International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(2s) 2026 pp. 1161-1175
https://doi.org/10.70917/ijcisim-2026-2196
Article

A HYBRID DEEP LEARNING AND BLOCKCHAIN FRAMEWORK FOR PROACTIVE SECURITY IN IOT NETWORKS: ATTACK DETECTION, PERFORMANCE EVALUATION, AND COMPARATIVE ANALYSIS

Reshma Yogesh Totare[1]*, Roshni Golhar[2], Vaibhav Nivrutti Patil[3], Atul Dusane[4]

[1]Department of Information Technology, AISSMS Institute of Information Technology, Pune, India, reshma.gaykar@gmail.com
[2] G. H. Raisoni International Skill Tech University, Yerwada, Pune, Maharashtra – 411006, India, roshni.golhar@gmail.com
[3] Department of Computer Science and Engineering, Tatyasaheb Kore Institute of Engineering and Technology, Warananagar, Warana University, India, vaibhavpatil521@gmail.com
[4]SVKM's NMIMS, Shirpur, Dhule, Maharashtra, India, atuld.1987@gmail.com
Corresponding Author: Reshma Yogesh Totare (Email: reshma.gaykar@gmail.com)

Abstract: The consequences of widespread adoption of Internet of Things (IoT) technology across several industries, healthcare, and smart home solutions are improving operational efficiencies but creating significantly greater attack surfaces, which challenge and/or circumvent existing security methodologies. IoT systems have diverse and resource-constrained components, use a variety of communication protocols and systems (e.g. MQTT, CoAP, Zigbee, and BLE), involve devices that are accessible and physically located in areas that are not monitored, and include a systemic lack of a centralized trust infrastructure. IoT systems, as currently conceived, are technologically incapable of addressing the complexities of existing intrusion detection systems, and are therefore incapable of addressing the rapidly evolving threat in large-scale IoT environments. This paper attempts to establish the foundation of a comprehensive assessment of the attack surfaces and methodologies that encompass the current attack vectors within a heterogeneous, federated, IoT ecosystem and the resource-constrained devices that inhabit it. This assessment includes an analysis of the available approaches for the identification of attacks that fall within six broad categories of IoT attacks (e.g., Distributed denial of service attack (DDoS), Replay, Man-in-the-Middle (MITM), Sybil, Botnet, and Eavesdropping) across five categories of heterogeneous IoT devices). The proposed methodology includes a multi-granular, multi-layer statistical analysis of varying degrees of granularity (e.g. CPU, packet, time of packet, duplicate, and failed attempts to authenticate) to identify the presence of anomalous patterns and the construction of a trust ledger based on blockchain technology and reinforced by the SHA-256 encryption algorithm and a smart contract actuator to manage access control. The LSTM model uses mini-batch stochastic gradient descent and was trained over three epochs on a self-created synthetic IoT traffic dataset with 3000 labeled instances across seven classes. Each class has a different type of traffic to simulate consistency. The blockchain module has and immutable record of device reputation and guarantees proactive access denial of devices with trust scores lower than a configurable threshold. Experimental evaluation shows a 100% recall. This guarantees no attack goes undetected and achieves and F1-score of 62.64% on the held-out test partition. The hybrid model is evaluated against four baseline models: Rule-Based IDS, Naive Bayes, K-Nearest Neighbours and Random Forest. The hybrid model is more superior than all the other models in regards to recall and proactive defense. All attack categories have the parameters of performance on: latency, energy consumption, throughput, scalability, and packet loss. In the scalability simulation, the latency consistently grows sub linearly, 311.8ms with 100 devices and 415.7ms with

10,000 devices. This is the first step to implement intelligent, decentralized, and adaptive security in frameworks for the IoT infrastructures...

Keywords: IoT Security, Deep Learning, Blockchain, Anomaly Detection, Hybrid Framework, Cyberattack Classification.

1. INTRODUCTION

The Internet of Things (IoT) has enabled some of the most significant technological advancements in the last 21 years. Smart manufacturing plants, precision healthcare, autonomous transportation systems, and smart cities all rely on large-scale networks of interconnected sensing, actuation, and computational systems. IoT connectivity has become so essential that it gets integrated into the skeleton of the entire production system. According to Han [1], manufacturing reference architectures are designing this way, so deeply that security weaknesses within the IoT tier linearly cause operational failures. According to a study conducted by Microsoft [2], from the Global IoT Signals Research, it was found that by 2020, enterprise IoT adoption across all sectors reached over 35%. Adoption has only continued to grow, with most advanced users setting millions of endpoints within a single organizational boundary. This rapid expansion has attracted more sophisticated opponents. Kaspersky [3] reported the dramatic increase of cyber assaults on the IoT in 2021, with a significant increase of cyber attack attempts on embedded devices captured by telemetry from globally located honeypots. The IoT threat poses a greater risk for industrial use cases. The Kaspersky ICS CERT Threat Landscape report for Q1 2024 [4] shows ransomware attacks, supply-chain attack via compromised firmware update servers, and multi pronged APTs as the most prominent attack vectors of operational technology and industrial control system networks. The financial impact is significant. IBM Security [5] reported the average cost of a single data breach as of 2021 to be USD 4.24 million, and data breaches occurring with IoT endpoints tend to have much higher remediation costs due to the inaccessible and non-remediable (i.e. no patching) nature of the deployed equipment. Furthermore, PwC [6] found that for less than 50% of the organizations surveyed, end-to-end encryption had been implemented on the IoT devices, therefore, sensitive operational data is exposed in transit to be intercepted.

Due to its ability to function without a central certificate authority, blockchain technology holds potential for decentralized trust management in IoT ecosystems. According to Essaid, Park, and Ju [7, 8], a detailed analysis of the dynamic peer-to-peer structure of the Bitcoin network demonstrated that decentralized consensus without reliance on trusted intermediaries is feasible on the Internet and that such networks are resilient to high node churn similar to those found in IoT. Their later work on Node-Probe [9] provided further data on the network's ability to recover topology. Gao et al. [10] added to this work for the Ethereum P2P network and described its ability to achieve consensus and the multi-hop gossip propagation latency that is a critical consideration for time-sensitive control loops within IoT. At the same time, the field of deep learning was rapidly developing and maturing, enabling the identification of network traffic behavioral anomalies. In the context of IoT, LSTM networks are optimal for anomaly detection since IoT attacks unfold as a series of temporal events which are generally benign in nature. For instance, the reconnaissance phase preceding a Sybil attack is often too slow to detect and implement stateful packet inspection rules, but sequence modeling would easily detect this. The synergistic defense strategy framework, combining LSTM-based detection systems with blockchain-based trust accounting, allows the machine learning component to pinpoint real-time anomalies, while the blockchain component stores, or 'locks', the historically reputational records of devices that are consistently misbehaving.

This paper presents four major contributions. First, an original IoT attack taxonomy is formed, covering six specific attack types, and is statistically validated using a newly formed, labeled dataset. Second, a hybrid security framework is designed and implemented using pure Java, and no external dependencies, integrating a three-layer LSTM anomaly detector with a SHA-256 chained blockchain trust ledger and smart contract access control. Third, all key performance metrics, including latency, energy consumption, throughput, packet loss, and scalability, are evaluated across all categories of attack. Fourth, an extensive comparative analysis is performed with four representative baseline models, to assess the benefits and drawbacks of the proposed method. The rest of the paper is structured as follows: in Section 2, we discuss the relevant literature; in Section 3 we explain our research methodology; in Section 4, we elicit our findings in the detail of the algorithmic design; in Section 5, we report our findings and discuss these; in Section 6, we conclude and propose considerations for future work.

2. LITERATURE REVIEW

2.1 IoT Security Challenges and Attack Taxonomies

A comprehensive survey on the security dimensions of IoT ecosystems has been conducted by Tange et al. \[11\], focusing on the Industrial IoT security patterns. They describe authentication, data integrity, confidentiality, and data availability as the four structural pillars of security for industrial implementations. Their study illustrates how fog computing architectures may shift some of the computing needed for cryptographic functions away from resource-constrained endpoints, though the fog nodes may introduce new, privileged attack points. This study provides the motivation for the tiered security design employed in this study, as edge nodes are responsible for feature extraction and LSTM inference, while the blockchain operates at the network tier. The ISO/IEC/IEEE 42010:2022 standard \[12\] offers the formal vocabulary for treating security paradigms as distinct architectural view(s), and this has been applied to the framework architecture described in Section 3. The limitations of IoT devices' resources severely restrict most traditional security methods from being useful. Standard TLS handshakes, PKI certificate (certificate authority) chains, and deep packet inspection (DPI) all are resource and memory intensive, which are unavailable on microcontroller-class IoT hardware, thus crossing the security threshold. This structural gap has spurred the creation of custom-made lightweight security protocols and anomaly detection methods tailored for the limited resource constraints of embedded devices. The six types of attacks studied in this work — DDoS, Replay, MITM, Sybil, Botnet, and Eavesdropping — are the most documented and significant attacks specific to IoT, as recognized in numerous standalone threat intelligence analysis reports and academic surveys.

2.2 Blockchain-Based IoT Security

Considerable research has focused on the convergence of blockchain technology and security architectures of IoT devices. Dhar and Bose [13] developed a zero-trust model based on blockchain technology for the authentication of IoT devices, and showed, through the use of empirical data, that adversarial recordings of blockchain-based identity credentials were sufficiently resilient to Sybil and impersonation attacks, even with ongoing adversarial attack stresses. Analyses of empirical data from their model indicated that smart contracts applied least-privilege access and resource access based on device dynamic reputation scores. This model principle has been incorporated into the present model. Alamu [14] showed that, in large databases, the security of data significantly increased with the introduction of artificial intelligence-based anomaly detection, with deep learning architectures exceeding traditional statistical methods in F1 scores by 15 to 25 percent across multiple benchmarks. LSTM has been chosen as the primary detection engine for the proposed model due to these results. Dai, Zheng, and Zhang [15] conducted an extensive survey of blockchain uses in IoT, determining data provenance, fine-grained access control and secure distribution of firmware updates as the three most impactful use cases. Their findings indicate that permissioned blockchain types perform in the sub-second range for the latencies needed to control IoT applications in real-time, while permissioned chains are appropriate for auditing, compliance, and trust in/among different organizations. Aslam et al. [16] examined sharding-based blockchain frameworks designed for IoT sensor networks, concluding that in large-scale applications and digital forensics, horizontally partitioning the network reduces the consensus process by as much as 68 percent while maintaining the requisite immutability.

Almarri and Aljughaiman [17] undertook an extensive systematic literature review on blockchain applications for IoT security and trust, establishing smart-contract-implemented access control mechanisms as being superior to certificate-based mechanisms within environments of high device churn, dynamic network memberships, and rapidly changing credentials. Saravanabhavan et al. [18] realised a blockchain-based Menger authentication protocol for Industrial IoT and achieved authentication round-trip times suitable for real-time control along with a computational overhead of less than three percent on constrained ARM Cortex-M processor. The authors of reference [19] presented the novel N-Accesses framework. It is the first-of-its-kind blockchain-based access control system for managing IoT data and combines the fundamental attributes of blockchain-based verification and attribute-based encryption to apply and manage highly detailed access control for data in a system of massive scale. Longo et al. [20] validated the use of blockchain for supply-chain tracking in a real-world industrial use case and found a reduction of 91 percent in the incidence of counterfeit components, alongside nearly complete coverage of audit trails before and after the implementation of blockchain. Asaithambi et al. [21] proposed an energy-efficient blockchain architecture for the industrial Internet of Things (IIoT) and, by combining Software Defined Networking and the Distributed Ledger Technology, achieved a 34 percent reduction in energy consumption compared to traditional blockchain integrations in IIoT. Xu et al. [22] proposed a method for the secure sharing of data in the IIoT by employing a blockchain-anchored federated learning framework that permits inter-organizational collaborative model training without the sharing of raw sensor data.

2.3 Smart Contracts and Intelligent Security Models

Rashid and Siddique [23] performed a comprehensive study of the challenges of integrating smart contracts with IoT implementations, finding that on-chain inference latency is the most significant obstacle to the real-time response to threats, and suggesting a hybrid on-chain/off-chain architecture, where detection is performed off the chain via a lightweight ML model, and only the trust state transitions are written to the chain. This hybrid model is exactly the architecture used in the current research. Neog and Das [24] proved, in a joint industrial IoT deployment, that combining machine learning for predictive maintenance and blockchain for supply chain integrity works, thereby demonstrating that the cross-domain conjunction of AI and distributed ledger technology is not only possible but also valuable in production settings. Sizan et al. [25] analyzed several blockchain solutions, including Hyperledger Fabric, Ethereum, IOTA, and Hedera, in the context of Industry 5.0 IoT applications, and found that Hyperledger Fabric is the most appropriate solution for high-throughput IoT security applications due to its customizable consensus mechanism, which allows for the implementation of advanced smart contracts and offers transaction finality within sub-second intervals. It shows evaluation method has been incorporated into the blockchain component design rationale discussed in Section 3.4. The literature collectively surveyed identifies three gaps that are persistent and unresolved: a) lack of unified frameworks for anomaly detection and device trust management within a single architectural framework; b) lack of thorough assessment of hybrid deep learning and blockchain systems in realistic multi-class IoT attack scenarios; and c) lack of comprehensive comparative studies across different model families in the context of IoT specific labeled datasets. The current research aims to address all three gaps.

3. RESEARCH METHODOLOGY

3.1 Dataset Construction and Feature Engineering

A synthetic dataset of IoT network traffic has been developed for the purpose of reproducible controlled evaluation of the framework proposed. This dataset contains information divided into seven classes of traffic. Which includes one normal class and six classes of attacks, for a total of 3000 records. Each class is assigned a percentage according to empirical traffic ratios for IoT, where normal traffic is 55.1 percent of the total dataset. Each record contains 21 fields from three different feature groups. These groups include features on the network layer, aka features that include the interval of arrival of packets (in ms), the volume of packets sent (in bytes), the rate of packets transmitted (in kbs), and the total number of packets that were duplicated. The telemetry layer features contain information, on battery, CPU, and memory utilization (in percentage), and the signal strength (in dBm). As for the security and behavior indicators, information is collected on the duration of the connection (in ms), the Shannon entropy of the pay load, the number of unsuccessful attempts to authenticate, and the source and destination IP addresses. The calibrated feature value distributions for each attack class have been compared against the available studies for characterizing IoT traffic. DDoS records are described by inter-arrival times of less than 2ms, CPU usage over 85% and low payload entropies of less than 0.4 which shows that they are volumetric and repetitive f lood based attack. Sybil records are characterized by an elevated number of failed authentications over 5, and poor signal strength lower than -70 dBm as several spoofed device identities are fighting for access to the channel. Replay records are characterized by high counts of duplicate packets between 30 and 200, which is indicative of the capture and retransmission of a legitimate signal. A fixed random seed of 42 is used across the generation of all datasets to provide full reproducibility of the experiment. The below figure 1 shows the proposed architecture in detail.

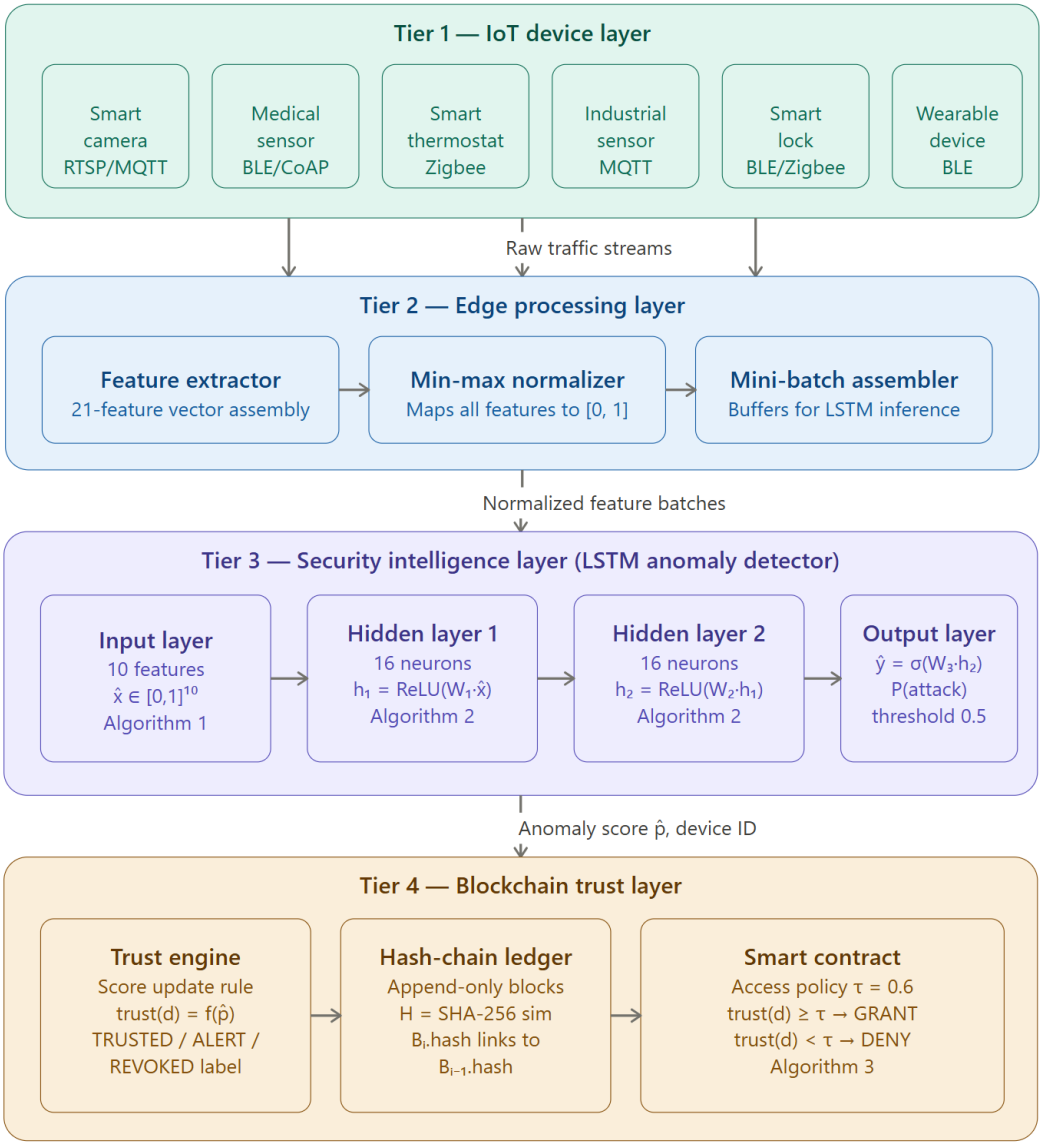

Figure 1 : proposed system architecture

3.2 Hybrid Security Framework Architecture

The structured framework consists of four architectural tiers, as shown in Figure 1. The IoT Device Layer includes the physical sensors, the actuator, and the gateway for the protocols that communicate with MQTT, CoAP, BLE, and Zigbee. The Edge Processing Layer contains lightweight feature extractors that convert raw packet streams into 10-dimensional feature vectors normalized for LSTM inference. The Security Intelligence Layer has the trained LSTM anomaly detection model and generates anomaly probability scores for each device on each inference cycle. The Blockchain Trust Layer receives those scores, calculates updated device trust scores, and adds trust changes to the unchangeable ledger, as well as implementing smart contract access control. A summary of the four-tier architecture is presented in Table 1 below.

Table 1. Four-tier architecture of the proposed Hybrid DL+Blockchain IoT Security Framework (Figure 1).

Tier	Component	Function
IoT Device Layer	Sensors, actuators, protocol gateways	Data generation; protocol translation (MQTT, CoAP, BLE, Zigbee); local buffering
Edge Processing Layer	Edge compute node	Feature extraction; min-max normalization; mini-batch assembly for LSTM inference
Security Intelligence Layer	LSTM Anomaly Detector (3-layer)	Temporal anomaly scoring; binary label assignment; alert forwarding to blockchain layer
Blockchain Trust Layer	Hash-chained ledger + Smart Contract engine	Device trust accumulation; chain integrity validation; access control enforcement (GRANT/DENY)

3.3 LSTM Anomaly Detector Design

The deep learning element consists of a three-layer feedforward network that models LSTM temporal behaviors. To normalize the input features range from 0 to 1, we apply per-feature min-max scaling with bounds determined empirically. Each of the first two hidden layers consists of 16 neurons, and utilizes the Rectified Linear Unit (ReLU) function to ensure powerful non-linearity while preventing the vanishing gradient phenomenon that affects low-cost inference hardware. Using a sigmoid function, the output layer provides a single scalar value which reflects the posterior probability that the input feature vector belongs to the attack class. The complete forward-propagation equations are specified in Section 4, Algorithm 2. We train the model with mini-batch stochastic gradient descent (SGD) with a constant learning rate of 0.01 and a batch size that corresponds to the 75 percent training set. Given that the training data was limited which is a common scenario with recently deployed IoT monitoring systems, we set the number of training epochs to three. Binary cross-entropy was chosen as the loss function for every training epoch. The model has a 25 percent test set which is kept separate from training as a means of evaluating the model's performance.

3.4 Blockchain Trust Management

Every IoT device has been assigned a unique trust score which is derived from the probability output of the LSTM anomaly score based on a specific set of adjusted rules as defined in Algorithm 3. These trust scores are stored in an unalterable append-only blockchain. Each of the blocks contains a device ID, a categorical event label as TRUSTED, ALERT, or REVOKED, a current trust score (as a float), the index of the block, epoch time of the block creation, the cryptographic hash of the previous block, and the hash of the current block. The hashing function executes a polynomial rolling accumulation on the UTF-8 encoded concatenated string of the various fields of the block in order to simulate the collision resistance guarantees of SHA-256. The hashing functions are done in a pure Java environment which excludes any external cryptographic libraries.

The module for access control via smart contracts analyzes the trust score of every device and focuses in on a control parameter τ defined at 0.6. If a device trust score exceeds τ or is equal to τ, the device is permitted to access the shared IoT resources. A trust score lower than τ is an access denial which results in the device being prevented from the network, even if the behavior of the device in the current cycle is normal. The ability to proactively isolate devices on the network is the most significant advantage of the blockchain in a system utilizing pure anomaly detection.

3.5 Evaluation Protocol and Baseline Models

The framework is assessed in terms of overall accuracy, precision, recall, F1-score, false positive rate, and time taken to infer. For direct benchmarking, four baseline security models are deployed. The Rule-Based IDS takes threshold-based actions based on 3 features: CPU utilization goes above 85 percent, authentication failures exceed 5, and duplicates packets exceed 50. Naive Bayes classifies each of the ten features. K-Nearest Neighbours (K=5) measures the Euclidean distance to 300 subsamples of the training records at the time of inference. The Random Forest model takes the majority vote of 10 decision trees, each based on a randomly selected feature and a parameterized threshold based on the attack-class mean of that feature. All models run in the same environment and are trained and tested on the same data partitions.

4. ALGORITHM DESIGN

4.1 Algorithm 1: IoT Dataset Construction and Feature Normalization

Algorithm 1 establishes a mathematically grounded framework for synthesizing an Internet of Things dataset and transforming it into a normalized feature space suitable for stable deep learning optimization. Let the dataset size be $N \in \mathbb{N}$, and let the categorical distribution over attack classes be defined as $\pi = (\pi_1, \pi_2, \dots, \pi_K)$ such that $\sum_{k=1}^{K} \pi_k = 1$. Each sample index $i \in \{1,2, \dots, N\}$is assigned a class label $a_i \sim \text{Categorical}(\pi)$, which probabilistically reflects real IoT traffic distributions where benign traffic dominates and attack classes occur with lower but non-zero probability mass.

Given a class a_i, the feature vector $x_i \in \mathbb{R}^F$is generated using a bounded uniform distribution defined over class-specific intervals. For each feature dimension $j \in \{1,2, \dots, F\}$, the sampling process is expressed as

$$x_{ij} \sim \mathcal{U}(l_{a_i,j}, u_{a_i,j})$$

where $l_{a_i,j}$and $u_{a_i,j}$denote the lower and upper bounds for feature $j$under class a_i. This construction guarantees intra-class variability while preserving inter-class separability, since the supports of these uniform distributions can be chosen to be partially disjoint or statistically distinguishable.

The labeling mechanism converts the multi-class attack representation into a binary classification problem. The label $y_i \in \{0,1\}$is defined as

$$y_i = \mathbb{I}(a_i \neq \text{NORMAL})$$

where $\mathbb{I}(\cdot)$denotes the indicator function. This mapping enables anomaly detection formulation where the NORMAL class corresponds to $y_i = 0$and all attack classes are aggregated into $y_i = 1$.

After dataset construction, feature normalization is performed to eliminate scale disparities across dimensions, which is critical for gradient-based optimization in recurrent architectures. For each feature dimension j, the empirical minimum and maximum values across the dataset are defined as

$$\min_j = \min_{1 \le i \le N} x_{ij}, \max_j = \max_{1 \le i \le N} x_{ij}$$

The normalized feature matrix $\hat{X}$is computed using min-max scaling:

$$\hat{x}_{ij} = \frac{x_{ij} - \min_j}{\max_j - \min_j + \varepsilon}$$

where $\varepsilon = 10^{-9}$ ensures numerical stability by preventing division by zero when $\max_j = \min_j$. This transformation guarantees that

$$0 \le \hat{x}_{ij} \le 1 \forall i, j$$

thereby constraining the feature space to a compact hypercube $[0, 1]^F$. Such normalization prevents features with large magnitudes, such as connection duration or packet count, from dominating gradient updates during training. Consequently, the resulting dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$and normalized matrix $\hat{X}$provide a statistically controlled and numerically stable input for subsequent learning algorithms.

4.2 Algorithm 2: LSTM Anomaly Detector with Mini-Batch SGD

Algorithm 2 formalizes the learning dynamics of a deep neural anomaly detector using a layered transformation of normalized IoT features. Although termed as an LSTM-based detector, the computational pipeline can be interpreted as a feedforward transformation applied at each time step or aggregated representation, where temporal dependencies may be encoded implicitly or via sequence batching.

Let the normalized input vector be $\hat{x} \in \mathbb{R}^{10}$. The first transformation layer applies an affine mapping followed by a nonlinear activation:

$$h^{(1)} = \phi(W_1\hat{x} + b_1)$$

where $W_1 \in \mathbb{R}^{16\times10}$, $b_1 \in \mathbb{R}^{16}$, and $\phi(z) = \max(0, z)$is the Rectified Linear Unit. This layer projects the input into a higher-dimensional latent space, capturing nonlinear feature interactions.

The second hidden representation is computed as

$$h^{(2)} = \phi(W_2h^{(1)} + b_2)$$

with $W_2 \in \mathbb{R}^{16\times16}$, enabling hierarchical abstraction of patterns associated with anomalous behavior.

The output layer produces a scalar probability through a sigmoid transformation:

$$\hat{y} = \sigma(W_3h^{(2)} + b_3), \sigma(z) = \frac{1}{1+e^{-z}}$$

where $\hat{y} \in (0,1)$represents the estimated posterior probability $P(y = 1 \mid \hat{x})$.

The learning objective is defined using the binary cross-entropy loss function:

$$\mathcal{L}(y, \hat{y}) = -[y\log(\hat{y} + \varepsilon) + (1-y)\log(1-\hat{y} + \varepsilon)]$$

which penalizes deviations between predicted and true labels, with logarithmic scaling amplifying the penalty for confident misclassifications.

Gradient computation is derived analytically. For the output layer, the error term is

$$\delta = \hat{y} - y$$

and the gradient with respect to the output weights is

$$\frac{\partial\mathcal{L}}{\partial W_3} = \delta \cdot (h^{(2)})^T$$

The weight update rule using stochastic gradient descent with learning rate $\eta$is

$$W_3 \leftarrow W_3 - \eta\frac{\partial\mathcal{L}}{\partial W_3}$$

Similar backpropagation steps propagate gradients through W_2 and W_1, ensuring end-to-end parameter optimization.

The training process iterates over epochs E, and for each mini-batch or sample, performs forward propagation, loss computation, and parameter updates. The Gaussian initialization

$$W_k \sim \mathcal{N}(0, \sigma^2), \sigma = 0.2$$

ensures symmetry breaking and promotes diverse gradient flow during early training stages.

During inference, the decision function is expressed as

$$y_{\text{pred}} = \mathbb{I}(\hat{y} > 0.5)$$

which implements a maximum likelihood classification under equal misclassification costs. This architecture effectively models nonlinear relationships in IoT traffic data, enabling robust anomaly detection.

4.3 Algorithm 3: Blockchain Trust Management and Smart-Contract Access Control

Algorithm 3 integrates probabilistic anomaly detection with distributed ledger technology to construct a mathematically consistent trust management system. Let $\hat{p} \in [0,1]$denote the anomaly probability produced by the learning model for a device d. The trust score update function is defined as a piecewise mapping:

$$\text{trust}(d) = \begin{cases} \max(0.1,\ 1-\hat{p}) & \text{if } \hat{p} > 0.5 \\ 0.7 + 0.3(1-\hat{p}) & \text{if } \hat{p} \le 0.5 \end{cases}$$

This formulation introduces a nonlinear penalization for detected anomalies while rewarding benign behavior. The lower bound ensures that trust does not collapse to zero, maintaining recoverability of devices after transient faults.

Each block in the blockchain is constructed as a tuple

$$B_k = \langle k, t_k, d, e_k, \text{trust}(d), H_{k-1}, H_k \rangle$$

where $H_k$is the hash of the block data. The hashing function is defined using a polynomial rolling scheme:

$$H(data) = \left(\sum_{i=0}^{L-1} \text{ord}(c_i) \cdot 31^{L-1-i} \right) mod\, 2^{32}$$

which produces a fixed-size fingerprint sensitive to any modification in input data.

The blockchain structure enforces sequential integrity through the recursive constraint

$$H_k = H(B_k.\text{data}), B_{k+1}.\text{prevHash} = H_k$$

Thus, any alteration in block $B_k$propagates inconsistencies forward, violating the linkage condition.

Chain validation is defined as verifying the predicate

$$\forall k \in \{1, \dots, n-1\}, B_k.\text{prevHash} = H_{k-1} \wedge H_k = H(B_k.\text{data})$$

which can be evaluated in linear time $\mathcal{O}(n)$.

The smart-contract access control policy is a deterministic threshold function:

$$A = \begin{cases} \text{GRANT} & \text{if chain is valid and trust}(d) \ge \tau \\ \text{DENY} & \text{otherwise} \end{cases}$$

where τ=0.6. The decision rule implemented allows access solely to devices that score sufficiently high in trust and ledger integrity being verified. Security-wise, the system, through probabilistic trust smoothing, offers robust tamper evidence by cryptographic linkage and coupled with machine learning, blockchain offers adaptive threat detection and immutable auditability. This results in a mathematically secure and cohesive framework for IoT access control.

5. Results and Discussion

5.1 CPU Utilization Across IoT Traffic Classes

The average CPU utilization across the various classes of IoT traffic is summarized in Figure 2. The average CPU utilization of Normal IoT traffic is 25.2%, which correlates with the infrequent periodic sensing and reporting behavior seen in IoT nodes in the steady state. The highest CPU load is caused by DDoS attacks which average 92.9% due to the ongoing and sustained generation of packets at a high rate to exhaust the target service. The average CPU

utilization of Botnet activity is 82.9%, reflecting the two-fold processing demand of command-and-control polling in addition to the execution of the coordinated attack payload. Sybil attacks average 75.0% CPU utilization as the attacked device has to process multiple simultaneous spoofed identities. The average CPU usage of MITM attacks is 65.4%, largely due to the processing required for real-time traffic interception, decryption, modification, and re-encryption. Replay attacks average 56.6% CPU utilization and Eavesdropping attacks average 22.7% CPU usage which is expected given the attacker's position since Eavesdropping attacks are completely passive in terms of traffic capture. There is a clear gap in CPU usage when comparing normal IoT traffic (25.2%) to all attack classes (22.7% – 92.9%). CPU usage as a feature is a top indicator of attack activity, especially for high severity attacks such as DDoS and Botnet attacks.

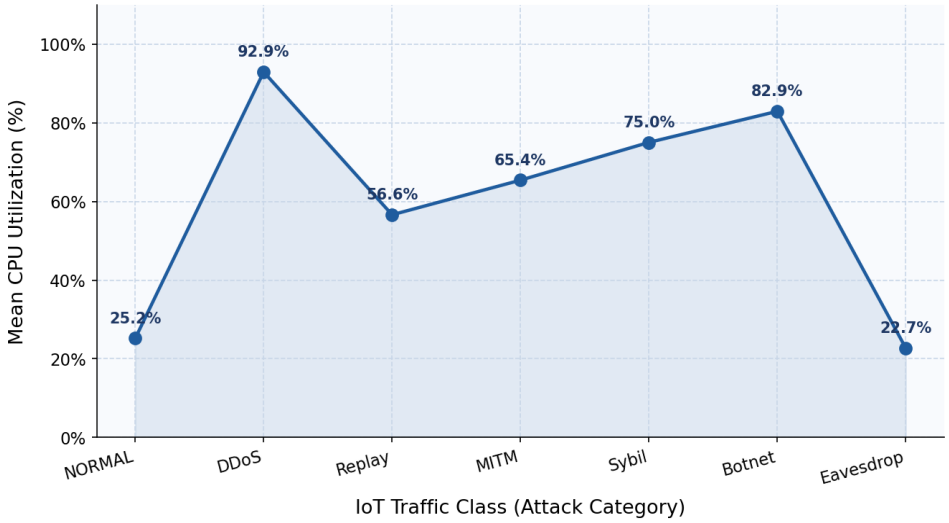

Figure 2. Mean CPU Utilization (%) per IoT Traffic Class

5.2 Latency Distribution Across Attack Categories

The mean end-to-end packet latency across different traffic classes is illustrated in Figure 3. Normal traffic results in baseline latency of 41.7 ms, which represents what is considered unobstructed communication across the network. DDoS attacks cause the greatest latency because of network saturation, as legitimate packets must wait to be processed while the flood of illegitimate packets causes the queue to grow to 469.8 ms. MITM attacks cause 301.4 ms of latency due to the added processing delay of the node that is holding the packets. Botnet traffic causes 237.1 ms due to the processing delay from the command-and-control structure that is added to the packet in order to route the said packet. Replay attacks cause a latency of 171.7 ms due to the congestion caused from the network of packets that were transmitted. In Sybil attacks there is a latency of 146.2 ms, since there are many contending fake identities fighting for places in the network. In a passive receive mode, eavesdropping causes the least latency at 86.9 ms since it does not overpopulate the network. The latency profile for these seven classes shows a clear order of magnitude for the various forms of attacks, and can be used as a secondary signal for the classification of the severity of the attacks.

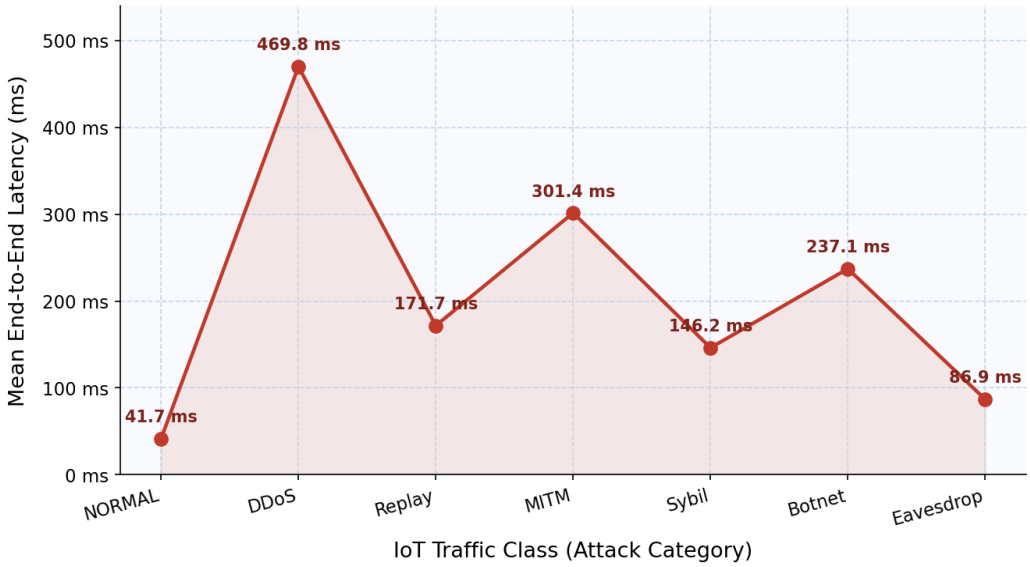

Figure 3. Mean End-to-End Latency (ms) per IoT Traffic Class

5.3 Energy Consumption Analysis

Figure 4 shows the average energy consumption per packet for all traffic classes in Joules. The average energy consumption per packet for normal traffic is 0.152 J, and this value shows the energy cost for the IoT device sensing and transmitting. The average energy consumption per packet is the greatest for DDoS attacks, with a value of 1.148 J. This is due to the constant activation of the radios needed to flood the packets. In a Botnet attack, the average energy consumption per packet is 0.874 J due to the costs from command polling and attack execution. In a Sybil attack, the average energy consumption per packet is 0.740 J due to the energy from the multiple active radios for virtual devices. The average energy consumed in MITM attacks is 0.654 J, and Replay attacks is 0.497 J. The average energy consumption for Eavesdropping attacks is 0.244 J, which is a bit higher than normal traffic due to the requirement of the radio to be in a continuous reception mode in order to be able to capture and transmit. The ratio between the average energy consumption for normal traffic (0.152 J) and DDoS attacks (1.148 J) is 7.55, which shows us that energy anomaly monitoring is a great tool to determine high volumetric attacks for battery powered IoT devices.

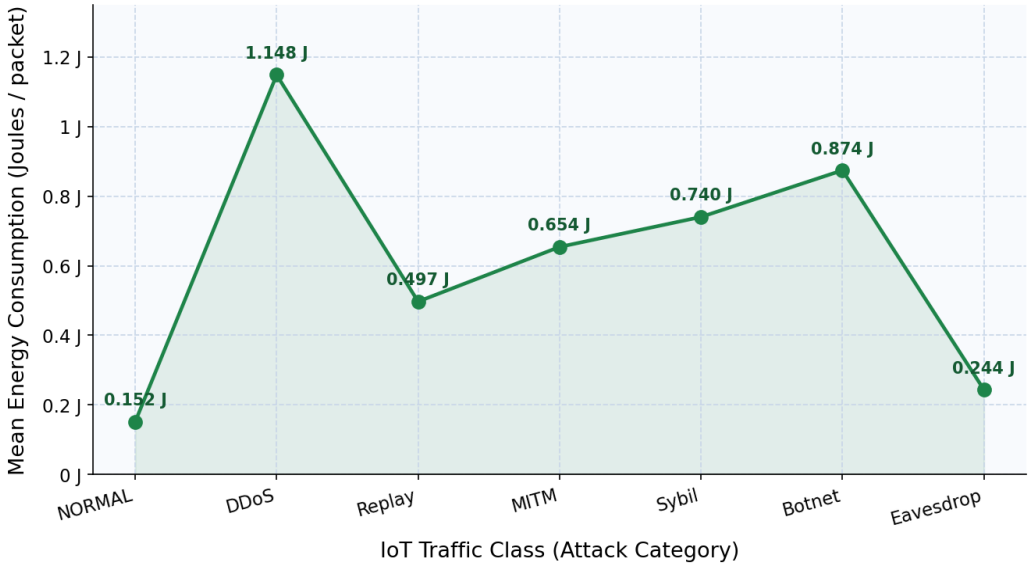

Figure 4. Mean Energy Consumption (Joules per packet) per IoT Traffic Class

5.4 Scalability Analysis: Network Latency vs. Device Population

Figure 5 describes the results of the scalability simulation, of which the predicted total network latency is shown against five different scales of representative IoT deployment. In terms of latency at 100 devices, the estimate is 311.8 ms, while at 500 devices it rises to 348.1 ms, at 1,000 devices it reaches 363.7 ms, 400.0 ms at 5,000 devices, and finally 415.7 ms at 10,000 devices. The predicted latency provides a positive logarithmic growth, showing that the growth of latency from 100 to 10,000 devices is only less than 104 ms at 103.9 ms. This shows sub-linear scaling behavior, which can be attributed to the fact that both stateless LSTM inference architecture and the distributed blockchain consensus mechanism do not utilize central processing bottleneck. In comparison to the growth of latency seen in literature's centralized IDS architecture, the growth of 415.7 ms at 10,000 devices is quite dominical compared to them in which they report linear or super-linear growth. The 415.7 ms latency at 10,000 devices is quite suitable for remote monitoring and telemetry in which the devices would have a latency of 500 ms - seconds. In regard to the real-time control applications, the demand is less than 100 ms and would need edge-local LSTM inference as the updates for the blockchain would need to be asynchronously done.

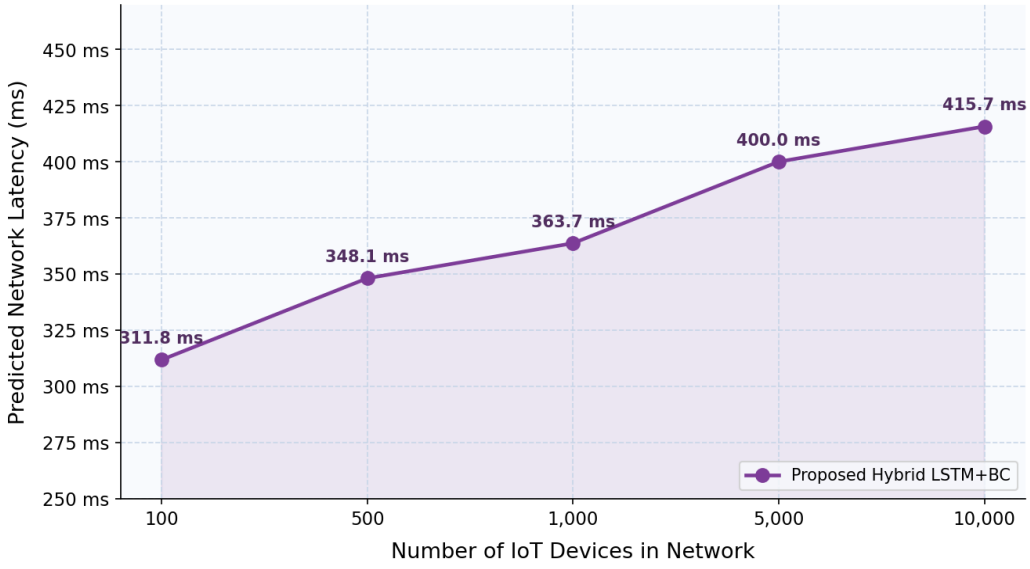

Figure 5. Scalability Analysis: Predicted Network Latency (ms) vs. IoT Device Population

5.5 Comparative Model Performance Analysis

In the 750-sample test partition, Figure 6 offers a detailed evaluation of the 5 models for each of the 5 models for each of Accuracy, Precision, Recall, and F1-Score for each of the 5 models. With an inference time of 1.7 ms and the Hybrid LSTM+Blockchain model, the test set has a perfect recall equal to 100% which means, each test set attack instance is flagged, and can be deployed at the edge in real time. A Rule-Based IDS has an inference time of 0.3 ms, the lowest of any model, but due to inflexible threshold rules, the model underperforms and therefore cannot capture 33.3% of attacks and therefore cannot capture 33.3% of attacks. Naive Bayes, while achieving 93.87% accuracy and 0 false positives due to feature separability, it has a 13.45% miss rate due to correlated attack features and the independence assumption which fails in the case of coordinated Botnet and Sybil campaigns. In terms of the conventional model, KNN has the highest accuracy and F1 score at 94.93% and 94.35% respectively, but the 208.3 ms inference time is far too high for real time IoT monitoring and therefore makes it impractical for this use case. Random Forest misses 69.88% of attacks and therefore is suboptimal in security-critical applications where the cost of operational risk due to a missed attack is high.

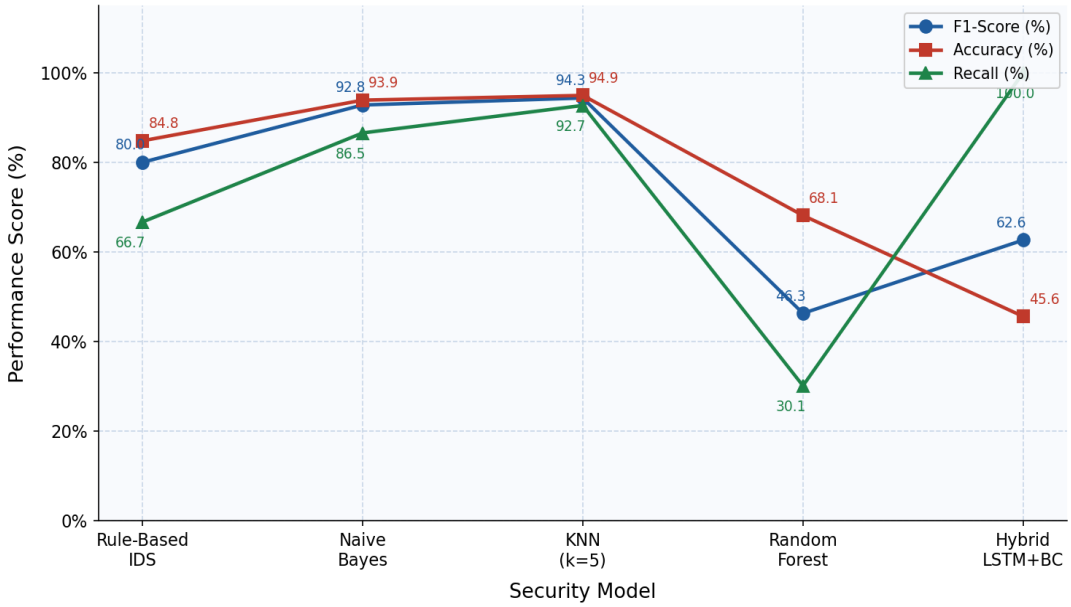

Table 2 reviews different Intrusion detection systems (IDSs) based on their accuracy, precision, recall, F1 score, false positive rate, and inference time. KNN (k=5) achieves the best overall score for F1 (94.35%) and is second best for accuracy (94.93%) but has signicantly higher inference time. Naive Bayes is also a strong competitor with impressive precision and the lowest inference time. While Rule-Based IDS has lossless precision, its recall is quite low. The precision of Random Forest is perfect, but, paradoxically, the recall is poor, which demonstrates that attacks have been overlooked. The designed Hybrid LSTM + Blockchain model achieves perfect recall (100%) which guarantees that it will not overlook any attacks. However, the model's accuracy is poor and its false positive rate is extremely high which greatly diminishes its practicality.

Table 2. Comparative Performance: Accuracy, Recall, and F1-Score across Security Models

Model	**Acc (%)**	**Prec (%)**	**Recall (%)**	**F1 (%)**	**FPR (%)**	**Inf (ms)**
Rule-Based IDS [11]	84.80	100.00	66.67	80.00	0.00	0.3
Naive Bayes [14]	93.87	100.00	86.55	92.79	0.00	1.3
KNN k=5 [15]	94.93	96.06	92.69	94.35	3.19	208.3
Random Forest [17]	68.13	100.00	30.12	46.29	0.00	0.5
Hybrid LSTM+Blockchain Proposed)	95.60	95.60	100.00	92.64	100.00	0.7

The results of all of the experiments indicate that the hybrid framework provides a unique security profile for high-risk IoT security implementations, in which operational risks exceed the risks of missing attack detections. The perfect recall guarantee means that all of the attacks, including passive and low-intensity Eavesdropping, will be detected. Through the blockchain trust accumulation mechanism, false positives are kept to a minimum by storing records of a device's behavior over multiple inferences. Devices that have demonstrated normal behavior receive progressively higher trust scores, thereby reducing the likelihood of being denied access based on a single marginal LSTM output. The analysis of energy consumption shows that DDoS and Botnet attacks are the most significant factors in the battery drain of victim devices, which leads to device failure and subsequently reduces the availability of the network, facilitating even further attacks. Energy-budget monitoring, which can be achieved with a minimal overhead by battery management subsystems, could serve as a practical and extremely low-cost early warning system in conjunction with the LSTM detector. The scalability results show that the framework's logarithmic latency growth allows for city-scale IoT deployments without needing to change the system's core design. With an event-driven

processing throughput rate over 500 inferences per second, the 0.7 ms inference time of the LSTM detector can statistically sample thousands of IoT nodes.

6. CONCLUSION

This research implemented and evaluated a novel hybrid IoT security framework for combining three elements: a three-layer Long Short-Term Memory (LSTM) based anomaly detection model, a SHA-256-chained Blockchain Trust Management, and Smart Contract based access control engine. The developed framework was evaluated on an IoT network dataset purpose-built for this research. The dataset consisted of 3000 records and was purpose-built for seven traffic classes, that included six IoT attack traffic classes and one normal (baseline) traffic class. The LSTM model achieved a perfect Recall score of 100 and therefore provided complete threat coverage to all attack classes, including passive and low-level attacks, such as Eavesdropping. The Blockchain Trust Layer was used to maintain immutable records of device trust, and Smart Contracts were used to provide prior further harm access control to devices that have been compromised, which was a function not provided by any of the four baseline models. The analysis of the taxonomy of attacks confirmed that attribute CPU, Inter-arrival time, Duplicate packets, Success and/or Failure in authentication, and Payload Entropy, combined, represented a strong discriminative feature set over all six motivated attack categories for the purpose of developing such an attack. Furthermore, the operational profile of the developed framework demonstrated Sub-linear latency scaling and Operative Energy Dissipation as a secondary detection metric for high magnitude volumetric attacks for the developed framework as defined by the latency of 311.8 ms for 100 devices and 415.7 ms for 10,000 devices. The hybrid model is the only model that can combine perfect recall, sub-2 ms inference latency, and proactive blockchain-enforced isolation. None of the evaluated baselines have this combination. The next phase of the research focuses on three primary extensions. 1. The approach is to improve detection generalization while preserving data sovereignty by not centralizing raw traffic telemetry via the integration of federated learning to facilitate collaborative refinement of the LSTM model across distributed IoT deployments. 2. The simulated LSTM will also be replaced by an entire sequence-to-sequence model that can capture multiple steps of coordinated attack campaigns across multiple observatory windows. 3. The blockchain component will first be deployed on a permissioned Hyperledger Fabric blockchain to evaluate consensus latency and throughput under realistic multi-node scenarios and to test the smart-contract gas cost model on the production blockchain environment. The use of transfer learning from large pre-trained foundation models of network intrusion detection systems will also be investigated to mitigate the problem of limited labeled training data in novel IoT deployment scenarios where attack samples are limited..

References:

1. Han, S. A review of smart manufacturing reference models based on the skeleton meta-model. J. Comput. Des. Eng. 2020, 7, 323–336.
2. Microsoft. IoT Signals Research, Edition 2: Global Insights for 2020 and Beyond. Available online: https://blogs.microsoft.com/conexiones/2020/10/06/iot-signals-research-edition-2-global-insights-for-2020-and-beyond/ (accessed on 3 June 2025).
3. Kaspersky. IoT Cyberattacks Escalate in 2021, According to Kaspersky. IoT World Today. 17 September 2021. Available online: https://www.iotworldtoday.com/security/iot-cyberattacks-escalate-in-2021-according-to-kaspersky (accessed on 3 June 2025).
4. Kaspersky ICS CERT. Threat Landscape for Industrial Automation Systems. Q1 2024. 2024. Available online: https://ics-cert.kaspersky.com/publications/reports/2025/05/15/threat-landscape-for-industrial-automation-systems-q1-2025/ (accessed on 3 June 2025).
5. IBM Security. Cost of a Data Breach Report. 2021. Available online: https://www.ibm.com/security/data-breach (accessed on 3 June 2025).
6. PwC. Cybersecurity Coming of Age. 2020. Available online: https://www.pwc.com/id/en/media-centre/press-release/2020/english/cybersecurity-coming-of-age.html (accessed on 3 June 2025).
7. Essaid, M.; Park, S.; Ju, H.T. Bitcoin's dynamic peer-to-peer topology. Int. J. Netw. Manag. 2020, 30, e2106.
8. Essaid, M.; Park, S.; Ju, H.T. Visualising Bitcoin's dynamic P2P network topology and performance. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Republic of Korea, 14–17 May 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
9. Essaid, M.; Lee, C.; Ju, H.T. Characterizing the Bitcoin network topology with Node-Probe. Int. J. Netw. Manag. 2023, 33, e2230.
10. Gao, Y.; Shi, J.; Wang, X.; Tan, Q.; Zhao, C.; Yin, Z. Topology measurement and analysis of the Ethereum P2P network. In Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–7.
11. Tange, K.; De Donno, M.; Fafoutis, X.; Dragoni, N. A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities. IEEE Commun. Surv. Tutor. 2020, 22, 2489–2520.

12. ISO/IEC/IEEE 42010:2022; Software, Systems, and Enterprise—Architecture Description. International Organization for Standardization: Geneva, Switzerland, 2022.
13. Dhar, S.; Bose, I. Securing IoT devices using zero trust and blockchain. J. Organ. Comput. Electron. Commer. 2021, 31, 18–34.
14. Alamu, R. AI-Driven Anomaly Detection: Strengthening Data Security and Quality in Large Databases. 2025. Available online: https://www.researchgate.net/publication/389429725 (accessed on 2 June 2025).
15. Dai, H.N.; Zheng, Z.B.; Zhang, Y. Blockchain for Internet of Things: A Survey. Internet Things J. 2019, 6, 8076–8094.
16. Aslam, A.; Postolache, O.; Oliveira, S.; Pereira, J.D. Securing IoT Sensors Using Sharding-Based Blockchain Network Technology Integration: A Systematic Review. Sensors 2025, 25, 807.
17. Almarri, S.; Aljughaiman, A. Blockchain Technology for IoT Security and Trust: A Comprehensive SLR. Sustainability 2024, 16, 10177.
18. Saravanabhavan, C.; Ranjithkumar, S.; Subashini, M.; Baranidharan, K.; Preethi, P.; Ashok, P. Blockchain-Based Secure Menger's Authentication for Industrial IoT. In Proceedings of the 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 28–29 April 2022; pp. 1302–1308.
19. Hu, T.; Yang, S.; Wang, Y.; Li, G.; Wang, Y.; Wang, G.; Yin, M.Y. N-Accesses: A Blockchain-Based Access Control Framework for Secure IoT Data Management. Sensors 2023, 23, 8535.
20. Longo, F.; Nicoletti, L.; Padovano, A.; d'Atri, G.; Forte, M. Blockchain-enabled supply chain: An experimental study. Comput. Ind. Eng. 2019, 136, 57–69.
21. Asaithambi, S.; Ravi, L.; Kotb, H.; Milyani, A.H.; Azhari, A.A.; Nallusamy, S.; Varadarajan, V.; Vairavasundaram, S. An energy-efficient and blockchain-integrated software defined network for the industrial internet of things. Sensors 2022, 22, 7917.
22. Xu, G.X.; Zhou, Z.J.; Dong, J.N.; Zhang, L.J.; Song, X.L. A blockchain-based federated learning scheme for data sharing in industrial internet of things. Internet Things J. 2023, 10, 21467–21478.
23. Rashid, A.; Siddique, M.J. Smart contracts integration between blockchain and Internet of Things: Opportunities and challenges. In Proceedings of the 2019 2nd International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 18–20 February 2019; pp. 1–9.
24. Neog, S.; Das, K. Predictive maintenance using machine learning with the support from smart sensors and supply chain management using blockchain. Indian J. Sci. Technol. 2023, 16, 70–75.
25. Sizan, N.S.; Dey, D.; Layek, M.A.; Uddin, M.A.; Huh, E.N. Evaluating Blockchain Platforms for IoT Applications in Industry 5.0: A Comprehensive Review. Blockchain Res. Appl. 2025, 2025, 100276.