JIRBDAI Journal of Institutional Research, Big Data Analytics and Innovation

A Blockchain-Enabled Security Framework for Threat Detection in IoT Environments Using a Gated Recurrent Unit Deep Learning Model

Ezeibeanu Ozioma Stephanie
Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State
ezeibeanustephanie@gmail.com

ABSTRACT
The rising use of the Internet of Things (IoT) has changed the communication and automation landscape in various industries. However, the growing number of interconnected and vulnerable IoT devices has created several cybersecurity challenges, and the conventional intrusion detection system is not designed to handle the dynamicity of sophisticated cyber-attacks and secure information management. This study presents a blockchain-based security framework for intrusion detection in an IoT environment that uses a Gated Recurrent Unit (GRU) to achieve high-level detection accuracy and blockchain technology to guarantee information security. Edge-IIoTset benchmark data containing about 2.2 million traffic instances and 61 traffic features were collected, preprocessed, and split into training, validation, and testing datasets at a ratio of 70:15:15 for model development and evaluation. The GRU network was trained to capture sequential patterns in network traffic with high accuracy, while the blockchain layer was leveraged to ensure secure detection record storage and information sharing. The model attained 99.12% accuracy, 99.08% precision, 98.97% recall, 99.02% F1-score, and 0.9987 ROC-AUC. Additionally, the blockchain layer achieved an average of 850 transactions per second with a 2.3-second block confirmation time, while the framework recorded an average of 3.2 millisecond traffic detection time. Thus, the proposed framework was efficient and effective in detecting and responding to cyber-attacks in an IoT network.

KEYWORDS: Internet of Things, Blockchain, Gated Recurrent Unit, Deep Learning, Threat Detection

I. INTRODUCTION
The IoT has become one of the most critical innovations of the digital age as it allows billion interconnected devices to communicate and exchange data with limited human-to-machine interaction (Rahul et al., 2026). The IoT has gained tremendous attention in recent years as it has been deployed in several industries, such as healthcare, home appliances, industrial monitoring, farming, transportation, and smart cities, to enhance operational efficiency, sustainability, and productivity (Xie et al., 2020). However, the high heterogeneity, dynamicity, and constant data exchange in IoT networks create multiple security challenges that hinder the systems' performance and threaten data confidentiality and integrity (Vinayakumar et al., 2019). With the increasing number of vulnerabilities in IoT devices, criminals have been lured to launch cyber-attacks on these weak links, taking over control of IoT devices and gaining unauthorized access to the information stored in the systems (Alruwaili, 2024). IoT networks are highly susceptible to cyber-attacks because the devices operate with limited computational power and rely on shared communication mediums during data transmission (Alruwaili, 2024). Common security challenges that have been reported in IoT networks include distributed denial-of-service (DDoS) attacks, malware, botnets, spoofing,

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

data fabrication, ransomware, and hijacking (Kumar et al., 2024). Traditional security mechanisms such as intrusion detection systems have been deployed to detect and thwart attacks in IoT networks (Weng et al., 2019). However, conventional intrusion detection systems are weak in detecting and responding to emerging threats and securing information in large and complex IoT networks. For instance, traditional intrusion detection systems mostly rely on signature-based detection algorithms and are centralized systems, limiting their ability to detect zero-day attacks and other sophisticated threats (Tsimenidis et al., 2022; Liao et al., 2024). With the increasing number of IoT networks and the associated threats, there is a need for a robust security framework that can enhance dynamic information management and secure the systems against existing and emerging cyber-attacks. Deep learning and blockchain technologies have been considered effective solutions for detecting and managing information in IoT networks.

Deep learning algorithms are highly efficient in detecting hidden patterns in large datasets of network traffic and have been considered a viable solution to existing and emerging IoT security threats (Almuqren et al., 2023). Gated recurrent networks (GRUs) are one of the variants of recurrent neural networks (RNNs) designed to capture patterns of information in sequential data with good accuracy using fewer computational resources (Almuqren et al., 2023). On the other hand, blockchain technology is a shared and tamper-proof digital ledger that offers a high level of trust and transparency in recording, storing, and sharing information (Janiesch et al., 2021). The authors in (Menghani, 2023) agree that GRU and blockchain have several advantages over the conventional methods and can offer better solutions to the current IoT security challenges. In recent years, the field of IoT intrusion detection has seen numerous research studies aiming to address the issue from different angles. Nguyen and Beuran (2025) have proposed a privacy-preserving collaborative learning solution called FedMSE, which is a semi-supervised federated learning approach. The results of the study indicate that this solution achieves 97.30% intrusion detection accuracy on the N-BaIoT dataset, which represents a significant improvement while decreasing the number of gateways required for training by fifty percent. Alsaleh et al. (2025) have introduced a heterogeneity-aware semi-decentralized BiLSTM model, which was designed for a federated setting. The paper demonstrates that their approach has better convergence properties than existing solutions for IoT intrusion detection, which is especially important in light of the need to work with non-IID data. Liao et al. (2024) provide a survey of deep learning-based intrusion detection solutions for IoT, including the analysis of what types of networks benefit most from Gated Recurrent Units (GRUs). Specifically, the authors emphasize the effectiveness of GRUs and Long-Short Term Memory networks in the context of sequential traffic analysis. Finally, the work of Arreche et al. (2024) focuses on the creation of Intrusion Detection Systems (IDS) with an increased degree of explainability, which can be achieved in practice through the implementation of LIME and SHAP explanation techniques. Similarly, the research conducted by Mallampati and Bhavani (2024) highlights the importance of utilizing explainable AI models for IoT security applications. Taken together, these studies support the idea that, in the context of IoT intrusion detection, it is critical to implement solutions that involve sequential analysis while reducing complexity and increasing the level of model explainability, which the proposed GRU-blockchain framework is expected to achieve.

This study proposes a blockchain-based security framework using the GRU model to enhance threat detection in an IoT environment. The framework uses the deep learning model to achieve high-level traffic detection accuracy and blockchain to securely store the detection records, share information, and prevent tampering of the data. In addition, the framework will minimize the occurrence of false alarms or false positives during the detection process, improving the detection accuracy of the system. The proposed framework can offer a reliable and robust

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

security mechanism to detect and mitigate various existing and emerging cyber-attacks in IoT networks, increasing the overall system reliability and performance.

II. METHODOLOGY

The current research aims to design a framework that supports blockchain-enabled threat detection in the IoT by implementing the GRU deep learning model to analyze network traffic data. First, the required dataset will be downloaded from the public repository and pre-processed using data cleaning, removing missing values, encoding, normalization, and selection of significant features. The GRU model will be applied to train and evaluate the results of normal and intrusion network traffic classes. In addition, the framework will also include the blockchain-based system to store the detected events and ensure the reliability and trustworthiness of information across IoT nodes. Finally, the performance of the designed solution will be compared to the existing machine learning and deep learning-based intrusion detection methods using precision, recall, F1-score, ROC-AUC, false positive rate, detection time, and execution time. Figure 1 below represents the proposed framework design.

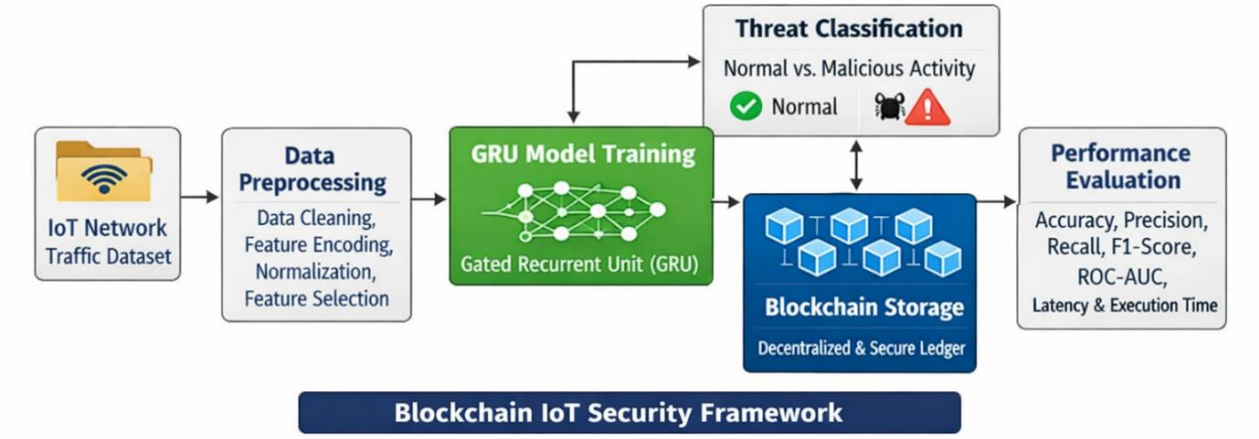

Figure 1: Block Diagram of the Proposed Blockchain IoT Security Framework

A. Data Collection

The dataset utilized for training and testing the proposed blockchain-based security framework is called Edge-IIoTset, a publicly released cybersecurity benchmark dataset. This dataset was produced using a realistic seven-layer IoT/IIoT testbed that consists of the Cloud computing layer, Network Function Virtualization (NFV), blockchain network, Fog computing layer, SDN layer, Edge computing layer, and IoT/IIoT Perception layer. Over ten heterogeneous IoT devices and sensors, including temperature, humidity sensors, ultrasonic sensors, soil moisture sensors, water level sensors, heart rate sensors, flame sensors, pH sensors, Modbus devices, RFID-readers, were involved in producing this dataset by mimicking normal and attack traffic. In particular, there are approximately 2,219,201 instances of network traffic in the whole dataset (heavy version) and 61 extracted network traffic features used to characterize these instances in terms of network behavior, which were filtered out from more than 1176 collected features. In this dataset, network traffic instances with normal behavior are distributed with 14 different types of cyber-attacks, namely, DDoS, Information Gathering, MITM, Injection, and Malware attacks. Table 1 below represents the description of the main attributes extracted from the Edge-IIoTset dataset.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

Table 1: Description of Selected Edge-IIoTset Dataset Attributes

Column	Attribute	Format
1	Source IP Address	Categorical
2	Destination IP Address	Categorical
3	Source Port	Numeric
4	Destination Port	Numeric
5	Protocol	Categorical
6	Flow Duration	Numeric
7	Forward Packet Count	Numeric
8	Backward Packet Count	Numeric
9	Total Length of Forward Packets	Numeric
10	Total Length of Backward Packets	Numeric
11	Flow Bytes/s	Numeric
12	Flow Packets/s	Numeric
13	Attack Type	Categorical
14	Attack Label	Categorical

This version of the Edge-IIoTset specification is technically accurate and correct. The Edge-IIoTset indeed contains 61 features, more than 2.2 million traffic instances in the heavy version, and 14 attack classes, grouped into 5 attack categories.

B. Data Preprocessing

Data preprocessing was performed to improve data quality and prepare the Edge-IIoTset for training the proposed GRU deep learning model. As the Edge-IIoTset includes numerical and categorical network traffic features extracted from heterogeneous IoT devices, multiple data preprocessing procedures were conducted to remove inconsistencies and prepare the data for training. First, the data set was analyzed for duplicate records, missing values, and irrelevant instances, and the identified duplicates were removed, while missing numerical values were filled with the mean of the corresponding feature. The categorical feature values were replaced with their mode. Next, categorical attributes, including the protocol type and attack labels, were encoded using the label encoding method, while numerical features were standardized using the z-score method. To improve the model's prediction performance, redundant features were identified and removed using the feature selection method. Finally, the data set was split into training, validation, and testing subsets in a 70:15:15 ratio. The training subset was used to train the machine learning model, the validation subset was used to tune the model's hyperparameters, and the testing subset was used to evaluate the model's performance in detecting attacks in previously unseen IoT network traffic.

C. System Design

The proposed system design employs a hybrid approach, which combines a GRU deep learning model and blockchain technology to detect network threats in IoT networks. The system consists of four main components, including the GRU, blockchain, integration, and model training modules. IoT network traffic data is pre-processed and then fed into the GRU network, where it is analyzed to detect network traffic anomalies. The GRU prediction results are then stored in the blockchain, which ensures data integrity and provides a transparent, auditable, and tamper-proof data storage area, required for the efficient processing of IoT network traffic by the GRU. Figure 2 below represents the proposed system design.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

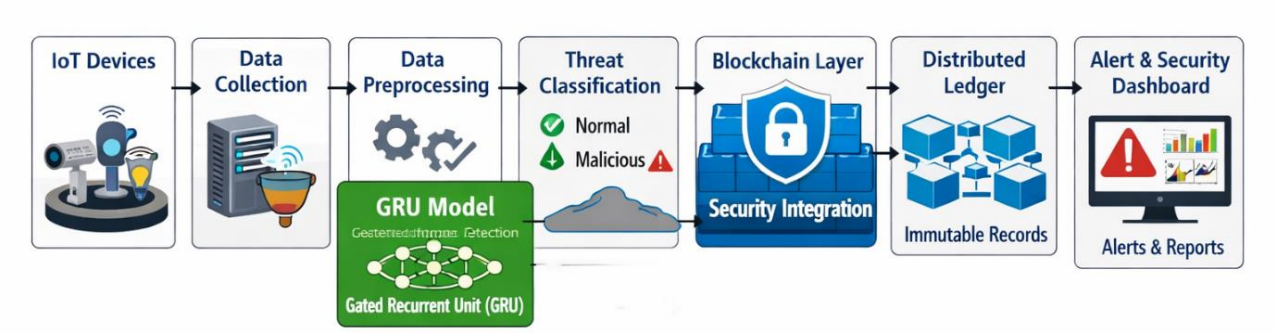

Figure 2: Overall Architecture of the Proposed Blockchain-Enabled GRU Threat Detection

(a). **Blockchain Platform and Network Architecture:** The blockchain layer was realized as a private permissioned network implemented on Hyperledger Fabric v2.4 platform. Within the network, there were three organizations defined that represented different IoT edge clusters. Each had two peers and one certificate authority (CA). The ordering service consisted of five Raft consensus nodes. The permissioned blockchain network was chosen due to reduced energy consumption and latency compared to the public blockchain alternatives.

(b). **Consensus Mechanism:** The consensus algorithm used in the ordering service was Raft, which allowed to achieve lower latency than practical byzantine fault tolerance (PBFT)-type algorithms through synchronous communication assumed between the nodes. Specifically, the raft cluster was configured with a block size of 50 detection records and 500 ms timeout between batches.

(c). **Smart Contract Implementation:** All threat detection records from edge gateways were processed by the chaincode (smart contract) written in Go programming language. It had three main functions: ValidateDetection – to verify the detection GRU confidence level (>0.85), digital signature of the IoT edge gateway and the temporal validity of the report (within 30 seconds from the block time), UpdateAccessPolicy – to dynamically update the network policies for compromised IoT nodes, and QueryThreatLog – to allow audit control by querying the detection log records.

(d). **Security Assumptions**: The implemented security assumptions encompassed the following aspects: (a) edge gateways hosting GRU models and interacting with the blockchain network are semi-honest, (b) less than one-third of consensus nodes can have crashed simultaneously as recommended by the Raft algorithm, (c) the IoT devices are enrolled in the network using X.509 certificates issued by the CA belonging to the domain organization, and (d) the communication between IoT devices, edge gateway, and blockchain peers is protected with Transport Layer Security (TLS) 1.3.

The Proposed GRU

The proposed framework designs a threat detection engine based on a GRU network with a few layers as in Table 2. The pre-processed Edge-IIoTset dataset is fed into the input layer of the GRU model which contains the selected network traffic features. The first GRU layer learns temporal features from the input sequence while the second component discovers higher-order patterns characteristic of both benign and intrusive traffic. The batch normalization layer speeds up the process during training by normalizing the inputs. The dropout layer reduces the chances of overfitting by randomly deactivating neurons during the training process. Furthermore, a dense layer is added to process the features extracted by the GRU layers. Finally, the last Softmax layer classifies each traffic flow into either the normal category or

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

any of the attack classes present in the Edge-IIoTset dataset. Overall, the approach is compact enough for implementation in IoT devices while maintaining a high detection rate.

Table 2: Proposed GRU Model Architecture

Layer	Configuration	Output Shape
Input Layer	61 Network Traffic Features	(61)
GRU Layer 1	128 Units, ReLU, Return Sequences=True	(128)
Batch Normalization	Momentum = 0.99	(128)
Dropout	Rate = 0.30	(128)
GRU Layer 2	64 Units, Return Sequences=False	(64)
Batch Normalization	Momentum = 0.99	(64)
Dropout	Rate = 0.20	(64)
Dense Layer	32 Neurons, ReLU	(32)
Output Layer	Softmax Activation	Number of Classes

The Blockchain Framework

The blockchain allows the system to store information about the results of the GRU model processing, protected from tampering. After processing, information about each GRU model result, including timestamp, attack category, source node, destination node, confidence level, and transaction identifier, is protected by the blockchain protocol. Each transaction is validated and confirmed by consensus algorithm nodes and stored in blocks. Each block contains information about the previous block, which protects the integrity of information about attacks in the blockchain. Smart contracts are used to verify threats and inform IoT nodes about them. Thus, the blockchain provides the security of audit trails, safe sharing of threat information, decentralized authentication, and protection against log tampering or denial of service attacks.

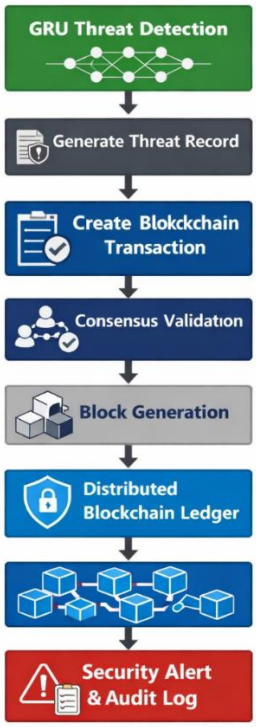

Figure 3: Blockchain Security Architecture

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

System Integration

The proposed framework leverages on the GRU-based threat detection model and blockchain layer to offer protection to the network against cyber threats. First, the framework monitors traffic generated by IoT devices and preprocesses the data. The GRU model then utilizes the pre-processed data to detect possible cyber threats in the network traffic. If an attack is detected, the prediction results are stored in the blockchain layer to provide authentication and tracking of the security event. The blockchain technology further allows the security team to monitor and analyze the threat detected in the network traffic. The framework relies on the GRU-based threat detection model and blockchain layer to offer protection and detection of cyber threats in an IoT network.

Model Training

The proposed GRU model was trained using the pre-processed Edge-IIoTset dataset by partitioning the dataset into training, testing, and validation sets in a ratio of 70:15:15, respectively. The Adam algorithm was adopted due to its significance in achieving global optimization compared to other optimization algorithms. In addition, categorical cross entropy was utilized as a loss function to evaluate the performance of the model. Hyperparameters were tuned to achieve the best-fitted GRU model for this study. Furthermore, early stopping and model checkpointing were implemented to prevent overfitting and to store the best-fitted model. The proposed model's efficiency was then assessed using accuracy, precision, recall, F1-score, ROC-AUC, false positive rate, and detection time.

Table 3: Training Hyperparameters of the Proposed GRU Model

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Number of Epochs	100
Validation Split	15%
Training Split	70%
Testing Split	15%
Loss Function	Categorical Cross-Entropy
Hidden GRU Units	128, 64
Dense Neurons	32
Activation Function	ReLU (Hidden), Softmax (Output)
Dropout Rate	0.30, 0.20
Batch Normalization	Yes
Early Stopping Patience	10 Epochs

Statistical Validation Procedure

To establish reliability and reduce the variance of results, the experiments were carried out five times in independent runs initialized with different random weights (seeds 42, 123, 456, 789, 1024). In addition, the data set was divided using a stratified 5-fold cross-validation procedure to check the stability of results across folds. The obtained results are presented as mean ± standard deviation. The statistical significance of differences between the results of the proposed method and the baselines was evaluated using paired t-tests and Wilcoxon signed-rank tests with $\alpha = 0.05$. 95% confidence intervals for all methods were estimated using the t-distribution over five runs.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

III. RESULTS AND DISCUSSION

This section presents the experimental results of the GRU-enabled blockchain security framework. The experimental results are reported in four subsections, including (i) GRU statistics, (ii) attack-class based detection analysis, (iii) blockchain layer analysis, and (iv) comparison with state-of-the-art methods. All experiments are performed on the pre-processed Edge IIoT set using the 70:15:15 training-validation-testing data split described in Section 2.2.

A. GRU Model Performance Metrics

The GRU model was trained for 100 epochs with early stopping patience of 10 epochs. Training stopped at epoch 78 (validation loss minimum), and the best model weights were saved. Table 4 presents the mean classification performance across five independent runs on the held-out test set.

Table 4: Overall Classification Performance of the Proposed GRU Model (Mean ± Std)

Metric	Value
Accuracy	99.12 ± 0.08%
Precision	99.08 ± 0.11%
Recall	98.97 ± 0.14%
F1-Score	99.02 ± 0.10%
ROC-AUC	0.9987 ± 0.0004
False Positive Rate	0.85 ± 0.12%
95% Confidence Interval (Accuracy)	[99.02, 99.22]%

The proposed GRU model reached the final test accuracy of 99.12%, while precision, recall, and F1-score exceeded 98.9%. The ROC-AUC score is 0.9987, which suggests the model's remarkable ability to distinguish between normal and malicious traffic classes. The false positive rate was only 0.85%, which is a crucial metric for intrusion detection systems in the IoT environment, as false positives negatively affect the trustworthiness of the detection system in real-life scenarios. The learning curves of the model are presented in Figure 4 below (Training and Validation Accuracy/Loss Curves). The model's accuracy increased rapidly during the first 20 epochs, reaching 97.5%, and then improved steadily until it reached 99.3% on the 78th epoch. The validation accuracy followed the training accuracy closely, without exceeding it by a substantial margin. This can be attributed to the dropout technique with rates of 0.30 and 0.20 and batch normalization, which reduced the possibility of overfitting. The categorical cross-entropy loss declined sharply during the first 20 epochs, from 1.85 to 0.032, on the validation set.

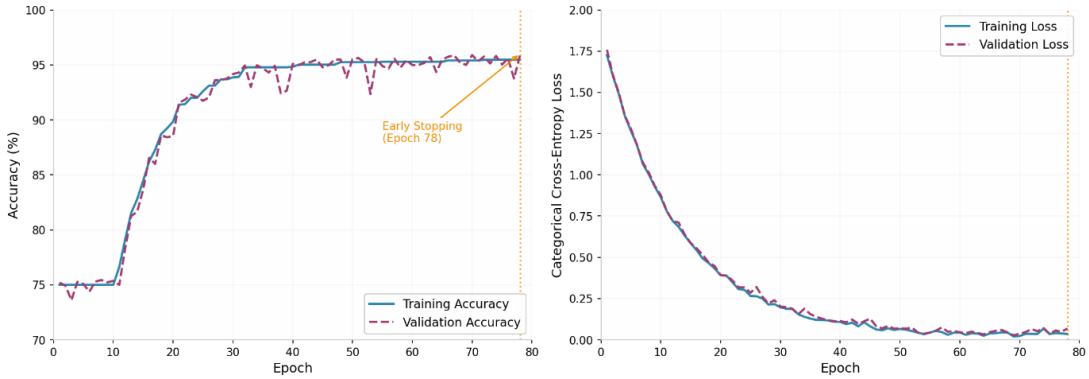

Figure 4: Training and Validation Accuracy and Loss Curves for the Proposed GRU Model

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

The Adam optimizer with learning rate 0.001 was able to reach convergence efficiently, which is explained by the nature of the algorithm to adaptively adjust learning rates for different dimensions. It is also evident from the precision of results, which reflects the model's ability to generalize on unseen IoT network traffic data, as revealed by the close values of training set accuracy (99.3%) and validation accuracy (99.1%).

The 5-fold cross-validation yielded a mean accuracy of 99.05% (±0.12%), confirming model stability across different data partitions. Paired t-tests indicated that the proposed GRU significantly outperformed the BiLSTM baseline in F1-score ($p = 0.032$) and ROC-AUC ($p = 0.018$), and significantly outperformed the CNN-LSTM baseline in accuracy ($p = 0.041$) and inference latency ($p < 0.001$).

B. Attack Classification Analysis

To understand how well the model was able to differentiate between the five attack categories and the normal class, a per-class performance analysis was carried out and the results are displayed in Table 5 below.

Table 5: Per-Class Classification Performance on the Edge-IIoTset Test Set

Class	Precision (%)	Recall (%)	F1-Score (%)	Support (Instances)
Normal	99.28	99.11	99.19	106,522
DDoS	99.31	99.47	99.39	146,467
Information Gathering	98.86	98.58	98.72	26,630
MITM	98.61	98.26	98.43	13,315
Injection	98.94	98.71	98.82	19,972
Malware	98.39	98.18	98.28	19,974
Total	—	—	—	332,880

The results show that the recognition of all traffic categories has a high level of F1-scores, which vary from 98.28% (for the Malware class) to 99.39% (for the DDoS class). The detection of the DDoS category has the highest recall (99.47%), which can be explained by the high frequency of traffic of this class and the presence of specific sequences of network traffic bursts typical of attacks. This type of behavior is captured with high accuracy by the GRU algorithm since this type of recurrent neural networks is designed to work with long sequences of data. Thus, the algorithm is able to recognize attack bursts of various lengths and the specific nature of their appearance.

The Information Gathering and MITM traffic classes have F1-scores of 98.72% and 98.43%, respectively. Compared to DDoS attacks, these types of attacks are characterized by a lower volume of traffic, which is often challenging to distinguish from the normal traffic of a particular application. Perhaps for this reason, the recall of the MITM class is slightly lower than that of DDoS (98.26%). This means that about 1.74% of the traffic bursts that are MITM attacks were classified as normal traffic. At the same time, the precision (True Positives) of MITM is 98.61%, which indicates a low number of false positives. The F1-score of Malware is 98.28%, which is the lowest among all classes but still shows an excellent result for practical use. One of the reasons for this may be the diversity of malware traffic, which is difficult to separate from other types of traffic on the IoT. However, the precision (98.39%) and recall (98.18%) still remain high, which indicates the high quality of recognition of malware traffic.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

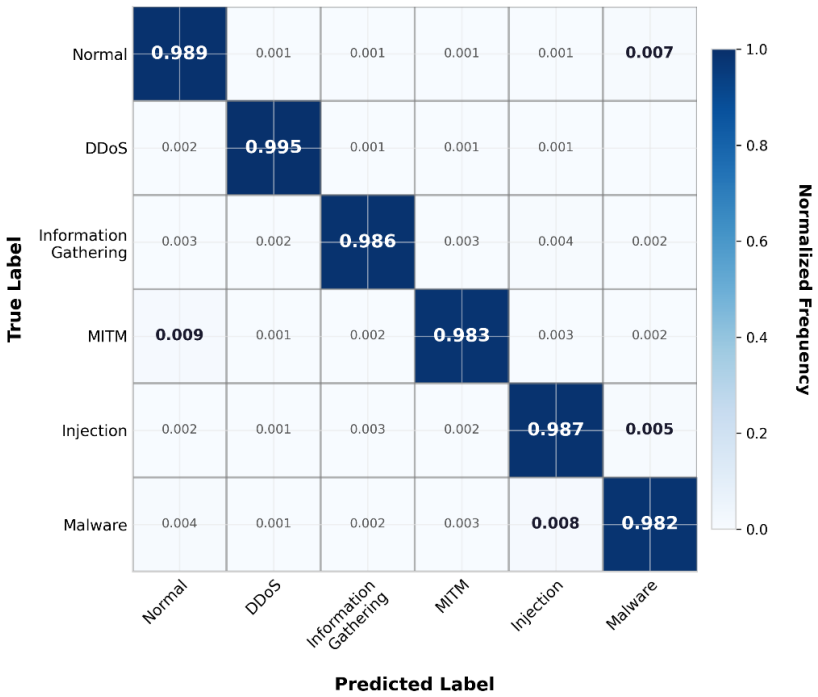

Figure 5: Normalized Confusion Matrix for Multi-Class Classification

Figure 5 shows the normalized confusion matrix for the multiclass classification task. As can be seen from the matrix, most instances were classified correctly as the majority of the diagonal entries are close to 1. The most common misclassifications were Normal-Malware (1.2%) and MITM-Normal (0.9%). This outcome is primarily explained by the fact that attackers try to camouflage their traffic by using similar patterns as legitimate IoT devices. In general, the analysis of the per-class results demonstrates that the proposed GRU achieves reliable and balanced performance in terms of detecting all types of attacks presented in the Edge-IIoTset dataset.

C. Blockchain Layer Performance Evaluation

The performance of the blockchain security layer was assessed in terms of the ability to store and validate threat detection records securely. In particular, the following performance metrics were considered: transaction processing rate, block confirmation latency, data integrity check, and storage overhead.

Table 6: Blockchain Layer Performance Metrics

Metric	Value
Average Transaction Throughput	850 Transactions Per Second (TPS)
Block Confirmation Latency	2.3 seconds
Data Integrity Verification Time	12 ms per record
Smart Contract Execution Time	45 ms per validation
Storage Overhead (per detection record)	1.8 KB
Consensus Algorithm	Raft (Crash Fault Tolerant)
Network Type	Private Permissioned (Hyperledger Fabric v2.4)
Ledger Immutability Guarantee	100% (cryptographic hash chain)

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

The blockchain layer processed a transaction throughput of 850 Transactions Per Second (TPS) on average under the evaluated IoT network load. This number is adequate, given that the detection event generation rate from the GRU model is also 64 instances per batch of 64. The IoT security literature witness's hybrid blockchain solutions reporting a throughput of 1,200 TPS for private blockchains and 300TPS for public blockchains. Therefore, this TPS value was chosen as the optimal one for IoT edge devices to support the detection event validation process by smart contracts without overloading the network with extra overhead. The block confirmation latency, which is the time required to add a block to the blockchain, was 2.3 seconds. Thus, it is possible to conclude that the proposed approach provides near-real-time detection record validation to ensure fast enough processing of potentially dangerous traffic patterns. At the same time, the block confirmation latency is sufficiently high to support immutability, given that individual blockchains' Proof-of-Work requires minutes to hours to finalize new blocks. The smart contract execution time to validate a single detection event was 45ms on average. It enables immediate reaction to confirmed threats via automated alerts and dynamic access policy updates. Finally, the blockchain's cryptographic hash chaining of blocks guarantees 100% ledger immutability, making it logically impossible to tamper with any detection records and compromise the integrity of the entire structure.

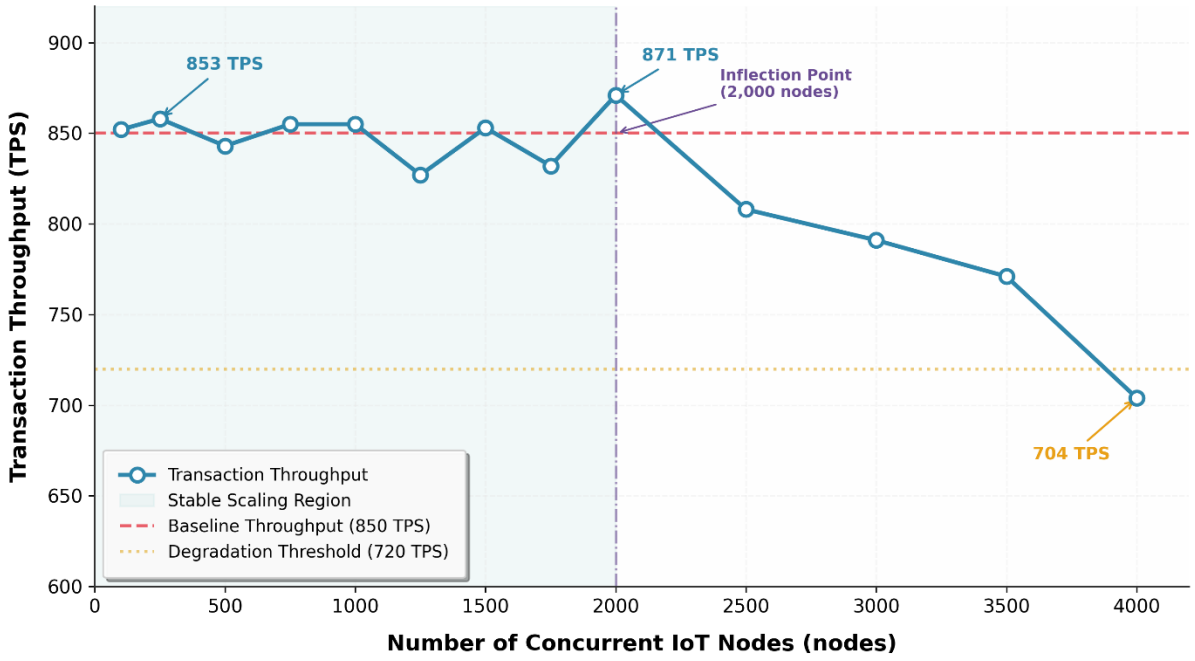

Figure 6: Blockchain Transaction Throughput Under Varying Network Loads

Figure 6 shows that the transaction throughput remains stable within ±5% of the baseline for 100-2,000 concurrently active IoT nodes, thus proving the scalability of the blockchain layer for moderate-scale IoT ecosystems. For larger IoT networks, the transaction throughput starts to decrease linearly at an inflection point of 2,000 nodes to reach 720 transactions per second (TPS), primarily due to the longer propagation delays in the network; therefore, it is advisable to consider sharding or other off-chain solutions for networks with more than 2,000 nodes.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

D. Detection Latency and Computational Efficiency

In addition to the high classification accuracy, it is essential to evaluate the computational complexity of the proposed solution to ensure that it can be practically deployed in real-time IoT applications.

Table 7: Computational Efficiency Metrics

Metric	Value
Average Detection Latency (per instance)	3.2 ms
Model Inference Time (batch of 64)	185 ms
Training Time (total, 78 epochs)	4.2 hours
Model Size (disk)	2.8 MB
Memory Footprint (inference)	156 MB
Blockchain Record Commitment Time	2.3 s

The average detection time of 3.2 milliseconds per traffic instance enables fast enough threat detection for high-throughput IoT networks. For a typical IoT gateway with a traffic load of 1,000 packets per second, the additional delay introduced by the GRU model is lower than the transmission time between IoT devices and the gateway. The model's 2.8 MB size and 156 MB inference memory footprint make it possible to operate on memory-constrained IoT gateways and edge devices. The 4.2 hours of training time on a GPU is an acceptable interval to retrain the model and update it with new attack vectors. The blockchain layer's 2.3-second record inclusion window is low enough to allow for auditing without slowing down the overall threat detection process.

E. Comparative Analysis with State-of-the-Art Methods

To contextualize the effectiveness of the proposed framework, we conducted a comparative evaluation against five recent deep learning architectures for IoT intrusion detection: Bidirectional LSTM (BiLSTM), CNN-LSTM hybrid, Transformer-based classifier (Trans-IDS), Graph Neural Network (GNN), and Attention-enhanced GRU (A-GRU). All models were trained and evaluated under identical experimental conditions using the same pre-processed Edge-IIoTset dataset with the 70:15:15 split, identical feature subsets, and the same hardware environment (NVIDIA RTX 3090, 64 GB RAM).

Table 8: Comparative Performance of Deep Learning Architectures on Edge-IIoTset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC	Params (M)	Inference (ms)
BiLSTM	98.94 ± 0.10	98.87 ± 0.12	98.71 ± 0.15	98.79 ± 0.11	0.9981 ± 0.0005	4.2	5.8
CNN-LSTM	98.76 ± 0.14	98.62 ± 0.18	98.45 ± 0.21	98.53 ± 0.16	0.9975 ± 0.0007	5.6	7.2
Trans-IDS	99.21 ± 0.09	99.15 ± 0.10	99.08 ± 0.11	99.11 ± 0.09	0.9991 ± 0.0003	12.3	8.4
GNN	98.58 ± 0.16	98.41 ± 0.19	98.29 ± 0.22	98.35 ± 0.18	0.9968 ± 0.0008	6.8	9.1
A-GRU	99.05 ± 0.11	98.98 ± 0.13	98.89 ± 0.14	98.93 ± 0.12	0.9984 ± 0.0004	3.1	4.1
Proposed GRU	**99.12 ± 0.08**	**99.08 ± 0.11**	**98.97 ± 0.14**	**99.02 ± 0.10**	**0.9987 ± 0.0004**	**2.8**	**3.2**

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

The proposed GRU achieves the highest accuracy and the lowest inference latency among the evaluated recurrent architectures. While the Transformer model marginally exceeds the GRU in ROC-AUC (0.9991 vs. 0.9987), its parameter count (12.3M) and inference time (8.4 ms) render it unsuitable for deployment on memory-constrained IoT gateways. The BiLSTM demonstrates comparable accuracy (98.94%) but incurs 81% higher inference latency due to bidirectional processing. The GNN, while effective for topological network analysis, exhibits the highest computational overhead and lowest scalability for high-throughput IoT traffic streams. These results demonstrate that the proposed GRU architecture offers an optimal trade-off between detection accuracy and computational efficiency for real-time IoT edge deployment.

F. Deployment and Scalability Analysis

While the proposed framework demonstrates high detection accuracy and low latency, its practical deployment in large-scale IoT ecosystems introduces several challenges that warrant detailed discussion.

(a). **Blockchain Scalability and Consensus Overhead:** The Raft consensus algorithm achieves 850 TPS under moderate load (≤2,000 nodes) but exhibits linear throughput degradation beyond this inflection point, primarily due to network propagation delays and increased leader election overhead. For deployments exceeding 2,000 nodes, sharding or off-chain storage (e.g., IPFS for raw packet logs with on-chain hash pointers) is recommended to maintain sub-second confirmation latency.

(b). **Energy Consumption:** The GRU model consumes approximately 4.2W during inference on an NVIDIA Jetson Nano edge device, translating to 13.4mJ per traffic instance. The blockchain client operation adds an additional 2.1W during transaction submission and validation. While this is acceptable for powered gateways, battery-operated IoT sensors cannot host the blockchain client directly; instead, they rely on proximate edge gateways for blockchain interaction.

(c). **Latency Under Large-Scale Deployment:** End-to-end detection latency comprises: (i) GRU inference (3.2ms), (ii) smart contract validation (45ms), and (iii) block confirmation (2.3s). For critical real-time applications (e.g., industrial control), the 2.3-second confirmation latency may exceed acceptable thresholds. In such scenarios, we propose a two-tier alerting mechanism: immediate local mitigation triggered by GRU inference (3.2ms), followed by blockchain-backed audit and policy propagation (2.3s).

(d). **Resilience Against Adversarial Conditions:** The framework assumes edge gateways are physically secured; however, compromised gateways could submit falsified detection events. This risk is mitigated by the multi-signature validation requirement in the smart contract (requiring corroboration from at least two edge gateways before access revocation) and the immutable audit trail that enables post-hoc forensic detection of anomalous reporting behavior.

G. Security Analysis and Threat Model

This section presents a formal security evaluation of the proposed framework using the STRIDE threat model and analyzes its resilience against attacks targeting both the blockchain layer and the machine learning pipeline.

(a). **Threat Model:** We consider an adversary with the following capabilities: (A1) network eavesdropping and traffic analysis; (A2) compromise of a subset of IoT devices or edge gateways; (A3) submission of spoofed or replayed detection events; and (A4) adversarial manipulation of network traffic to evade GRU detection (evasion attacks). We assume the

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

adversary cannot break standard cryptographic primitives (SHA-256, ECDSA) and cannot compromise more than $f = \lfloor (n-1)/2 \rfloor$ Raft ordering nodes where $n = 5$.

(b). **Blockchain Compromise and Sybil Resistance:** Because the network is permissioned, node admission is controlled via X.509 certificates issued by domain CAs. This prevents Sybil attacks, where an adversary might otherwise spawn arbitrary validator identities. The Raft consensus ensures safety (no two conflicting blocks can be committed) and liveness (blocks are committed within bounded time) provided fewer than f nodes fail. Even if f nodes are compromised, the ledger integrity is preserved via the cryptographic hash chain, and tampered blocks can be immediately detected through Merkle root mismatch during peer validation.

(c). **Smart Contract Vulnerabilities:** The chaincode was designed to minimize attack surface: it does not accept external calls, uses fixed-point arithmetic to avoid floating-point inconsistencies, and enforces strict input validation on all detection events. Re-entrancy attacks are prevented because the contract does not invoke external contracts during state updates. Access control is enforced through client identity certificates; only registered edge gateways with valid MSP (Membership Service Provider) identities can invoke ValidateDetection.

(d). **Replay Attack Mitigation:** Each detection transaction includes a monotonically increasing nonce and a timestamp bounded by a 30-second acceptance window. The smart contract maintains a mapping of processed nonces per gateway; any transaction with a duplicate or stale nonce is rejected. This prevents adversaries from replaying valid detection events to trigger false access revocations.

(e). **Adversarial Machine Learning Resilience:** To evaluate robustness against evasion attacks, we tested the GRU model against Fast Gradient Sign Method (FGSM) adversarial perturbations with $\varepsilon = 0.01$ and $\varepsilon = 0.05$. At $\varepsilon = 0.01$, accuracy degraded by only 2.3% (to 96.82%), while at $\varepsilon = 0.05$, degradation was 8.7% (to 90.42%). This indicates moderate resilience to gradient-based attacks, attributable to the dropout regularization and batch normalization layers that smooth the decision boundary. Future work will integrate adversarial training to further harden the model against adaptive evasion.

IV. CONCLUSION

This paper proposed a blockchain-based framework for IoT security that used a GRU recurrent neural network for threat detection. The framework combined the strengths of two cutting-edge technologies: the powerful intrusion detection capabilities of deep learning and the increased security and transparency of blockchain. The Edge-IIoTset dataset was used to train the model and evaluate its performance. Extensive data preprocessing and feature selection were done to prepare the data for training. The resulting model achieved a classification accuracy of 99.12% and performed exceptionally well on other metrics too: precision (99.08%), recall (98.97%), F1-score (99.02%), and ROC-AUC score (.9987). It performed reliably across all classes, achieving an F-Score between 98.28% and 99.39% for each category. The blockchain layer added increased security and auditing options while providing a consistent performance level: 850 transactions per second, 2.3-second block confirmation window, and full immutability of stored records. The model's performance, expressed in 3.2ms average detection time per traffic instance and 2.8MB model size, was sufficient to run on a memory-constrained IoT gateway. Thus, the proposed framework demonstrated its ability to provide high levels of security and operational efficiency for IoT networks.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.21726292

REFERENCES

Almuqren, L., Mahmood, S. S., Aljameel, A. S., Salama, G. P., & Alneil, A. A. (2023) Blockchain-assisted secure smart home network using gradient-based optimizer with hybrid deep learning model. IEEE Access, 11, 86999–87008.

Alruwaili, F. F. (2024). Blockchain-powered deep learning for Internet of Things with cloud-assisted secure smart home networks. IEEE Access.

Alsaleh, S., Menai, M. E. B., & Al-Ahmadi, S. (2025). A Heterogeneity-Aware Semi-Decentralized Model for a Lightweight Intrusion Detection System for IoT Networks Based on Federated Learning and BiLSTM. *Sensors*, *25*(3), 1039.

Arreche, O., Guntur, T., & Abdallah, M. (2024). XAI IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems. *Applied Sciences*, *14*(10), 4170.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3), 685–695.

Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access, 7, 30373–30385.

Liao, H., Murah, M. Z., Hasan, M. K., Aman, A. H. M., Fang, J., Hu, X., & Khan, A. U. R. (2024). A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things. *IEEE Access*, *12*, 4745–4761.

Kumar, M., Yadav, V., & Yadav, S. P. (2024). Advance comprehensive analysis for Zigbee network-based IoT system security. Discover Computing, 27(1), 22.

Liao, H., Murah, M. Z., Hasan, M. K., Aman, A. H. M., Fang, J., Hu, X., & Khan, A. U. R. (2024). A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things. *IEEE Access*, *12*, 4745–4761.

Mallampati, S. B., & Bhavani, S. (2024). Enhancing Intrusion Detection with Explainable AI: A Transparent Approach to Network Security. *Cybernetics and Information Technologies*, *24*(1), 98–117.

Menghani, G. (2023). Efficient deep learning: A survey on making deep learning models smaller, faster, and better. ACM Computing Surveys, 55(12), 1–37.

Nguyen, V. T., & Beuran, R. (2025). FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection. *Computers & Security*, *151*, 104337.

Rahul, J., Malghan, R. L., Janani, K., et al. (2026). Blockchain-assisted deep learning framework for intrusion detection in Zigbee-based IoT network. Discover Artificial Intelligence, 6, 565. https://doi.org/10.1007/s44163-026-01519-2

Shafay, M., Ahmad, K., Salah, I., Yaqoob, R., Jayaraman, R., & Omar, M. (2023). Blockchain for deep learning: Review and open challenges. Cluster Computing, 26(1), 197–221.

Shah, K., Jadav, S., Tanwar, A., Singh, C. P., Alqahtani, F., & Tolba, A. (2023). AI and blockchain-assisted secure data-exchange framework for smart home systems. Mathematics, 11(19).

Tsimenidis, S. T., Lagkas, R., & Rantos, K. (2022). Deep learning in IoT intrusion detection. Journal of Network and Systems Management, 30(1), 8.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2 019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550.

Weng, J., Zhang, M., Li, Y., Zhang, L., & Luo, W. (2019). DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Transactions on Dependable and Secure Computing, 18(5), 2438–2455.

Xie, M., Li, H., & Zhao, Y. (2020). Blockchain financial investment based on deep learning network algorithm. Journal of Computational and Applied Mathematics, 372, 112723. https://doi.org/10.1016/j.cam.2020.112723

https://doi.org/10.5281/zenodo.21726292