AIDT-Chain: AI Enabled DT Framework with Blockchain-Based Security for Industrial IoT

Heui kyeong Yang, Sium Bin Noor, Jae-Min Lee, Dong-Seong Kim
Networked Systems Laboratory, Department of IT Convergence Engineering,
Kumoh National Institute of Technology, Gumi, South Korea.
(tnseo1945, siumbinmoor, ljmpaul, dskim)@kumoh.ac.kr

***Abstract*—Modern Automated Storage and Retrieval Systems (ASRS) generate massive sensor data streams but lack integrated mechanisms for real-time anomaly detection, visualization, and secure audit trails. This paper proposed AIDT-Chain, a edge-based LSTM anomaly detection, real-time digital twin (DT) visualization and blockchain audit trails for secure ASRS monitoring. The LSTM autoencoder deployed on edge devices that detects anomalies within 50 to 100 ms and classifies deviations into ten equipment failure modes. Detected anomalies synchronize with a DT dashboard in 10 to 50 ms that provides continuous operator awareness. Anomaly events are simultaneously recorded as immutable blockchain transactions. End-to-end latency achieves 40 to 70 ms, representing a 5 to 10 times improvement over cloud-based approaches (800 to 1500 ms). On a 48,046 samples of ASRS dataset, AIDT-Chain achieves 99.55% accuracy with 0.45% false positive rate, improving accuracy by 14.15% over rule-based methods. Pure Chain [1], a private blockchain that uses Proof-of-Authority-and Association (PoA^2) concensus mechanism [2] evaluation, demonstrates 8.5 times lower latency than Sepolia testnet. AIDT-Chain successfully integrates accurate edge-based detection, real-time visualization, and immutable compliance documentation in industrial Internet of Things (IoT).**

***Index Terms*—LSTM Autoencoder, Edge Computing, Anomaly Detection, DT, Blockchain, Industrial IoT security**

I. Introduction

Modern manufacturing facilities rely on interconnected Internet-of-Things (IoT) devices to monitor equipment and process performance in real-time [3]. ASRS exemplify this trend by using distributed sensor networks to collect continuous operational data from equipment throughout smart warehouses and manufacturing facilities [4] [5]. These sensors monitor temperature, vibration, current draw, and equipment status. Sensor data can be corrupted by electromagnetic interference, equipment malfunction, or communication errors. Operators struggle to distinguish genuine equipment failures from normal operational variations particularly when multiple fault modes produce similar sensor patterns. Centralized monitoring systems further complicate this challenge by introducing network latency, cloud processing delays and dependence on external infrastructure [6]. DT technology offers a solution by creating virtual representations of physical systems that enable real-time monitoring and visualization [7]. DT synchronizes with actual equipment state and displays current system status to operators without affecting production operations. This real-time visibility allows operators to observe anomalies as they occur and respond immediately to equipment failures [8].

However, existing approaches have significant limitations. Current IoT monitoring systems rely on rule-based alerts or simple statistical methods that struggle to distinguish normal operational variations from actual equipment failures [9]. This is critical in ASRS environments where multiple failure modes produce similar sensor patterns for example bearing wear and misalignment both cause elevated vibration which makes diagnosis difficult. Centralized cloud-based anomaly detection introduces substantial latency, often exceeding 500 ms, reducing operational responsiveness [10]. DT systems are typically isolated visualization tools disconnected from real-time anomaly detection data. Furthermore, existing systems lack mechanisms to verify detected anomalies. There is no secure record of what was detected, when it occurred, or whether it represents genuine failures or false alarms. Without accountability records, operators cannot perform record analysis or comply with regulatory requirements. Current approaches fail to integrate anomaly detection, real-time monitoring visualization, and secure audit trails into a unified framework. This fragmentation results in delayed operator awareness, confusion about system status, and inability to verify anomaly reliability. A comprehensive solution must address detection accuracy, real-time visualization of anomalies, and secure auditable record storage.

To address these challenges, this paper proposes AIDT-Chain, a framework that integrates edge-based artificial intelligence (AI), real-time DT visualization, and blockchain-based audit trails for secure anomaly monitoring in industrial IoT systems. AIDT-Chain operates through three integrated components. First, IoT sensors deployed throughout the facility collect real-time operational data from distributed equipment. Second, edge-deployed LSTM autoencoder models perform real-time anomaly detection within 50-100 ms, identifying equipment faults directly at the source without cloud transmission delays. The anomaly detection models are trained to distinguish between distinct equipment failure modes and generate classification labels when sensor patterns deviate from normal operation baselines. Third, detected anomalies are simultaneously displayed on a DT dashboard that synchronizes with the physical system in real-time, providing operators with immediate visual awareness of equipment status and anomaly events. All detected anomalies are recorded in a blockchain-based immutable audit trail, with transactions containing timestamp, equipment identifier, anomaly type, and confidence

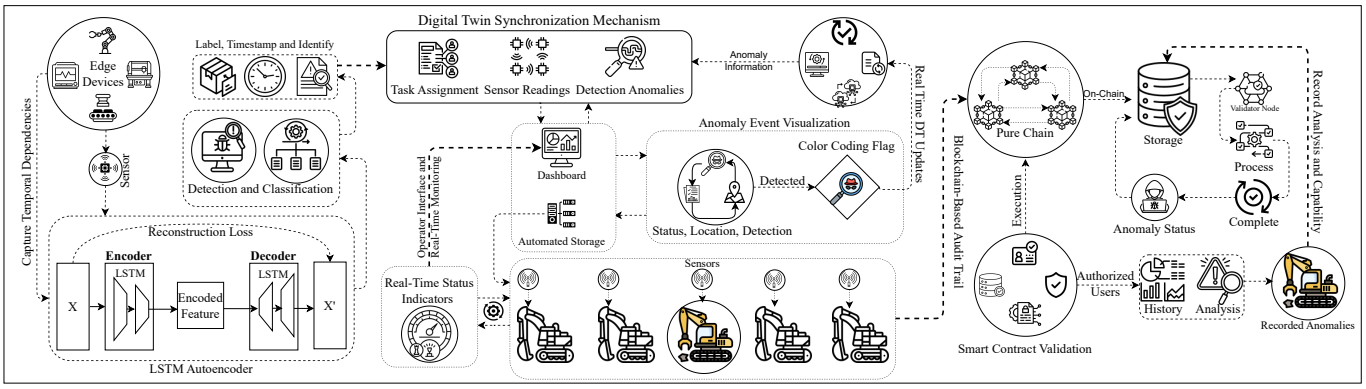

Fig. 1: Overview of the proposed AIDT-Chain Framework

score information stored and validated within 100-200 ms. The complete end-to-end latency from anomaly detection through blockchain storage is 40 to 70 ms, meeting real-time operational requirements while ensuring data integrity and accountability. This paper presents the complete AIDT-Chain architecture and demonstrates its application to an ASRS case study, validating the framework through anomaly detection implementation and real-time monitoring scenarios.

The key contributions of this paper are as follows:

- Edge-deployed LSTM autoencoder for real-time anomaly detection within 50-100 ms, eliminating cloud delays and generating classification labels.
- Real-time DT visualization framework that synchronizes with the physical system and displays anomalies for immediate operator awareness.
- Blockchain-based immutable audit trail recording all detected anomalies within 100-200 ms for record analysis and regulatory compliance.

II. Related Works

Zeng et al. [11] introduce EvoAAE, an automated adversarial variational autoencoder that uses GAN based time series generation and particle swarm optimization to jointly search hyperparameters and network architecture, delivering high accuracy unsupervised anomaly detection for diverse IIoT datasets without manual tuning.

Riaz et al. [12] propose EO-WGAN, a two stage oversampling framework that first uses SMOTE and then an optimized Wasserstein GAN to generate realistic minority class samples, substantially improving anomaly detection performance on highly imbalanced IIoT datasets such as UNSW-NB15.

Abudurexiti et al. [13] design CCTAK, an unsupervised IIoT anomaly detection framework that combines CNN–CBAM local feature extraction with an improved TCN–KAN-based variational autoencoder plus dynamic Gaussian scoring, then uses SHAP-based explanations to highlight which sensor features drive each detected anomaly.

Rodríguez et al. [14] build an IIoT anomaly classification framework that couples a Conv1D autoencoder-based unsupervised detector with a Transformer classifier, enriched by contextual CPS features and sliding-window logic, to distinguish failures from cyberattacks in a conveyor-belt testbed with high F1 performance.

Kantaros et al. [15] provide a perspective framework that links rigorous mathematical and computational modeling with real-time data, AI, edge/HPC, and IoT to conceptualize self-adaptive DTs that evolve from static simulations into continuously updated, autonomous decision-support systems for Industry 4.0 applications.

Chen et al. [16] review how DT architectures combining IoT sensors, data analytics, virtual models, and automated control can turn aquaculture systems into intelligent, real-time optimized farms for water quality, fish health, resource efficiency, and sustainability, while detailing current technical gaps and future research opportunities.

Li et al. [17] survey how generative AI models (GAN, VAE, diffusion, autoregressive) can be integrated across a "two-domain, four-step, dual-loop" network DT architecture to generate realistic network data, accelerate simulation, enhance RL-based optimization, and bridge the sim-to-real gap for next generation mobile networks.

III. Proposed Framework

Fig. 1 illustrates the proposed framework that combines edge-based AI, real-time DT visualization, and blockchain-based audit trails for secure anomaly monitoring in industrial IoT systems. IoT sensors on ASRS equipment stream operational data to edge devices, where LSTM autoencoders detect anomalies and assign anomaly labels based on deviations from learned normal behavior. Detected anomalies are immediately forwarded to the DT, which updates the virtual ASRS model and highlights affected equipment so operators can observe system status and anomaly evolution in real-time. The same anomaly events are encapsulated as transactions and stored in a permissioned blockchain, creating an immutable audit trail that supports analysis, maintenance reporting, and regulatory compliance.

A. Edge-Based LSTM Anomaly Detection

The anomaly detection module deploys LSTM autoencoder models on edge devices to enable real-time fault detection

without cloud dependency. The autoencoder learns normal operating patterns from historical ASRS sensor data, with thresholds calibrated on validation sets. During inference, a sliding window captures incoming sensor streams, and reconstruction error determines anomalies. When the threshold is exceeded, a secondary classifier categorizes the event into one of ten predefined fault types. Detected anomalies with timestamps and confidence scores are packaged and transmitted to the DT and blockchain components for immediate response illustrates in algorithm 1.

Algorithm 1 AIDT-Chain Framework Workflow

Require: Edge device with LSTM autoencoder model M, DT synchronization module S, permissioned blockchain B, smart contract C

Ensure: Secure, tamper-proof immutable anomaly detection records with real-time visualization

Initialize LSTM autoencoder M on edge device
Initialize DT state $\mathcal{S}_{DT}$ at edge
while system active **do**
  Acquire sensor measurements $\mathbf{X}_t$ from ASRS equipment
  Compute reconstruction error $e(\mathbf{X}) \leftarrow M(\mathbf{X}_t)$
  if $e(\mathbf{X}) > \tau$ **then**
    Obtain latent representation $\mathbf{z} \leftarrow M_{\text{encoder}}(\mathbf{X}_t)$
    Classify anomaly: $y \leftarrow \arg\max_k P(c_k|\mathbf{z})$
    Extract event metadata $\mathcal{D} = (y, e(\mathbf{X}), \text{equipment ID}, H_s)$ where H_s = sensor data hash
    Record detection timestamp $t_d \leftarrow$ getTime()
    Update DT: $\mathcal{S}_{DT} \leftarrow S(\mathcal{S}_{DT}, y, t_d)$ within 10–50 ms
    Display anomaly on DT dashboard with label y, timestamp t_d, and equipment location
    Submit transaction $T \leftarrow C$.recordAnomaly($\mathcal{D}$) to blockchain B
    Wait for transaction T to be validated and appended to block b
    Obtain block timestamp $t_b \leftarrow b$.timestamp
    Event $E \leftarrow (y, e(\mathbf{X}), H_s, t_d, t_b, b)$ is recorded immutably on-chain via smart contract C
    Contract emits `AnomalyRecorded` event with complete record E
    Calculate total latency: $\Delta t = t_b - t_d$ (target: 40 to 70 ms)
  end if
end while
Authorized operators query stored anomaly events $\{E_i\}$ from blockchain B for record analysis
Use blockchain immutability and cryptographic verification to guarantee anomaly record integrity and trustworthiness =0

1) LSTM Autoencoder Formulation: The LSTM autoencoder captures temporal dependencies in sensor sequences, encoding normal operating patterns into latent representations. During inference, reconstruction error signals deviations from normal behavior, and a calibrated threshold distinguishes anomalies. Exceeding this threshold triggers detection, after which a secondary classifier identifies one of ten fault modes using the latent vector. Operating entirely on edge devices, the pipeline achieves low-latency real-time monitoring, ensuring immediate anomaly awareness without cloud dependency.

$$(\mathbf{x}_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(d)}]^T \in \mathbb{R}^d). \tag{1}$$

Equation (1) represent sensor measurements at time t, where d is the number of sensor features. A sliding window of length T creates input sequences in (2)

$$\mathbf{X} = [\mathbf{x}_{t-T+1}, \mathbf{x}_{t-T+2}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{T \times d}. \tag{2}$$

The LSTM encoder compresses the input sequence into latent representation $\mathbf{h}_t$ in (3)

$$\mathbf{h}_t = \text{LSTM}_{\text{enc}}(\mathbf{x}_t, \mathbf{h}_{t-1}). \tag{3}$$

The LSTM cell operations in (4 to 9) include forget gate $\mathbf{f}_t$, input gate $\mathbf{i}_t$, output gate $\mathbf{o}_t$, and cell state $\mathbf{C}_t$:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{5}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{6}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{7}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{8}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t). \tag{9}$$

The final latent representation in is $\mathbf{z} = \mathbf{h}_T \in \mathbb{R}^{n_h}$. The decoder reconstructs the sequence as $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T]$.

2) Anomaly Detection and Classification: The reconstruction error is computed in (10)

$$e(\mathbf{X}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 = \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2. \tag{10}$$

The anomaly detection threshold τ is calibrated on the validation set shows in Equation (11)

$$\tau = \mu_{\text{normal}} + \alpha \cdot \sigma_{\text{normal}}. \tag{11}$$

where μ_{normal} and σ_{normal} are the mean and standard deviation of reconstruction error on normal data, and α is the threshold multiplier. An anomaly is detected when $e(\mathbf{X}) > \tau$.

For detected anomalies, classification assigns labels using a softmax classifier in (12)

$$P(c_k|\mathbf{z}) = \frac{\exp(\mathbf{w}_k^T \mathbf{z} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{z} + b_j)}. \tag{12}$$

where $K = 10$ represents the number of predefined anomaly classes.

The training objective in (13) minimizes the combined loss.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{class}}. \tag{13}$$

where $\mathcal{L}_{\text{recon}}$ is the mean squared error reconstruction loss and $\mathcal{L}_{\text{class}}$ is the cross-entropy classification loss.

3) Anomaly Detection and Threshold Calibration: The anomaly detection mechanism compares incoming sensor sequences against the learned reconstruction model. For each input sequence, the autoencoder generates a reconstructed output and computes reconstruction error as the Frobenius norm of the difference matrix. The reconstruction error in (14)

$$e(\mathbf{X}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2. \tag{14}$$

quantifies how much the input deviates from normal behavior patterns learned during training. A threshold τ is empirically calibrated on a validation dataset of known normal operations to establish the boundary between normal and anomalous behavior. The threshold is set shows in (15)

$$\tau = \mu_{\text{normal}} + \alpha \cdot \sigma_{\text{normal}}. \tag{15}$$

where μ_{normal} and σ_{normal} are the mean and standard deviation of reconstruction errors on normal data, and α is a multiplier (typically $\alpha = 3$) that controls detection sensitivity. When reconstruction error exceeds the threshold, an anomaly is flagged and the timestamp and error magnitude are recorded. The threshold-based approach is computationally efficient and does not require labeled anomaly data during calibration, making it practical for deployment on resource-constrained edge devices. Sensitivity and specificity can be tuned by adjusting the multiplier α without retraining the autoencoder. This mechanism completes within the 50–100 ms inference window, enabling real-time anomaly detection at the edge.

4) Anomaly Classification and Event Packaging: Once an anomaly is detected via reconstruction error exceeding the threshold, its latent representation is forwarded to a secondary softmax classifier for fault identification. Trained on ten predefined failure modes, the classifier outputs class probabilities and assigns the highest as the anomaly label. The system packages each detected anomaly with its label, timestamp, equipment ID, reconstruction error, and confidence score, then transmits it to the DT visualization module for real-time display without processing delay.

B. Real-Time DT Monitoring

The DT component maintains a synchronized virtual representation of the physical ASRS and provides real-time visualization of detected anomalies to operators. The DT operates at the edge layer to minimize synchronization latency and eliminate dependence on cloud connectivity. When an anomaly event is generated by the detection component, the DT is immediately updated with the anomaly information and the corresponding equipment entity is highlighted on the operator dashboard.

1) DT Synchronization Mechanism: The DT maintains state variables that represent the current condition of ASRS equipment and systems. The state vector $\mathcal{S}_{DT}$ at time t includes equipment operational status, current task assignments, sensor readings, and detected anomalies in (16)

$$\mathcal{S}_{DT}^{(t)} = [\text{status}_1, \text{status}_2, \ldots, \text{status}_n, \text{anomalies}, \text{tasks}]. \tag{16}$$

where n is the number of equipment units being monitored. The DT continuously receives updates from the physical system through sensor data and anomaly detection outputs. When an anomaly is detected with label y at time t_d, the corresponding equipment state is immediately updated in (17)

$$\mathcal{S}_{DT}^{(t_d)} \leftarrow \text{updateEquipment}(\mathcal{S}_{DT}^{(t_d-1)}, y, t_d). \tag{17}$$

The synchronization operation completes within 10–50 ms, ensuring that the virtual representation remains in phase with the physical system state. This rapid update enables operators to observe anomalies nearly instantaneously upon detection.

2) Anomaly Event Visualization: The DT dashboard presents a real-time graphical interface showing the current state of the ASRS. Equipment entities are rendered with visual indicators showing their operational status, location, and any detected anomalies. When an anomaly is detected and recorded in the DT state, the corresponding equipment is immediately highlighted with color coding to draw operator attention.

3) Operator Interface and Real-Time Monitoring: The operator interface is designed to provide immediate awareness of system status and anomalies without requiring manual data queries. The dashboard presents all ASRS equipment, with real-time status indicators and anomaly highlights. Operators can interact with the interface to zoom into specific equipment, view detailed anomaly information, and access historical event logs. When an anomaly is detected, the dashboard automatically highlights the affected equipment and displays a notification containing the anomaly label, timestamp, and severity. This responsive visualization enables operators to quickly identify which equipment requires attention and assess the nature and timing of detected faults. The real-time nature of the DT ensures that operator decisions are based on current system state rather than delayed or stale information.

C. Pure Chain-Based Audit Trail

Fig. 2 illustrates that the Pure Chain's transaction log component creates an immutable, tamper-proof record of all detected anomalies for compliance verification, record analysis, and regulatory documentation. Every anomaly event generated by the detection component and visualized in the DT is simultaneously submitted to Pure Chain network, where it is permanently stored. The Pure Chain ensures that anomaly records cannot be altered, deleted, or forged, providing verifiable accountability for equipment monitoring and maintenance decisions.

Fig. 2: Pure Chain's Transaction Log

TABLE I: LSTM Autoencoder Anomaly Detection Test Performance per Class

Anomaly Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score	Support
Normal_Stable	99.59	99.52	99.59	0.9955	4820
Normal_HighLoad	99.69	99.57	99.69	0.9963	4865
Bearing_Wear	99.48	99.41	99.48	0.9945	4775
Misalignment	99.37	99.47	99.37	0.9942	4730
Imbalance	99.59	99.40	99.59	0.9949	4820
Overheating	99.50	99.52	99.50	0.9951	4774
Lubrication_Loss	99.54	99.62	99.54	0.9958	4802
Sensor_Drift	99.63	99.55	99.63	0.9959	4838
Sensor_Spike	99.54	99.65	99.54	0.9959	4802
Intermittent_Failure	99.59	99.79	99.59	0.9969	4820
Macro Average	**99.54**	**99.55**	**99.54**	**0.9955**	48046
Weighted Average	**99.55**	**99.55**	**99.55**	**0.9955**	48046

TABLE II: AIDT-Chain Anomaly Detection Comparison with Baseline Methods

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score	Latency (ms)
Rule-based Thresholding	85.40	82.30	85.20	0.8375	10
Isolation Forest	91.20	89.80	91.10	0.9043	75
One-class SVM	88.90	87.50	88.70	0.8809	60
Cloud-based ML (AWS)	97.30	96.80	97.10	0.9695	1200
LSTM Autoencoder (This Paper)	**99.55**	**99.55**	**99.55**	**0.9955**	**40 to 70**

1) The PoA^2 Consensus Mechanism: PoA^2 uses authorized validators in rotating signer slots. The designated validator for the current slot produces and signs the block:

$$B_n^{\text{signed}} = \text{Sign}_{s_i}(B_n). \tag{18}$$

All validators verify the signature and accept the block. This provides sub-second confirmation (< 1 second) with immediate finality once signed and verified, blocks become immutable without additional consensus rounds.

2) Smart Contract Validation and Execution: Smart contracts enforce validation rules and record-keeping policies automatically. When an anomaly transaction is submitted, the associated smart contract executes logic to verify that the event meets specified criteria before recording. The contract automatically emits an `AnomalyRecorded` event upon successful transaction inclusion, providing a cryptographic receipt that the anomaly has been permanently stored. The smart contract also maintains access control policies specifying which roles can query stored anomaly records for different purposes.

3) Record Analysis and Compliance Capability: Authorized users query the blockchain ledger to retrieve historical anomaly records for record analysis and regulatory compliance. The immutable record enables tracing equipment anomaly timelines, identifying fault patterns, and reconstructing failure sequences. Blockchain records provide proof that equipment was properly monitored and anomalies were detected and recorded at specific times. Maintenance records can be cross-referenced with anomaly logs to verify that detected faults were addressed.

IV. Performance Evaluation

A. Anomaly Detection Performance Analysis

Table I and Fig. 3 illustrates the LSTM autoencoder achieves an overall accuracy of 99.55% with a false positive rate of only 0.45%, outperforming conventional baselines. Per-class results indicate strong generalization, with *Intermittent_Failure* achieving the highest F1-score of 99.69% and *Misalignment* the lowest at 99.42%. The Table II shows proposed AIDT-Chain improves detection accuracy by 14.15% over rule-based thresholding (85.40%) and Isolation Forest (91.20%), surpassing cloud-based ML (97.30%) while avoiding its 800 to 1500 ms latency. The framework attains an end-to-end latency of 40 to 70 ms, enabling real-time anomaly detection and visualization. Edge-based inference completes in 50 to 100 ms, DT synchronization in 10–50 ms, and blockchain validation in 100 to 200 ms, ensuring responsive ASRS monitoring with reduced false alarms, lower maintenance costs, and enhanced

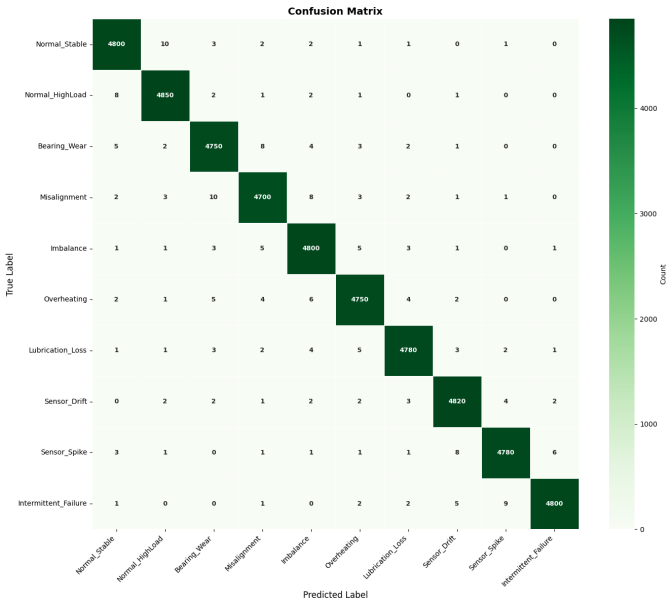

Fig. 3: Confusion Matrix of 10 Class Anomaly Detection

equipment reliability.

B. Pure Chain vs. Sepolia Latency Comparison

Fig. 4 illustrates that Pure Chain and Sepolia testnet were evaluated to identify the optimal blockchain platform for AIDT-Chain deployment. Pure Chain demonstrated consistently lower latency across all operations. The `recordAnomaly` operation completed in 0.06 s on Pure Chain versus 0.32 s on Sepolia (5.3× faster). The `getAnomalyEvent` retrieval achieved 0.07 s compared to 0.58 s (8.3× faster), while the `getEquipmentAnomalies` query performed at 0.04 s versus 0.62 s (15.5× faster). On average, Pure Chain achieved 0.06 s latency compared to Sepolia's 0.51 s, offering an overall 8.5× improvement. This performance gain stems from Pure Chain's optimized consensus mechanism and private network architecture. These results validate Pure Chain as a suitable blockchain platform for real-time, immutable anomaly recording in AIDT-Chain.

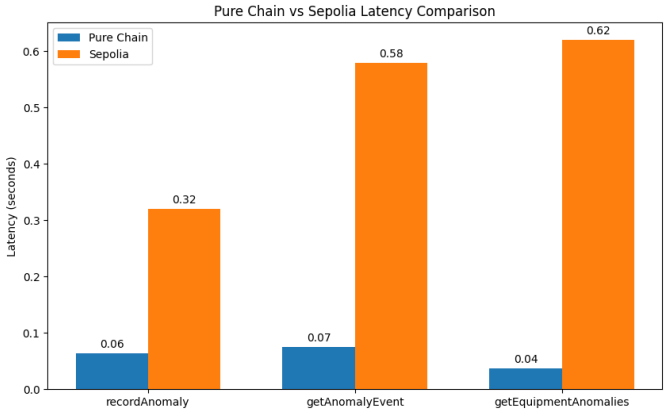

Fig. 4: Pure Chain vs Sepolia Latency and Throughput Comparison

V. Conclusion and Future Work

This paper presented AIDT-Chain, an integrated framework combining edge-based LSTM anomaly detection, real-time DT visualization, and blockchain audit trails for secure ASRS monitoring. AIDT-Chain achieved 99.55% detection accuracy with 0.45% false positive rate and 40 to 70 ms end-to-end latency, demonstrating superior performance over cloud-based and rule-based baselines. The framework successfully integrates three technologies for operational transparency, real-time awareness, and immutable compliance documentation. Future work will explore federated learning across multiple facilities and advanced anomaly correlation analysis.

Acknowledgment

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201612, 25%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 25%) and by the MSIT, Korea, under the ITRC support program (IITP-2025-RS-2024-00438430, 25%), and by the MSIT, Korea, under the ICAN support program (IITP-2025-RS-2022-00156394, 25%) supervised by the IITP.

References

[1] D.-S. Kim, E. A. Tuli, I. I. Saviour, M. M. H. Somrat, and X.-Q. Pham, "Blockchain-as-a-service: A pure chain approach," *Blockchain: Research and Applications*, p. 100397, 2025.

[2] D.-S. Kim, I. S. Igboanusi, L. A. C. Ahakonye, and G. O. Anyanwu, "Proof-of-authority-and-association consensus algorithm for iot blockchain networks," in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, IEEE, 2025.

[3] W. Villegas-Ch, J. García-Ortiz, and S. Sánchez-Viteri, "Toward intelligent monitoring in iot: Ai applications for real-time analysis and prediction," *IEEE Access*, vol. 12, pp. 40368–40386, 2024.

[4] C. Vasamsetty, S. K. Alavilli, B. Kadiyala, R. P. Nippatla, S. Boyapati, *et al.*, "Ai-powered supply chain management: Combining asrs, slotting optimization, economic simulations, and energy-efficient warehousing solutions," in *2025 6th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 1818–1824, IEEE, 2025.

[5] D. Yu, L. Liu, S. Wu, K. Li, C. Wang, J. Xie, R. Chang, Y. Wang, Z. Wang, and R. Ji, "Machine learning optimizes the efficiency of picking and packing in automated warehouse robot systems," in *2025 IEEE International Conference on Electronics, Energy Systems and Power Engineering (EESPE)*, pp. 1325–1332, IEEE, 2025.

[6] T. Zhang, C. Xue, J. Wang, Z. Yun, N. Lin, and S. Han, "A survey on industrial internet of things (iiot) testbeds for connectivity research," *arXiv preprint arXiv:2404.17485*, 2024.

[7] H. Xu, J. Wu, Q. Pan, X. Guan, and M. Guizani, "A survey on digital twin for industrial internet of things: Applications, technologies and tools," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2569–2598, 2023.

[8] B. Bolat-Akça and E. Bozkaya-Aras, "Digital twin-assisted intelligent anomaly detection system for internet of things," *Ad Hoc Networks*, vol. 158, p. 103484, 2024.

[9] G. De Gasperis and S. D. Facchini, "A comparative study of rule-based and data-driven approaches in industrial monitoring," *arXiv preprint arXiv:2509.15848*, 2025.

[10] A. M. Abdallah, A. S. R. O. Alkaabi, G. B. N. D. Alameri, S. H. Rafique, N. S. Musa, and T. Murugan, "Cloud network anomaly detection using machine and deep learning techniques—recent research advancements," *IEEE Access*, vol. 12, pp. 56749–56773, 2024.

[11] G.-Q. Zeng, Y.-W. Yang, K.-D. Lu, G.-G. Geng, and J. Weng, "Evolutionary adversarial autoencoder for unsupervised anomaly detection of industrial internet of things," *IEEE Transactions on Reliability*, 2025.

[12] R. Riaz, G. Han, K. Shaukat, N. U. Khan, H. Zhu, and L. Wang, "A novel ensemble wasserstein gan framework for effective anomaly detection in industrial internet of things environments," *Scientific Reports*, vol. 15, no. 1, p. 26786, 2025.

[13] Y. Abudurexiti, G. Han, F. Zhang, and L. Liu, "An explainable unsupervised anomaly detection framework for industrial internet of things," *Computers & Security*, vol. 148, p. 104130, 2025.

[14] M. Rodriguez, D. P. Tobon, and D. Munera, "A framework for anomaly classification in industrial internet of things systems," *Internet of Things*, vol. 29, p. 101446, 2025.

[15] A. Kantaros, T. Ganetsos, E. Pallis, and M. Papoutsidakis, "From mathematical modeling and simulation to digital twins: Bridging theory and digital realities in industry and emerging technologies," *Applied Sciences*, vol. 15, no. 16, p. 9213, 2025.

[16] J. Chen, Y. Xu, H. Li, X. Zhao, Y. Su, C. Qi, K. Qu, and Z. Cui, "The application of digital twin technology in the development of intelligent aquaculture: Status and opportunities," *Fishes*, vol. 10, no. 8, p. 363, 2025.

[17] T. Li, Q. Long, H. Chai, S. Zhang, F. Jiang, H. Liu, W. Huang, D. Jin, and Y. Li, "Generative ai empowered network digital twins: Architecture, technologies, and applications," *ACM Computing Surveys*, vol. 57, no. 6, pp. 1–43, 2025.