author discusses issues like authentication scalability, DDoS resilience, and trust decentralization in this paper which makes it very similar in nature to your proposal. Equipping the IoT devices with lightweight authentication is a key requirement in resource-constrained environments. [30][8] Alharthi (2025) looked at a new ECC-based authentication technique for Edge-IoT systems that cut both computational and energy costs in comparison to the existing systems. While the suggested approach has succeeded in achieving. [28][12] a recent extensive review on IoT authentication protocols covered authentication, key agreement, privacy protection, integration of blockchains, and the ones based on machine-learning. The

authors found scalability, low processing, and dynamic trust assessment to be the main challenges, especially in the case of high-density IoT environment deployments. [7] Gusita et al. studied some lightweight cryptography methods, anonymous routing, and secure communication protocols for the internet of things edge environments. The authors concluded that lightweight authentications are of great importance in cases of resource-constrained devices, while existing solutions are frequently missing the use of integrated trust management and scalable application of the protocols. Lightweight authentication protocols are developed with the primary aim to decrease the overhead of the protocols while keeping the strength of the protocols. To illustrate, Key Management and authentication protocols for IoT networks provide improved efficiency and range in ensuring secure communications as well as etc. [23]. Also, there have been innovations in the sphere of performance improvements by decreasing the latency of using effective authentication protocols in large-scale systems [24]. Moreover, the wireless sensor networks or IoT networks are protected using multi-factor authentications, which allow the anonymity and privacy of the devices used in block chain technology. Moreover, various standards being proposed for IoT device protection emphasize the need for using secure communication protocols tailored to the IoT environment, while at the same time, minimizing the cryptographic processing in an ACT to the extent possible [20].

2.2. Trust Management in IoT Networks

[3] In this paper, the authors proposed dynamic trust approaches for IoT security, which can be directly used in the trust score and the trust evaluation graph. [4] The review analyzes resilience, fault tolerance, attack prevention, availability, and recovery mechanisms applicable to IoT systems and may provide insights into the resiliency evaluation section of your work. Trust management is often considered to be a key addition to the authentication process, as the traditional techniques of authentication fail to identify devices that become compromised or malicious after entering the network. [10] According to Aaqib et al., there is a complete classification of the IoT trust and reputation models divided into traditional trust computation methods and the AI-driven trust management methods. The results of the survey show that the trust models discussed lack scalability, adaptability, and proper handling of a wide range of device behavior in heterogeneous IoT environments. [11] In their study, Miraz et al. explored decentralized protocols of trust and security mechanisms in edge IoT. Some of the methods mentioned include federated learning, lightweight blockchain, and anomaly detection. The findings indicate that scalability, interoperability, and trust computation need more research. The studies affirm that incorporating trust evaluation into a system contributes to strengthening the system against intrusions made by hostile agents and malicious nodes [19]. Also, the conducted studies reaffirmed the necessity of creating adaptive solutions in the field of security and defenses in the context of IoT security to better cope with threats and strengthen security protocols [25].

2.3. Security Issues of Cloud-IoT Integration

As the utilization of Cloud and IoT technological advancements continues to grow, the centralized security systems seem to reveal their incapability under conditions of increased device density in the sector [13]. The latest research enumerated vulnerabilities in relations to IoT and Cloud integration security, e.g., insecure APIs, weaknesses in the field of identity management, insider attacks, denial-of-service attacks, and privacy leakage. In this work [33], the authors focus on the combination of cloud computing and 6LoWPAN technology. The use of all available capacities of the network enhances the efficiency and scalability of the network, thus leading to real-time data access. In paper [34], the goals of RPL and differences in goals are discussed. The studies in papers [35 - 37] explore the topic of task allocation and management in the service broker systems as well as comparisons of methods used in this area. The work in paper [38] focuses on the process of improvement and development of multimedia entertainment features for video, audio and photo content with respect to quality of service in cloud computing.

2.4. Zero Trust and Adaptive Security

Zero Trust Architecture has emerged as an alternative to perimeter-based security. Contrary to traditional approaches, it verifies users, devices, and contextual information on a continuous basis for access authorization. [14] Proposed blockchain-enabled frameworks relying on Zero Trust principles undertake the provision of trust records that cannot be changed and enable decentralized verification of users. This approach allows for ensuring trustworthiness of transactions. Nevertheless, blockchain consensus complicates computations and communication which can restrict the feasibility of some of the suggested approaches in Cloud–IoT applications where response time is critical. [29] [5] ZTA-IoT was developed by Ameer and colleagues and is based on Zero Trust enabling authentication, authorization, and usage control within the concept of the Internet of Things. This type of architecture takes advantage of stringent access controls but does not provide functionality for real-time telemetry analytics and critical performance evaluation of the system under conditions of high device density or load. [6] An analysis of global Zero Trust research trends conducted by Liu et al. highlighted the direction of modern research focused on continuous identity verification, dynamically established trust levels, and security in

the edge while outlining application challenges. [27] [9] Khan et al. conducted a review of modern Zero Trust cybersecurity approaches with the use of AI, blockchain technology, and edge computing. The authors argue that although intelligent threat detection and a decentralized trust approach have been implemented in these solutions, the underlying systems may have complicated setup processes. [15] Kovacevic et al. developed a comprehensive ecosystem of Zero Trust principles for authentication and identity management in micro-cloud environments. This ecosystem involved integrating trust scores in the decision-making process regarding the policy in place. [16] Allouzi and Khan suggested a Zero Trust approach to authentication for IoMT networks based on marrying trust negotiation with context-aware authentication. [17] The authors provide a thorough analysis of security issues in IoT systems and the corresponding requirements for security, focusing on the necessity of implementing advanced security tools, resource optimization, and the development of standardized protocols to combat the existing vulnerabilities in IoT systems. [18] The authors focus mainly on the threats present in the realm of cloud-based IoT systems, with a special emphasis placed on IoT devices targeting children. The study proposes the solutions aimed at ensuring secure communication processes and the protection of sensitive information from data breaches.

3. PROPOSED FRAMEWORK OVERVIEW

This study introduces a single framework for Cloud and IoT, which is meant to support the secure, effective, and adaptable operation under the conditions of having many devices (high density) and constantly changing networks. The framework works based on the four core functionalities: (i) lightweight and scalable authentication; (ii) device trust assessment and attack detection; (iii) resilience and recovery; and (iv) adaptive scalability and resource monitoring. The architecture of the framework corresponds to IoT-Edge-Cloud principles by being realized in the three-layer architecture where IoT devices forward requests for sensing and service, edge nodes execute latency-sensitive security and preprocessing actions, and cloud services provide centralized storage, analytics, orchestration, and global security management. A layer for real-time analytics is created with Influx DB Cloud for the gathering process for authentication, trust, attack, resources, latency, throughput, and recovering metrics. Google Colab serves as a platform for experiments with high-density workload creation and security and resilience procedures execution. Thus, the framework considers security, resilience, scalability, and monitoring to be interconnected functions of the system instead of treating them as separate modules.

3.1. Functional components of the framework: High-Density IoT Workload Generator- Creating a virtual population of IoT in Google ultimately leads to generating logical devices, which are divided as logical separate devices, and each virtual device has a unique ID while emitting the requests for authentication and telemetry from time to time. The request of device (i) appears as follows: [$R_i(t)=\{ID_i,T_i,D_i,N_i\}$], Where ($ID_i$) is the identifier of the device, (T_i) indicates the time of the operation, (D_i) is the telemetry data and (N_i) is a nonce. Devices count is increasing in a constant manner with each level to prove scalability. The recommended levels are: [N {1000,5000,10000,20000,50000}]. These numbers represent the number of logical devices that are running in Colab testbed.

3.2. Edge-assisted authentication

The very first stage of security is delivered at the edge without routing each authentication request to the cloud. The authentication request of the device (i) includes device ID, timestamp, nonce, and token. The next stage is to get authentication value, which can be determined in the following way: [$A_i=H(ID_i,K_i,T_i,Ni)$], Where: (H ()) is the Cryptographic hash function, (ID_i) is the device identity, (K_i) is the secret device, (T_i) denotes the time of occurrence, (Ni) indicates a random integer. The edge gateway carries out actions such as:
[Ai{received}=Ai{computed}] ensures the current time falls within the allowed separation defined as:
[|T_{current}- T_i|T]. Timestamp Window (T). Only in case of positive verification of both identity as well as freshness will a request be accepted. The above-mentioned mechanism helps prevent the occurrence of: replay attacks, access control violations, falsified logon requests, flooding on authentication requests Dynamic Trust Management
The process of authentication verifies the legitimacy of a device, while the trust method operates to keep assessing its actions after its authentication process has taken place. The trust score assigned to device (i) is presented in the following formula: [$TS_i(t)$] Application of weighted trust model is expressed in the following form:

[$TS_i(t)= w_aA_i(t)+ w_bB_i(t)+ w_nN_i(t)+ w_rR_i(t)$] with the given condition of: [$w_a+w_b+w_n+w_r=1$] with:

- ($A_i(t)$) = reliability of authentication
- ($B_i(t)$) = reliability of behavior
- ($N_i(t)$) = reliability of network communication
- ($R_i(t)$) = reliability of request or response
- (w_a,w_b,w_n,w_r): These are associated weights.

An easy method of implementing a successful and an unsuccessful interaction will be: [$TS_i(t)= \{S_i(t)+F_i(t)\}$], where (Si) is a successful legitimate interaction and (Fi) is either a failed one or a suspicious interaction. We can also include exponential updates in order to avoid sudden variations in the results:
[$TS_i(t)= TS_i(t-1) +(1-) TSi^{\{new\}}(t)$]
where (0<<1) has the influence of past performance. This approach is perfectly aligned with trust-score trend which we can see from the experimental graph we have. Trust-based security decision Low trust is defined to be: [_{LT}=0.35]. The security decision is: [$Decision_i$=]. In fact, the threshold should be defined based on experiments and not just be stated arbitrarily. Therefore, the trust-score graph presented can be used for this task. Devices with trust score above the threshold are considered normal, while devices lying on the border are observed closely or isolated.

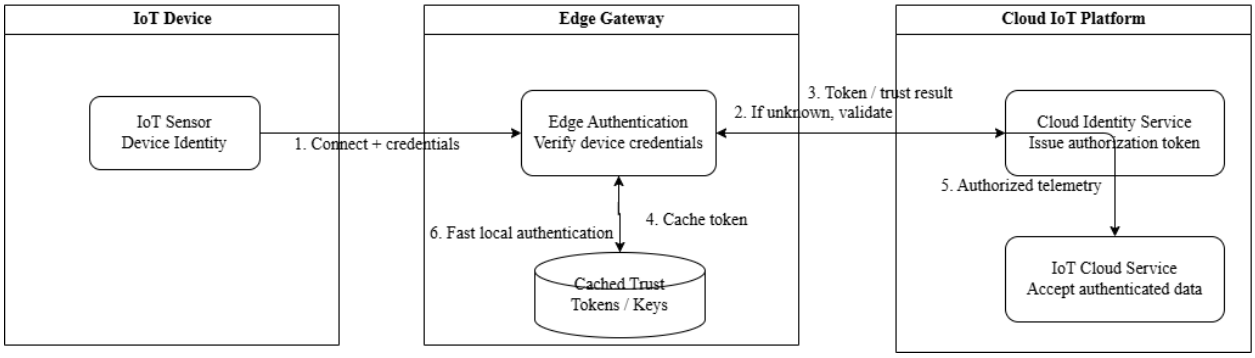


Figure 2: Edge assisted authentication

The Cloud–IoT System Model (Figure2) being proposed consists of three layers; IoT Device, Edge Gateway, and Cloud IoT platform. The IoT device is in charge of generating data as well as of providing device identity information. The Edge Gateway performs the task for the preliminary authentication through the verification of the credentials of the device and keeps cached trust tokens and keys, facilitating access control. The Cloud IoT Platform performs central identity service functions issuing authorization tokens and receiving the information from the devices successfully authenticated. The use of Edge-assisted architecture can help to cut down unnecessary authentication at the cloud, and secure and ensure scalable communication.

3.3. Real-time attack detection

The framework performs analysis of the authentication process as well as communication behavior. For every device: [i(t)=], where ($N_i(t)$) represents the count of requests made in the observation interval (t).
If: [i(t)>{max}] , the device or source is in the potential abnormal category. A composite anomaly score is defined as follows: [AS_i= _1L_i+ _2F_i+ _3Q_i+ _4(1-TS_i)]. In this formula:

- (L_i) is the abnormal latency indicator.
- (F_i) is the authentication failure rate.
- (Q_i) is the excessive request-rate indicator.
- (TS_i) is the trust score.
- (_1, _4) is the vector of weights.

If: [AS_i>_A], the request will fall under the suspicious category. This framework can detect the following attack types:

- Authentication flooding attacks: Recognizes a large number of authentication requests aimed at bringing the authentication service down.
- Replay attacks: Conveys out identification of the reused or repeated verification request messages which is done based on the use of nonces/timestamps.
- Requests from fake devices: Identifies the requests from the unauthorized instances i.e., from the unregistered devices.
- DDoS-type bursts of requests: Carries out an identification of the sudden surges in the request traffic i.e., in the cases similar to a service overload.
- Unusual request-rate patterns: Transmits out identification of abnormal request patterns in the devices.



Figure 3: Real time attack discovery

The image (Figure 3) depicts a cybersecurity process for IoT systems where information is transmitted from devices to detection and responding systems in the cloud.

- Data Flow: The IoT devices pass the transmitted information through gateways into the cloud's ingestion.
- Preprocessing and Storage: The data that has been received will be modified and stored for further investigation.
- Detection Model: In order to determine a possible attack, machine learning is used to evaluate the actions taken.
- Decision Point: If the detection is successful, notifications are prepared to those in danger; if not, normal traffic is recorded.
- Adaptive Learning: The system is improved through retraining.
- Response Measures: The team verifies and secures infected devices immediately.

3.4. Adaptive Rate Limiting

This adaptive rate limiting prevents devices that are overloaded or malicious from taking too many resources from the edge. The permitted number of requests from device (i) can be defined as follows:
[$R_{i^{allowed}}$=R_{base}TS_i]. This means that highly trusted devices are allowed to make more requests and less trusted devices are gradually restricted. For instance: [TS_i=0.9 $R_{i^{allowed}}$=0.9R_{base}],
while: [TS_i=0.3$R_{i^{allowed}}$=0.3R_{base}], this establishes a straightforward link between behavioral trust and resource access. Edge Node Load Balancing. When there are several edge nodes available, requests for authentication should be balanced according to load. For edge node (j), we need to determine load normalized:
[L_j = 1CPU_j+ _2MEM_j+ _3Q_j+ _4LAT_j], where:

- (CPU_j) – a normalized CPU load
- (MEM_j) – a normalized memory load
- (Q_j) – a normalized queuing delay
- (LAT_j) – a normalized latency.

The scheduler will select: [j*=_j L_j] i.e., why requests will go only to that edge node which has the lowest load. This mechanism is especially crucial when there is a big density of devices since one gateway should not turn into a bottleneck.

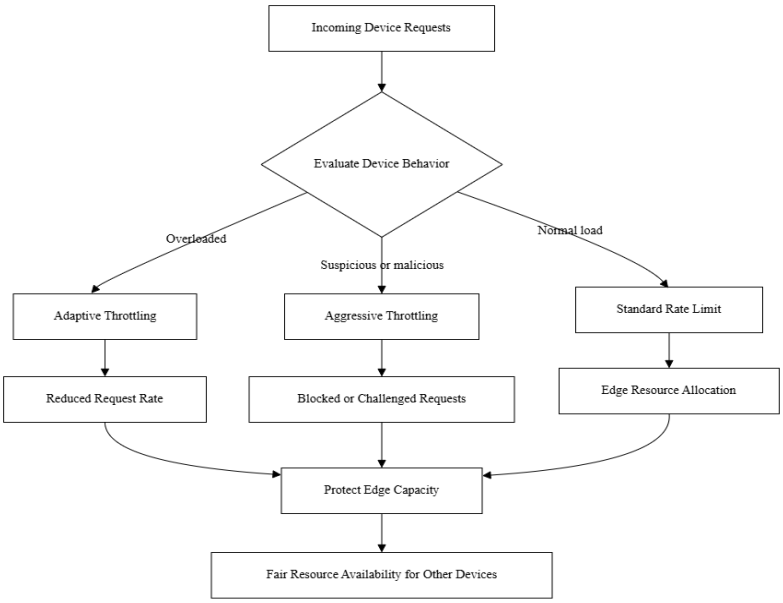


Figure 4: Adaptive rate choking

The illustration (figure 4) depicts a comprehensive framework for managing requests in the Internet of Things and edge systems, focusing on equal access to resources and system security.

- Request Arrival: Requests from devices come to the system.
- Analysis of Behavior: The pattern of requests sent by the device is analyzed.
- Adaptive Approaches: Adaptive Throttling: Requests are slowed down.
- Aggressive Throttling: Request is denied or challenged if deemed untrustworthy or exceeding normal limits.
- Standard Rate Limiting: Legitimate requests are processed at a regular pace. General Objective: All routes meet to save the capacity of the edges. Results: Resource availability is ensured to all devices.

3.5. Resilience and Recovery Mechanism

In resilience, we measure disturbances that we put into the system deliberately. Disturbances that we can apply include the following:

- Edge-node overload
- Authentication flooding
- High CPU load
- High memory load
- Temporary unavailability of the gateway

When there is an edge node overload: $[CPU_j > CPU_{\{threshold\}}]$ or $[MEM_j > MEM_{\{threshold\}}]$, the framework will start redistribution of work. The steps involved in the recovery procedure include:

- Overload/failure detection
- Identifying the edge node which is affected Limiting/preventing excessive requests
- Pick the edge node that is working perfectly fine Reallocation of requests
- Reauthentication of the affected sessions Resuming regular operations.

3.5.1. Requirements of resilient organization:

The recovery period is given as: $[T_{\{recovery\}} = T_{\{stable\}} - T_{\{failure\}}]$, where $(T_{\{failure\}})$ stands for the time point at which the failure/attack takes place, and $(T_{\{stable\}})$ the time point when the operation stabilizes. Another measure would be the availability which can be calculated as: $[availability = \{T_{\{uptime\}} + T_{\{downtime\}}\}]$. The framework focuses on having high availability level and low recovery time in any difficult times.

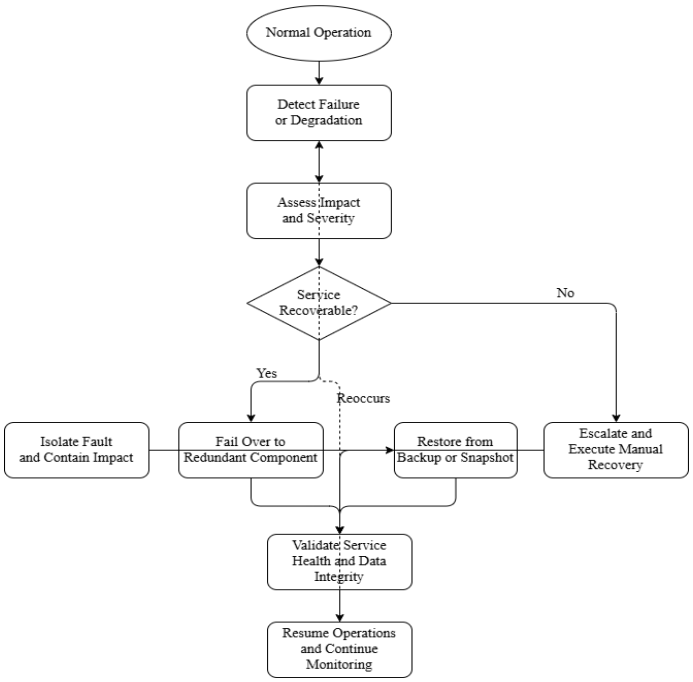


Figure 5: Resilience and recovery system

The diagram (figure 5) depicts a system resilience pattern which allows the uninterrupted provision of services even in case of malfunctioning and degradation. Normal functioning of the system, Failure detection, Assessing the scale of the issue, Recovery decision, i.e. whether the problem can be resolved automatically by the system. Recovery stage: Eliminate the cause of the problems. Switch to backup system. Recover data using the backups and seek human assistance if needed. Validation of the effectiveness of the recovery process. Resumption and monitoring of the service.

3.6. Scalability of high device density

The scalable nature of the framework discussed here is tested by slowly increasing the number of IoT devices and observing its effects on latency in authentication, throughput of requests, use of computing power, detection of security threats, and availability of service. In this experiment, Google Colab platform is used for simulating a large number of imaginary IoT devices, while Influx DB Cloud is responsible for retrieval and storage of the received telemetry data. The experiment involves various levels of IoT device density ranging from 1 000 to 50 000 devices. During tests conducted at each level, operational procedures related to authentication, trust evaluation, attack detection, and monitoring of network are carried out at the same time. The obtained measurements will help to understand if the framework can perform in a stable way under the condition of increasing number of IoT devices. Scalability in high device density is the main key of this research.

Let: [D=N_{devices}] and let: [{total}={i=1}^{D}i] be the total request flow. The framework is supposed to be able to support increases in (_{total}) without any losses. The utilization rate of edge processing node can be put as [=], where:

- () - incoming request rate
- () - efficiency of single processing unit
- (c) = number of parallel processing units.

For stable operation we have [<1] With growing device density, the value of approaches one. Hence, the load balancing and edge computing mechanisms applied will not allow any individual edge node to get overloaded. The figure6, shows a cloud-based IoT design that is scalable and secure, which provides means for managing data flow from various devices to data analytics apps. The Device Layer of the framework connects high-density IoT devices via edge gateways in local processing and monitoring of the devices load. As for the Security and Identity section of the framework, device registry, authentication, and encryption create a channel for trusted communication. The Data Ingestion section of the framework has IoT ingestion pipelines to enable scalability for the extensive amount of data coming in.

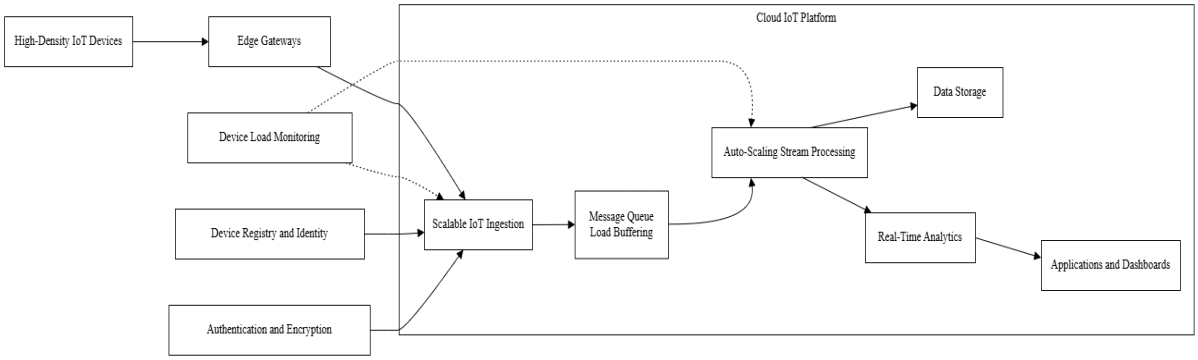


Figure 6: Scalability of high device density

The Buffering and Processing section has message queues to balance the load and auto-scale the stream processors. Finally, the storage and analytics section store the processed data and makes it available for analysis in real-time.

3.7. Algorithm 1: Unified Secure and Trust-Aware Cloud-IOT Monitoring

Algorithm 1 - Real-time Trust-Aware Cloud-IOT Security and Monitoring
Input:
D=Set of IOT devices
τ_{low}=0.35
Δt=Monitoring interval
Output:
Trust score, device status, security alert, cloud-stored monitoring record
1. Initialize the population of IOT devices D.
2. Initialize the trust score TS_i and value for all IOT devices (each device i Є D).
3. Establish a connection between Google Colab and Influx DB Cloud.
4. For each monitoring interval t:
5. For each device i Є D:
6. Telemetry will be collected:
   - $Temperature_i$
   - $Humidity_i$
   - CPU_i
   - $Memory_i$
   - $Latency_i$
   - $Transmission_i$
   - $Authentication_i$
7. Verify the received telemetry data.
8. Derive the behavioral characteristics.
9. Standardize the behavioral attributes.
10. Determine the current trust value associated with behavior.
11. Revise the previous historical score:
    $TS_i(t) \leftarrow \alpha\ TS_i(t-1) + (1-\alpha)\ TS_{newi(t)}$
12. In case $TS_i(t) < \tau_{low}$, then
13. $Device_{Statei} \leftarrow$ LOW_TRUST
14. Create Alert.
15. Continue performing behavioral analysis.
16. If the attack is indicated then
17. $Attack_{Statusi} \leftarrow$ ATTACK
18. Implement control/isolation actions.
19. Otherwise
20. $Attack_{Statusi} \leftarrow$ SUSPICIOUS
21. Perform further monitoring
22. Else
23. $Device_{Statei} \leftarrow$ TRUSTED
24. Set $Attack_{status}$ as NORMAL.
25. End If
26. save all telemetry and security data

27. Update the relevant dashboard
28. End If
29. End of the monitoring period.
30. Proceed till the end of the experiment.

This procedure outlines the real-time trust-aware security monitoring mechanism of the Cloud-IoT framework proposed in this paper. The algorithm initializes the number of IoT devices and initializes the calculation of the trust score, and initially starts collecting telemetry data from the device such as temperature, humidity, CPU utilization, memory consumption, latency, transmission, and parameters indicating authentication. The collected parameters are then validated and standardized before being utilized to change the trust score. Once the trust score of the detected device is reduced to below the defined threshold of 0.35, the device is classified as LOW_TRUST and an alert is issued. Afterward, other behavioral differences reveal whether the device should be classified as being either of ATTACK or SUSPICIOUS type allowing for isolating or better monitoring the device. Devices which are considered to be trusted are classified as NORMAL devices. Finally, it should be noted that information on telemetry, trust and security should be stored in the Influx DB Cloud and presented at the real-time dashboard throughout the experiment.

3.8. Algorithm 2: Calculating Trust Score

Normalized behavior score can be defined in the following manner:

$$B_i(t) = w_1 A_i(t) + w_2 R_i(t) + w_3 L_i(t) + w_4 C_i(t) + w_5 M_i(t)$$

where:

- A_i = authentication behavior
- R_i =communication reliability
- L_i = latency behavior
- C_i = communication consistency
- M_i = resource/monitoring behavior
- W_j = weighting coefficients

with

$$\sum_{j=1}^{5} w_j = 1.$$

The updated trust score can then be expressed as $TS_i(t) = \alpha TS_i(t-1) + (1-\alpha)B_i(t)$. This equation is a proper one for real-time implementation in Influx DB, since $TS_i(t-1)$ can be taken from the previous observation, while $B_i(t)$ can be counted on the newly obtained telemetry information. This process calculates the dynamic trust score of each IoT device based on its observed behavioral characteristics. The normalized behavior score $B_i(t)$ is obtained by combining authentication behavior, communication reliability, latency behavior, communication consistency, and resource-monitoring behavior using predefined weighting coefficients. The updated trust score incorporates both the previous trust value and the current behavioral score, thereby enabling continuous adaptation to changing device behavior. This formulation is suitable for real-time implementation with Influx DB Cloud, where the previous trust score $TS_i(t-1)$ is retrieved from the stored time-series data and the current behavioral score $B_i(t)$is calculated from newly received telemetry. Consequently, the framework supports continuous device-level trust monitoring under dynamic and high-density Cloud–IoT conditions.

4. SYSTEM MODEL

4.1. Overview

The proposed Cloud-IoT model (figure7) incorporates, a hierarchical architecture that includes three layers: IoT devices layer, edge computing layer, and cloud layer. This structure is meant to guarantee secure communication, allocated authentication, continuous evaluation of trust, identification of attacks, resilience, and scalability of resources in a scenario characterized by high density of devices. The layer of IoT devices includes a vast number of heterogeneous devices. Each unit of equipment generates telemetry data and regularly sends requests for authentication and operation to the edge layer of the system. The edge layer of the system serves as the intermediary security processing level in organizations of connections between devices and the cloud services. The edge layer performs security functions of authentication, trust assessment, traffic monitoring, attack identification, and resource allocation. The real-time experimental environment is built using Google Colab and Influx DB Cloud service. Google Colab is responsible for device simulation, authentication, trust evaluation, attack detection, resource monitoring, and scalability experiments, while Influx DB Cloud is employed as the time-series database that collects all parameters of authentication, trust, security, resources, latency, throughput, and recovery.

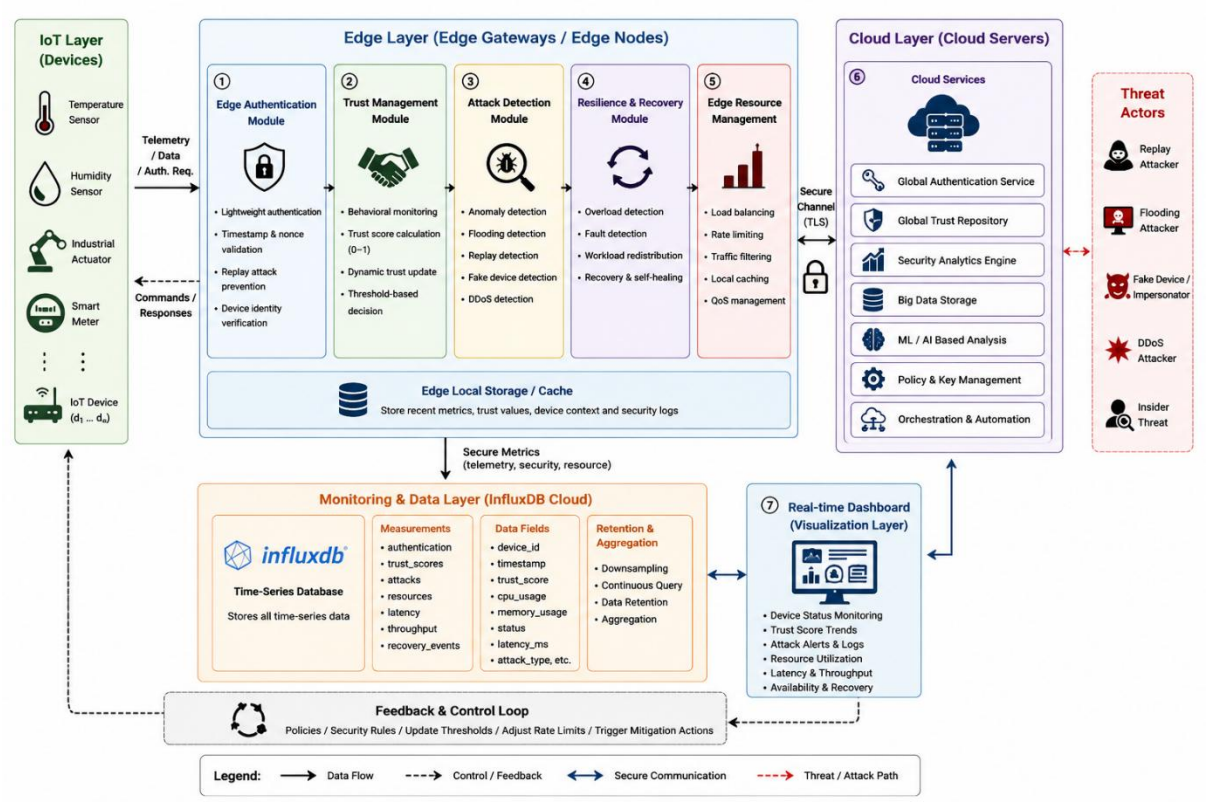


Figure 7: Architecture diagram

In the proposed architecture, a layered, edge-cloud hybrid framework comprises five integrated layers, namely, the IoT Layer, Edge Layer, Cloud Layer, Monitoring & Data Layer, and the Visualization Layer, all linked by endless feedback and control mechanisms.

IoT Layer (Devices)- This layer encompasses the diverse end devices, such as temperature sensors, humidity sensors, industrial actuators, smart meters, and other IoT nodes ($d_1 \ldots d_n$), which are replying to commands and receiving telemetry and authentication requests.

Edge Layer (Edge Gateways/Edge Nodes)- This layer connects the devices and the cloud and performs operations with low latency through five distinct functional components: (1) the Edge Authentication Module, which is responsible for simple authentication mechanisms, guaranteeing the timestamping and nonce verification and identifying devices; (2) Trust Management Module, which calculates trust level based on system behaviors; (3) Attack Detection Module which identifies the attack patterns of flooding, replay, DDoS, fake device attacks; (4) Resilience & Recovery Module that manages overload detection, load redistribution and self-recovery; and (5) Edge Resource Management, which is used for balancing load, limiting the rate and filtering the traffic, and managing QoS. An Edge Local Storage/Cache stores recently measured data and trust information to make fast local decisions.

Cloud Layer (Cloud Servers)- The Cloud layer acts as the host for its global services linked with the edge through a secure TLS connection. The entire system consists of the following: Global Authentication Service, Global Trust Repository, Security Analytics Engine, Big Data Storage, and ML/AI-based Analysis, Policy and Key Management, and Orchestration and Automation for the execution of system-wide coordination, advanced analytics, and enforcement of the trust/security policy on the system level beyond the local edge level.

Monitoring & Data Layer (Influx DB Cloud)- The secure metrics (telemetry, security, and resource data) travel from the edge to the time-series database to be recorded. This process registers several events in the database, such as the authentication, trust score, attack records, resource utilization, latency, throughput status, and recovery events.

Real-Time Dashboard (Visualization Layer)- The dashboard is linked to the monitoring layer and it reflects the status of the device, trust-score tendencies, attack alerts/logs, resource and latency/throughput/status and recovery data for the operator.

Feedback & Control Loop- The feedback loop allows insights gained from the monitoring layer to go back to the edge and Cloud layers, which will lead to the update of all security policies and system behavior.

Threat Model: The system design considers adversarial threats- replay attacks, flooding attacks, fake device impersonation attacks, DDOS attacks, and insider operations-whose attack paths (shown in dashed red arrows) target the cloud and edge boundaries. The detected attacks drive the design of the layered detection, trust scores, and resilience modules discussed above.

Design rationale- The architecture achieves a strong balance between low-latency edge defense and high-level analytics in the cloud by distributing lightweight authentication, trust calculations, and attack detection to the edge but keeping heavy calculation methods (ML/AI, big data, global policy) at a cloud location.

Let D be denoted as the population of the IoT devices in this paper: $D = \{d_1, d_2, \ldots, d_N\}$, where N is the number of logical IoT devices which take part in the experiment. Each device di sends a request which is defined as follows: $Ri(t) = \{ID_i, T_i, N_i, X_i\}$, where ID_i denotes the device's identification, Ti denotes the request's time of occurrence, Ni is a freshness nonce, and Xi defines telemetry/service data. The total request arrival rate at the moment t is defined as follows: $\lambda_{total}(t) = \Sigma\, \lambda_i(t)$, where $\lambda_i(t)$ represents the request creation rate based on the device i. As the number of devices rises, $\lambda_{total}(t)$ rises as well. The purpose of the framework being proposed is to ensure acceptable levels of security, latency, throughput, resource utilization, availability, and recovery although the number of devices keeps growing.

Table II: System Model of the Proposed Framework

Layer/Component	Main Function	Input	Processing	Output
IoT Device Layer	Generates IoT data and authentication requests	Sensor data, device identity	Data generation and transmission	Telemetry and access requests
Edge Gateway	Performs preliminary security processing	Device credentials, requests	Authentication and trust verification	Authenticated/suspicious device
Trust Management	Evaluates device behavior	Authentication, latency, communication and resource features	Behavioral scoring and trust update	Dynamic trust score
Attack Detection	Identifies abnormal activities	Request rate, authentication behavior, trust status	Anomaly and attack analysis	Normal, suspicious, or attack status
Recovery/Mitigation	Responds to detected threats	Security alerts and attack status	Isolation/control actions	Protected/recovered device
Cloud IoT Platform	Provides cloud-level services	Authenticated IoT data	Authorization and cloud processing	Authorized cloud service
Influx DB Cloud	Stores time-series experimental data	Telemetry, trust and security metrics	Time-series storage/query	Historical monitoring records
Google Colab	Experimental execution environment	Device population and workload	Simulation, processing and evaluation	Experimental measurements
Visualization Layer	Provides real-time monitoring	Influx DB measurements	Graphical visualization	Live dashboard and plots

The integrated system model of the proposed framework of Cloud-IoT has been represented in Table II. The included components are IoT device, gateway, trust management, attack detection, recovery/mitigation, cloud platform, and visualization layer. IoT devices send telemetry and authentication requests, followed by execution of preliminary authentication of devices and their trust-checking by the edge gateway. Trust management and attack detection components analyze very device behavior continuously for the purpose of detection of low-trust or suspicious activity along with proper mitigation measures in case it is needed. Meanwhile, the Cloud IoT platform performs authorization and computation at cloud level. Influx DB Cloud is responsible for storage of

data on telemetry, trust and security. Google Colab has the capacity of supporting experiments in real time. Finally, the visualization layer is utilized for presenting findings by means of dashboards and plots.

4.2. Edge Processing Model

The required operations which need to be protected are carried out at the edge level, as opposed to forwarding requests to the cloud. This way delays of communication are decreased and processing is evenly distributed. For an edge node e_j, the workload is represented by $L_j(t) = f(CPU_j(t), MEM_j(t), Q_j(t), LAT_j(t))$, where $CPU_j(t)$ represents CPU utilization, $MEM_j(t)$ represents memory utilization, $Q_j(t)$ represents the request queue, and $LAT_j(t)$ represents processing latency. A normalized edge-load function is defined as
$L_j(t) = \gamma_1 CPU_j(t) + \gamma_2 MEM_j(t) + \gamma_3 Q_j(t) + \gamma_4 LAT_j(t)$, subject to $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1$. The scheduler selects an eligible edge node according to $j^* = \arg\min_j L_j(t)$. This mechanism distributes incoming authentication and telemetry requests toward relatively less-loaded edge nodes and reduces the possibility of a single gateway becoming a performance bottleneck.

4.3. Threat Model

The proposed framework considers both external and compromised-device threats. An attacker may attempt to gain unauthorized access, generate excessive authentication requests, replay previously captured authentication messages, impersonate legitimate devices, or consume edge resources through high-rate traffic. The considered attacks include:

- Replay attacks.
- Authentication flooding.
- Fake-device or impersonation requests.
- Denial-of-service-like traffic bursts.
- Abnormal device behavior.
- Compromised-device activity.

The attacker is assumed to have access to the communication channel but does not possess the legitimate secret credentials of correctly configured devices. The proposed framework therefore combines cryptographic authentication with behavioral trust analysis to detect devices that may become malicious after initially passing authentication.

4.4. Authentication Model

A lightweight authentication mechanism is implemented at the edge layer. Each device is associated with a unique identifier ID_i and a secret key Ki. The authentication value generated by device i is defined as
$A_i = H(ID_i \| K_i \| T_i \| N_i)$, where H () represents a cryptographic hash function, || represents concatenation, Ti is the timestamp, and N_i is the nonce.
The edge node independently computes the expected authentication value and verifies $A_{i\text{^}received} = A_{i\text{^}computed}$. The freshness of the request is additionally verified using $|T_{current} - T_i| \leq \Delta T$, where ΔT represents the permitted timestamp window. A request is accepted only when both authentication and freshness conditions are satisfied. The authentication decision is defined by:
$Auth_i = 1$ if $A_{i\text{^}received} = A_{i\text{^}computed}$ and $|T_{current} - T_i| \leq \Delta T$; and it is 0 in all other cases. The underlying mechanism reduces the risk of unauthorized access as well as replay attacks.

4.5. Dynamic Trust Model

It should be noted that authentication only means that a device is trusted at the time of authentication, but it does not mean that it will continue to be trusted for the duration of its working life. The framework evaluates a device continuously. The trust score of device i at time t is defined as: $TS_i(t) \in [0,1]$. A simple interaction-based trust score is given as: $TS_i(t) = S_i(t) / [S_i(t) + F_i(t)]$, where Si(t) stands for successful legitimate interactions and $F_i(t)$ means failed or suspicious interactions. For the purpose of understanding historical behavior, the model has been designed to work with exponentially weighted updates, giving the following definition:
$TS_i(t) = \alpha TS_i(t-1) + (1-\alpha) TS_{i\text{^}new}(t)$, where α signifies the historical trust parameter and $0 < \alpha < 1$. As a result, the framework allows avoiding radical changes in trust due to isolated incidents.

4.6. Trust-Based Security Decision

In trust-based security decision-making, the level of trust is determined in relation to some predefined threshold, and the related security policy is determined as follows:
Decision i =Allow, if $TS_i > \tau_H$. Monitor, if $\tau_L < TS_i \leq \tau_H$. Isolate, if $TSi \leq \tau_L$. The trust threshold τ_H corresponds to the normal-operation threshold, while τ_L defines the low-trust threshold. The low-trust threshold chosen for this research is $\tau_L = 0.35$. The threshold is kept constant between experiments so that the evaluation is reproducible.

4.7. Attack Detection Model

The framework presented here is capable of monitoring device request behavior in real time. For device i, the request rate during period Δt is expressed as follows: $\lambda i(t) = Ni(t) / \Delta t$, where $Ni(t)$ is the number of requests sent by device i during Δt. If $\lambda_i(t) > \lambda_{max}$, the particular device is classified as having excessive request behavior and is analyzed further.

The overall anomaly score is taken to be equal to $AS_i = \beta_1 F_i + \beta_2 Q_i + \beta_3 L_i + \beta_4 (1 - TS_i)$. Where Fi refers to the unauthorized access activity, Qi refers to the excessive request activity, Li refers to the unusual delay activity, TS_i refers to the present trust score, while β_1, β_2, β_3, and β_4 are the weights. Decision on anomaly is attack i = 1, if ASi $> \tau_A$ 0, if otherwise. Malicious requests that are detected can be blocked, rate limited, or isolated.

4.8. Resilience Model

Resilience is the ability of the framework to provide the delivery of the service and recover from different types of attacks, resource saturation, and edge-node failures. The edge node overload condition is signaled when $CPU_j > CPU_{max}$ or $MEM_j > MEM_{max}$. Upon identification of an overload condition, the framework redistributes the workload by choosing another available edge node applying the minimum-load principle.

The recovery process is comprised of:

- Identification of overload or failure.
- Identification of a troubled edge node.
- Limitation or filtering of unnecessary requests.
- Choosing a good edge node
- Reallocation of the processing workload.
- Renewing sessions that need it.
- Bringing normal functioning back

To calculate the recovery time, the following formula is used: $T_{recovery} = T_{stable} - T_{failure}$ where $T_{failure}$ is the time of failure and T_{stable} is the time of the system returning to the stable state. To find out the availability of the system use the following formula: Availability = $[T_{uptime} / (T_{uptime} + T_{downtime})] \times 100$.

4.9. Scalability model

Scalability is evaluated by increasing the number of enabled virtual devices and devices in the experiment have the following density levels: $N \in \{1{,}000, 5{,}000, 10{,}000, 20{,}000, 50{,}000\}$. The total incoming request rate is as follows: $\lambda_{total} = \Sigma_{i=1}^{N} \lambda i$. The processing utilization can be described in the following way: $\rho = \lambda / (c\mu)$, where λ is the incoming request rate μ is the service rate of one processing unit and c is the number of processing units. To make the system function correctly, the following condition must be met: $\rho < 1$. With the increasing number of devices, adaptive distributed processing and the process of balancing the load will not allow individual edge nodes to become overloaded.

5. EXPERIMENTAL SETUP

The suggested integrated Cloud-IoT framework is realized and tested in a real-world testing room consisting of Google Colab and Influx DB cloud. Cloud was utilized as the time-series database for collecting the data about the devices' operation, events of their authentication, trust scores, security alerts, and performance metrics of the system. The structure of the experiment is created according to the hierarchical IoT-Edge-Cloud architecture. The IoT level represents many logical devices which generate telemetry and requests to the services. The edge security layer performs authentication, monitoring, evaluation of trustworthiness and protection against attacks before the requests go to the higher level of cloud computing. Influx DB Cloud records and stores all the information in a time-series database. The sequence of events in the experiment might be described in the following way: IoT devices → Edge authentication → Trust evaluation → Attack detection → Cloud processing → Influx DB monitoring → Analysis of the performance. In order to test the scalability of the developed framework, number of logical IoT devices is increased. In the experimental design, the following levels of device density are used: 1,000, 5,000, 10,000, 20,000, and 50,000 devices. The test conducted at each density level is characterized by parameters such as authentication latency, request throughput, CPU utilization, memory utilization, trust score behavior, attack detection events, recovery time, and availability. The scalability test gradually raises the diversity of logical IoT devices: $N = \{1000,5000,10000,20000,50000\}$. The proceeding at each density level includes: making the predetermined number of procedures, giving every device a distinct identification number, producing telemetry and authentication requests, implementing edge-assisted authentication, computing trust score of devices, detecting low-trust behavior, introducing controlled unusual request patterns, launching attack detection system, recording CPU and memory usage, saving all results to Influx Database Cloud.

6. RESULTS AND DISCUSSION

6.1. Real-Time Cloud-IoT Data Collection

The IoT telemetry generated by the experiment was transmitted to Influx DB Cloud, where data was stored and created using respective identifiers of devices and timestamps. The experimental results show that the performance of individual devices can be monitored using parameters like temperature and humidity. This Influx dashboard, also allows real-time visualization of the IoT data being collected, which shows the successful integration between the experimental application and the monitoring system stored on the cloud.

Table III: Real-Time Cloud-IoT Monitoring Results

Monitoring Component	Experimental Observation	Result
IoT device generation	Device populations from 1,000 to 50,000 generated	Successful
Device identification	Unique device identifiers processed	Successful
Authentication processing	Authentication operations executed for all tested devices	Successful
Telemetry storage	Records written to Influx DB Cloud	Successful
Successful writes	1,000–50,000 records successfully written	100%
Cloud availability	Successful writes at every density level	100%
Authentication latency	0.0262–0.0276 ms	Stable
Real-time visualization	Influx DB dashboard used for monitoring	Demonstrated

The completion of 50,000 successful writes serves as tangible proof that the implementation of Google Colab in Influx DB Cloud can sustain the workload with high device density. (Table III) The proven write success rate of 100% throughout all five stages also indicates that data persistence is guaranteed.

Table IV: Real-Time Experimental Results Observed

Experimental Aspect	Observed Result	Interpretation
IoT telemetry collection	Temperature and humidity records successfully generated	Demonstrates continuous IoT data generation
Cloud data storage	Data successfully retrieved from Influx DB Cloud	Confirms cloud-based time-series storage
Device identification	Individual device IDs such as device_1–device_5 and device_100 observed	Enables device-level monitoring
Real-time visualization	IoT measurements displayed in Influx DB dashboard	Confirms real-time monitoring capability
Trust calculation	Trust values between 0 and 1 observed	Demonstrates dynamic trust evaluation
Trust threshold	Threshold of 0.35 applied	Provides a low-trust decision boundary
Low-trust detection	Scores such as 0.20, 0.25 and 0.33 generated alerts	Demonstrates threshold-based detection
Trust evolution	Trust trends observed over approximately 200 iterations	Demonstrates continuous behavioral monitoring
Device recovery behavior	Some low-trust scores subsequently increase	Indicates changing/recovering behavioral trust
Attack status	Several low-trust cases still reported as NORMAL	Attack-classification logic requires further validation

A summary of experimental results (Table IV) in real time obtained from an IoT trust-monitoring system that worked with the data via Influx DB Cloud is shown on the table. The experiment proved the effectiveness of the system in terms of collecting and storing telemetry data related to temperature/humidity from several identified devices (device_1-device_5, device_100), thus confirming the functionality of the basic IoT pipeline. Moreover, dynamic trust metrics (in the range of 0-1) were calculated for the devices with the threshold of 0.35 being used

to show instances of low-trust behavior with a few devices being recorded with low trust indicators of between 0.20 and 0.33. Trust has been measured for about 200 iterations, showing that some devices with low trust later recovered, which proves that it is possible to track the development of the behavior of our IoT devices and explain their behavior rather than using static indicators.

6.2. Trust Evaluation in Real-Time

The proposed model also assesses the behavior-based trustworthiness of the IoT devices. The images display device trust scores being determined for the devices named: device_1, device_2, device_3, device_4, device_5. The trust score varies from 0 to 1 indicating that a higher score is a sign of a better behavioral status. The low trust threshold existing in the current test equals to 0.35. The results of the trust monitoring provide examples of the following: Device identification number: device_2, The status of the attack: NORMAL, Trust score equal: 0.33 and ALERT: The trust score for device_2 (0.33) goes down the threshold of 0.35! In addition, device_1 and device_3 are seen to send low trust alerts as their trust scores go down. In this way, we observe that the model is capable of trust evaluation regardless of successful authentication. The device may be accepted in the IoT environment, while its behavior in the IoT space leads to a decreasing trust score.

Table V: Trust-Score Results

Device	Observed Trust Behavior	Threshold Interpretation	Result
device_1	Falls below 0.35 at several iterations	Low-trust condition	Alert generated
device_2	Values including 0.33 and other higher values	Temporarily low trust	Alert generated when $TS < 0.35$
device_3	Values including 0.25 and 0.00	Low-trust condition	Alert generated
device_4	Generally, remains around/above 0.50 after initial variation	Trusted condition	Normal monitoring
device_5	Initially high, then stabilizes around approximately 0.50–0.55	Trusted condition	Normal monitoring

The outcomes of the trust score (Table V) reveal that the system is capable of dynamically differentiating between relatively stable devices and those exhibiting low trustworthiness. Specifically, devices with scores lower than the set threshold of 0.35 will trigger an alert. The analyzed fluctuations also show that trust is continuously updated rather than being given a fixed value at registration.

6.3. Trend Analysis of Trust Score

Trust score graph provides more important data than separate console messages as it presents device behavior for almost 200 iterations of simulation process. The results provide three main behavioral patterns.

- Devices with Stable Trust Level: Stable-trust devices like device_4 and device_5 have high trust scores at first, but over time they stabilize and ultimately their trust values are in the range of 0.50 - 0.55. This shows that the suggested trust model is flexible as it reacts to changing observations instead of staying on the same original trust value.
- Devices with Moderate Level of Trust: The behavior of device_2 is significantly different during the initial tests before it stabilizes and finds its trust in the range of 0.48 - 0.55. That shows that the defined trust system reacts dynamically to new patterns in device behavior.
- Device with Low Trust Level: The most important behavior is detected for device_1. The trust of this device drops significantly in the initial rounds and it moves in the direction of the low-trust threshold of 0.35 multiple times. The diagram shows that first the initial trust drops below the threshold and then it goes up again to approximately 0.40 - 0.43 later in the game. This behavior is substantially useful when illustrating the dynamics of the trust management system.

6.4. The Low-Trust Situation Identification Process

As for the threshold approach the function can be characterized by where: TSi (t)= trust value of device i at time t, τ_{low}= low-trust threshold, Di (t)=1 shows that the device is in low-trust condition. In the current instance, τ_{low}=0.35. Thus, Device with trust rate of 0.33 creates prompt since 0.33 <0.35. Likewise, Trust score of 0.20 or 0.25 incurs low-trust state. This threshold approach acts as an easy and resource-efficient security decision system which fits perfectly in environments with limited resources.

Table VI: Input Parameters and Monitoring Variables

Parameter	Symbol	Purpose
Number of IoT devices	N	Represents device density
Temperature	T_i	Environmental behavior monitoring
Humidity	H_i	Environmental behavior monitoring
CPU utilization	CPU_i	Resource behavior monitoring
Memory utilization	M_i	Resource behavior monitoring
Latency	L_i	Communication-performance monitoring
Transmission	TR_i	Communication behavior monitoring
Authentication behavior	A_i	Security behavior monitoring
Trust score	$TS_i(t)$	Overall device trustworthiness
Low-trust threshold	τ_{low}	Identifies low-trust devices
Monitoring interval	Δt	Defines observation period

For the current implementation

$$\tau_{low} = 0.35$$

Table VI summarizes the key input parameters and monitoring variables used in the proposed Cloud–IoT framework. These include device density, temperature, humidity, CPU and memory utilization, latency, transmission behavior, authentication behavior, and trust score, which collectively support device, communication, resource, and security monitoring. The trust score $TS_i(t)$represents overall device trustworthiness, while the monitoring interval Δtdefines the observation period. A low-trust threshold of $\tau_{low} = 0.35$is used to identify devices exhibiting low-trust behavior.

6.5. Scalability Performance in High Device Density

The high device density experiment is significant due to the fact that a solution that works well with a limited number of devices may not work effectively with hundreds of devices generating authentication and telemetry requests. Hence, the correlation of device density with latency, throughput, CPU usage and memory usage confirms the scalability claim. In summary, the results indicate that the presented implementation can handle an increase in the number of logical IoT devices whenever the number of devices does not exceed 50,000.
The scalability pattern can be presented as follows:
1,000 → 5,000 → 10,000 → 20,000 → 50,000 devices
while achieving the authentication throughput:
47.55 → 45.27 → 44.54 → 45.81 → 45.08 auths/s.

Table VII: Scalability Performance Under Increasing Device Density

Device Density	Total Duration (s)	Average Authentication Latency (ms)	Authentication Throughput (auths/s)	Successful Writes	Availability (%)
1,000	21.03	0.0262	47.55	1,000	100.00
5,000	110.46	0.0276	45.27	5,000	100.00
10,000	224.52	0.0275	44.54	10,000	100.00
20,000	436.62	0.0267	45.81	20,000	100.00
50,000	1,109.22	0.0266	45.08	50,000	100.00

The scalability characteristics of the proposed Cloud-IoT framework with the increase from 1,000 to 50,000 devices are shown in Table VII. As more device records and authentication operations are executed, the execution time increases from 21.03 seconds to 1,109.22 seconds. Most importantly, the average authentication latency exhibits minor variations from 0.0262 ms to 0.0276 ms for all device densities considered. At 50,000 devices, the measured latency is equal to 0.0266 ms and is approximately equal to a 1.53% increase from the 1,000-device case. In other words, the authentication computations do not show a significant increase in latency when the number of devices increases in this experiment. Operation throughput also demonstrates the same figures that are equal to 44.54–47.55 auths/s. Although a slight decrease was observed compared to the case of 1,000 devices, the

throughput remains stable at about 45.08 auths/s at high device densities. Moreover, all records have been successfully created and written, providing the necessary 100% availability at all densities used in experiments.

IoT Data (first 5 rows):

	result	table	_time	device_id	humidity	temperature
0	_result	0	2026-06-08 04:14:23.972703+00:00	device_0	30.069255	39.459633
1	_result	0	2026-06-08 04:23:16.571646+00:00	device_0	44.215584	23.684204
2	_result	0	2026-06-08 04:47:43.672302+00:00	device_0	30.139667	23.743467
3	_result	0	2026-06-08 04:53:16.276492+00:00	device_0	58.786479	20.686527
4	_result	1	2026-06-08 04:14:24.015479+00:00	device_1	71.134882	27.107757

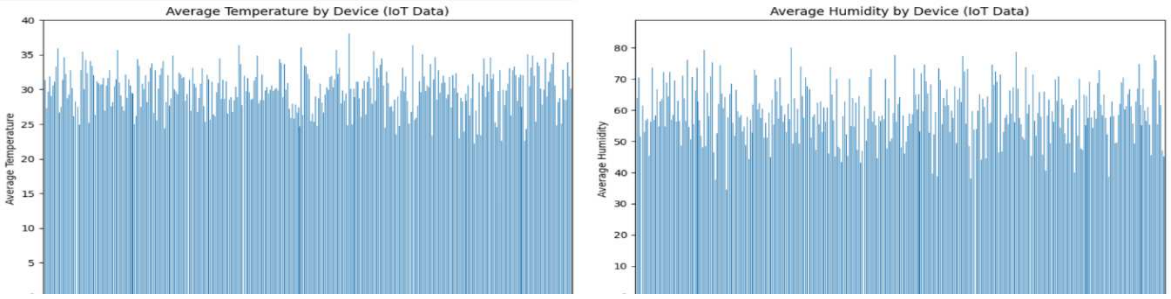


Figure 9a: Visualizing IoT metrics from Influx dashboard

Such IoT data analysis results (figure9a&9b) illuminate the way sensor readings can be logically processed and visualized in order to monitor environments. The dataset used includes readings like device ID, humidity, and temperature, while the sampled dataset shows the differences in readings received from different devices. Comparative analysis of average temperature and humidity for each device is depicted through bar charts, while the results of this analysis help to identify trends and anomalies. Thus, this visualization shows the potential of IoT platforms to turn raw data from sensors into usable information.

	result	table	_time	device_id	humidity	temperature
0	_result	0	2026-06-08 04:15:19.843657+00:00	device_1	89.318788	23.438092
1	_result	0	2026-06-08 04:16:53.410488+00:00	device_1	37.088829	38.626835
2	_result	0	2026-06-08 04:19:02.875936+00:00	device_1	40.940452	36.243387
3	_result	0	2026-06-08 04:20:42.051824+00:00	device_1	62.340736	35.554141
4	_result	0	2026-06-08 04:30:36.065466+00:00	device_1	59.469043	38.748692

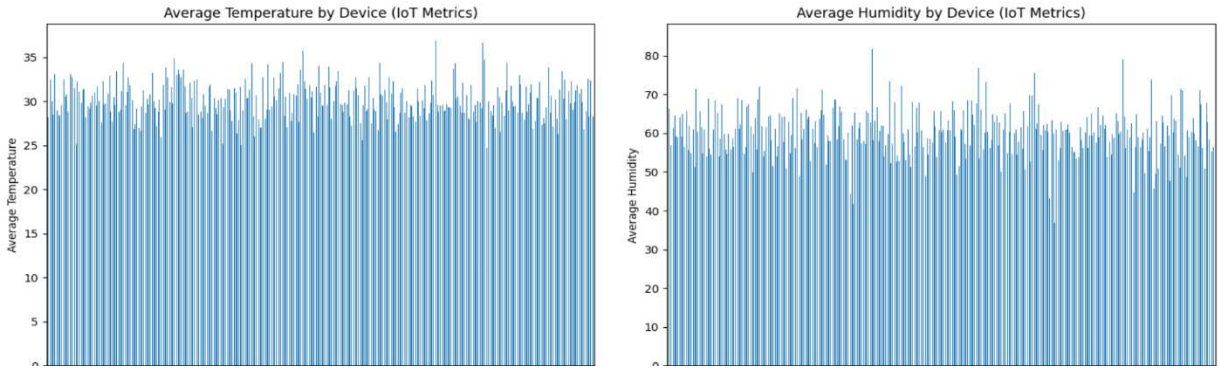


Figure 9b: Visualizing IoT metrics from Influx dashboard

This Cloud-IoT analytics dashboard features metrics of the Internet of Things such as humidity, memory utilization, and temperature. The time series graph displays the change of sensor values over time for a particular period. The dashboard allows one to apply parameters for filtering results, e.g., by measurement type or device ID. The possibility of working with raw data, exporting to CSV, and changing time frames shows how IoT cloud solutions provide means for analyzing and presenting information about the operation of devices and their environments.

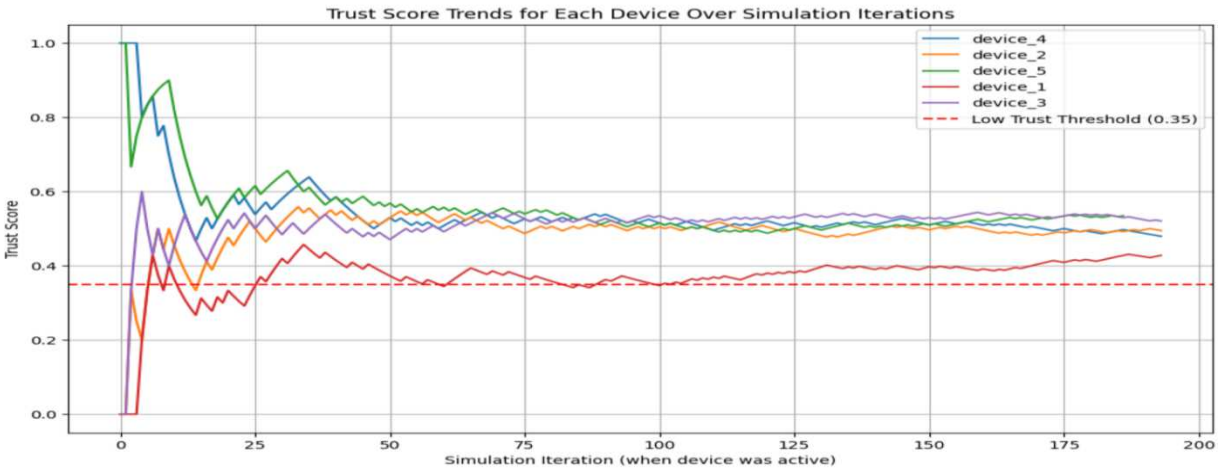


Figure 10: Trust score trends

The following graph (figure10) shows the trends of device trust scores during simulations. It can be seen that different IoT devices have a different level of trust. While some devices show stable high values of trust throughout the whole testing period, others demonstrate consistently low performance. Over time, the differences reduce, and it becomes easier to identify the most trusted devices.

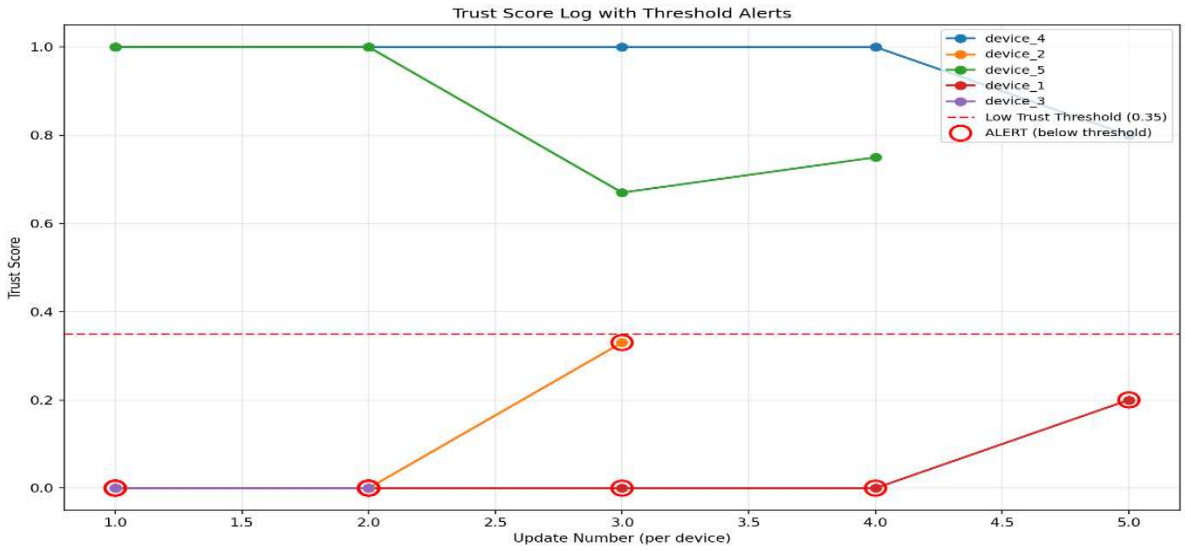


Figure 11: Trust score log with threshold alerts

This infographic of logged trust score (figure11) highlights device monitoring with alert thresholds. One can notice from the line chart the trust scores for several devices as well as when the points have scored below the designated low trust level of 0.35. Various devices like device_1, device_2, and device_3 make the alerts generated, while device_4 and device_5 are more stable. The given example shows that the unique nature of trust metrics helps in problem detection and will ensure the increase of IoT security and stability.

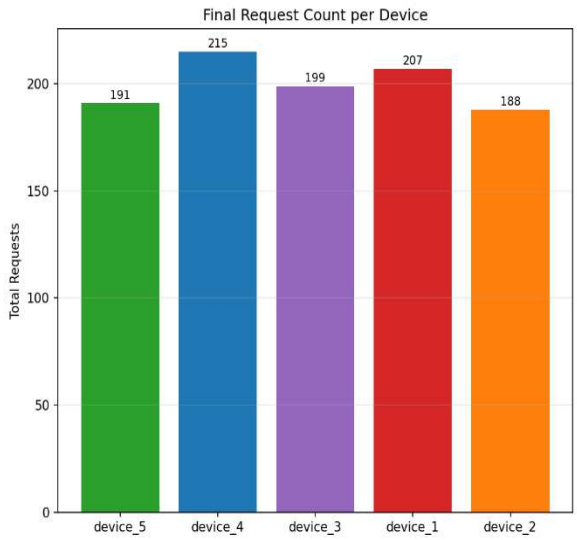


Figure 12a: Final request count per device

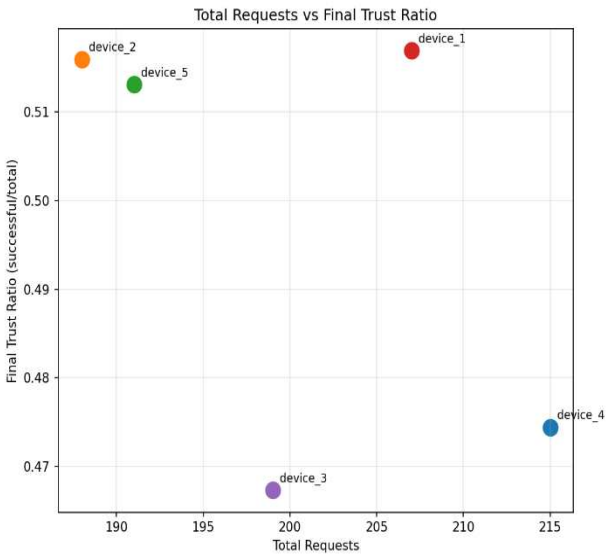


Figure 12b: Total requests Vs Final trust ratio

In the box plot (figure12a & 12b), which shows the distribution for devices and the attack status, device_1 has the highest mean normal-trust score (approximately 0.59) throughout the period of normal operations and the attacks (approximately 0.53), showing that this device is the most attack-resistant. The case for device_4 is the opposite one: it has the lowest mean trust score during attacks (0.47). Likewise, for all devices the spread decreases drastically during attacks when compared to the period of normal operation. This shows that despite device_4 having the total amount of requests of 215, its final trust ratio is one of the lowest; at the same time, device_1 has the highest final trust ratio while receiving a smaller number of requests (207 compared to device_4).

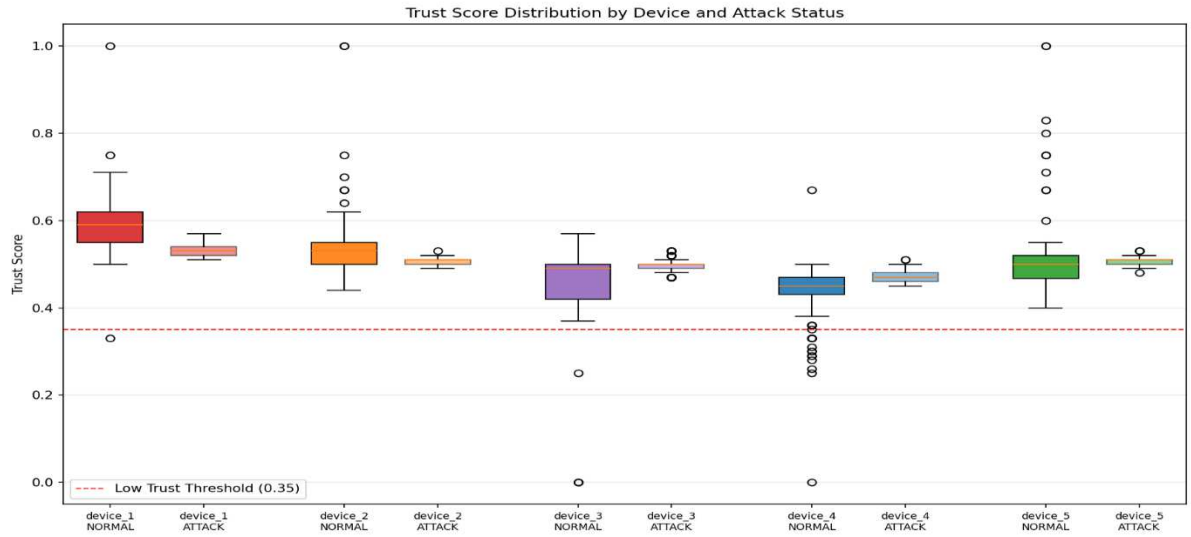


Figure 13: Trust score distribution

The data presented (figure13) utilizes a process of averaging over 10 instances for each piece of technology. This allows for the elimination of short-term discrepancies between instances of data collection with vertical lines representing the first recorded attack for each device. The key facts highlighted in this statement: Each of the devices recorded trust scores that lay evenly within the range of 0.45 to 0.55 early on, indicating that all five devices reached a steady state. Device 1, however, keeps scoring above the average (approximately 0.55 to 0.65) even after the attack starts, clearly declaring it to be the safest device amongst the five technologies monitored. Device 4 shows the lowest trust score in general, coming close to crossing the threshold of 0.35 in the turbulent early period. The start of the attack is marked by the dotted lines that do not cause the graph to change in any way, indicating that the trust score remained unchanged despite the attacks.

An analysis of the experimental results reveals the four main properties of the framework from the point of efficiency. Talking about the security aspects of the proposed framework, the entitlement for an edge-based authentication provides the first step of authorization, while dynamic trust management process is responsible for

latter evaluation of device behavior. From the resilience point of view, the framework runs the process of monitoring of both the resource and security states and initiating procedures of mitigation and recovery when they differ from normal conditions. With the help of quantitative parameters of recovery time and availability, one can prove service continuity by analyzing the framework. In terms of scalability, one can measure the development of performance indicators under increasing number of logical devices from the point of view of authentication latency, throughput, CPU load and memory load. By combining Google Colab and Influx DB Cloud provides continuous operational monitoring of the framework. The outcome of the study supports the joining of the concepts of authentication, dynamic trust measures, cloud storage in real time, visualization technology, and scalability across multiple devices under one comprehensive framework. The results are in tune with the goal of the analysis and show through experimental observations that the suggested framework is capable of functioning securely and consistently, irrespective of the increase in the number of logical devices.

7. CONCLUSION AND FUTURE DIRECTIONS

This study introduced a Cloud-IoT system that was aimed at solving security, surveillance, and scalability issues present in large-scale implementations of IoT with significantly large device density. More specifically, the discussed system brought together edge-based authentication, dynamic trust control, surveillance, attack detection, and cloud monitoring in one cohesive solution. The implementation of the system was done using Google Colab and Influx DB Cloud. The results of the experiments showed that the developed system managed to function well with increasing number of devices from 1,000 to 50,000. The average latency was in the range of 0.0262 – 0.0276 ms, demonstrating limited variations in relation to authentication being processed in accordance with the density of devices under consideration. The throughput was approximately throughout 44.54 – 47.55 auth/sec, while the device records were successfully written to Influx Dashboard Cloud. Furthermore, the real-time trust management experiments revealed that device performance can be assessed through the use of dynamic trust score. If the trust score is below 0.35 threshold, the alert will be raised. Moreover, the integration with Influx Dashboard Cloud guarantees permanent storage and visualization of all related telemetry and security information. The results support the scalability of the suggested framework by providing a quantitative assessment, especially in terms of real-time monitoring, when up to 50,000 logical IoT devices are involved. However, it should be noted that the scalability experiment did not incorporate any test scenarios based on attack injection or failure of the system. This work can be developed in several directions. Firstly, it is planned to carry out controlled security experiments by utilizing authentication flooding, replay attacks, fake device requests, etc. in order to measure detection efficiency, reach false positive and false negative results in numerical terms. Secondly, the framework will be improved allowing the implementation of automatic elimination and recovery from the attack, such as putting the device into an isolated state, the revoking of credentials, the regulation of traffic and service restoration. Thirdly, further studies will involve using CPU load, memory usage, network traffic, and resource consumption across various densities of devices for a good evaluation of scalability. Fourthly, attempts will be made to run experiments involving a larger number of devices and longer run time of experiments to investigate the maximum workload that can be supported. Finally, the proposed method will be compared with other traditional or existing authentication methods for Cloud-IoT to estimate its contribution in terms of latency, throughput, security, resources consumption, resilience, and availability of the solution.

DECLARATIONS

Ethics, Consent to Participate, and Consent to Publish declarations: not applicable.

Data Availability

The datasets generated and analyzed during this study, including the experimental results for IoT devices, are publicly available in the Zenodo repository at [https://doi.org/10.5281/zenodo.23394618]

Research Funding

No funding was received for conducting this research work or for the preparation of this manuscript.

Clinical Trial Number

Clinical trial number: not applicable.

Ethics Approval

This research work does not involve human participants or animals.

Conflict of Interest

The authors declare that they have no conflict of interest regarding the publication of this paper.

REFERENCES

[1] P. Chithaluru, B. V. Jyothi, F. S. Alharithi, W. Ksiazek, M. Ramchander, A. Singh, and R. K. Rachavaram, "A scalable and secure federated learning authentication scheme for IoT," *Sci. Rep.*, vol. 16, Art. no. 7888, Feb. 2026. doi: 10.1038/s41598-026-37541-8

[2] G. Iswarya and C. Vennila, "A secure and scalable blockchain-assisted authentication framework for decentralized IoT data management," *Sci. Rep.*, vol. 16, Art. no. 16359, Apr. 2026. https://doi.org/10.1038/s41598-026-47192-4

[3] J. Alotaibi, "Energy-efficient trust management for secure IoT devices in information-centric wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 19, Art. no. 43, Feb. 2026. https://doi.org/10.1007/s12083-025-02194-3

[4] B. Alotaibi, "A review of resilient IoT systems: Trends, challenges, and future directions," *Appl. Sci.*, vol. 16, no. 4, Art. no. 2079, Feb. 2026. https://doi.org/10.3390/app16042079

[5] S. Ameer, L. Praharaj, R. Sandhu, S. Bhatt, and M. Gupta, "ZTA-IoT: A novel architecture for zero-trust in IoT systems and an ensuing usage control model," *ACM Trans. Priv. Secur.*, vol. 27, no. 3, Art. no. 22, pp. 1–36, Aug. 2024. https://doi.org/10.1145/3671147

[6] C. Liu, R. Tan, Y. Wu, et al., "Dissecting zero trust: Research landscape and its implementation in IoT," *Cybersecurity*, vol. 7, Art. no. 20, 2024. https://doi.org/10.1186/s42400-024-00212-0

[7] B. Gușiță, A. A. Anton, C. S. Stângaciu, et al., "Securing IoT edge: A survey on lightweight cryptography, anonymous routing and communication protocol enhancements," *Int. J. Inf. Secur.*, vol. 24, Art. no. 149, 2025. https://doi.org/10.1007/s10207-025-01071-7

[8] N. Sarwar, R. S. Alharthi, H. M. Mujlid, and F. A. Almalki, "Enhancing IoT communication in fog computing: A lightweight remote user authentication and key management scheme utilizing ECC," *J. Cloud Comput.*, vol. 14, no. 1, Art. no. 86, 2025. https://doi.org/10.1186/s13677-025-00816-y

[9] I. U. Khan, F. M. Khan, Z. A. Haider, and F. Alturise, "Integrating AI, blockchain, and edge computing for zero-trust IoT security: A comprehensive review of advanced cybersecurity framework," *Comput. Mater. Continua*, vol. 85, no. 3, pp. 4307–4344, 2025. 10.32604/cmc.2025.070189

[10] M. Aaqib, A. Ali, L. Chen, et al., "IoT trust and reputation: A survey and taxonomy," *J. Cloud Comput.*, vol. 12, Art. no. 42, 2023. https://doi.org/10.1186/s13677-023-00416-8

[11] M. H. Miraz, K. A. U. Zaman, and M. N. M. Ali, "Decentralised trust and security mechanisms for IoT networks at the edge: A comprehensive review," *EAI Endorsed Trans. Internet of Things*, vol. 11, 2025. https://doi.org/10.4108/eetiot.10996

[12] A. N. Alsheavi, A. Hawbani, W. Othman, X. Wang, G. Qaid, L. Zhao, and M. A. Al-Qaness, "IoT authentication protocols: Challenges, and comparative analysis," *ACM Comput. Surv.*, vol. 57, no. 5, pp. 1–43, 2025. https://doi.org/10.1145/3703444

[13] M. Almutairi and F. T. Sheldon, "IoT–cloud integration security: A survey of challenges, solutions, and directions," *Electronics*, vol. 14, no. 7, Art. no. 1394, 2025. https://doi.org/10.3390/electronics14071394

[14] C. Bicer, I. Murturi, P. K. Donta, and S. Dustdar, "Blockchain-based zero trust on the edge," *arXiv preprint* arXiv:2311.16744, 2023. doi: 10.48550/arXiv.2311.16744

[15] I. Kovacevic, M. Stojkov, and M. Simic, "Authentication and identity management based on zero trust security model in micro-cloud environment," in *Disruptive Information Technologies for a Smart Society*, Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 481–489. https://doi.org/10.1007/978-3-031-50755-7-45

[16] M. A. Allouzi and J. Khan, "Enabling zero trust security in IoMT edge network," *arXiv preprint* arXiv:2402.10389, 2024. doi: 10.48550/arXiv.2402.10389

[17] J. Yalli, M. H. Hasan, T. Low, A. Ibrahim, D. A. Aliyu, U. Maiwada, S. Al-Selwi, and M. Rehman, "A systematic review for evaluating IoT security: A focus on authentication, protocols and enabling technologies," *IEEE Internet Things J.*, vol. 12, no. 21, pp. 18908–18928, 2025. doi: 10.1109/JIOT.2025.3545737

[18] N. Singh, R. Buyya, and H. Kim, "Securing cloud-based Internet of Things: Challenges and mitigations," *arXiv preprint* arXiv:2402.00356, 2024. doi: 10.48550/arXiv.2402.00356

[19] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013. doi: 10.1016/j.comnet.2012.12.018

[20] O. Garcia-Morchon et al., "Security considerations in the IP-based Internet of Things," IETF RFC 8576, Apr. 2019. doi: 10.17487/RFC8576

[21] M. Conoscenti, A. Vetrò, and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, 2016, pp. 1–6. doi: 10.1109/AICCSA.2016.7945805

[22] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016. doi: 10.1109/ACCESS.2016.2566339

[23] M. Wazid et al., "Design of secure user authenticated key management scheme for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018. doi: 10.1109/JIOT.2017.2780232

[24] A. Braeken, "Efficient and secure authentication protocol for IoT," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10229–10239, Dec. 2019. doi: 10.1109/JIOT.2019.2941199

[25] Y. Lu and L. Da Xu, "Internet of Things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2103–2115, Apr. 2019. doi: 10.1109/JIOT.2018.2869847

[26] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K. K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in IoT," *J. Netw. Comput. Appl.*, vol. 103, pp. 194–204, Feb. 2018. doi: 10.1016/j.jnca.2017.10.011

[27] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero trust architecture (ZTA): A comprehensive survey," *IEEE Access*, vol. 10, pp. 57143–57179, 2022. doi: 10.1109/ACCESS.2022.3174679

[28] M. A. Azad, S. Abdullah, J. Arshad, H. Lallie, and Y. H. Ahmed, "Verify and trust: A multidimensional survey of zero-trust security in the age of IoT," *Internet of Things*, vol. 27, Art. no. 101227, 2024. doi: 10.1016/j.iot.2024.101227

[29] T. Wan, B. Shi, and H. Wang, "A continuous authentication scheme for zero-trust architecture in industrial Internet of Things," *Alexandria Eng. J.*, vol. 122, pp. 555–563, May 2025. https://doi.org/10.1016/j.aej.2025.03.012

[30] Z. Ma, H. Wei, J. Jiang, B. Wang, H. Wang, and Z. Di, "A lightweight zero-trust authentication architecture for IoT via unified enhanced FAST-SM9 and dynamic re-authentication," *PLOS ONE*, vol. 20, no. 10, Art. no. e0332943, Oct. 2025. doi: 10.1371/journal.pone.0332943
[31] V. Padmavathi and R. Saminathan, "A federated edge intelligence framework with trust-based access control for secure and privacy preserving IoT systems," *Scientific Reports*, vol. 15, Art. no. 35832, 2025, doi: 10.1038/s41598-025-19712-1.
[32] S. Deepthi, K. A. Tran, and G. Deepa, "Secure device authentication with blockchain-based trust technique for IoT," *SN Computer Science*, vol. 7, no. 7, Art. no. 744, 2026, doi: 10.1007/s42979-026-05328-3.
[33] P. Suganya and C. H. Pradeep Reddy, "Integration of 6LoWPAN and cloud data centers: The role of link capacity between sensor networks and cloud data centers," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 2158–2164, 2020, doi: 10.35940/IJITEE.A4715.119119.
[34] P. Suganya and C. H. Pradeep Reddy, "A survey and analysis on various objective functions defined for RPL in 6LoWPAN," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 403–411, 2019. 10.35940/ijrte
[35] P. J. Kumar *et al.*, "A suite of load balancing algorithms and service broker policies for cloud: A quantitative analysis with different user grouping factor in cloud," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 2983–2989, 2019. doi: 10.35940/ijrte. A1939.058119
[36] P. J. Kumar, N. Sivakumar, J. Prabhu *et al.*, "Analysis of heterogeneous device characteristics in round robin-based load balancing algorithm with closest data center as service broker policy in cloud," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 1627–1630, 2019. doi: 10.35940/ijitee. I8273.078919
[37] P. J. Kumar, J. Prabhu, N. Sivakumar *et al.*, "An analysis on ECSE load balancing algorithm with CDC as a service broker policy in cloud with heterogeneous communication and device characteristics," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 1631–1635, 2019. doi: 10.35940/ijitee. I8274.078919
[38] P. J. Kumar, P. Ilango *et al.*, "MQRC: QoS-aware multimedia data replication in cloud," *International Journal of Biomedical Engineering and Technology*, vol. 25, pp. 250–266, 2017, doi: 10.1504/IJBET.2017.087727.

Author Contribution Statement

Revathi R: developed the research, created the proposed work, performed the literature search, and drafted the manuscript including the figures, and wrote the methodology and results sections.
Dr. P J Kumar: reviewed the manuscript and provided significant intellectual contributions. All authors wrote, read and approved the final manuscript.

Authors Biography

Ms. R. Revathi is pursuing Ph.D in School of computer science engineering and information systems at VIT University, Vellore campus from July 2024 and designated as Teaching cum Research Assistant. She has 10+ Years of Teaching Experience in Engineering Colleges. She secured distinction in both UG and PG degree. She has handled various Computer science & Information Technology subjects like Communication Networks, Computer Architecture, Mobile Communication, Design Analysis & Algorithm, Object Oriented Programming, Software Project & Management, Object Oriented Analysis & Design, Software Testing, Communication Networks, etc. She has published 3 papers in UGC International Journals and 3 papers in Scopus Indexed Journals and also presented 14 papers in various National and International Conferences. She has completed 22 FDPs, 12 online courses and certified as 5S-Internal Auditor. She was awarded as Best Young faculty Award, by SKCT in December 2020.

Dr. P. J. Kumar is an Associate Professor (Grade 2) in the School of Computer Science Engineering and Information Systems, Department of Smart Computing, at Vellore Institute of Technology (VIT), India. He received his B.E. degree in Electrical and Electronics Engineering from Madras University in 1998, his M.E. degree in Computer Science and Engineering from Anna University in 2008, and his Ph.D. in the areas of MANET and Cloud Computing from VIT in 2018. His research interests include mobile ad hoc networks, cloud computing, machine learning algorithms, the Internet of Things, 6LoWPAN, and distributed systems. He has published research across these domains and holds an Indian patent titled "Method and system for improving replica allocation time in mobile ad hoc network" (Application No. 201841020867). Dr. Kumar has received the VIT Research Award for five consecutive years and serves as a reviewer for IEEE Access, IEEE conferences, Inder Science Publishers, and Hinda Wi journals.