Harisankar R Nair, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Blockchain-Enhanced LSTM Based Machine Learning for Real-Time Iot Security: Anomaly Detection and Trust Management at the Edge

Harisankar R Nair, Duane Chembakassery, Prassanna J
School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai, India.

Abstract- The rapid proliferation of Internet of Things (IoT) devices has introduced significant security challenges, including unauthorized access, data breaches, and distributed denial-of-service (DDoS) attacks. Traditional security mechanisms often fall short due to the resource constraints of IoT devices and the evolving nature of cyber threats. This paper presents a novel Machine Learning (ML)-powered anomaly detection system, integrated with Blockchain-Based Trust Management, to enhance IoT security. The proposed system utilizes a Raspberry Pi 4 as an edge device to monitor real-time network traffic, extract relevant features, and employ an Autoencoder/LSTM-based ML model for anomaly detection. When an anomaly is detected, the system triggers an alert, mitigates threats using firewall rules, and logs security events on a private blockchain. A trust score mechanism is implemented via smart contracts to dynamically assess device behavior, enabling automated device quarantine and adaptive security measures. Experimental evaluations demonstrate that the LSTM model achieves an accuracy of 98.85%, with high precision and recall, effectively identifying network anomalies. The blockchain-based trust management framework ensures tamper-proof security event logging while dynamically adjusting device trust scores based on anomaly detection results. The results indicate that this integrated approach enhances both detection accuracy and accountability in IoT networks, making it a viable solution for real-time, decentralized IoT security.

Keywords: IoT Security, Blockchain, Automated Threat Mitigation.

I. INTRODUCTION

The Internet of Things (IoT) enables large numbers of smart devices to communicate across domains such as healthcare, industrial automation, smart homes, and smart cities. However, this connectivity also expands the attack surface of IoT systems. Because many IoT devices have limited computation, memory, and power, conventional security mechanisms are often difficult to deploy in a uniform and effective manner. Centralized intrusion detection systems can further introduce a single point of failure, which is undesirable in adversarial environments.

Machine-learning-based anomaly detection has emerged as a promising solution for identifying previously unseen or evolving attacks. In parallel, blockchain has been explored as a decentralized mechanism for maintaining trustworthy and tamper-resistant device records. Although both directions are individually promising, many existing hybrid frameworks use blockchain only as a passive logging layer after anomaly detection. They do not tightly couple anomaly evidence, trust adaptation, and edge-side response into a single operational loop.

This paper presents an edge-centric IoT security framework that integrates LSTM-based anomaly detection with blockchain-based trust management and automated mitigation. The key distinction of the proposed system is that blockchain is not used merely for immutable storage; instead, it serves as an active trust-control layer that updates device trust scores from anomaly evidence and supports security decisions such as monitoring, alerting, and device isolation. A second distinguishing feature is edge deployment: traffic analysis and anomaly scoring are executed on a Raspberry Pi 4 with Google Coral support, enabling near-real-time operation close to the data source. A third contribution is closed-loop security orchestration, in which anomaly detection, trust updating, blockchain validation, and physical/network response are integrated into one pipeline.

The main contributions of this work are as follows. First, an LSTM-based anomaly detection module is designed for temporal modelling of IoT traffic behaviour. Second, a blockchain-backed trust model is introduced to maintain auditable and tamper-resistant trust evolution for each device. Third, the framework supports automated response through alarms, dashboard notifications, and device isolation. Fourth, a Petri-net-based workflow representation is used to describe the end-to-end security process in a structured manner.

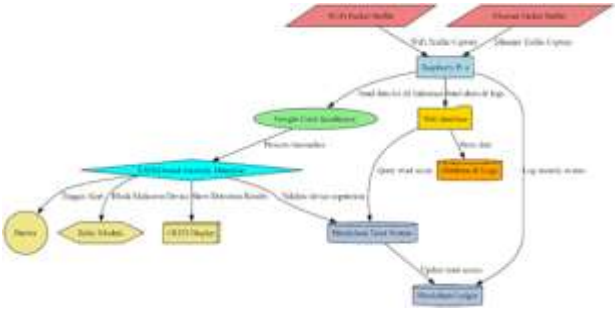

Fig. 1. Multi-layered security architecture of the proposed system.

II. RELATED WORK

Machine Learning and Blockchain for IoT Security
Machine learning has been widely used for anomaly detection in IoT networks because rule-based methods are often ineffective against zero-day and adaptive attacks. Prior work has employed Decision Trees, Logistic Regression, Random Forests, Support Vector Machines, Autoencoders, and other deep learning architectures to detect malicious traffic patterns [1]–[3], [10]. Classical models are often lightweight and interpretable, but their ability to model temporal dependencies is limited. Deep learning models, particularly sequential architectures, are better suited to traffic streams with temporal structure, although they may increase computational cost.

Blockchain-based trust management has also received significant attention for IoT security [4]–[7], [9]. These approaches improve traceability and resilience by storing trust-related events in an immutable ledger. However, they also introduce nontrivial challenges, including transaction latency, storage growth, smart-contract overhead, and limited throughput under large device populations [6]. These issues are especially important in real-time IoT settings, where decision latency directly affects security response quality.

Hybrid approaches combine machine learning with blockchain to improve both detection and accountability [7], [8]. Yet many such systems still treat the blockchain as a passive backend for archival logging. In contrast, the present work couples anomaly scores with trust-score evolution and response actions at the edge. This tighter integration is the primary novelty of the framework.

Table 1. Comparative positioning of representative approaches and the proposed framework.

Ref.	Core idea	ML component	Blockchain role	Edge deployment / response	Limitation relative to this work
[1]	ML-based anomaly detection in IoT	DT/LR/SVM/RF	Not central	Limited discussion of real-time response	No decentralized trust layer
[2]	Attack classification for IoT devices	SVM	Not used	Higher computational cost	No trust adaptation or edge mitigation loop
[3]	Deep-learning-based anomaly detection	Autoencoder	Not used	High training demand	No blockchain-backed accountability

Ref.	Core idea	ML component	Blockchain role	Edge deployment / response	Limitation relative to this work
[4]	Blockchain-based trust management	Not central	Reputation storage and update	Limited real-time mitigation	High overhead and latency
[5]	Smart-contract-based IoT protection	Not central	Secure logging and access control	Limited detection integration	Slow transactions and contract overhead
[6]	Survey of blockchain trust in IoT	Not central	Trust-management overview	No deployment architecture	Highlights scalability concerns
[7]	Blockchain and social-IoT trust	ML-assisted	Immutable trust storage	Limited direct edge response	Privacy and integration complexity
[8]	ML-enhanced IoT anomaly detection	ML with blockchain	Trust/log storage	Resource-intensive integration	Limited end-to-end control loop
This work	Edge anomaly detection with active trust control	LSTM-based temporal detector	Dynamic trust scoring, validation, immutable audit trail	Raspberry Pi 4/Coral-based inference, alerts, dashboard, isolation	Current evaluation is limited to NSL-KDD and trust simulation

Research Gap

Existing literature supports the effectiveness of both ML-based anomaly detection and blockchain-based trust management. However, three gaps remain. First, few studies clearly define blockchain as an active decision layer rather than a passive log. Second, real-time edge deployment is often discussed conceptually but not emphasized as a practical system design choice. Third, many papers report classification accuracy alone and provide limited discussion of system-level trade-offs such as latency, throughput, and energy. This work addresses the first two gaps directly and identifies the third as an important direction for future evaluation.

Furthermore, a detailed analysis of existing literature reveals a stark division between machine learning implementations and blockchain applications. Studies focusing primarily on ML-based anomaly detection often lack a decentralized trust layer and fail to establish an edge mitigation loop. For instance, while classical models and advanced deep learning architectures prioritize classification accuracy, they do not translate these detections into automated, accountable security actions at the device level. Conversely, research centered on blockchain-based trust management frequently encounters challenges related to transaction latency and smart contract overhead. In many of these architectures, the blockchain acts merely as a passive archival tool for reputation storage rather than an active component of the detection pipeline. Even in hybrid approaches that combine both technologies, the integration remains resource-intensive and lacks a complete end-to-end control loop. Therefore, a critical need exists for an architecture that tightly couples temporal anomaly detection with dynamic trust scoring at the edge.

III. PROPOSED METHODOLOGY

System Overview

The proposed system consists of five main stages: traffic capture, preprocessing and feature extraction, anomaly scoring, trust-score management, and automated response. Let the feature sequence generated for a device over a monitoring window be

$$X = [x_1, x_2, \ldots, x_L], x_t \in \mathbb{R}^F,$$

where Lis the sequence length and Fis the number of extracted traffic features per time step. The feature sequence is processed by an LSTM-based anomaly detector f_θ (·), parameterized by θ, to estimate whether the observed traffic deviates from learned normal behaviour.

The overall pipeline operates as follows. Network packets are captured from IoT devices and converted into feature sequences. The anomaly detector produces an anomaly score for each sequence. This score is then used to update the device trust score maintained through a blockchain-backed trust module. Based on the updated trust state, the system triggers appropriate actions such as continued monitoring, alert generation, or device isolation. In this way, the framework forms a closed-loop security pipeline rather than a simple detect-and-log architecture.

Traffic Capture and LSTM-Based Anomaly Detection

Traffic capture and preprocessing.

IoT traffic is passively monitored using packet-sniffing interfaces over Wi-Fi or Ethernet. The captured packets are forwarded to an edge device, namely a Raspberry Pi 4, where relevant traffic descriptors are extracted. Typical features include packet size, connection frequency, communication duration, and behavior-pattern indicators. These features are normalized and arranged into temporal sequences for model input.

Anomaly detection model.

The detector is based on LSTM because network traffic exhibits sequential dependence over time. Unlike purely feedforward classifiers, LSTM can preserve temporal context across observations, which is useful for identifying evolving attack behavior.

Let X^=f_θ (X)denote the reconstructed or predicted representation of the input sequence. The anomaly score is computed using mean squared reconstruction error:

$$A(X) = \frac{1}{LF} \| X - \hat{X} \|_2^2.$$

Here, A(X)is the anomaly score, Xis the original feature sequence, and X^is the output of the LSTM-based model. Larger values of A(X)indicate greater deviation from learned normal behavior. A decision threshold δis then used to classify the sequence:

$$\text{label}(X) = \begin{cases} \text{normal}, & A(X) \le \delta, \\ \text{anomalous}, & A(X) > \delta. \end{cases}$$

This formulation standardizes the notation that was previously described only informally. It also makes clear that anomaly detection is driven by reconstruction deviation rather than direct signature matching.

Algorithm 1 Anomaly Detection Using LSTM-Based Autoencoder

Input: Network traffic features X, trained autoencoder model $f(X)$
Output: Anomaly Score A_s
Initialize LSTM-based autoencoder with trained parameters.
Extract feature vector X from network traffic.
Generate reconstructed output X from the autoencoder: $X = f(X)$
Compute Anomaly Score using Mean Squared Error (MSE):
$A_s = \|X - X\|^2$
if $A_s > \tau$ **then**
    Classify as anomaly.
else
    Classify as normal traffic.
end if
Forward anomaly detection results to the Blockchain Trust System.
Return A_s

Fig. 2. Anomaly Detection Algorithm using LSTM-Based Autoencoder.

Blockchain-Based Trust Management

Trust representation.

Let $T_d^{(k)} \in [0,1]$denote the trust score of device d after the k-th observation or transaction, where larger values indicate higher trustworthiness. The anomaly score produced by the detection model is normalized to A_d^((k))∈[0,1]before trust updating. The trust score is updated as

$$T_d^{(k+1)} = \alpha T_d^{(k)} + (1-\alpha)\left(1 - A_d^{(k)}\right),$$

where α∈(0,1)is the trust-retention factor. This equation has an intuitive interpretation. When the anomaly score is small, the second term is large and the device retains or improves its trust state. When the anomaly score is high, the second term decreases and the device is penalized. Thus, trust

evolves gradually rather than changing abruptly after a single event.

Why trust scoring matters.

A single anomalous observation may arise from transient noise or unusual but benign behavior. Therefore, trust management provides temporal smoothing and prevents overreaction to isolated events. At the same time, repeated abnormal behavior causes cumulative trust degradation, making the system more robust against persistent threats.

Blockchain validation.

To preserve integrity and auditability, trust updates are recorded on a blockchain. Let H_kdenote the hash of the current block and H_(k-1)the previous block hash. A compact block representation can be expressed as

$$H_k = SHA256(H_{k-1} \,||\, T_d \,||\, TX)$$

where TX_krepresents transaction data and ts_kis the timestamp for the k-th trust update. The concatenation operator is denoted by ‖. Because each block depends on the previous block hash, any unauthorized modification breaks the chain consistency.

Algorithm 2 Trust Score Update Mechanism

Input: Previous trust score T_{prev}, Anomaly Score A_s, Trust decay factor α
Output: Updated Trust Score T_d
Retrieve last recorded trust score T_{prev} from blockchain.
Compute updated trust score:
$T_d = \alpha T_{prev} + (1-\alpha)(1-A_s)$
Update T_d on the blockchain.
if $T_d < \tau$ **then**
    Flag device as low trust.
end if
Return T_d

Fig. 3. Trust Score Update Algorithm.

Algorithm 3 Blockchain-Based Trust Score Validation

Input: Updated trust score T_d, Previous block hash B_{prev}, Transaction data TX
Output: Blockchain hash H
Collect trust update transaction details.
Compute cryptographic hash:
$H = SHA-256(T_d \| B_{prev} \| TX)$
Store the generated hash H in the blockchain ledger.
Verify integrity by cross-checking with previous transactions.
Return H

Fig. 4. Blockchain-Based Trust Validation Algorithm.

Automated Threat Mitigation and Security Workflow

Decision policy.

The updated trust score determines the system response through two thresholds, τ_1and τ_2, with τ_1<τ_2:

$$\text{Response}(d) = \begin{cases} \text{Normal operation}, & T_d \geq \tau_2, \\ \text{Alert and monitor}, & \tau_1 \leq T_d < \tau_2, \\ \text{Isolate device}, & T_d < \tau_1. \end{cases}$$

This policy ensures that devices are not disconnected purely on the basis of a single raw anomaly score. Instead, the system combines model evidence and trust history before taking action.

Response mechanisms.

When a device enters a suspicious or untrusted state, the framework can activate a buzzer alert, display the status on an OLED screen, log the event to the dashboard, and, when necessary, isolate the device using a relay module. These actions translate analytical outputs into operational security control.

Algorithm 4 Security Response Decision

Input: Trust Score T_d, Anomaly Score A_s, Predefined thresholds τ_1, τ_2
Output: Security action decision
if $T_d > \tau_1$ **then**
    Allow device to operate normally.
else if T_d is between τ_1 and τ_2 **then**
    Flag the device for monitoring.
else if $T_d < \tau_2$ **then**
    Isolate the device from the network using the relay module.
    Trigger buzzer and OLED alerts for immediate security notification.
    Log the event in the blockchain for auditability.
end if
Return action decision.

Fig. 5. Security Response Decision Algorithm.

Dashboard support.

A web-based dashboard enables administrators to monitor trust scores, view anomaly logs, inspect blockchain-validated events, and intervene manually if required. This human-in-the-loop element is important for practical deployment because highly sensitive environments may require confirmation before device isolation.

Petri-net interpretation.

The workflow is also represented through a Petri net to capture the logical progression of events. Tokens represent the state of a device or traffic event, while transitions represent processing stages such as capture, detection, trust update, validation, and response. This modeling choice improves conceptual clarity and demonstrates that the proposed

framework is not just a collection of modules, but a coordinated control process.

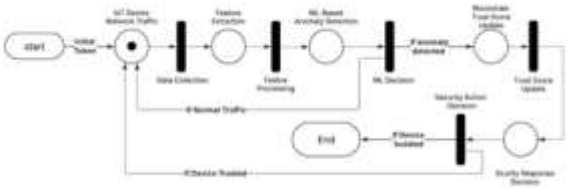

Fig. 6. Petri Net Model of security workflow.

Scalability Considerations

A key reviewer concern is blockchain scalability. This concern is valid because trust updates for large IoT deployments can create bottlenecks in block generation, validation latency, storage growth, and smart-contract execution cost. In practical terms, if every packet-level event were written directly on-chain, the ledger could quickly become a throughput bottleneck and undermine real-time security response.

The present framework is therefore best interpreted as a permissioned or private-chain security architecture suited to managed IoT environments rather than a public-chain design for internet-scale deployment. For such deployments, scalability can be improved by recording compact trust events instead of raw traffic traces, batching multiple trust updates into a single block, maintaining detailed packet logs off-chain, and storing only hashes or summarized security evidence on-chain. Consensus simplification in trusted administrative domains can also reduce latency. These considerations do not invalidate the framework; rather, they define the deployment envelope within which blockchain-backed trust remains practical.

IV. RESULTS AND DISCUSSION

LSTM-Based Anomaly Detection Performance

The anomaly detection model was evaluated using the NSL-KDD dataset [11]. After preprocessing and feature scaling, the LSTM-based model was trained on temporalized traffic samples and tested on held-out data. The model achieved an accuracy of 98.85%, and the corresponding precision, recall, and F1-score values were also high, indicating strong discrimination between normal and anomalous traffic.

To keep the paper focused on methodology and analysis, implementation-level code listings are omitted from the main text and may be placed in an appendix or an external repository. This revision addresses the concern that the original manuscript contained excessive implementation detail in the results section.

Accuracy: 0.9885537301373553

	precision	recall	f1-score	support
0	0.99	0.99	0.99	15450
1	0.99	0.99	0.99	14254
accuracy			0.99	29704
macro avg	0.99	0.99	0.99	29704
weighted avg	0.99	0.99	0.99	29704

Fig. 7. Accuracy Metrics of the LSTM-Based Anomaly Detection Model.

Confusion matrix analysis.

The confusion matrix in Fig. 8 provides a more informative view than accuracy alone. A strong concentration of samples along the main diagonal indicates that the detector correctly identifies most instances of both normal and anomalous traffic. The off-diagonal entries correspond to false positives and false negatives.

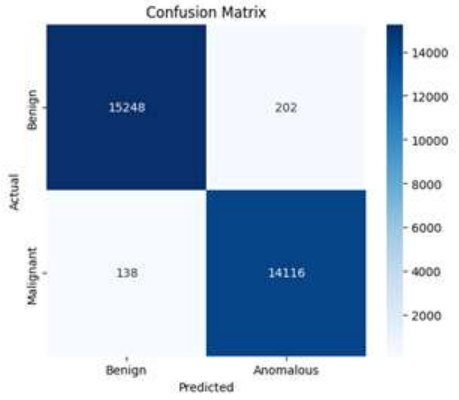

Fig. 8. Confusion Matrix Representing Classification Performance.

In the context of this work, false positives are especially important because they may unnecessarily reduce device trust and trigger avoidable alerts or isolation. False negatives are equally critical because

they allow malicious behavior to persist undetected. Therefore, the confusion matrix supports the claim that the model is suitable not only for classification, but also for downstream trust-based control.

ROC analysis.

The ROC curve in Fig. 9 illustrates the trade-off between true positive rate and false positive rate across decision thresholds. The curve's shape toward the upper-left region indicates strong separability between normal and anomalous traffic. This is important because anomaly detection thresholds often need to be tuned for different IoT environments. A favorable ROC profile suggests that the proposed detector can maintain good sensitivity without excessively inflating false alarms.

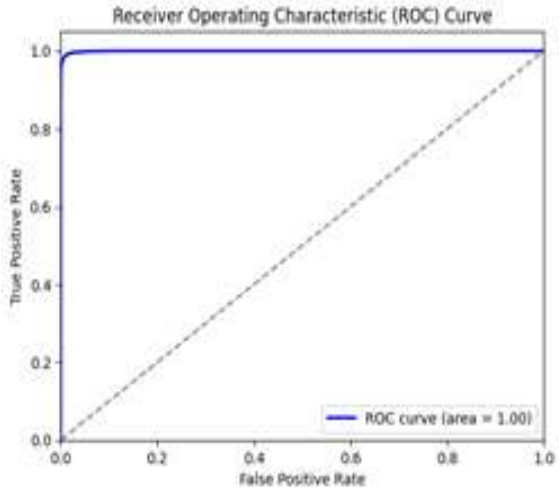

Fig. 9. ROC Curve.

Blockchain Trust Simulation and System-Level Discussion

The blockchain-based trust mechanism was simulated using a local Ganache environment. Trust scores were updated from anomaly outputs and recorded through a smart-contract-based workflow. This setup demonstrates the feasibility of combining anomaly evidence, trust adaptation, and tamper-resistant logging within a unified framework.

Trust score evolution.

Fig. 10 shows how trust changes over successive transactions. The gradual downward trend under repeated abnormal activity reflects the effect of the trust-retention factor α, which prevents unstable trust oscillations while still penalizing persistent anomalies. This is desirable in practice because transient irregularities should not immediately cause device removal, whereas repeated suspicious behavior should produce a steady and explainable trust decline.

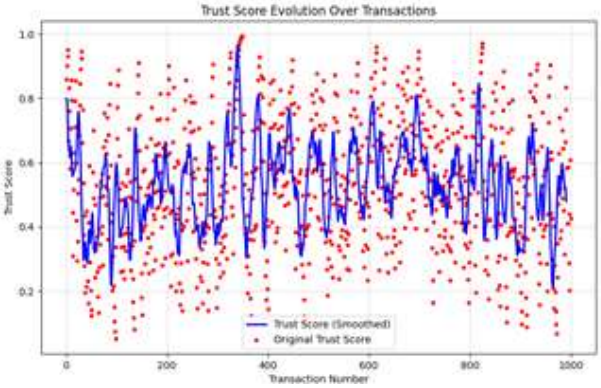

Fig. 10. Trust Score Evolution Over Successive Transactions in the Blockchain Framework.

Anomaly-trust coupling.

Fig. 11 highlights the direct relationship between anomaly score and trust degradation. Higher anomaly scores produce stronger trust penalties, which means the blockchain layer is not independent of the ML detector but operationally driven by it. This supports the central claim of the paper: the proposed framework is a coupled ML-blockchain security system, not a simple side-by-side integration of two unrelated components.

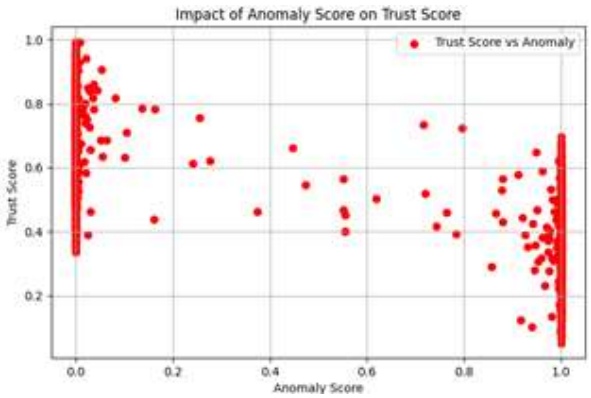

Fig. 11. Impact of Anomaly Score on Trust Score in the Blockchain-Based Trust Mechanism.

System-level scope.
The current experimental section validates detection effectiveness and trust-score evolution. However, broader system-level benchmarking remains an important extension. In particular, comparisons with CNN-based, pure Autoencoder-based, and Transformer-based anomaly detectors, evaluation on more recent IoT datasets such as CICIDS and IoT-23, and measurement of latency, throughput, and energy usage would further strengthen external validity. These are not claimed as completed results in the current study and are therefore identified as high-priority future work rather than presented as unsupported findings.

Overall, the results show that the anomaly detector provides strong classification performance, while the blockchain-backed trust module adds accountability, temporal consistency, and auditable response logic. Together, these properties make the framework suitable for real-time, decentralized IoT security in controlled edge deployments.

V. CONCLUSION

This paper presented a blockchain-enhanced LSTM-based framework for real-time IoT security at the edge. The system combines temporal anomaly detection, blockchain-backed trust management, and automated mitigation within a unified security loop. The novelty of the approach lies in the active use of blockchain for trust control, the emphasis on edge deployment, and the coupling between anomaly evidence, trust adaptation, and operational response. The experimental results show that the LSTM-based detector achieves strong anomaly-classification performance on NSL-KDD, while the trust simulation demonstrates that abnormal behaviour can be translated into explainable and tamper-resistant trust degradation over time.

The confusion matrix and ROC analysis further indicate that the detector is suitable for security applications in which both missed detections and false alarms have operational consequences. At the same time, the study has clear scope boundaries. Blockchain scalability remains a practical challenge as IoT network size grows, especially with respect to transaction latency, storage overhead, and throughput. In addition, broader comparative evaluation against CNN, Autoencoder, and Transformer baselines, as well as benchmarking on newer datasets such as CICIDS and IoT-23, would provide stronger evidence of generality. Future work will therefore focus on three directions: expanding comparative experiments, quantifying latency/throughput/energy for edge deployment, and optimizing blockchain design for scalable trust management in larger IoT environments.

VI. DECLARATIONS

Funding
The authors declare that no financial support, external grants, or corporate funding was received for the design, execution, authorship, and/or publication of this article. This research was conducted independently as part of the authors' academic and investigative pursuits at the School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai.

Conflicts of Interest
The authors declare that they have no known competing financial interests, personal relationships, or professional affiliations that could have appeared to influence the objectivity of the work reported in this paper. Furthermore, the authors declare no commercial affiliations with or endorsements from the hardware and software brands utilized in this study (such as Raspberry Pi, Google Coral, or Ganache), which were selected strictly for experimental, open-source, and academic demonstration purposes.

Data Availability
The network traffic data analysed during the current study is derived from the publicly accessible NSL-KDD dataset, which is openly available in the IEEE Dataport repository (DOI: 10.21227/425a-3e55). No proprietary, confidential, or restricted institutional data was generated or utilized in the training and evaluation of the anomaly detection models presented in this manuscript.

Code Availability

The custom Python scripts used for training the LSTM-based anomaly detection model, as well as the Solidity smart contract source code designed for the blockchain trust management simulation, are available from the corresponding author upon reasonable academic request. The codebase and simulated environment parameters are provided strictly to facilitate research reproducibility, peer verification, and non-commercial academic use.

REFERENCES

1. Sharma and H. Babbar, "Machine Learning-Based Anomaly Detection in the Internet of Things," 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023, pp. 1-6, doi: 10.1109/ASIANCON58793.2023.10270100.
2. A. Obaidli, D. Mansour, S. M. Abdulhamid, N. B. Halima and A. Al-Ghushami, "Machine Learning Approach to Anomaly Detection Attacks Classification in IoT Devices," 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 2023, pp. 1-6, doi: 10.1109/ICAISC56366.2023.10085349.
3. K. Mithran and C. Gopi, "Anomaly Detection in IoT Sensor Networks using Machine Learning," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 2022, pp. 1-7, doi: 10.1109/IC3SIS54991.2022.9885575.Lahbib, K. Toumi, A. Laouiti, A. Laube and S. Martin, "Blockchain based trust management mechanism for IoT," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-8, doi: 10.1109/WCNC.2019.8885994.
4. V. Cp, S. Kalaivanan, R. Karthik and A. Sanjana, "Blockchain-based IoT Device Security," 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India, 2022, pp. 1-6, doi: 10.1109/AISP53593.2022.9760674.
5. Y. Liu, J. Wang, Z. Yan, Z. Wan and R. Jäntti, "A Survey on Blockchain-Based Trust Management for Internet of Things," in IEEE Internet of Things Journal, vol. 10, no. 7, pp. 5898-5922, 1 April1, 2023, doi: 10.1109/JIOT.2023.3237893.
6. M. Amiri-Zarandi and R. A. Dara, "Blockchain-based Trust Management in Social Internet of Things," 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 2020, pp. 49-54, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00024.
7. V. N and P. T. V. Bhuavneswari, "ADBIS: Anomaly Detection to Bolster IoT Security Using Machine Learning," 2023 IEEE 3rd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), Bhubaneswar, India, 2023, pp. 1-6, doi: 10.1109/AESPC59761.2023.10390100.
8. J. Panduro-Ramirez, R. S. V. Rama Swathi, A. Juyal, J. M. Ruiz-Salazar, H. S. Bedi and D. Verma, "Blockchain-based Trust
9. Management System in Implementation in Social Internet of Things," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 78-83, doi: 10.1109/IC3I56241.2022.10072999.
10. X. -W. Wu, Y. Cao and R. Dankwa, "Accuracy vs Efficiency: Machine Learning Enabled Anomaly Detection on the Internet of Things," 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), BALI, Indonesia, 2022, pp. 245-251, doi: 10.1109/IoTaIS56727.2022.9975889.
11. Ghulam Mohi-ud-din, December 29, 2018, "NSL-KDD", IEEE Dataport, doi: https://dx.doi.org/10.21227/425a-3e55.