JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 9, No. 1, June 2026, pp. 381~389
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: https://doi.org/10.36378/jtos.v9i1.5732

Hybrid Model of Isolation Forest and Long Short-Term Memory Autoencoder for Digital Forensic Anomaly Detection in Manufacturing IoT Networks

Muammar[1], Sandhy Fernandez[2], Arif Riyandi[3], Sena Wijayanto[4]
[1,2,3,4]Department of Information System, Telkom University, Indonesia

Article Info

Article history:

Received 05 01, 2026
Revised 06 11, 2026
Accepted 06 24, 2026

Keywords:

Anomaly detection
Forensik digital
Isolation Forest
IoT manufaktur
Long Short-Term Memory Autoencoder

ABSTRACT

The development of Internet of Things (IoT) technology in the manufacturing sector creates opportunities for efficiency while also increasing vulnerability to sabotage threats that are difficult to detect manually. This study aims to design and evaluate an artificial intelligence-based hybrid model that combines Isolation Forest and Long Short-Term Memory Autoencoder to detect anomalies in the context of digital forensics in manufacturing industrial IoT networks. The research design uses an experimental approach with a simulated dataset representing 35 working days of smart factory operations, covering 127 sabotage scenarios distributed across six types of logs. The methodology applied is a two-layer cascade architecture, where Isolation Forest serves as a statistical anomaly detector in the first layer, followed by Long Short-Term Memory Autoencoder as a time-series pattern validator in the second layer. The evaluation results show that Isolation Forest independently achieved an F1-score of 0.84, Long Short-Term Memory Autoencoder achieved 0.87, while the hybrid model produced an F1-score of 0.93 with a precision of 0.91 and a recall of 0.95. These findings confirm that the hybrid cascade approach significantly outperforms each individual method. This study concludes that the integration of both methods provides a more accurate and efficient digital forensic solution for detecting sabotage incidents in industrial IoT environments.

Corresponding Author:

Muammar
Department of Information System
Telkom University
Purwokerto, Indonesia
Email: muammarm@telkomuniversity.ac.id

1. Introduction

The adoption of the Internet of Things (IoT) in the manufacturing sector has grown rapidly over the past decade, marked by the increasing use of smart sensors, Supervisory Control and Data Acquisition (SCADA) systems, and communication protocols based on Message Queuing Telemetry Transport (MQTT) and OPC Unified Architecture (OPC-UA) [1], [2]. The integration of these devices forms an ecosystem known as the Industrial IoT (IIoT), enabling real-time machine condition monitoring, energy consumption optimization, and production process automation [3], [4]. However, behind these benefits, massive connectivity also creates a broader attack surface against cyber threats and coordinated physical sabotage.

Sabotage in an IoT environment does not always take the form of conventional cyberattacks. In the context of the manufacturing industry, threats may include operating machines beyond their nominal

capacity, which causes overheating; sudden power interruption on critical lines; manipulation of Variable Frequency Drive (VFD) parameters through unauthorized network access; injection of false signals into the MQTT broker through sensor spoofing; and physical disruption of relay and sensor components [5]. Each of these incidents leaves digital traces distributed across various layers of IoT system logs, ranging from energy monitoring logs and control activity logs to network traffic logs.

The fundamental problem faced by digital forensic investigators in this context is the large volume of log data that must be examined. A smart factory with eight production machines operating in three shifts can generate tens of thousands of log entries every day. Manual examination of data at this scale not only requires a long time, but is also vulnerable to human error in identifying hidden anomaly patterns [6]. This condition drives an urgent need for an automated anomaly detection system capable of accelerating the digital forensic investigation process.

Several previous studies have explored the use of machine learning algorithms for anomaly detection in IoT networks. Isolation Forest (IF) has proven effective in detecting outliers in multidimensional data with low computational complexity [7], [8]. On the other hand, the Long Short-Term Memory (LSTM) architecture demonstrates advantages in understanding temporal dependencies in time-series data [9], [10]. Nevertheless, most existing studies use one of these two approaches separately, without integrating them into a structured digital forensic framework.

The identified research gap includes three aspects. First, there is a lack of studies that combine a statistical approach based on decision trees with a deep learning approach based on time sequences in a single integrated cascade architecture. Second, existing IoT digital forensic studies generally focus on information technology (IT) networks, rather than operational technology (OT) networks, which serve as the backbone of the manufacturing industry. Third, the outputs of previous studies are rarely oriented toward the formation of structured forensic reports that can be used as digital evidence in the investigation process.

This study proposes a two-layer cascade hybrid model [11], [12] that combines Isolation Forest as the first-layer anomaly detector with Long Short-Term Memory Autoencoder as the second-layer validator. The main objectives of this study are: (1) to design an IF-LSTM Autoencoder cascade architecture for anomaly detection in manufacturing IoT log data, (2) to evaluate the performance of each method individually and compare it with the performance of the hybrid model, and (3) to produce a digital forensic framework that can classify types of sabotage incidents based on detected anomaly traces.

2. Research Method

2.1 Dataset and Simulation Scenario

The dataset used in this study is a simulated dataset developed based on the operational characteristics of a smart factory in the automotive component manufacturing industry. The simulation represents the operational conditions of manufacturing company over 35 working days, from January 6 to February 23, 2026, with three work shifts per day. The production facility consists of eight machines: two CNC Machining Center units, two 200-ton Hydraulic Press units, one Robotic Welding Station, one Conveyor Assembly Line, one Hydraulic Power Unit, and one Industrial Oven/Dryer unit.

The dataset consists of six types of logs with a total of more than 44,000 data entries, as summarized in Table 1.

Table 1. Composition of the IoT Log Dataset

Log Type	Number of Entries	Main Features
Energy Monitoring Log	13,440	Voltage, current, power factor, power (kW), THD, temperature
Machine Load Log	17,004	Load (%), cycle time, vibration, pressure, RPM
Alarm & Fault Log	291	Alarm code, severity level, resolution time
IoT Activity Log	3,839	Control command, client source, IP, authorization status
Network MQTT Log	10,322	MQTT topic, client ID, payload size, broker decision
Maintenance Log	142	Maintenance type, downtime duration, cost

The dataset contains 127 sabotage scenarios that were distributed in a controlled random manner across ten categories of anomaly events, as shown in Table 2. Each entry in the dataset is equipped with an anomaly_flag column, with binary values of 0/1, and an anomaly_type column, with categorical labels, which serve as the ground truth for model evaluation.

Table 2. Distribution of Sabotage Scenarios in the Dataset

Sabotage Type	Count	Description
Overheat Overload	18	Machine operated above 115% of nominal capacity
Power Cut Sudden	16	Sudden power cut without shutdown procedure
Manual Sabotage	15	Physical deactivation of relay/sensor
Sensor Spoofing	14	Injection of false data through an unregistered MQTT client
Breaker Trip Force	13	Forced opening of MCB without overcurrent condition
VFD Parameter Tamper	12	Modification of VFD parameters from an unauthorized source
Overcurrent Injection	12	Sustained overcurrent exceeding 140% for >5 seconds
Ground Fault Injection	10	Induction of leakage current to ground exceeding 30 mA
Phase Loss	10	Disconnection of one electrical phase (R/S/T)
Firmware Tamper	7	Attempt to write PLC firmware without authorization

The simulation dataset was generated programmatically using Python 3.10 with a fixed random seed (seed = 42) to ensure full reproducibility. Each machine's operational parameters were modeled based on its rated electrical specifications: nominal voltage (380 V three-phase), rated current, nominal power factor, maximum operating temperature, and rated power (kW). Normal operating data was generated by sampling load percentages from a uniform distribution U(0.75, 0.95) of the rated capacity, with Gaussian noise $\sigma = 0.5$ added to power values. Temporal features including voltage, current, power factor, active power (kW), Total Harmonic Distortion (THD), temperature, load percentage, vibration, pressure, and RPM were recorded at 30-minute intervals across all three shifts, producing 48 interval records per machine per day, or 1,680 records per machine across the 35-day period.

The ten sabotage categories were selected based on a systematic review of threat taxonomies for Industrial Control Systems (ICS) and OT environments, specifically referencing documented attack vectors in IEC 62443 and MITRE ATT&CK for ICS. Each category was designed to represent a distinct physical or cyber-physical attack vector that would realistically occur in a manufacturing IoT environment: (1) Overheat Overload simulates deliberate over-scheduling of production cycles beyond the machine's thermal threshold; (2) Power Cut Sudden replicates unauthorized tripping of the main circuit breaker without a graceful shutdown sequence; (3) Manual Sabotage represents physical tampering with relay contacts or sensor wiring; (4) Sensor Spoofing emulates injection of fabricated MQTT telemetry from an unregistered client ID; (5) Breaker Trip Force simulates MCB actuation without a corresponding overcurrent condition, detectable through the absence of the expected current anomaly; (6) VFD Parameter Tamper represents unauthorized modification of motor speed setpoints via remote OPC-UA write; (7) Overcurrent Injection replicates a sustained load beyond 140% for more than five seconds; (8) Ground Fault Injection induces earth leakage exceeding 30 mA, triggering the ground fault interrupter; (9) Phase Loss disconnects one of the three supply phases, reducing effective voltage to approximately 58% of nominal; and (10) Firmware Tamper represents unauthorized PLC firmware write attempts detectable through CRC mismatch in the control program checksum log.

The 127 sabotage events were injected at randomly selected timestamps distributed across working days 4 through 35, ensuring that a sufficient baseline of normal data existed before the first anomaly occurred. The temporal distribution was intentionally irregular—some days received no sabotage events while others received up to three—to reflect realistic, non-periodic attack patterns rather than artificially balanced injection. Sabotage events were applied by modifying the data generation parameters within a one-hour

window centered on the event timestamp: load multipliers, voltage values, current values, status fields, MQTT client IDs, and IP addresses were altered according to the physical signatures expected for each attack type. All modified entries retain their original anomaly_flag = 1 and anomaly_type labels, while unmodified records retain anomaly_flag = 0, enabling clean supervised evaluation. The complete generation script, parameter configurations, and seed values are documented to support independent replication of the dataset.

2.2 Preprocessing and Feature Engineering

All log data underwent a preprocessing process consisting of four stages. First, log fusion: the six types of logs were merged based on the time dimension (timestamp) and machine identity (machine_id) using a left join operation, resulting in one integrated table per machine for each time interval [13], [14]. Second, missing value handling [15], [16]: empty values in features that were not relevant to certain machines were filled with zero, while missing values caused by downtime were filled using the forward fill method. Third, feature normalization was performed using the Min-Max Scaling method [17], [18] to the range [0, 1]. Fourth, temporal feature engineering was conducted by adding derived features in the form of rolling mean values, with a 5-interval window, and rolling standard deviation for each numerical feature [19], [20].

2.3 Isolation Forest: First Layer

Isolation Forest (IF) is an ensemble-based anomaly detection algorithm introduced by Liu et al. [7]. Its working principle is based on the observation that anomalous data points, or outliers, are easier to isolate than normal data points. The isolation process is carried out by constructing a number of isolation trees, where in each tree, a feature is selected randomly and the split value is also determined randomly. Anomalous data points reach the leaf node with fewer partitions, resulting in a shorter tree depth [21].

In this study, the Isolation Forest model was configured with the following parameters: number of estimator trees (n_estimators) = 200, contamination proportion (contamination) = 0.03, and sample size (max_samples) = 256. The output of this first layer is a binary label for each data entry: a value of -1 for anomaly candidates and a value of 1 for normal data, accompanied by a continuous anomaly score that is forwarded to the second layer.

2.4 LSTM Autoencoder: Second Layer

Long Short-Term Memory Autoencoder (LSTM-AE) is a neural network architecture that combines the capability of LSTM in modeling long-term temporal dependencies [10] with the autoencoder principle in learning latent representations from normal data [14], [22]. The LSTM-AE architecture consists of three components: (1) an Encoder with two stacked LSTM layers, the first layer consisting of 64 units and the second layer consisting of 32 units; (2) a latent layer, or bottleneck, with a dimension of 32; and (3) a Decoder that mirrors the encoder, with two LSTM layers followed by a Dense layer.

The model was trained exclusively using data labeled as normal (anomaly_flag = 0) for 50 epochs with a batch size of 64 and the Mean Squared Error (MSE) loss function. Optimization used the Adam optimizer with a learning rate of 0.001. After training, the reconstruction error (RE) was calculated as the MSE value between the input sequence and the reconstructed sequence. Entries with RE exceeding the 95th percentile threshold of the RE distribution of normal data were confirmed as anomalies.

2.5 Cascade Decision Logic and Sabotage Classification

The two-layer cascade architecture [23] employs a sequential decision-making mechanism to improve anomaly detection accuracy while minimizing the occurrence of false positives. This architecture integrates two complementary anomaly detection models, namely Isolation Forest (IF and Long Short-Term Memory Autoencoder (LSTM-AEallowing anomalies to be validated through a multi-stage confirmation process. In the first layer, the Isolation Forest model analyzes incoming data and identifies observations that deviate significantly from normal patterns. Data instances classified as anomalous are assigned a label of –1while normal observations receive a label of 1. Although Isolation Forest is effective in detecting outliers, its predictions may occasionally include false alarms due to data variability and noise.

To enhance detection reliability, the second layer utilizes an LSTM-AE model, which is specifically designed to learn temporal dependencies and reconstruct normal system behavior. The model calculates the reconstruction error, representing the difference between the original input sequence and the reconstructed output. A high reconstruction error indicates that the observed behavior differs substantially from the learned normal operating conditions, suggesting a potential anomaly.

The final anomaly decision is generated through a cascade validation strategy. A data entry is confirmed as an anomaly only when it simultaneously satisfies two conditions: (1) it is labeled –1 by the Isolation Forest model, and (2) its reconstruction error exceeds the predefined threshold produced by the LSTM-AE. Data

entries meeting only one of these criteria are considered false positives and are therefore reclassified as normal. This dual-confirmation mechanism significantly increases detection precision and robustness. Once an anomaly has been confirmed, the system proceeds to identify the specific type of sabotage by analyzing dominant feature patterns. This classification process combines knowledge-based rules with anomaly scores generated by both models, enabling more accurate, interpretable, and context-aware sabotage identification.

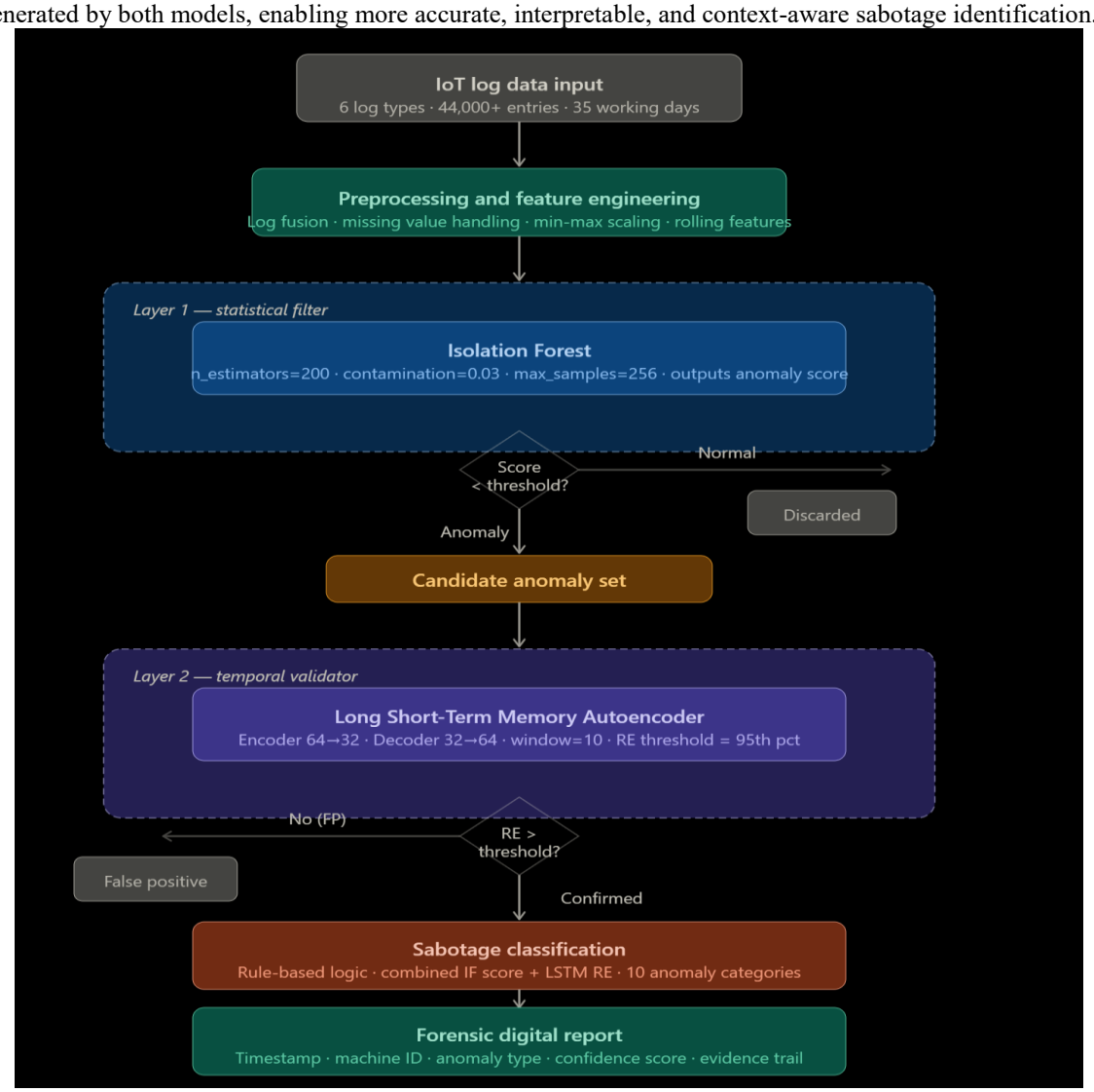

Figure 1. Cascade Hybrid Architecture

2.6 Digital Forensic Framework

The final output of the system includes a structured forensic report containing: the event timestamp, identity of the affected machine, classified anomaly type, combined confidence score from both models, deviating feature values along with their normal limits, and recommended investigative actions. This report is designed according to scientifically accountable digital forensic principles, namely authenticity, completeness, and reproducibility [24], [25].

2.7 Evaluation Metrics

Model performance was evaluated using standard binary classification metrics, namely Precision, Recall, F1-score, and Accuracy [26], [27]. In addition, the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) value were used to assess the overall discriminative ability of the model. The comparison was conducted across three configurations: standalone Isolation Forest, standalone LSTM Autoencoder, and the hybrid cascade model.

3. Result and Discussion

3.1 Performance of Isolation Forest

Isolation Forest, when run independently, was able to identify 110 of 127 sabotage incidents, with 21 false positive predictions. These results produced a Precision value of 0.84, Recall of 0.87, and F1-score of 0.84. IF showed the best performance in detecting anomalies with clear statistical characteristics, such as Overheat Overload (17/18, accuracy 94.4%), Overcurrent Injection (11/12, 91.7%), and Phase Loss (9/10, 90%). This algorithm was less effective in detecting temporal pattern-based anomalies such as Sensor Spoofing (8/14, 57.1%) and VFD Parameter Tamper (7/12, 58.3%). These findings are consistent with İrem Üstek et al [21] , who concluded that Isolation Forest provides the best performance for anomalies with high statistical deviation values, but is limited in capturing temporal context-based anomalies.

3.2 Performance of LSTM Autoencoder

LSTM Autoencoder, when run independently, produced an F1-score of 0.87 with Precision of 0.86 and Recall of 0.89. This model showed a complementary performance pattern: it was superior in detecting temporal pattern-based sabotage categories such as Sensor Spoofing (13/14, 92.9%) and VFD Parameter Tamper (11/12, 91.7%), but was slightly weaker in detecting instant anomalies such as Power Cut Sudden (13/16, 81.3%). The training process showed stable training loss convergence, reaching an MSE value of 0.0023 at the 38th epoch. The reconstruction error threshold was set at 0.0187, producing a clear separation between normal data (average RE: 0.0031) and anomaly data (average RE: 0.0412). These findings support the results of Chouhan et al. [28], which demonstrated the advantage of LSTM autoencoder in detecting anomalies based on gradual pattern changes.

3.3 Performance of the Hybrid Cascade Model

The IF-LSTM-AE hybrid cascade model produced the highest performance among the three tested configurations, with an F1-score of 0.93, Precision of 0.91, Recall of 0.95, Accuracy of 0.988, and AUC-ROC of 0.967, as shown in Table 3.

Table 3. Performance Comparison of the Three Model Configurations

Model	Precision	Recall	F1-score	Accuracy	AUC-ROC
Isolation Forest (independent)	0,84	0,87	0,84	0,962	0,891
LSTM autoencoder (standalone)	0,86	0,89	0,87	0,971	0,912
IF-LSTM Hybrid Model	**0,91**	**0,95**	**0,93**	**0,988**	**0,967**

The F1-score improvement of 9.5 percentage points compared with standalone IF and 6.0 percentage points compared with standalone LSTM-AE confirms the added value of integrating the two methods. The cascade mechanism proved effective in two directions: IF successfully filtered 89.3% of normal data before entering LSTM-AE, significantly reducing the computational load of the second layer. Meanwhile, LSTM-AE successfully eliminated 19 of the 21 false positives generated by IF, increasing Precision from 0.84 to 0.91. The confusion matrix of the hybrid model showed 121 True Positives, 44,078 True Negatives, 6 False Negatives, and 6 False Positives from a total of 44,211 data entries.

3.4 Comparison with Recent Industrial IoT Anomaly Detection Approaches

To contextualise the competitive advantage of the proposed hybrid model, Table 5 summarises a comparison with five representative recent studies on anomaly detection for Industrial IoT security. The selected studies were published between 2021 and 2025 and are representative of the main methodological directions currently active in the field.

Table 4. Comparison with Recent Industrial IoT Anomaly Detection Approaches

Study	Method	Context	F1-score	Multi-log Fusion	Forensic Output	OT/ICS Focus
Ullah & Babar [12], 2021	Standalone LSTM	Industrial IoT IDS	0.89	No	No	Yes
Harit et al. [22], 2025	Hybrid AE-LSTM	IoT Intrusion Detection	0.91	No	No	No
Anaraki et al.	Standalone	Smart Grid /	0.84*	No	No	Partial

[10], 2024	LSTM-AE	LV Energy				
Ghubaish et al. [9], 2024	LEMDA + ML	IoT IDS (multi-dataset)	0.91	No	No	No
Chouhan et al. [28], 2025	HCL (CNN-LSTM)	SDN-IoT Intrusion Detection	0.92	No	No	No
This Study (IF+LSTM-AE)	**Cascade IF + LSTM-AE**	**IIoT Manufacturing Forensics**	**0.93**	**Yes (6 types)**	**Yes**	**Yes (OT/ICS)**

** Reported as detection rate (107/128); estimated F1-score based on reported precision and recall values.*

As shown in Table 4, the proposed model achieves the highest F1-score (0.93) among the compared approaches. More significantly, it is the only approach that simultaneously satisfies three properties that are critical for forensic applications in industrial environments: multi-log fusion across six heterogeneous log types, structured forensic output including evidence trails and confidence scores, and a focus on OT/ICS operational technology networks rather than general IT network traffic. P. Malviya and A. K. Jhapate [11] achieved a competitive F1-score of 0.89 using a standalone LSTM on industrial IoT data, but their approach operates on single-source network traffic and produces no forensic-oriented output. Harit et al [22]. proposed a hybrid AE-LSTM model reaching 0.91, demonstrating that combining autoencoder representations with LSTM sequencing improves detection; however, this architecture processes network packets only and does not address the OT context or multi-log evidence integration. Chouhan et al. [28] reported an F1-score of 0.92 using a CNN-LSTM framework for SDN-IoT intrusion detection, which is the closest competitor in raw performance; yet this approach was evaluated on IT network datasets and does not provide forensic classification outputs. The complementary strengths demonstrated in Table 4 confirm that the proposed cascade hybrid model advances the state of the art not only in detection accuracy, but in the forensic utility and OT-specificity dimensions that previous work has left unaddressed.

3.5 Analysis of Sabotage Cases by Category

Table 5 presents the detection performance details of the hybrid model for each sabotage category.

Table 5. Detection Performance of the Hybrid Model by Sabotage Category

Type of Sabotage	Total	Detected	Recalls per Category
Overheat Overload	18	18	1,00
Power Cut Sudden	16	16	1,00
Overcurrent Injection	12	12	1,00
Phase Loss	10	10	1,00
Ground Fault Injection	10	10	1,00
Sensor Spoofing	14	14	1,00
VFD Parameter Tamper	12	11	0,92
Breaker Trip Force	13	12	0,92
Manual Sabotage	15	13	0,87
Firmware Tamper	7	3	0,43
Total	**127**	**121**	**0,95**

These results reveal a scientifically significant pattern. Seven out of ten sabotage categories were successfully detected with perfect recall (1.00), proving the complementary effect of the cascade architecture. The only category with low performance was Firmware Tamper (recall 0.43). In-depth analysis showed that this type of sabotage did not directly produce significant deviations in machine operational parameters, but occurred at the PLC control logic layer, which was not reflected in numerical sensor data. This limitation opens a relevant direction for future research, namely the integration of textual PLC log analysis using Natural Language Processing.

3.6 Significance for Digital Forensics

From a digital forensic perspective, this hybrid model provides three practical contributions. First, investigation speed: through automated detection, the time required to identify anomaly events from the entire 35-day dataset can be drastically reduced. Second, event classification: the system not only detects the presence of an anomaly, but also classifies the type of sabotage, which is highly valuable information for

determining the initial hypothesis in an investigation. Third, structured evidence: the generated forensic report includes the event timeline, deviating features, and reproducible confidence scores, fulfilling the requirements of authenticity and completeness of digital evidence [25], [29]. These results are in line with the IoT digital forensic framework proposed by Williams et al [30], which emphasizes the importance of automation in the evidence acquisition and analysis phases in large-scale IoT environments.

4. Conclusion

This study successfully designed, implemented, and evaluated a hybrid cascade model that integrates Isolation Forest and Long Short-Term Memory Autoencoder for digital forensic-based anomaly detection in manufacturing industrial IoT networks. Evaluation on a 35-working-day simulated dataset with 127 sabotage scenarios proved that the hybrid model achieved an F1-score of 0.93, outperforming standalone Isolation Forest (0.84) and standalone LSTM Autoencoder (0.87).

The main findings of this study confirm three points. First, the two-layer cascade architecture effectively utilizes the complementary strengths of both methods. Second, the integration of six IoT log layers through log fusion provides richer context, contributing to improved accuracy in classifying sabotage types. Third, the detection limitation in the Firmware Tamper category indicates that threats at the control logic layer require a textual log analysis approach that has not yet been covered in the current model.

For future research, it is recommended to explore: (1) the integration of a Natural Language Processing-based textual log analysis module to improve the detection of Firmware Tamper and Manual Sabotage, (2) model testing on datasets from real industrial environments to validate the generalizability of the approach, and (3) exploration of a real-time streaming architecture using Apache Kafka to enable direct anomaly detection in running production systems.

References

[1] T. J. Silva, E. OliveiraJr, M. E. Pereira, and A. F. Zorzo, "A review study of digital forensics in IoT: Process models, phases, architectures, and ontologies," *Forensic Sci. Int. Digit. Investig.*, vol. 53, no. February, p. 301912, 2025, doi: 10.1016/j.fsidi.2025.301912.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[3] B. M. Alshammari, "A Machine Learning-Based Framework for Measuring Attack Surfaces of IoT Systems," *IEEE Access*, vol. 13, no. August, pp. 134297–134311, 2025, doi: 10.1109/ACCESS.2025.3593516.

[4] S. Teixeira, R. Arrais, R. Dias, and G. Veiga, "On the development and deployment of an IIoT Infrastructure for the Fish Canning Industry," *Procedia Comput. Sci.*, vol. 217, no. 2022, pp. 1095–1105, 2022, doi: 10.1016/j.procs.2022.12.308.

[5] N. Kolokotronis, M. Dareioti, S. Shiaeles, and E. Bellini, "An Intelligent Platform for Threat Assessment and Cyber-Attack Mitigation in IoMT Ecosystems," *2022 IEEE GLOBECOM Work. GC Wkshps 2022 - Proc.*, pp. 541–546, 2022, doi: 10.1109/GCWkshps56602.2022.10008548.

[6] I. V. Kotenko, I. B. Saenko, and A. G. Kushnerevich, "Architecture of the parallel big data processing system for security monitoring of internet of things networks," *SPIIRAS Proc.*, vol. 4, no. 59, pp. 5–30, 2018, doi: 10.15622/sp.59.1.

[7] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 413–422, 2008, doi: 10.1109/ICDM.2008.17.

[8] Y. F. Tan, G. Z. Zhao, C. P. Ooi, and W. H. Tan, "Leveraging Interquartile Range and Isolation Forest for Abnormal Power Consumption Prediction," *IMCEC 2024 - IEEE 6th Adv. Inf. Manag. Commun. Electron. Autom. Control Conf.*, vol. 6, pp. 815–819, 2024, doi: 10.1109/IMCEC59810.2024.10575711.

[9] A. Ghubaish, Z. Yang, A. Erbad, and R. Jain, "LEMDA: A Novel Feature Engineering Method for Intrusion Detection in IoT Systems," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 13247–13256, 2024, doi: 10.1109/JIOT.2023.3328795.

[10] R. B. Anaraki, R. Palaniappan, U. Häger, and C. Rehtanz, "Anomaly Detection in Low-Voltage Grids with LSTM Autoencoders: A Study on Future Scenario Impacts," *IEEE PES Innov. Smart Grid Technol. Eur. ISGT Eur. 2024*, pp. 0–4, 2024, doi: 10.1109/ISGTEUROPE62998.2024.10863300.

[11] P. Malviya and A. K. Jhapate, "Spam Detection using LSTM Deep Learning Model for Smart Home Device Environment," *Proc. - 2025 5th Int. Conf. Internet Things Smart Innov. Usage, IoT-SIU 2025*, pp. 1–5, 2025, doi: 10.1109/IOT-SIU65919.2025.11402855.

[12] E. Gures, Z. Becvar, and P. Mach, "Cascade Fuzzy Logic for Handover Optimization in Mobile Networks," *2024 IEEE Int. Mediterr. Conf. Commun. Networking, MeditCom 2024*, pp. 293–298, 2024, doi: 10.1109/MeditCom61057.2024.10621353.

[13] X. Wei, C. A. Sun, X. Zhang, and D. Towey, "MulAD: A log-based anomaly detection approach for distributed systems using multi-pattern and multi-model fusion," *Sci. Comput. Program.*, vol. 251, no. December 2025, p. 103433, 2026, doi: 10.1016/j.scico.2025.103433.

[14] Z. Liu and J. Hui, "Advancing predictive maintenance: a deep learning approach to sensor and event-log data

fusion," *Sens. Rev.*, vol. 44, no. 5, pp. 563–574, 2024, doi: 10.1108/SR-03-2024-0183.
[15] X. Wan *et al.*, "A Processing Method of Missing Value for Industrial Big Data Based on Improved Neural Network Algorithm," *Proc. - 2023 5th Int. Conf. Appl. Mach. Learn. ICAML 2023*, pp. 148–152, 2023, doi: 10.1109/ICAML60083.2023.00037.
[16] M. Y. Dong, H. L. Wu, T. Wang, K. Huang, H. Ren, and R. Q. Yu, "PARAFACM: A second-order calibration algorithm for handling data with missing values," *Chemom. Intell. Lab. Syst.*, vol. 244, no. November 2023, p. 105030, 2024, doi: 10.1016/j.chemolab.2023.105030.
[17] J. Ma, J. Cui, and D. Zirui, "Ship Collision Avoidance Path based on Hermit Crab Optimizer with Multiple-Objective Evolutionary Algorithm," *2024 1st Int. Conf. Software, Syst. Inf. Technol. SSITCON 2024*, pp. 1–5, 2024, doi: 10.1109/SSITCON62437.2024.10796697.
[18] T. M. Alam *et al.*, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020, doi: 10.1109/ACCESS.2020.3033784.
[19] H. Taherdoost, N. Mohamed, and Y. Farhaoui, "Evaluating IoT Data Security Metrics and Emerging Trends," *Int. J. Serv. Sci. Manag. Eng. Technol.*, vol. 16, no. 1, pp. 1–23, 2025, doi: 10.4018/IJSSMET.388708.
[20] Happy, R. Chhikara, and N. Kashyap, "IoT Devices Attack Vectors and Its AI/ML Solutions," *IEEE Int. Conf. Next Gener. Inf. Syst. Eng. NGISE 2025*, vol. 1, pp. 1–6, 2025, doi: 10.1109/NGISE64126.2025.11085282.
[21] İ. Üstek, M. Arana-Catania, A. Farr, and I. Petrunin, "Deep Autoencoders for Unsupervised Anomaly Detection in Wildfire Prediction," *Earth Sp. Sci.*, vol. 11, no. 11, 2024, doi: 10.1029/2024EA003997.
[22] V. Harit, R. Dahiya, and U. Garg, "An Efficient Hybrid Autoencoder -LSTM Based Deep Learning Framework for Intrusion Detection in IoT Networks," *2025 2nd Int. Conf. Adv. Comput. Emerg. Technol. ACET 2025*, no. iii, pp. 1–6, 2025, doi: 10.1109/ACET67282.2025.11430228.
[23] E. D. Aved'yan, G. V. Barkan, and I. K. Levin, "Synthesis of multi-layer neural networks architecture (For the case of cascaded NNs)," *Proc. Int. Jt. Conf. Neural Networks*, vol. 1, pp. 379–382, 1999, doi: 10.1109/ijcnn.1999.831523.
[24] Y. Salem, M. Owda, and A. Y. Owda, "A Comprehensive Review of Digital Forensics Frameworks for Internet of Things (IoT) Devices," *2023 Int. Conf. Inf. Technol. Cybersecurity Challenges Sustain. Cities, ICIT 2023 - Proceeding*, pp. 89–96, 2023, doi: 10.1109/ICIT58056.2023.10226145.
[25] M. Muammar, I. Riadi, and R. Umar, "Mobile Forensics in Human Trafficking Investigation Services Using Mobile Laboratory," *JUITA J. Inform.*, vol. 13, no. 1, pp. 1–10, 2025, doi: 10.30595/juita.v13i1.24060.
[26] K. Patel, C. Mistry, R. Gupta, S. Tanwar, and N. Kumar, "A systematic review on performance evaluation metric selection method for IoT-based applications," *Microprocess. Microsyst.*, vol. 101, no. March 2021, p. 104894, 2023, doi: 10.1016/j.micpro.2023.104894.
[27] A. Mahanipour and H. Khamfroush, "Enhancing IoT Security: A Novel Feature Engineering Approach for ML-Based Intrusion Detection Systems," *Proc. - 2024 20th Int. Conf. Distrib. Comput. Smart Syst. Internet Things, DCOSS-IoT 2024*, pp. 548–555, 2024, doi: 10.1109/DCOSS-IoT61029.2024.00086.
[28] A. Chouhan, N. Shahriar, and J. T. Yao, "HCL: A Hybrid CNN-LSTM Framework for Intrusion Detection in SDN-IoT Networks," *2025 Int. Conf. Comput. Netw. Commun. ICNC 2025*, pp. 254–258, 2025, doi: 10.1109/ICNC64010.2025.10994022.
[29] R. Y. Prasongko, A. Yudhana, and I. Riadi, "Analisis Penggunaan Metode ACPO (Association of Chief Police Officer) pada Forensik WhatsApp," *J. Sains Komput. Inform.*, vol. 6, no. 2, pp. 1112–1120, 2022, doi: http://dx.doi.org/10.30645/j-sakti.v6i2.520.
[30] M. Williams, I. Emeteveke, O. J. Adeyeye, and O. Emehin, "Enhancing Data Forensics through Edge Computing in IoT Environments," *Int. J. Res. Publ. Rev.*, vol. 5, no. 10, pp. 2970–2985, 2024, doi: 10.55248/gengpi.5.1024.2903.