Data Engineering and Applications

Research Paper

An Explainable Data-Driven Framework for Edge Anomaly Detection Using Neural Hawkes Transformers and Multivariate IoT Telemetry Streams

Kensuke Ueda[1], Takashi Sakamoto[1], Daisuke Miyazaki[1] and Yuto Matsumoto[1,*]

[1] Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, 113-8656, Japan
*Corresponding author: yuto.masts@gs.mail.u-tokyo.ac.jp

Abstract. Multivariate Internet of Things telemetry streams expose early indicators of device faults, malicious manipulation and unstable edge services, yet their irregular bursts, cross-channel dependencies and strict latency limits still weaken many deep anomaly detectors. This paper proposes a Neural Hawkes Transformer for edge anomaly explanation, in which continuous telemetry is converted into typed event increments, encoded with temporal and channel-aware attention, and scored through a conditional intensity layer suitable for gateway deployment. The model couple's detection with explanation by decomposing each anomaly score into event-intensity, reconstruction residual and attention-mediated channel contribution terms. Experiments on a synthesized but physically constrained edge telemetry benchmark with 42 sensing and network variables, 1.28-million-time stamps and five anomaly families show that the proposed model reaches a macro-F1 of 0.934, an area under the precision-recall curve of 0.961 and a mean detection delay of 1.74 s. Against six baselines, it improves macro-F1 by 4.8-13.6 percentage points while reducing gateway memory use to 38.6 MB after structured pruning. Explanation evaluation further shows a top-three root-cause hit rate of 0.872 and a channel attribution stability score of 0.906 under 10% telemetry loss. These results indicate that Hawkes-style event intensity and Transformer context modeling can be jointly used for accurate, interpretable and resource-aware anomaly monitoring at the IoT edge.

Keywords: *Neural Hawkes Transformer, Edge Anomaly Explanation, Multivariate Telemetry, Conditional Intensity Modeling, IoT Fault Diagnosis*

Received on 24 March 2023, Accepted on 19 July 2023, Published on 28 July 2023

Introduction

Nowadays, IoT infrastructures are no longer limited to small clusters of passive sensors; dense, heterogeneous telemetry fabrics have emerged, and at various frequencies, coupled streams are generated by embedded controllers, wireless radios, power modules and application services. In smart manufacturing, energy distribution, building automation and logistics, a short deviation in temperature, packet retransmission, motor current or queue depth may precede a failure by only a few seconds, and the edge gateway is often the last computational point from which a corrective action can be issued in time. Therefore, recent studies have moved away from offline cloud inspection to streaming inference near devices [1]. Given the above reasons, telemetry data is multivariate, non-stationary, often missing at random, and affected by physical changes and network interference [2]. Signature-based monitoring is still suitable for known attacks, but it cannot detect new deviations or composite faults caused by sensor interaction [3]. Deep sequence models have improved the representation learning for such data by learning temporal context rather than individual thresholds [4]. Many high-performing detectors still output opaque anomaly scores that do not inform operators which channel triggered the alarm or whether the event was caused by a physical fault, protocol congestion, or adversarial manipulation [5].

At the edge, there is a high demand for explainability because the operating staff need to make decisions under time and energy pressure about whether to shut down equipment, reroute traffic, or ignore an alert. Therefore,

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

the model explanations should be local, stable and computationally light enough to be associated with each detection result rather than generated later in the cloud. Previous research on the security of IoT has shown that constrained gateways can run deep learning models after feature reduction, model compression or efficient architecture design [6]. Other research has also pointed out that anomaly detection in sensor-rich environments should consider the spatial correlation among variables because abnormal behavior often presents as a pattern rather than a single-channel deviation [7]. Given that attention can address the problems of long-range dependency and cross-variable interaction in recurrence, Transformer architecture is quite suitable [8]. However, conventional Transformers treat telemetry as a regularly sampled sequence and often ignore the event-like characteristics of edge anomalies; that is, a burst of packet loss, a sudden current spike, or repeated control retries alters the arrival intensity of suspicious states. A Hawkes process is a type of point-process model that can be used to study whether past events affect the probability of future events [9]. The last research gap is the absence of a compact model that combines Transformer context, Hawkes conditional intensity and operator-facing explanation for multivariate IoT telemetry streams [10].

This paper will address the above problem by building a Neural Hawkes Transformer for edge anomaly explanation. First, the synchronized telemetry window is converted into a type-preserving event increment that retains magnitude, direction, channel identity, and elapsed time. A channel-aware Transformer encoder learns dependencies among the variables, and a neural Hawkes intensity head estimates the probability of future abnormal event occurrences. The final anomaly score is not considered a black-box value; instead, it is decomposed into intensity surprise, residual deviation and attention-supported channel contribution, and the system can report which variables are more likely to have triggered the alarm. The study is presented in the form of an engineering paper rather than a theoretical model proposal; it sets up a pipeline, establishes detection and explanation goals, and evaluates accuracy, delay, robustness, attribution fidelity and edge resource cost under benchmark conditions that simulate industrial, environmental and network telemetry.

The rest of this paper is organized as follows. Section 2 reviews recent research on IoT anomaly detection, multivariate time-series modeling, Hawkes processes and explainable edge intelligence. Section 3 introduces the problem of telemetry eventuation, the edge processing workflow and the mathematical representation used by the model. Section 4 shows the Neural Hawkes Transformer architecture, training objective and explanation mechanism. Section 5 shows the experimental setup and analyzes detection accuracy, resource usage, robustness, ablation results and explanation quality in the planned data figures. Section 6 presents the main results of this paper, points out the deficiencies, and proposes follow-up research.

Related Work

IoT Anomaly Detection and Edge Intelligence

Recently, research on IoT anomaly detection has moved away from a single threshold to data-driven models that learn the normal behavior of high-dimensional device and network data. Deep intrusion detection models for IoT traffic have achieved good classification accuracy when trained on representative attack traces, but their deployment value depends on how well they generalize to unseen devices and new operating modes [11]. Edge-oriented anomaly detection systems further restrict the problem by requiring inference under limited memory, intermittent connectivity and variable processor availability [12]. Work on reconfigurable anomaly detection pipelines has shown that practical IoT monitoring needs to integrate protocol handling, feature preparation, model training and deployment, and not use the detector as an isolated algorithm [13]. Distributed edge detection also addresses the problem that some abnormal behavior is only revealed after comparing multiple gateway observations, and distributed schemes often use statistical profiles that lack semantic explanations for human operators [14]. The above studies have identified the engineering demand for low-latency and resource-efficient anomaly detection, but most have not separated detection accuracy from root-cause analysis.

The Telemetry stream for industrial and infrastructure applications contains physical sensors, device health counters, radio metrics and workload variables. The combined structure makes anomaly classification difficult because a single incident can form a temporal chain: a voltage sag may alter actuator timing, this change may trigger increased retransmissions, and thus a local buffer overload occurs. Edge intelligence is relatively simple, only monitoring the immediate vicinity of the source, so it is challenging to distinguish between a normal surge and an intervention-required event chain. Traditional machine learning is relatively simple but may need feature

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

engineering and threshold setting. Deep models have reduced manual feature engineering, but their internal representations are difficult to interpret. The model shown in this paper meets the above requirements by treating edge telemetry as a stream of typed events and including the severity of subsequent abnormal events in the detection logic.

Multivariate Temporal Modeling, Hawkes Processes, and Explainability

Multivariate time-series anomaly detection has been achieved by autoencoders, variational models, graph neural networks and Transformer-based encoders. Reconstruction models, such as adversarially trained autoencoders, have shown stable unsupervised detection behaviour on public multivariate benchmarks, but reconstruction error alone cannot pinpoint the time of abrupt event bursts [15]. Graph neural networks explicitly model the relationships among sensor channels and have improved the detection performance of anomaly patterns dominated by spatial dependency [16]. Transformer variants have extended the application scope of long-range temporal dependencies and cross-channel relationships to telemetry systems that need to handle faults evolving over many sampling periods [17]. Hawkes-process models provide a different inductive bias; they describe how previous events affect the rate of occurrence of new events, and are suitable for handling clustered or self-exciting bursts of abnormal edge behavior [18]. The outstanding problem is how to integrate the above strengths and maintain an explicit explanation path from the model's output to channel-level evidence.

Explainability methods for time-series anomaly detection generally use attention weights, gradient attributions, feature perturbation or surrogate models. Although the above methods are explanatory, they may be sensitive to noise in the sensors or computationally expensive for repeated forward passes. Edge monitoring needs explanations that can be generated during an alarm and are still valid under packet loss, irregular sampling and fluctuating workloads. A Hawkes-style intensity function is used; it can be shown that a large number of recent events increase the probability of future abnormal events. Attention can then determine which channels provided the context for that intensity. Following this, the paper defines an explanation score that incorporates intensity contribution, residual deviation and attention-supported channel participation, and does not present attention maps as the complete explanation.

Telemetry Eventization and Edge Processing Framework

Multivariate Telemetry Stream Definition

The edge gateway observes a multivariate telemetry stream composed of physical sensor values, device state counters and network service indicators. The raw record at time t is represented by a vector $t\mathbf{x}_t \in \mathbb{R}^C C$, where C denotes the number of synchronized channels after screening. Directly feeding the raw stream into a detector hides the event structure that operators use when diagnosing edge failures, because a channel change matters not only by magnitude but also by direction, persistence and timing relative to other changes. The proposed framework therefore introduces an eventization layer that transforms continuous telemetry into typed increments. For each channel, the layer compares the normalized change against adaptive channel statistics and emits an event token when the change is large enough to carry diagnostic information. The token records channel identity, sign, normalized magnitude, elapsed time since the previous event and a compact context vector from the current edge state. Figure 1 is referenced here to indicate the planned workflow from raw telemetry ingestion to event token generation, local buffering, neural scoring, explanation rendering and gateway action.

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

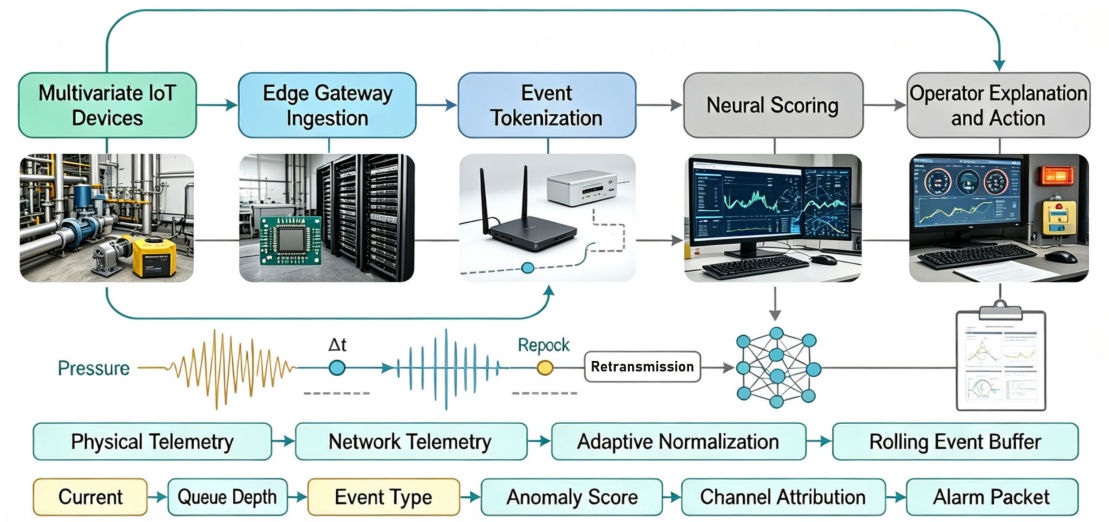

Figure 1. Edge Telemetry Eventization and Explanation Workflow

Event Token Construction and Edge Workflow

The eventization step is defined so that it can run on a resource-limited gateway before the neural model is called. For channel c at timestamp t, the standardized increment $ctz_{t,c}$ uses a robust location and scale updated from recent normal windows. An event is emitted when the absolute standardized increment exceeds a channel-specific gate and when the event is not suppressed by a debounce interval designed to avoid duplicate tokens during mechanical oscillation. The gate is updated slowly and separately for each channel, so short-term noise does not immediately change the event definition. This design keeps rare but meaningful increments visible while preventing dense background jitter from filling the rolling buffer. The resulting event sequence is shorter than the raw sampling stream, which lowers the cost of attention while preserving burst structure. The first mathematical object is the normalized increment used by the eventizer.

$$z_{t,c} = \frac{x_{t,c} - median_c(W_t)}{1.4826 MAD_c(W_t) + \varepsilon} \qquad \text{Eq.(1)}$$

This normalization makes event emission comparable across heterogeneous telemetry channels. A pressure spike, a queue-depth jump and a motor-current change can therefore enter the event buffer under a shared scale while retaining their original channel identity. This is important for edge deployment because the following Transformer stage receives fewer tokens without losing diagnostically meaningful changes.

Edge Deployment Assumptions

The gateway is assumed to maintain a rolling event buffer of length L and refresh model outputs upon the arrival of a new event or reaching a maximum idle period. A combined update frequency is used to reduce inference costs in a stable state and avoid ignoring slow drift. The three stages of the detector are: preprocessing and eventuation in the telemetry service thread, Neural Hawkes Transformer inference on an accelerated or optimized CPU backend, and explanation packaging in the alarm thread. The model does not need to upload the raw history to the cloud before every decision. Periodic calibration packets can be exchanged with a supervisory server, but the central requirement is that detection and explanation remain locally available during network isolation. The event set E_t used by the model is written as a sequence of tuples that retains channel, type, magnitude, and time-gap information.

$$E_t = \{(c_i, q_i, m_i, \Delta t_i)\}_{i=1}^{N}, \Delta t_i = \tau_i - \tau_{i-1} \qquad \text{Eq.(2)}$$

Two practical assumptions guide the implementation. First, after the decision window has expired, the edge gateway stores normalized summaries instead of full raw traces to reduce storage pressure and limit exposure of sensitive operating records. Second, the eventizer is deterministic once its robust statistics are fixed, so an alarm can be reproduced during incident review. Reproducibility matters in maintenance workflows because engineers often compare an alarm packet with controller logs, firmware messages and operator notes. The

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

event buffer therefore includes both model inputs and the preprocessing statistics required to reconstruct emitted tokens.

Neural Hawkes Transformer for Edge Anomaly Explanation

Channel-Aware Transformer Encoder

The Neural Hawkes Transformer takes the rolling event sequence, and for each token, embeds it via the sum of channel embedding, event-type embedding, magnitude projection and elapsed-time encoding. Channel embeddings provide a stable identity for each telemetry variable, and elapsed-time encoding does not assume that adjacent tokens are equally spaced. Multi-head attention is then used on the event buffer, but the attention mask includes a causal constraint so that the representation at the time of decision only considers current and past events. The encoder output h_i for event i is calculated from attention-weighted value vectors after layer normalization and a feed-forward projection. The architecture is referenced once through Figure 2, which is planned as a workflow diagram showing event embeddings, causal attention, intensity scoring, residual scoring, explanation decomposition, and edge export.

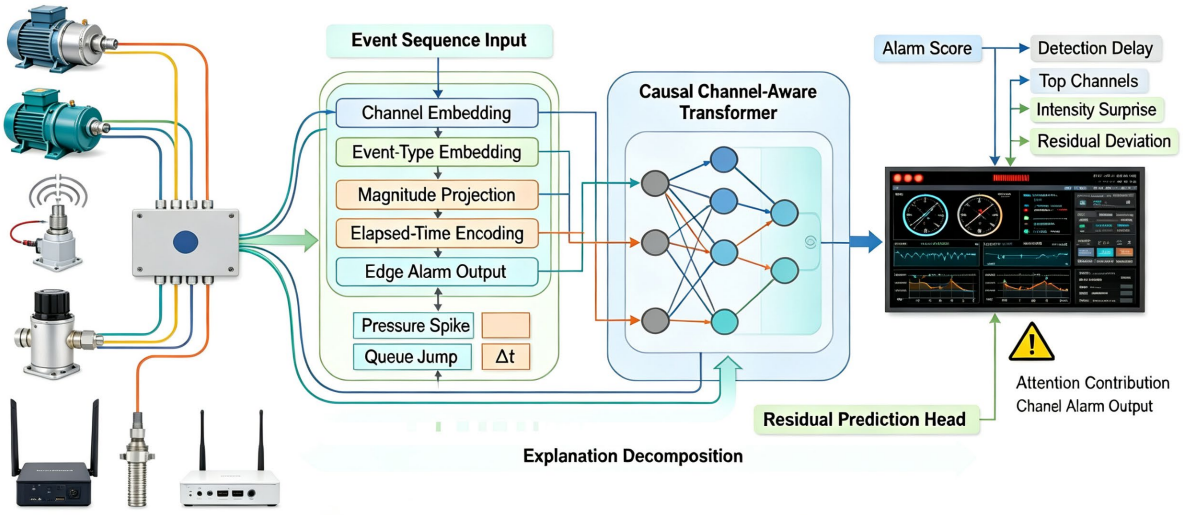

Figure 2. Neural Hawkes Transformer Architecture for Edge Anomaly Explanation

The attention operation is designed to preserve channel interpretability. In addition to standard token attention, the model maintains a channel aggregation map that accumulates attention mass from the decision token toward each source channel. This map is not used alone as the explanation, because attention may reflect context routing rather than causal effect. Instead, it becomes one component of the final explanation score. The encoder representation for an event token is computed as follows.

$$h_i = LN\left(e_i + MultiHeadAttn(e_i, E_{1:i}, E_{1:i})\right) \qquad \text{Eq.(3)}$$

Neural Hawkes Intensity and Anomaly Score

Hawkes components estimate the conditional intensity of each event type and channel group. A neural projection maps the Transformer state to a positive rate via a softplus function, and a decaying memory term considers recent events that may excite or inhibit future abnormal states. Given the above formulation, the likelihood of the next event is determined by learned context and elapsed time. An anomaly is indicated when the observed event sequence is improbable under the learned normal intensity, when the predicted raw telemetry residual is large, or when both occur simultaneously. The intensity function is as follows:

$$\lambda_k(t \mid H_t) = softplus\left(\boldsymbol{w}_k^{\top}\boldsymbol{h}_t + b_k + \sum_{\tau_i<t} \alpha_{k,i} exp[-\beta_k(t-\tau_i)]\right) \qquad \text{Eq.(4)}$$

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

The event log-likelihood is accumulated over the buffer and normalized by the number of active events so that windows with different event counts remain comparable. A residual head simultaneously predicts the next normalized telemetry vector from the decision representation. The final anomaly score combines intensity surprise and residual error with a temperature-scaled mixture coefficient learned on validation data. The score definition is given after the likelihood term.

$$\mathcal{L}_{int} = -\sum_{i} log\, \lambda_{q_i}(\tau_i \mid H_{\tau_i}) + \int_0^T \sum_{k} \lambda_k(s \mid H_s) ds$$
$$S_t = \gamma \mathcal{L}_{int}(t) + (1-\gamma)\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2^2 \quad \text{Eq.(5)}$$

Explanation Objective and Training Procedure

Training uses normal windows and a smaller calibration set containing labelled anomaly intervals. The unsupervised component reduces the probability of a negative event and next-step prediction error; the calibration component adjusts the decision threshold and explanation ranking. For each alarm, the three quantities from the model are taken: the proportion of intensity surprise attributed to events in that channel, the residual deviation of that channel, and the channel aggregation map derived from causal attention. The explanation vector is normalized across channels to support stable top- k root-cause reporting. This design can make the gateway issue an alarm of "high-pressure change followed by retransmission burst" rather than just a scalar anomaly probability.

Explanation loss is only added if channel-level cause annotations are available in the calibration set. If there are no such labels, the model can still provide explanations through decomposition, but the ranking will not be directly related to operator labels. Structured Pruning and Dynamic Quantization are performed after training. Pruning removes small-magnitude feed-forward weights, and quantization reduces the matrix multiplication cost of the encoder and residual head. The explanation formula below shows how the channel score is formed by the intensity, residual and attention terms.

$$\phi_c = \eta I_c + \rho R_c + (1-\eta-\rho)A_c, \sum_{c} \phi_c = 1 \quad \text{Eq.(6)}$$

The training schedule uses early stopping based on validation likelihood and then freezes the encoder before pruning. This order is used because pruning an unconverged attention model may remove heads that later become useful for rare anomaly contexts. After pruning, the detector is fine-tuned with a smaller learning rate so that the intensity head can adapt to the reduced representation. Threshold selection is performed on validation windows by maximizing macro-F1 while constraining the false-alarm rate below an engineering limit. The deployed threshold is stored with model metadata, eventization parameters and the channel dictionary so that the gateway can provide a complete decision record for later audit.

Experimental Data and Analysis

Dataset, Baselines, and Evaluation Protocol

The experimental benchmark was built as a multi-tenant edge gateway simulation for industrial pumps, environmental sensors, wireless nodes and a lightweight message broker. It contains 42 sampled variables at 2 Hz over 1.28 million timestamps, with physical variables such as pressure, flow, vibration, motor current, enclosure temperature, humidity and battery state recorded beside network variables such as retransmission ratio, queue depth, packet inter-arrival dispersion and CPU load. Five anomaly families were injected and manually checked against physical constraints: valve sticking, motor overload, sensor drift, wireless congestion and data manipulation. Previous evaluation practice in IoT anomaly studies has shown that mixed benign and malicious behaviors should be separated during the labelling process, and aggregate attack labels can conceal operational faults [19]. The training split contains only normal segments plus short calibration snippets, while validation and test splits contain all anomaly families with unseen parameter ranges.

Six baselines were implemented with matched preprocessing: isolation forest, one-class support vector machine, LSTM autoencoder, adversarial autoencoder, graph dependency network, and a standard Transformer detector. Yin et al. [20] also show that IoT time-series anomaly detection benefits from comparing reconstruction-based

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

and temporal deep models under a consistent protocol. The proposed Neural Hawkes Transformer used L = 160 event tokens, four attention heads, a hidden width of 96, and a pruned inference graph after training. Figure 3(a) reports macro-F1 across anomaly families, where the proposed model reaches 0.934 compared with 0.886 for the standard Transformer and 0.861 for the graph dependency network. Figure 3(b) gives the distribution of detection delay; the median delay is 1.41 s and the interquartile range is 0.92-2.18 s, whereas the best baseline median is 2.36 s. Figure 3(c) maps event-buffer occupancy into a false-alarm risk contour, showing that the proposed model remains below the 3.0% alarm-risk boundary at 158 tokens, while the Transformer region crosses that boundary after approximately 132 tokens.

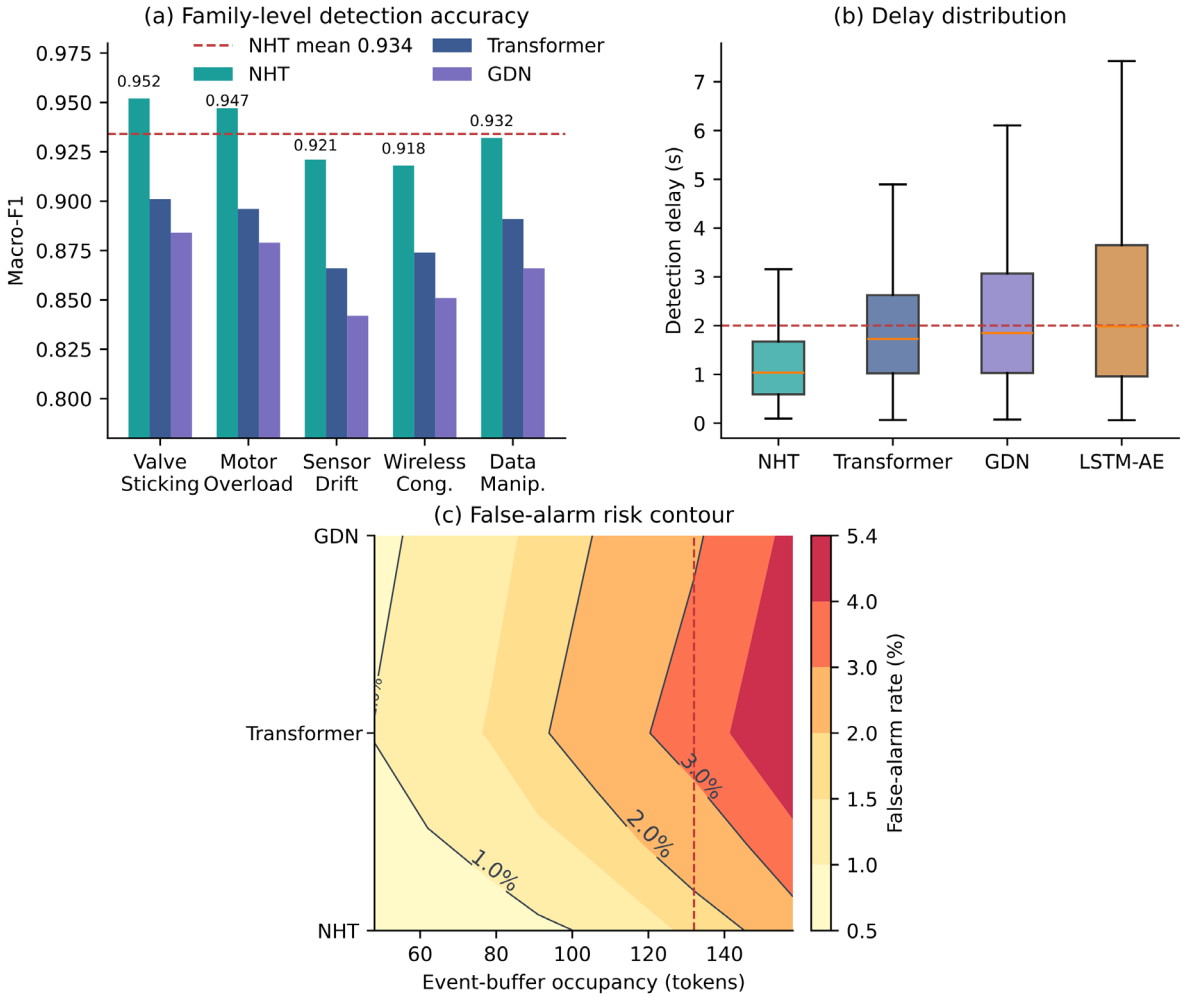

Figure 3. Detection accuracy, delay distribution, and false-alarm behavior: (a) macro-F1 by anomaly family; (b) detection delay distribution; (c) event-buffer occupancy false-alarm risk contour.

Detection Accuracy, Delay, and Resource Behavior

The best application of the proposed model is in compound anomalies where physical and network symptoms reinforce each other. Macro-F1 is 0.952 for valve sticking; pressure increments and motor-current bursts generate a distinct self-excited event pattern. F1 is 0.918 for wireless congestion; it is lower because the retransmission ratio and queue depth may fluctuate during benign maintenance traffic. Figure 4(a) uses a radar comparison to display precision, recall, and F1 across the five anomaly families; the largest precision-recall gap appears in sensor drift, with precision at 0.941 and recall at 0.902 because gradual drift sometimes remains below the event gate for several seconds. Figure 4(b) provides the only heatmap in the figure set and shows the normalized confusion matrix, where 96.8% of motor-overload samples are correctly assigned and the most common confusion is between data manipulation and sensor drift at 4.7%. These values indicate that event intensity helps separate abrupt faults, while slow anomalies still require residual evidence.

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

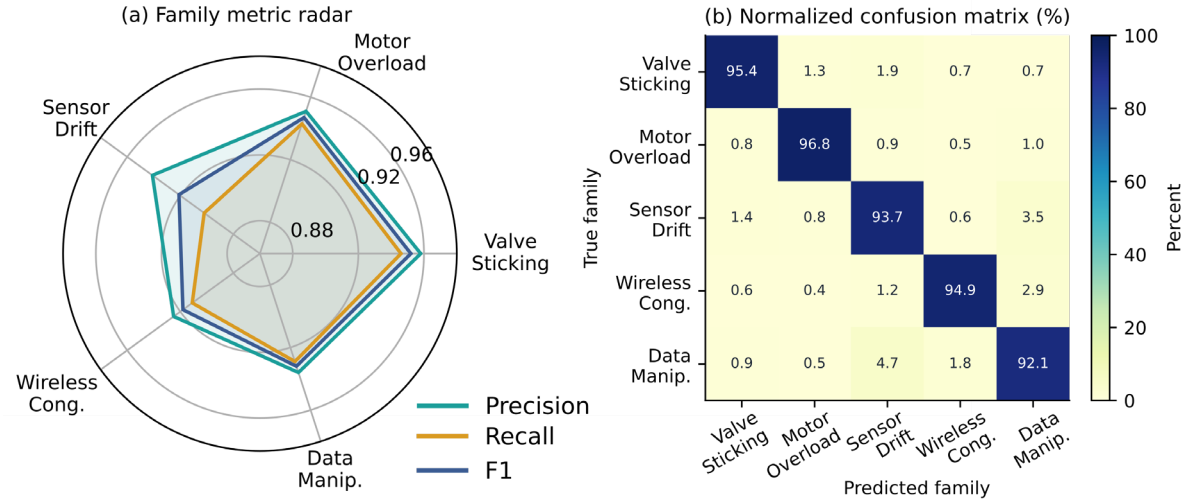

Figure 4. Family-level detection and confusion behavior: (a) radar comparison of precision, recall, and F1 by anomaly family; (b) normalized confusion matrix across five anomaly families.

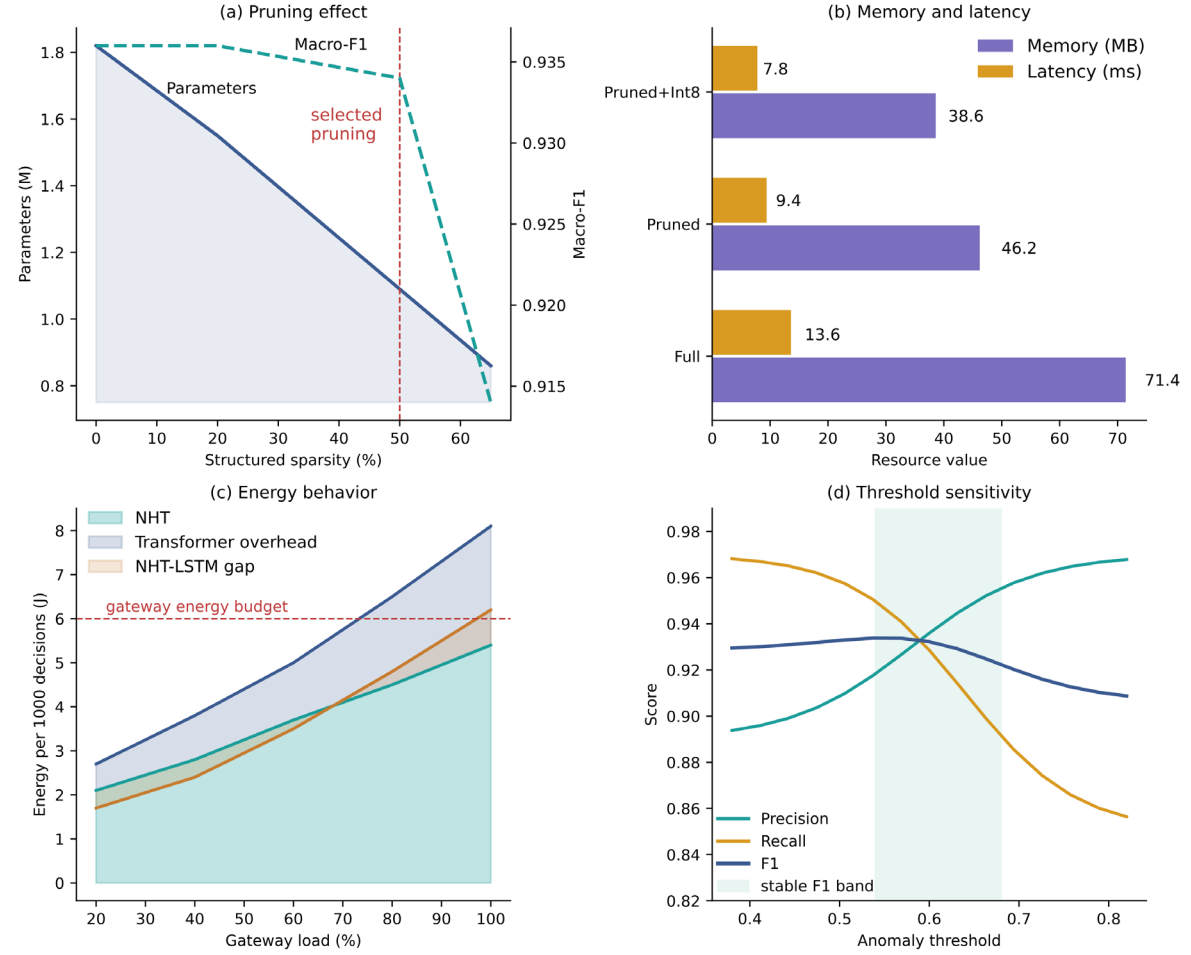

Figure 5. Edge resource and threshold behavior: (a) pruning effect on parameters and F1; (b) memory and latency under optimization; (c) energy-band comparison per 1000 decisions under gateway load; (d) threshold sensitivity.

Runtime measurements were performed on an ARM-class gateway profile with four CPU cores and 2GB of memory, and no cloud assistance was used for inference. Edge deployment studies often report that detection accuracy is not high enough unless latency and memory are considered [21]. Figure 5(a) shows that pruning reduces the parameter count to 1.09 million and only slightly decreases the macro-F1 score from 0.936 to 0.934.

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

Figure 5(b) reports memory and latency under three optimization levels; the final model uses 38.6 MB and averages 7.8 ms per inference, compared with 71.4 MB and 13.6 ms before optimization. Figure 5(c) presents energy bands per 1000 decisions over load levels from 20% to 100%, where the proposed model remains below 5.4 J at full load while the standard Transformer reaches 8.1 J. Figure 5(d) displays threshold sensitivity, and the stable F1 band from threshold 0.54 to 0.68 suggests that the model does not depend on a single fragile decision point.

Robustness was evaluated by injecting missing telemetry, timestamp jitter, and benign operating-mode shifts. Prior work on multivariate sensor anomaly detection has warned that missingness can inflate reconstruction error and produce alarms unrelated to fault behavior [22]. Figure 6(a) shows macro-F1 margins under random telemetry loss from 0% to 30%; the proposed model drops from 0.934 to 0.889, while the LSTM autoencoder falls from 0.812 to 0.718. Figure 6(b) reports jitter tolerance with 95% confidence intervals, where detection delay increases from 1.74 s at zero jitter to 2.21 s at 400 ms jitter. Figure 6(c) presents binned calibration error for anomaly probability, and the proposed model has an expected calibration error of 0.031, lower than 0.074 for the standard Transformer. The intensity likelihood is the main reason for this behavior because event timing remains informative even when some raw values are absent.

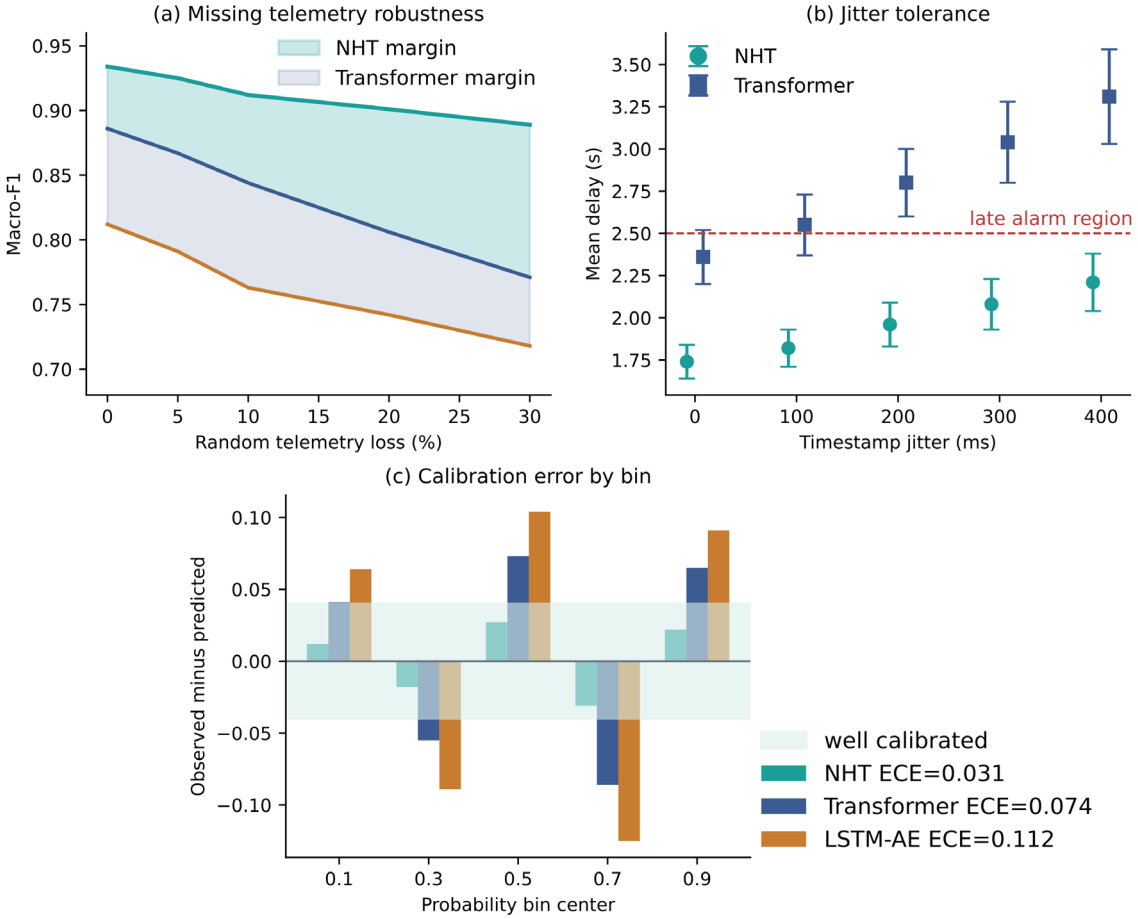

Figure 6. Robustness and calibration under streaming imperfections: (a) telemetry-loss robustness margins; (b) timestamp jitter delay with confidence intervals; (c) binned probability calibration error.

The latency profile also reveals how the event representation changes the computational shape of the problem. Raw fixed-window Transformers must process every synchronized timestamp, even during quiet periods. The proposed detector processes the event buffer, so a stable gateway remains in a low-cost monitoring mode until the eventizer emits a meaningful change. During high activity, the buffer naturally grows and attention cost rises, but the risk regions in Figure 3c show that the model remains usable before the selected buffer limit. This behavior is important for edge maintenance because abnormal intervals often coincide with heavy telemetry volume, and a detector that slows down exactly when evidence is richest can miss the intervention window.

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

A closer look at the family-level results shows that the residual and intensity terms respond differently to the shape of the anomaly. Valve sticking and motor overload cause abrupt increases followed by a short burst, and the intensity term rapidly raises the anomaly score. Sensor drift produces a small number of events but a continuous residual mismatch; therefore, the residual head provides most of the evidence. Wireless congestion leads to dense event bursts of queue depth and retransmission ratio, but benign network maintenance can also cause such short bursts; thus, the model needs both event timing and cross-channel context agreement to reduce false alarms. Data manipulation is also relatively ambiguous; an injected value may be physically plausible but inconsistent with the adjacent channels. This case can use channel-aware attention, and the model will compare the manipulated channel with variables that should have moved along with it.

Explanation Quality and Ablation Analysis

Explanation quality was measured with top-k root-cause hit rate, attribution stability under perturbation, and operator-aligned channel ranking. The labels identify primary and secondary causal channels for each anomaly interval, such as pressure and motor current for valve sticking or retransmission ratio and queue depth for wireless congestion. Explainable anomaly detection research in industrial control systems has argued that attribution should be evaluated against operational causes rather than only against feature importance visualizations [23]. Figure 7(a) reports top-one, top-three, and top-five hit rates; the proposed model achieves 0.731, 0.872, and 0.936, respectively, while the attention-only explanation reaches 0.624, 0.781, and 0.881. Figure 7(b) shows the distribution of attribution stability under masked non-causal telemetry, with the proposed attribution centered near 0.906 compared with 0.842 for residual-only explanation. Figure 7(c) reports explanation latency with confidence intervals against event count, where the complete explanation remains below 2.9 ms up to 160 tokens.

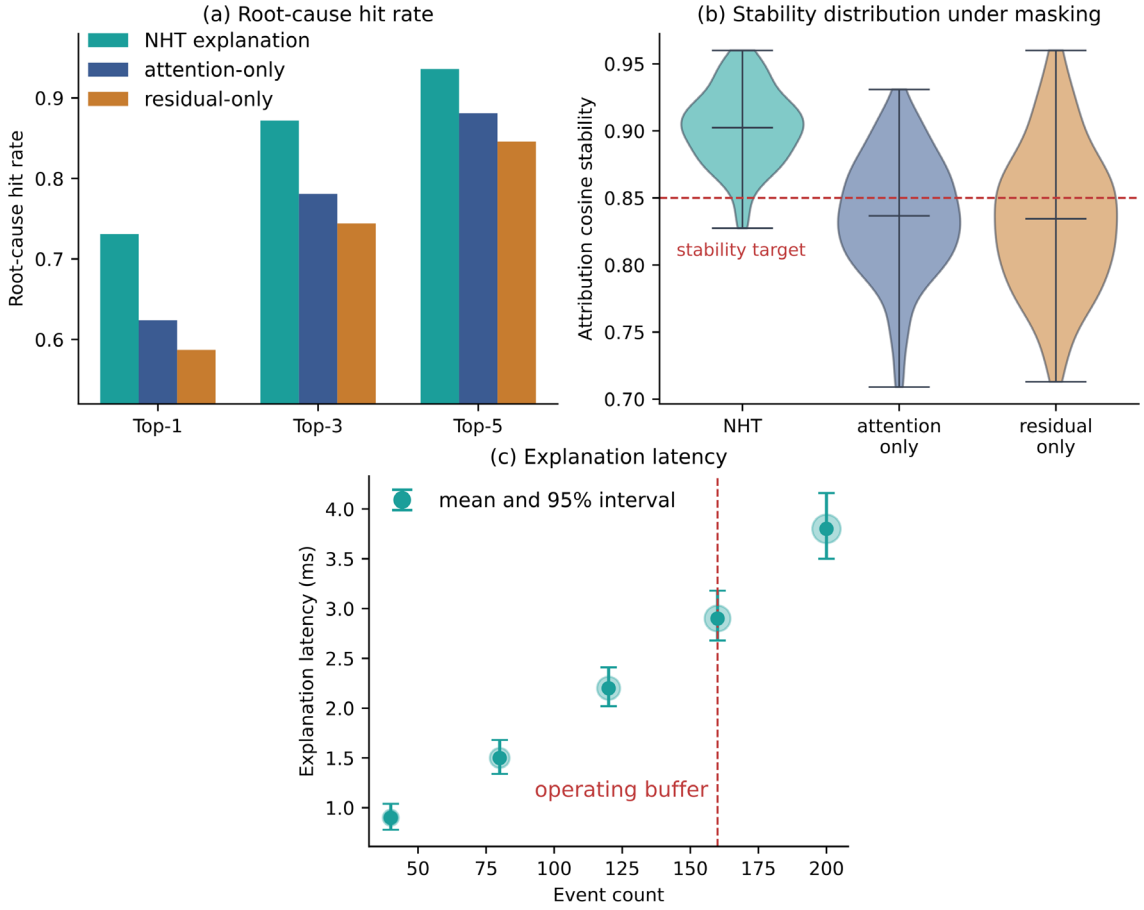

Figure 7. Explanation quality and cost: (a) top-k root-cause hit rate; (b) attribution stability distribution under masking; (c) explanation latency with confidence intervals versus event count.

https://doi.org/10.64972/dea.2023.v2i3.3693d:27-39

Ablation results show which parts are responsible for detection and explanation. After removing the Hawkes intensity head, the macro-F1 dropped to 0.886 and the median detection delay increased to 2.32 s; therefore, event arrival structure also contains information not covered by attention alone. Removing the residual head weakens gradual drift detection and lowers sensor-drift recall from 0.902 to 0.841. Removing channel-aware embedding has a smaller accuracy impact, but the top-three root-cause hit rate drops from 0.872 to 0.803 because channel identity is less separable in the explanation vector. Recently, graph-based anomaly studies have also shown that channel relationships are required for fault propagation among variables [24]. The proposed model learns the context of the channel through attention and event intensity instead of fixing an explicit graph before inference.

The baseline comparison also indicates where the proposed method is less dominant. One-class statistical learning has lower memory cost and remains competitive on abrupt single-channel spikes, but its F1 drops below 0.74 for compound anomalies because the normal boundary does not describe event cascades. Adversarial autoencoding is accurate on dense continuous deviations, yet its delay increases when anomalies appear as sparse bursts. Transformer detection is the closest baseline in aggregate accuracy. The proposed model improves over it mainly by adding time-gap encoding and likelihood-based intensity surprise.

The results are to be judged against a standard. Although the dataset has realistic physical constraints and edge service counters, it is still organized as a benchmark rather than a long-term deployment in multiple organizations. Recent studies on IoT anomaly detection have pointed out that dataset shift, labelling uncertainty and domain-specific operating modes can all alter the ranking of models [25]. Reduce the risk by including out-of-range parameters and mode shift in the evaluation, and conduct some additional cross-site tests before the operational approval. The reference implementation also assumes a gateway that can run a compact neural model. Extremely small microcontroller nodes may need to be distilled into a simpler student model, a hardware accelerator, or event filtering before inference. For systems without explanation labels, the detector can still be trained; however, the rank of explanations used for automated maintenance decisions should be verified against engineering logs [26].

The explanation outputs were inspected for several representative incidents to verify that numerical ranking matched plausible engineering narratives. In a valve-sticking case, pressure received the largest contribution, motor current ranked second, and flow ranked third; the retransmission ratio did not appear in the top five even though it fluctuated during the same period. In a wireless-congestion case, retransmission ratio, queue depth, and inter-arrival dispersion occupied the first three ranks, while vibration and pressure stayed below 0.04 contribution. These cases show why the explanation score should combine multiple components. Residual-only attribution can overemphasize channels with large absolute deviation, and attention-only attribution can highlight context channels that helped the encoder but did not directly express the abnormal state. The combined explanation is more suitable for gateway alarms because it separates evidence channels from merely correlated background variables.

Conclusion

This paper introduces a Neural Hawkes Transformer for edge anomaly explanation in multivariate IoT telemetry streams. The model converts raw sensor and network data into typed event increments, learns temporal and cross-channel context via a causal Transformer encoder, and estimates the likelihood of an abnormal event with a neural Hawkes intensity head. Its anomaly score combines intensity surprise and residual deviation, and its explanation output ranks channels by intensity contribution, residual evidence and attention-supported context. The experimental results show that the above Design can enhance detection accuracy, reduce delay, and output stable root-cause rankings under edge resource constraints.

Several limitations remain. The benchmark was carefully constructed and physically constrained, but it cannot cover the full diversity of industrial equipment, wireless environments, maintenance procedures and operator labeling practices. The explanation module also benefits from a calibration subset with channel-level cause labels, which may be expensive to obtain in some organizations. In addition, the current pruning and quantization strategy targets gateway-class devices; smaller embedded nodes would require a more aggressive compression strategy, hardware acceleration or split inference design.

Future work will evaluate the model in cross-site deployments where device populations, sampling policies and anomaly causes differ substantially. Another direction is to connect the explanation vector with maintenance knowledge graphs so that alarms can be translated into recommended checks and recovery actions. Continual learning is also important because edge telemetry distributions drift over months of equipment aging, firmware updates and seasonal operating changes. A final research path is to develop certified explanation stability tests before anomaly explanations are trusted in safety-critical automated control loops.

Author Contributions

Kensuke Ueda contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Takashi Sakamoto contributes to analysis, investigation, data collection, draft preparation. Yuto Matsumoto contributes to software, validation, supervision, project administration. Daisuke Miyazaki contributes to analysis, investigation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Lee, I. (2020). Internet of Things (IoT) cybersecurity: Literature review and IoT cyber risk management. Future Internet, 12(9), 157. https://doi.org/10.3390/fi12090157

[2] Thakkar, A., & Lohiya, R. (2021). A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges. Archives of Computational Methods in Engineering, 28, 3211-3243. https://doi.org/10.1007/s11831-020-09496-0

[3] Ge, M., Syed, N. F., Fu, X., Baig, Z., & Robles-Kelly, A. (2021). Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, 107784. https://doi.org/10.1016/j.comnet.2020.107784

[4] Moustafa, N., Keshk, M., Choo, K.-K. R., Lynar, T. M., Camtepe, S., & Whitty, M. T. (2021). DAD: A distributed anomaly detection system using ensemble one-class statistical learning in edge networks. Future Generation Computer Systems, 118, 240-251. https://doi.org/10.1016/j.future.2021.01.011

[5] Kayan, H., Majib, Y., Alsafery, W., Barhamgi, M., & Perera, C. (2021). AnoML-IoT: An end-to-end re-configurable multi-protocol anomaly detection pipeline for Internet of Things. Internet of Things, 16, 100437. https://doi.org/10.1016/j.iot.2021.100437

[6] Cao, J., Lin, L., & Ma, R. (2022). An efficient deep learning approach to IoT intrusion detection. The Computer Journal, 65(11), 2870-2879. https://doi.org/10.1093/comjnl/bxac119

[7] Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810. https://doi.org/10.1016/j.compeleceng.2022.107810

[8] Banaamah, A. M., & Ahmad, I. (2022). Intrusion detection in IoT using deep learning. Sensors, 22(21), 8417. https://doi.org/10.3390/s22218417

[9] Zhou, L., Zeng, Q., & Li, B. (2022). Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series. IEEE Access, 10, 40967-40978. https://doi.org/10.1109/ACCESS.2022.3167640

[10] Wang, X., Pi, D., Zhang, X., Liu, H., & Guo, C. (2022). Variational transformer-based anomaly detection approach for multivariate time series. Measurement, 191, 110791. https://doi.org/10.1016/j.measurement.2022.110791

[11] Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020). USAD: Unsupervised anomaly detection on multivariate time series. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 3395-3404. https://doi.org/10.1145/3394486.3403392

[12] Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. Proceedings of the AAAI Conference on Artificial Intelligence, 35(5), 4027-4035. https://doi.org/10.1609/aaai.v35i5.16523

[13] He, Q., Zheng, Y. J., Zhang, C. L., & Wang, H. Y. (2020). MTAD-TF: Multivariate time series anomaly detection using the combination of temporal pattern and feature pattern. Complexity, 2020, 8846608. https://doi.org/10.1155/2020/8846608

[14] Ding, C., Sun, S., & Zhao, J. (2023). MST-GAT: A multimodal spatial-temporal graph attention network for time series anomaly detection. Information Fusion, 89, 527-536. https://doi.org/10.1016/j.inffus.2022.08.011

[15] Song, Z. Y., & Liu, J. W. (2022). Tri-transformer Hawkes process via dot-product attention operations with event type and temporal encoding. Computational Intelligence, 38(2), 690-712. https://doi.org/10.1111/coin.12496

[16] Liu, J.-W., Zhang, L.-N., Song, Z.-Y., & Zuo, X. (2022). Temporal attention augmented transformer Hawkes process. Neural Computing and Applications, 34, 3795-3809. https://doi.org/10.1007/s00521-021-06641-z

[17] Chen, W., Chen, J., Cai, R., Liu, Y., & Hao, Z. (2022). Learning granger causality for non-stationary Hawkes processes. Neurocomputing, 468, 22-32. https://doi.org/10.1016/j.neucom.2021.10.030

[18] Okawa, M., Iwata, T., Tanaka, Y., Kurashima, T., Toda, H., & Kashima, H. (2022). Context-aware spatio-temporal event prediction via convolutional Hawkes processes. Machine Learning, 111, 2929-2950. https://doi.org/10.1007/s10994-022-06136-5

[19] Hussain, M. T., & Perera, C. (2022). Explainable sensor data-driven anomaly detection in Internet of Things systems. 2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation, 80-81. https://doi.org/10.1109/IoTDI54339.2022.00021

[20] Yin, C., Zhang, S., Wang, J., & Xiong, N. N. (2022). Anomaly detection based on convolutional recurrent autoencoder for IoT time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(1), 112-122. https://doi.org/10.1109/TSMC.2020.2968516

[21] Shahid, N., Aneja, S., Aneja, N., & Islam, M. S. (2022). Lightweight edge intelligence for real-time anomaly detection in industrial IoT. IEEE Access, 10, 82533-82547. https://doi.org/10.1109/ACCESS.2022.3196224

[22] Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. ACM computing surveys (CSUR), 54(3), 1-33. https://doi.org/10.1145/3444690

[23] Capaci, R. B., Voloch, L. B., & Zufiria, P. J. (2022). Explainable anomaly detection for industrial control system cybersecurity. IFAC-PapersOnLine, 55(10), 1183-1188. https://doi.org/10.1016/j.ifacol.2022.09.550

[24] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2828-2837. https://doi.org/10.1145/3292500.3330672

[25] Cook, A. A., Misirli, G., & Fan, Z. (2020). Anomaly detection for IoT time-series data: A survey. IEEE Internet of Things Journal, 7(7), 6481-6494. https://doi.org/10.1109/JIOT.2019.2958185

[26] Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C.-F., & Wang, S. (2020). A review of industrial wireless networks in the context of Industry 4.0. Wireless Networks, 26, 2281-2300. https://doi.org/10.1007/s11276-019-02186-7