Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080596

ARTICLE

MILOF-TCN: A Hierarchical Edge–Fog Framework for Monitoring Abnormal and Missing Patterns in Electric Vehicle Charging Data

Hwa-Young Jeong*

Humanitas College, Kyung Hee University, Seoul, Republic of Korea

*Corresponding Author: Hwa-Young Jeong. Email: hyjeong@khu.ac.kr

Received: 12 February 2026; Accepted: 25 May 2026; Published: 23 July 2026

ABSTRACT: The rapid growth of electric vehicle (EV) charging infrastructures has introduced new challenges in monitoring abnormal load behaviors under strict latency and resource constraints. Conventional anomaly detection approaches either rely on centralized processing or incur excessive false alarms, limiting their practical applicability in large-scale deployments. This paper proposes a hierarchical edge-fog anomaly detection framework that integrates lightweight edge-level filtering with a fog-level Temporal Convolutional Network (TCN) detector. The edge component suppresses non-informative patterns, while the fog layer performs temporal modeling on selectively forwarded data. This design enables controllable reduction of fog-level processing load. Under corrected end-to-end evaluation on real-world EV charging load data, the hierarchical pipeline should be interpreted as a system operating point rather than a uniformly superior detector. Relative to fog-only TCN-AE inference, the selected routing policy reduces fog workload by 34.9% and shortens average detection delay from 93.6 to 75.6 h, but increases false alarms per day from 0.88 to 7.29 and lowers F1 from 0.547 to 0.455. Sensitivity experiments over routing thresholds reveal a consistent trade-off among fog workload, alert burden, detection delay, and retained anomaly evidence. Additional routing diagnostics show that the primary source of performance degradation is information loss induced by filtering, rather than weakness of the fog detector on the forwarded subset. These findings suggest that hierarchical edge intelligence is a practical but calibration-sensitive direction for scalable anomaly monitoring in EV charging infrastructures.

KEYWORDS: Edge computing; fog computing; anomaly detection; electric vehicle charging; time-series monitoring; temporal convolutional networks; missing data patterns; hierarchical architecture

1 Introduction

1.1 Background and Motivation

The rapid adoption of electric vehicles (EVs) has accelerated the deployment of large-scale EV charging infrastructures, making reliable monitoring of charging load behaviors a critical operational requirement. As charging stations become increasingly distributed and heterogeneous, monitoring systems must continuously assess both abnormal load patterns and data quality issues, including missing or corrupted measurements.

In real-world deployments, EV charging data are frequently affected by diverse irregularities. Sudden load spikes may originate from abnormal charging behaviors, equipment malfunctions, or unauthorized usage, while missing data segments commonly arise from sensor faults, communication failures, or maintenance activities. Such irregularities not only degrade data reliability but also complicate real-time monitoring, particularly under strict latency and resource constraints. These challenges highlight the need

for monitoring frameworks that can jointly handle abnormal and missing patterns in a scalable and efficient manner. Recent EV charging studies have therefore examined session-level anomaly detection, data-driven station monitoring, and cyberattack-oriented anomaly detection, but these directions are typically treated separately from hierarchical edge–fog routing [1–3].

1.2 Challenges in Monitoring EV Charging Data

Traditional monitoring approaches predominantly rely on centralized data collection and analysis. While centralized methods can leverage expressive statistical or machine learning models [4,5], they often suffer from scalability limitations and delayed responses when applied to large-scale infrastructures. Moreover, centralized processing amplifies the impact of missing data and transient fluctuations, frequently leading to excessive false alarms or delayed detection of critical events.

From an operational perspective, monitoring systems must satisfy multiple, often conflicting requirements. They should detect abnormal and missing patterns in a timely manner, suppress spurious responses to minor fluctuations, and operate efficiently under constrained communication bandwidth and computational resources. These requirements make purely centralized solutions impractical for large-scale EV charging infrastructures and motivate the adoption of distributed monitoring architectures.

1.3 Related Work and Limitations

A wide range of approaches has been proposed for time-series anomaly detection, including statistical thresholding methods and classical machine learning techniques such as Isolation Forests [4–7]. While statistical methods offer computational efficiency, they often lack robustness to non-stationary behaviors and are highly sensitive to missing data. Recent IoT anomaly-detection surveys further show that this problem space increasingly spans edge deployment, deep learning, graph-based models, and security monitoring [8].

More recent studies have explored deep learning-based models, including recurrent, convolutional, and Transformer architectures, for temporal anomaly detection [9–13]. Although these models can capture complex temporal dependencies, their deployment in large-scale monitoring systems remains challenging due to high computational costs and sensitivity to noisy or incomplete data.

Most existing works treat missing data as a preprocessing issue to be imputed or discarded [14–16], rather than as an integral part of the monitoring problem. In contrast, recent studies have highlighted the importance of explicitly modeling missing patterns in time-series analysis [17–19].

In parallel, edge and fog computing paradigms have been investigated to address latency and scalability challenges in distributed systems [20–23]. Hierarchical anomaly detection has been studied for IoT data in distributed edge-computing environments, but applications to EV charging time series remain limited [24]. In contrast, prior EV charging studies have focused primarily on demand characterization and forecasting rather than hierarchical anomaly monitoring [25,26].

1.4 Our Approach and Contributions

In this work, we propose a hierarchical edge–fog monitoring framework for EV charging infrastructures that jointly addresses abnormal load behaviors and missing data patterns under practical deployment constraints. The proposed system integrates lightweight edge-level filtering with expressive fog-level temporal modeling, enabling early suppression of non-informative data while allowing explicit control of detection-performance trade-offs.

The main contributions of this study are summarized as follows:

- We design a hierarchical monitoring architecture that jointly distributes abnormal and missing pattern analysis across edge and fog layers, explicitly considering latency, communication, and computational constraints.
- We introduce a lightweight edge-level filtering mechanism using MiLOF micro-clusters that reduces fog-level processing load while exposing a tunable trade-off between alert burden and detection delay.
- Through corrected end-to-end experiments on real-world EV charging load data, we demonstrate that the proposed framework exposes a clear trade-off among fog workload, detection delay, false alarms, and retained anomaly evidence, and we quantify this coupling using routing diagnostics.

2 System Overview and Problem Formulation

This section presents an overview of the proposed hierarchical edge–fog monitoring system and formally defines the problem addressed in this study. The primary objective is to reliably monitor abnormal load behaviors and missing data patterns in electric vehicle (EV) charging infrastructures under practical deployment constraints, including limited latency, communication bandwidth, and computational resources.

2.1 System Overview

Fig. 1 illustrates the overall architecture of the proposed hierarchical monitoring framework, which consists of two primary computational layers: edge and fog, supported by periodic model updates from a backend infrastructure. The system is designed to perform real-time monitoring of EV charging loads by distributing computational responsibilities across these layers according to their operational characteristics.

At the **edge layer**, lightweight monitoring modules are deployed at individual charging sites. Each edge node continuously observes local charging load sequences and performs memory-efficient pre-filtering to identify potentially informative temporal segments. When abnormal patterns or data irregularities are detected, the edge node can immediately trigger local alerts for safety-critical responses, such as temporary shutdowns or isolation procedures.

The **fog layer** aggregates filtered data streams forwarded from multiple edge nodes. A Temporal Convolutional Network (TCN)-based analyzer models temporal dependencies across incoming sequences and produces refined anomaly scores. By operating on selectively forwarded data rather than raw streams, the fog layer significantly reduces processing overhead while preserving essential temporal information required for reliable monitoring.

For system evolution and offline optimization, model parameters and insights are periodically aggregated and processed through backend infrastructure (offline phase). Improved models can subsequently be deployed back to fog and edge nodes during scheduled updates. This design enables scalable monitoring while maintaining asynchronous adaptability to evolving load patterns. Edge–fog architectures have been explored to address latency and scalability constraints in distributed systems [21,23].

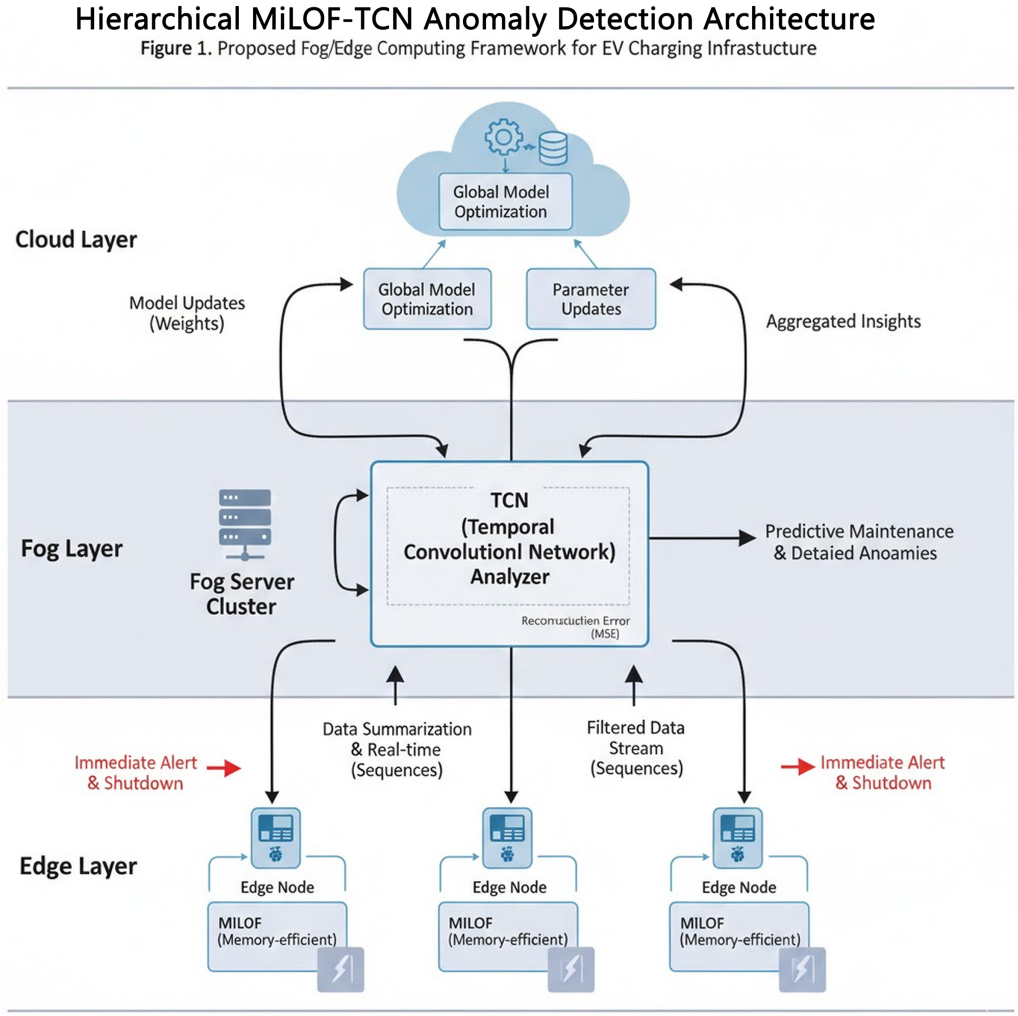

Figure 1: Overview of the proposed hierarchical edge–fog monitoring architecture for EV charging infrastructures. Lightweight edge nodes perform local pre-filtering and immediate response, while the fog layer conducts temporal modeling using a TCN-based analyzer on selectively forwarded data. Backend infrastructure supports periodic model optimization and updates.

2.2 Problem Formulation

Let $\{x_t\}_{t=1}^{T}$ denote the aggregated EV charging load observed at a site, where $x_t \in \mathbb{R}^+$. In implementation, each timestamp is represented by a multivariate feature vector

$$\mathbf{z}_t = \phi(x_t, \mathbf{c}_t, \mathbf{r}_t) \in \mathbb{R}^C, \tag{1}$$

where $\mathbf{c}_t$ denotes calendar features (hour, day-of-week, weekend indicator) and $\mathbf{r}_t$ denotes rolling statistics. To avoid leakage, edge-side standardization is fitted on train split only:

$$\tilde{z}_{t,c} = \frac{z_{t,c} - \mu_c^{\mathrm{tr}}}{\tilde{\sigma}_c^{\mathrm{tr}}}, \quad \mu_c^{\mathrm{tr}} = \frac{1}{|\mathcal{T}_{\mathrm{tr}}|} \sum_{t \in \mathcal{T}_{\mathrm{tr}}} z_{t,c}, \quad \tilde{\sigma}_c^{\mathrm{tr}} = \max\left(\sigma_c^{\mathrm{tr}}, 1\right). \tag{2}$$

The clamp $\tilde{\sigma}_c^{\mathrm{tr}} = \max(\sigma_c^{\mathrm{tr}}, 1)$ prevents unstable amplification when a feature has very small train-split variance. Due to sensor faults, communication failures, or abnormal charging behaviors, the observed sequence may contain both abnormal values and missing segments.

The monitoring task considered in this study is to identify time intervals that exhibit either *abnormal load behaviors* or *missing data patterns* in a timely and reliable manner. We consider a window-based formulation, where a sequence is divided into overlapping windows $\mathbf{X}_i = \{\tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_{i+1}, \ldots, \tilde{\mathbf{z}}_{i+L-1}\} \in \mathbb{R}^{L \times C}$ of fixed length L. For evaluation labels, the implementation uses "any" aggregation:

$$y_i = \mathbb{I}\left[\max_{t=i,\ldots,i+L-1} y_t = 1\right]. \tag{3}$$

For each window $\mathbf{X}_i$, the system aims to estimate an anomaly score $s_i \in \mathbb{R}$ that reflects the likelihood of abnormal or missing behavior within the window. A window is flagged as informative if its score exceeds a predefined threshold τ, i.e.,

$\mathbf{X}_i$ is informative if $s_i > \tau$.

The overall objective of the proposed hierarchical framework is twofold: (1) to accurately identify abnormal and missing patterns at the window and event levels, and (2) to minimize unnecessary data transmission and computation at the fog layer by filtering non-informative windows at the edge. This formulation naturally motivates a hierarchical design in which lightweight edge-level filtering and expressive fog-level temporal modeling operate in a complementary manner.

Formally, the monitoring objective can be expressed as a joint optimization problem. Let $\mathcal{I}$ denote the set of informative windows forwarded to the fog layer. The proposed framework aims to solve:

$$\min_{\mathcal{I}} \ \mathbb{E}\left[|\mathcal{I}|\right] \quad \text{subject to} \quad \mathrm{F1}(\mathcal{I}) \geq \alpha, \tag{4}$$

where $\mathbb{E}[|\mathcal{I}|]$ represents the expected fog-level processing load, and α is a predefined minimum detection performance requirement.

In practice, the framework addresses this trade-off through threshold-based gating rather than explicit constrained optimization. Edge and fog thresholds (τ_e, τ_f) are calibrated on validation sets independently to achieve desired operating points, enabling site-specific operational tuning without requiring joint optimization. This pragmatic approach trades theoretical optimality for robustness and interpretability, allowing operators to understand and adjust how edge filtering affects downstream fog-layer performance. The Pareto-style analysis in Section 6.2 can be interpreted as an empirical realization of Eq. (4), where routing thresholds instantiate feasible operating points under different workload and detection-performance constraints. Accordingly, the threshold sweep does not replace the formulation; rather, it provides the candidate operating points from which a deployment-specific constrained solution can be selected. The threshold sweep therefore serves as a practical approximation of the constrained operating-point search implied by Eq. (4).

3 Hierarchical Edge–Fog Monitoring Methodology

This section describes the proposed hierarchical monitoring methodology, detailing the edge-level filtering mechanism and the fog-level temporal modeling process. The overall design aims to jointly handle abnormal load behaviors and missing data patterns while minimizing unnecessary computation and communication overhead.

3.1 Edge-Level MiLOF Filtering with Micro-Clusters

At the edge layer, each node performs streaming anomaly scoring using a micro-cluster-based MiLOF approximation. Let $\mathbf{z}_t \in \mathbb{R}^C$ denote the feature vector at time step t. The detector maintains a set of micro-clusters $\mathcal{C} = \{c_j\}_{j=1}^{M}$, where each cluster stores center μ_j, count n_j, sum of squared errors S_j, and last update time t_j.

The cluster radius is defined as

$$r_j = \sqrt{\frac{S_j}{\max(1, n_j - 1)}}. \tag{5}$$

Given a new sample $\mathbf{z}_t$, let c_j be its nearest cluster. The sample is absorbed by c_j if

$$\|\mathbf{z}_t - \mu_j\|_2 \le \alpha r_j + \varepsilon, \tag{6}$$

where α is the radius factor and ε is a small slack constant. When accepted, the cluster is updated online as

$$\mu_j^+ = \mu_j + \frac{\mathbf{z}_t - \mu_j}{n_j + 1}, \qquad n_j^+ = n_j + 1, \tag{7}$$

$$S_j^+ = S_j + \|\mathbf{z}_t - \mu_j^+\|_2^2 + n_j\|\mu_j - \mu_j^+\|_2^2. \tag{8}$$

Otherwise, a new cluster is created.

For scoring, a LOF-like local density ratio is computed over cluster centers. The LOF-style density estimation follows the formulation introduced in Breunig et al. [27], adapted here to operate on streaming micro-cluster summaries. Let $\mathcal{N}_k(\mathbf{z}_t)$ be the k nearest centers to $\mathbf{z}_t$:

$$\text{reach}(\mathbf{z}_t, c) = \max(d(\mathbf{z}_t, c),\ k\text{-dist}(c)), \quad \text{lrd}(\mathbf{z}_t) = \frac{1}{\frac{1}{k}\sum_{c \in \mathcal{N}_k(\mathbf{z}_t)} \text{reach}(\mathbf{z}_t, c) + \varepsilon}. \tag{9}$$

The edge score is then

$$s_t^{(e)} = \frac{\frac{1}{k}\sum_{c \in \mathcal{N}_k(\mathbf{z}_t)} \text{lrd}(c)}{\text{lrd}(\mathbf{z}_t) + \varepsilon}. \tag{10}$$

To match the streaming implementation, scoring uses a warm-up rule:

$$s_t^{(e)} = \begin{cases} 1, & |\mathcal{C}_t| < M_{\min}, \\ \text{LOF-like}(\mathbf{z}_t, \mathcal{C}_t), & \text{otherwise}, \end{cases} \tag{11}$$

where $M_{\min}$ is the minimum number of clusters required for stable neighborhood scoring. After each update, stale clusters are pruned and capacity is capped:

$$\mathcal{C}_t \leftarrow \{c_j \in \mathcal{C}_t \mid t - t_j \le \Delta_{\max}\}, \qquad |\mathcal{C}_t| > M_{\max} \Rightarrow \mathcal{C}_t \leftarrow \text{Top}_{M_{\max}}(\mathcal{C}_t; n_j). \tag{12}$$

This streaming adaptation is inspired by memory-efficient local outlier detection in data streams [28].

The edge threshold is calibrated using validation-stream scores (after chronological train warm-up) with either quantile or mean-std calibration:

$$\tau_e = Q_{q_e}\left(\{s_t^{(e)}\}_{t\in\mathcal{V}}\right) \quad \text{or} \quad \tau_e = \mu_{s^{(e)},\mathcal{V}} + k_e\sigma_{s^{(e)},\mathcal{V}}. \tag{13}$$

The val-only protocol then applies a single fixed τ_e to all splits:

$$e_t^{(r)} = \mathbb{I}[s_t^{(e,r)} \geq \tau_e], \qquad r \in \{\text{tr}, \text{val}, \text{te}\}. \tag{14}$$

The complement of edge-flag rate is reported as the edge non-flag rate:

$$\text{NonFlagRate}_e^{(r)} = 1 - \frac{1}{T_r}\sum_{t=1}^{T_r} e_t^{(r)}, \qquad r \in \{\text{tr}, \text{val}, \text{te}\}. \tag{15}$$

In exported artifacts, this quantity is logged with the legacy key `edge_pass_rate_*`. Point-level edge flags are defined by $e_t = \mathbb{I}[s_t^{(e)} \geq \tau_e]$. For a window $\mathbf{X}_i$ of length L, we compute the edge-anomaly ratio

$$\rho_i = \frac{1}{L}\sum_{t=i}^{i+L-1} e_t. \tag{16}$$

The window is forwarded to the fog layer only when

$$\mathbf{X}_i \to \text{Fog} \quad \text{if} \quad \rho_i \leq \gamma, \tag{17}$$

where γ is the maximum allowed edge-anomaly ratio. This gate direction is intentional: windows with $\rho_i > \gamma$ are treated as edge-dominant incidents for immediate local handling, while fog analysis is reserved for lower-ratio windows that require additional temporal context.

3.2 Fog-Level Temporal Modeling

The fog layer receives forwarded windows $\mathbf{X}_i \in \mathbb{R}^{L\times C}$ and performs reconstruction-based scoring using a TCN autoencoder. Temporal convolutional networks have been widely used for sequence modeling [11]. Reconstruction-based anomaly detection is a common unsupervised approach for time-series monitoring [9]. The model uses stacked dilated causal convolutional blocks with residual connections, followed by a 1×1 projection layer for feature-space reconstruction:

$$\hat{\mathbf{X}}_i = \text{Proj}_{1\times 1}(\text{TCN}(\mathbf{X}_i)). \tag{18}$$

Training follows an unsupervised one-class protocol: only normal windows from the training split are used to optimize reconstruction fidelity. The optimization objective is the mean MSE over all timesteps and channels:

$$\mathcal{L}_{rec} = \frac{1}{N_{\text{tr}}}\sum_{i=1}^{N_{\text{tr}}}\frac{1}{LC}\sum_{t=1}^{L}\sum_{c=1}^{C}\left(X_{i,t,c} - \hat{X}_{i,t,c}\right)^2. \tag{19}$$

At inference, the fog-level anomaly score is defined by the same per-window reconstruction error:

$$s_i^{(f)} = \frac{1}{LC}\sum_{t=1}^{L}\sum_{c=1}^{C}\left(X_{i,t,c} - \hat{X}_{i,t,c}\right)^2. \tag{20}$$

Let $\mathcal{R}(B,k)$ denote the receptive field with B temporal blocks and kernel size k. Because each block contains two dilated causal convolutions with dilation 2^b, the effective receptive field is

$$|\mathcal{R}(B,k)| = 1 + 2(k-1)\sum_{b=0}^{B-1} 2^b. \tag{21}$$

This multi-scale temporal coverage enables the fog model to capture abrupt spikes, gradual drifts, and extended missing-pattern effects within a unified reconstruction framework.

3.3 Hierarchical Decision Strategy

The framework combines edge-level screening and fog-level reconstruction scoring in a two-stage decision pipeline. Edge scoring uses a permissive threshold to preserve high recall, while fog scoring uses a stricter threshold for precision-oriented final decisions.

For fog inference, the decision rule is

$$\hat{y}_i = \mathbb{I}\left[s_i^{(f)} \geq \tau_f\right]. \tag{22}$$

In the current experimental protocol, anomalies are injected only into the test split; therefore, the validation split is normal-only. The fog threshold is calibrated on normal-only validation windows:

$$\tau_f = Q_{q_f}\left(\{s_i^{(f)}\}_{i\in\mathcal{V}_{\text{normal}}}\right) \quad \text{or} \quad \tau_f = \mu_{s^{(f)},\mathcal{V}_{\text{normal}}} + k_f\sigma_{s^{(f)},\mathcal{V}_{\text{normal}}}. \tag{23}$$

The same quantile/mean-std calibration family is used for edge thresholding to maintain operational consistency across layers.

This hierarchy separates responsibilities: edge nodes provide immediate local handling for strongly irregular intervals, and the fog layer performs robust temporal discrimination on forwarded windows. As a result, the system balances low-latency responses with stable event-level detection performance. Graph-attention approaches have shown effectiveness in multivariate time-series anomaly detection by explicitly modeling dependencies among variables [29].

3.4 Module-Wise Algorithms

Algorithms 1–3 summarize the full execution path from streaming edge screening to event-level evaluation. These procedures are intentionally coupled rather than independent: Algorithm 1 controls upstream data selection, Algorithm 2 performs downstream temporal discrimination, and Algorithm 3 translates window predictions into operational outcomes. This decomposition clarifies the distinct functional role of each stage in the hierarchy. In our design, the edge stage is responsible for computational selectivity, the fog stage for representation-level anomaly scoring, and the event stage for operational interpretability.

Algorithm 1: Streaming edge MiLOF filtering and window gate

Require: Feature stream $\{\mathbf{z}_t\}_{t=1}^{T}$, window length L, stride h, neighbors k, radius factor α, gate ratio γ

1: Initialize MiLOF detector with empty micro-cluster set $\mathcal{C}$
2: **for** $t = 1$ to T **do**
3: $\quad s_t^{(e)} \leftarrow \textsc{ScoreMiLOF}(\mathbf{z}_t, \mathcal{C}, k)$
4: $\quad \mathcal{C} \leftarrow \textsc{PartialFit}(\mathbf{z}_t, \mathcal{C}, \alpha)$
5: **end for**

(Continued)

Algorithm 1 (continued)

$\tau_e \leftarrow \textsc{Threshold}(\{s_t^{(e)}\}_{t\in\mathcal{V}}, \text{quantile or mean-std})$
$e_t \leftarrow \mathbb{I}[s_t^{(e)} \geq \tau_e] \quad \forall t$
for each window $\mathbf{X}_i$ generated with (L, h) **do**
    $\rho_i \leftarrow \frac{1}{L}\sum_{t=i}^{i+L-1} e_t$
    if $\rho_i \leq \gamma$ **then**
        Forward $\mathbf{X}_i$ to fog
    else
        Keep interval for edge-side handling
    end if
end for

Algorithm 1 is designed as a high-recall front-end filter. The point-wise MiLOF score identifies locally isolated feature states, while the window gate ratio ρ_i regularizes short-term fluctuations that could otherwise trigger unstable forwarding. An important design choice is that windows dominated by edge irregularity are handled locally instead of being forwarded. These intervals are treated as edge-dominant incidents to prioritize low-latency local action; this choice trades off fog-side contextual analysis for immediate response, while reserving fog resources for structurally ambiguous windows. Accordingly, γ is an operational knob: smaller values prioritize fog efficiency, whereas larger values prioritize comprehensive fog inspection.

Algorithm 2: Fog TCN-AE training and scoring

Require: Forwarded windows $\mathcal{X}_{\text{tr}}, \mathcal{X}_{\text{val}}, \mathcal{X}_{\text{te}}$
Train TCN-AE on $\mathcal{X}_{\text{tr}}$ using $\mathcal{L}_{rec}$ and select best epoch by validation loss
Compute validation reconstruction scores $\{s_i^{(f)}\}_{i\in\mathcal{X}_{\text{val}}}$
Compute test reconstruction scores $\{s_i^{(f)}\}_{i\in\mathcal{X}_{\text{te}}}$
$\tau_f \leftarrow \textsc{Threshold}(\{s_i^{(f)}\}_{i\in\mathcal{X}_{\text{val}}}, \text{quantile or mean-std})$
$\hat{y}_i \leftarrow \mathbb{I}[s_i^{(f)} \geq \tau_f]$
return $\{s_i^{(f)}, \hat{y}_i\}$

Algorithm 3: Event-level aggregation and delay metrics

Require: Window outputs $\{(t_i, y_i, \hat{y}_i, g_i)\}$ where g_i is event group id (−1 for normal)
for each event group $g \geq 0$ **do**
    $t_g^{\text{start}} \leftarrow \min\{t_i \mid g_i = g,\ y_i = 1\}$
    $t_g^{\text{det}} \leftarrow \min\{t_i \mid g_i = g,\ y_i = 1,\ \hat{y}_i = 1\}$
    $\text{detected}_g \leftarrow \mathbb{I}[t_g^{\text{det}} \text{ exists}]$
    if $\text{detected}_g = 1$ **then**
        $\Delta_g \leftarrow t_g^{\text{det}} - t_g^{\text{start}}$
    end if
end for
DetectedRate $\leftarrow \text{mean}_g(\text{detected}_g)$
AvgDelay $\leftarrow \text{mean}_g(\Delta_g \mid \text{detected}_g = 1)$
FalseAlarms/day $\leftarrow \text{mean}_d \sum_{i:t_i\in d} \mathbb{I}[y_i = 0 \wedge \hat{y}_i = 1]$
return event-level metrics

Algorithm 2 formalizes the fog role as a one-class temporal discriminator. Because training uses only normal windows, anomaly evidence is encoded as reconstruction mismatch rather than class-specific posterior probability. This choice improves robustness when anomaly taxonomy is incomplete or evolving, which is typical in charging infrastructure monitoring. Threshold calibration on validation scores further makes the decision boundary explicit and reproducible, reducing ambiguity about how τ_f is selected and why different operating points produce different precision-recall and delay trade-offs.

3.5 Handling Abnormal and Missing Patterns

Abnormal load behaviors and missing data patterns have distinct temporal signatures. Abrupt anomalies usually appear as sharp deviations, whereas missing segments tend to create sustained irregularities or flat regions.

The proposed methodology addresses both cases by combining complementary mechanisms. Edge-level filtering rapidly reacts to sudden deviations and data discontinuities, while fog-level temporal modeling captures longer-term structural inconsistencies through reconstruction errors. This complementary design enables robust monitoring across diverse abnormal scenarios without relying on explicit pattern-specific rules.

3.5.1 Point-Wise Spikes and Drops

Point-wise anomalies, such as sudden load spikes or drops, manifest as sharp localized deviations from expected levels. These patterns are often caused by equipment malfunctions, transient charging events, or sensor faults. Point-wise anomalies are highly detectable at the edge level because they create locally extreme feature values that deviate significantly from the micro-cluster neighborhoods. The edge MiLOF scoring immediately flags such windows through high local density ratio scores, enabling rapid suppression of spurious signals and reducing unnecessary fog-layer processing. In reconstructive temporal models, point spikes typically result in elevated reconstruction errors because the model was trained on smooth normal patterns and thus poorly predicts sharp transitions.

3.5.2 Collective Shifts and Sustained Deviations

Collective shifts represent sustained changes in the load pattern, such as a base-level increase lasting hours or days. These anomalies include equipment failures, unusual operational modes, or charging demand changes that persist beyond transient fluctuations. At the edge, collective shifts may not be immediately obvious if the shift is gradual, because the shifted pattern can still remain localized relative to slowly-evolving micro-cluster centers. In such cases, windows with moderate edge flags (e.g., $0.2 < \rho_i < \gamma$) are forwarded to the fog layer, where the TCN-AE reconstruction model faces a key challenge: detecting whether the temporal sequence structure is abnormal despite resembling previously learned normal sequences. The fog detector's ability to distinguish sustained deviations depends on whether the shifted magnitude exceeds the model's learned tolerance bounds. If the shift is sufficiently large, reconstruction error increases; if the shift is subtle and occurs within the training-set variability, the fog model may assign low anomaly scores, making calibration of τ_f critical for operational performance.

3.5.3 Gradual Drifts

Gradual drifts represent slow, continuous changes in the time-series behavior over days or weeks. In EV charging contexts, drifts may arise from seasonal effects, equipment degradation, or evolving user behavior patterns. Unlike collective shifts (which step abruptly), drifts blend imperceptibly into the background.

Edge-level anomaly detection struggles with drifts because the feature statistics change so slowly that micro-cluster neighborhoods adapt over time; the edge threshold τ_e is fixed after validation, and thus cannot continuously re-calibrate to drifting means and variances. Consequently, the fog layer is the primary locus of drift detection. The TCN-AE model, trained on normal data, produces high reconstruction errors when the input sequence exhibits a systematic directional deviation that accumulates over the window length L. The multi-scale receptive field of the TCN (spanning blocks with increasing dilation) captures both short-term fluctuations and longer-term trends, enabling the model to recognize gradual inconsistencies. However, if drifts are very slow relative to the window length, both edge and fog may miss the anomaly if thresholds are not sufficiently conservative.

3.5.4 Missing Data Blocks

Missing data segments arise from sensor failures, communication outages, or maintenance stop periods. Missing values create data-quality anomalies that are distinct from behavioral anomalies but equally important for infrastructure reliability. At the edge, a window containing missing values exhibits irregular feature vectors (e.g., forward-filled or interpolated values create artificial patterns). The edge MiLOF detector flags such irregularities as local outliers because missing-imputed patterns deviate from normal micro-cluster statistics. At the fog, a TCN-AE trained exclusively on complete normal data encounters significant distributional shift when presented with imputed sequences, resulting in high reconstruction error even if the imputation strategy is reasonable. By jointly treating behavioral anomalies and missing-data patterns under a unified anomaly detection framework, the system avoids the need for separate missing-value handling modules and instead leverages the complementary strengths of edge (rapid detection of irregularity) and fog (global temporal context) layers. This design choice aligns with recent findings that missing patterns should be modeled as monitoring targets rather than preprocessed away [17].

4 System Implementation

This section describes the practical implementation details of the proposed hierarchical monitoring framework, including data preprocessing, anomaly and missing pattern injection, and the end-to-end inference pipeline. The implementation is designed to reflect realistic deployment conditions in EV charging infrastructures.

4.1 Data Preparation and Preprocessing

Raw EV charging load data are first preprocessed to construct time-series sequences suitable for window-based monitoring. The original measurements are temporally aligned and resampled to a fixed interval, after which missing values introduced by logging or communication failures are preserved rather than imputed to reflect realistic operating conditions.

Each time-series is normalized using statistics computed from the training set to ensure stable model behavior. Overlapping windows of fixed length L are then extracted with a predefined stride. This preprocessing pipeline enables consistent handling of both normal and irregular load patterns across different charging sites.

4.2 Injection of Abnormal and Missing Patterns

To evaluate the robustness of the proposed framework under diverse abnormal conditions, synthetic abnormal and missing patterns are injected into the test set. The injection process simulates realistic

operational issues commonly observed in EV charging infrastructures, such as sudden load spikes, sustained deviations, gradual drifts, and missing data blocks.

Injection parameters, including duration, magnitude, and occurrence probability, are randomly sampled within predefined ranges to avoid overfitting to specific anomaly shapes. This strategy ensures that the evaluation reflects heterogeneous and unpredictable behaviors encountered in real-world deployments.

4.3 End-to-End Inference Pipeline

During inference, each incoming timestep is first transformed into a feature vector and evaluated by the edge MiLOF module to produce $s_t^{(e)}$ and edge flag e_t. Windows are then generated on the original timeline, and each window is kept only if its edge-anomaly ratio ρ_i does not exceed γ. In our implementation, a single τ_e is calibrated on validation-stream edge scores and then fixed for train/validation/test gating.

Forwarded windows are scored by the fog TCN-AE using reconstruction error $s_i^{(f)}$. Final window-level predictions are obtained by comparing $s_i^{(f)}$ with τ_f. This implementation matches the training protocol in which the fog model is learned on normal windows only and thresholded on validation-score statistics.

The resulting pipeline provides immediate edge-side handling for strongly irregular intervals while reserving fog computation for windows that require temporal discrimination.

4.4 Computational and Operational Considerations

The implementation explicitly considers computational constraints at different system layers. At the edge, memory usage is bounded by the maximum number of micro-clusters (M), and each update is performed using nearest-center lookup and lightweight online statistics. This yields practical low-latency operation under edge hardware limits, aligning with recent findings on efficient edge AI inference and memory-efficient on-device time-series anomaly detection [30,31].

At the fog, TCN-AE inference is executed only on windows that pass edge gating, reducing effective workload compared to processing all windows centrally. The calibration-driven thresholds (τ_e, τ_f) provide an operational control knob for balancing detection quality against processing and communication cost.

This hierarchical execution strategy supports scalable deployment across many charging sites while maintaining bounded latency and resource usage.

5 Experimental Setup

5.1 Dataset and Experimental Setup

The proposed framework is evaluated using real-world electric vehicle (EV) charging load data provided by the Korea Electric Power Corporation. The dataset consists of hourly aggregated charging loads collected over multiple months. To reflect realistic abnormal behaviors observed in EV charging infrastructures, we inject diverse anomaly patterns, including point-wise spikes, collective shifts, gradual temporal drifts, and missing data segments.

The dataset is chronologically split into training, validation, and test sets to prevent temporal leakage. All models are trained exclusively on normal data, while evaluation is conducted on the test set containing injected anomalies.

5.2 Baselines and Comparison Models

We compare the proposed hierarchical approach against two representative unsupervised baselines:

- **Statistical baseline**, which detects anomalies using context-aware z-score deviations.
- **Isolation Forest**, a widely adopted classical anomaly detection method based on random partitioning.

The fog-only TCN-AE serves as a strong deep learning baseline, representing the upper bound of detection performance without routing constraints. This comparison isolates the effect of hierarchical routing from the underlying model capacity.

For this revision, all reported numbers are directly synchronized from the exported run artifacts (`metrics.json`) under the same dataset split and preprocessing pipeline. To strengthen statistical rigor, the main fog-only and corrected hierarchical configurations are evaluated over five random seeds $(3, 13, 23, 33, 43)$ on the fixed chronological split. We report mean ± standard deviation, paired t-tests over matched seeds, and 95% confidence intervals for the paired differences between fog-only TCN-AE and the selected routed operating point. The statistical comparison focuses on F1-score, average delay, false alarms per day, fog keep rate, and load reduction, because these metrics directly represent the intended operating-point trade-off. Since the data horizon is chronological and fixed, this analysis should be interpreted as seed-level robustness under the same temporal evaluation protocol rather than as an independent multi-split validation. Extensive multi-split evaluation remains future work because the current study focuses on routing behavior under a fixed chronological deployment scenario.

6 Evaluation Metrics

To comprehensively assess detection performance and operational feasibility, we employ evaluation metrics at both window and event levels. Detection accuracy is measured using the **F1-score** and **PR-AUC**, which are well-suited for highly imbalanced anomaly detection scenarios.

From an operational perspective, we further report the **event detection rate**, **average detection delay (hours)**, and **false alarms per day**, which directly reflect the suitability of the proposed method for real-world EV charging infrastructures. System efficiency is quantified using the **fog window keep rate**, representing the proportion of windows forwarded to the fog layer after edge-level filtering. Let $\mathcal{G}$ be anomaly event groups $(g \geq 0)$, t_g^{start} the first positive window-end timestamp in group g, and t_g^{det} the first detected positive window-end timestamp. Then

$$\text{DetectedRate} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mathbb{I}\left[t_g^{\text{det}} \text{ exists}\right], \tag{24}$$

$$\text{AvgDelay} = \frac{1}{|\mathcal{G}_{\text{det}}|} \sum_{g \in \mathcal{G}_{\text{det}}} \left(t_g^{\text{det}} - t_g^{\text{start}}\right), \qquad \mathcal{G}_{\text{det}} = \{g \in \mathcal{G} \mid t_g^{\text{det}} \text{ exists}\}, \tag{25}$$

$$\text{FalseAlarms/day} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \sum_{i:\, t_i \in d} \mathbb{I}[y_i = 0 \wedge \hat{y}_i = 1], \tag{26}$$

and the fog keep rate is

$$\text{KeepRate}_{\text{fog}} = \frac{|\mathcal{I}|}{N_{\text{all windows}}}. \tag{27}$$

These event-level metrics are computed following Algorithm 3, which aggregates window-level predictions into operationally interpretable outcomes.

Algorithm 3 connects model outputs to operational interpretation. Instead of stopping at window-level correctness, it quantifies whether each event is detected, when it is first detected, and how many false alerts are generated per day. This step resolves a typical question: why models with similar window-level scores may still behave differently in practice. By grouping windows into events and measuring first-detection delay, the evaluation explicitly reflects operational responsiveness and alert burden, which are primary decision criteria for real-world monitoring systems.

6.1 Detection Performance

Table 1 summarizes the corrected end-to-end results. Among the learned models, fog-only TCN-AE remains the strongest pure detection reference, attaining the best F1-score and the lowest false-alarm burden. The statistical baseline achieves the highest aggregate F1-score and PR-AUC under the selected threshold, but does so with substantially higher alert burden and without workload-aware routing. Isolation Forest shows weak practical utility under the selected operating threshold despite its non-zero PR-AUC.

Table 1: Main performance comparison under corrected end-to-end evaluation. Fog-only, corrected hierarchical, and triage routing are reported as mean ± std over five seeds; the statistical baseline and Isolation Forest are single-configuration references.

Method	F1-Score	PR-AUC	Avg. Delay (h)	False Alarms/Day
Statistical baseline	0.674 ± 0.000	0.990 ± 0.000	3.0 ± 0.0	7.04 ± 0.00
Isolation Forest	0.234 ± 0.000	0.263 ± 0.000	126.0 ± 0.0	9.75 ± 0.00
Fog-only TCN-AE	0.547 ± 0.068	0.791 ± 0.079	93.6 ± 4.7	0.88 ± 0.59
Corrected hierarchical	0.455 ± 0.057	0.318 ± 0.013	75.6 ± 16.2	7.29 ± 0.00
Triage routing	0.469 ± 0.017	0.550 ± 0.068	102.6 ± 1.2	3.14 ± 0.42

The corrected hierarchical pipeline should therefore be interpreted as a systems operating point rather than a uniformly superior detector. Relative to fog-only TCN-AE, the selected binary-routing configuration shortens average detection delay from 93.6 to 75.6 h and reduces fog workload, but it also increases false alarms per day from 0.88 to 7.29 and lowers F1 from 0.547 to 0.455. Triage routing pushes workload reduction further, but shifts the operating point toward stronger evidence loss and does not provide a better main-line configuration.

The paired statistical analysis in Appendix A Table A2 supports this interpretation. The F1 change (-0.092, 95% CI $[-0.231, 0.047]$, $p = 0.139$) and delay change (-18.0 h, 95% CI $[-39.8, 3.8]$, $p = 0.084$) do not justify a claim of uniform detection superiority. However, the increase in false alarms per day is statistically clear ($+6.41$, 95% CI $[5.59, 7.23]$, $p < 0.001$), and the fog keep-rate decrease from 1.000 to 0.651 corresponds to a deterministic 34.9% load reduction over the fixed test window set. Thus, the results statistically support an operating-point shift: reduced fog workload and earlier alerts are obtained at the cost of higher alert burden and lower aggregate F1-score. Nevertheless, the seed-level pattern, including consistent workload reduction, consistent false-alarm increase, and delay reduction in four of five matched seeds, supports interpreting the selected routing policy as a stable operating-point shift rather than an isolated seed artifact.

6.2 Efficiency and Trade-Off Analysis

Fig. 2 confirms a consistent workload–performance trade-off as the edge threshold varies. Such trade-offs between detection performance and system efficiency are commonly observed in distributed monitoring systems. Across the binary-routing sweep, higher fog keep rates generally recover more detection evidence,

whereas more aggressive filtering suppresses workload at the cost of larger degradation in F1 and operating stability. The representative corrected hierarchical point corresponds to a fog keep rate of 0.651, which yields a 34.9% fog-load reduction while shortening average delay relative to fog-only inference.

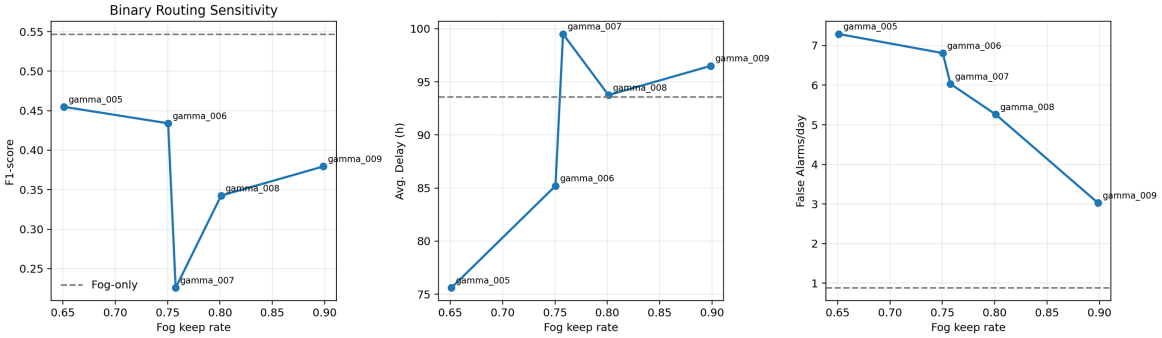

Figure 2: Binary-routing sensitivity under corrected evaluation. The three panels show how fog keep rate trades off with F1-score, average detection delay, and false alarms per day as the edge threshold varies.

To make the operating-point choice explicit, we apply a constrained Pareto selection rule to the binary-routing sweep. First, candidate thresholds are screened for Pareto efficiency with respect to average detection delay and false alarms per day. Event forwarded coverage, fog workload, and F1-score are then used as secondary operating-point selection criteria. Specifically, among Pareto-efficient candidates satisfying event forwarded coverage ≥0.90, we select the point with the minimum average detection delay, using false alarms per day and F1-score as tie breakers. This rule selects $\gamma = 0.05$, which is the corrected hierarchical operating point reported in Table 1. Appendix A Table A3 lists the candidate operating points and the selected configuration.

Table 2 clarifies why this trade-off arises. For the corrected hierarchical setting, the fog detector remains reasonably strong on the forwarded subset, attaining a forwarded-fog F1 of 0.692. Among forwarded windows, 45.3% are positive, whereas only 9.3% of filtered windows are positive. On the fixed test split, this corresponds to 163 positive windows retained for fog inference and 18 positive windows filtered out before fog processing. At the same time, event forwarded coverage remains high at 0.901, which makes this binary-routing point a representative operating point for the main paper. The marked decrease in PR-AUC nevertheless suggests that routing-induced filtering distorts the global score distribution seen by the end-to-end system, thereby degrading ranking-based metrics in addition to the thresholded operating point. The routing threshold can be interpreted as a system-level control variable that governs the balance between responsiveness and analytical depth.

The triage variant is useful as a supporting stress test rather than as the main proposed configuration. It reduces fog workload more aggressively, but only 30.7% of forwarded windows are positive, while 34.5% of filtered windows are positive. In absolute terms, the triage configuration retains 79 positive windows for fog inference and filters out 102 positive windows, lowering event forwarded coverage to 0.436. This confirms that more aggressive routing primarily amplifies evidence loss rather than improving detection performance. This analysis can be interpreted as a practical case study of how different routing configurations affect system behavior under operational constraints.

Table 2: Evidence-retention and workload diagnostics for the routing variants.

Method	Fog Keep Rate	Load Reduction (%)	Forwarded-Fog F1	Pos. Fraction in Forwarded	Pos. Fraction in Filtered	Event Forwarded Coverage
Corrected hierarchical	0.651 ± 0.000	34.9 ± 0.0	0.692 ± 0.092	0.453 ± 0.000	0.093 ± 0.000	0.901 ± 0.000
Triage routing	0.465 ± 0.000	53.5 ± 0.0	0.890 ± 0.056	0.307 ± 0.000	0.345 ± 0.000	0.436 ± 0.000

6.3 Detection Characteristics across Anomaly Patterns

The framework's behavior on different anomaly types illustrates the complementary roles of edge and fog layers in handling diverse irregular patterns. Although Table 1 reports overall aggregate metrics across multiple injected anomaly patterns (point spikes, sustained deviations, gradual drifts, and missing data blocks), the hierarchical mechanism is designed to handle each anomaly class differently based on its temporal signature. The following analysis describes the expected detection characteristics for each pattern type based on the system design.

For point-wise spikes and drops (Section 3.5), the edge-level MiLOF detector exhibits high sensitivity because isolated feature extremes create large local density ratios. These windows typically receive high edge scores and are filtered before fog processing when $\rho_i > \gamma$. This early suppression reduces fog workload on what are often transparent point anomalies, lowering communication and computation cost. The trade-off is that some point anomalies are treated as edge-dominant and do not reach the fog layer, which may reduce event-level detection rate if the anomaly is borderline.

For collective shifts and sustained deviations (Section 3.5), detection depends on the magnitude and duration of the shift relative to the training-set variability. Moderate shifts produce intermediate edge scores ($0.2 < \rho_i < \gamma$), allowing forwarding to the fog layer for temporal context. The fog TCN-AE, trained exclusively on normal windows, exhibits elevated reconstruction error when the shift is large enough to exceed the learned tolerance bounds. The reconstructive scoring strategy enables the fog layer to distinguish between anomalous magnitude shifts and normal temporal variability, particularly when the shift persists across multiple windows captured by the TCN's multi-scale receptive field.

For gradual drifts (Section 3.5), edge-level detection is inherently limited because micro-cluster centers adapt slowly over time. Consequently, the fog layer becomes the primary drift detector through its ability to recognize systematic errors in window-level reconstruction. The multi-scale dilated-convolution receptive field allows the TCN to capture trend-like deviations that accumulate over the window length L, making drift detection a strength of the fog-only baseline (Table 1, "Fog-only TCN-AE" row). The hierarchical pipeline's performance degradation relative to fog-only inference on drifts arises because aggressive edge filtering may suppress windows that exhibit mild drift signals, preventing the fog layer from accumulating sufficient temporal evidence.

For missing-data blocks (Section 3.5), the edge detector responds to irregularity in the imputed feature patterns, often triggering early filtering. However, some missing segments that are subtly imputed may not produce extreme edge scores. In such cases, the fog layer benefits from its trained tolerance for completeness signals: a sequence with missing-imputed values creates a distinct reconstruction signature that the model, trained on complete data only, recognizes as anomalous. By jointly addressing behavioral anomalies and data-quality issues under a unified framework, the system avoids separate preprocessing and imputation steps, aligning with recent evidence that missing patterns warrant explicit modeling rather than removal or forward-fill imputation [17–19].

The observed coupling among workload, detection rate, delay, and false alarms (Table 1) reflects the fundamental uncertainty in anomaly characterization: more aggressive edge filtering (lower γ) improves efficiency but increases the risk of filtering true anomalies that moderate edge scores incompletely represent. Conversely, permissive edge filtering (higher γ) preserves evidence but increases communication and fog-layer computation, potentially amplifying spurious responses to normal variability. This coupling motivates the calibration-driven approach to thresholding (τ_e, τ_f) rather than fixed rules, enabling site-specific tuning according to operational constraints.

6.4 Routing-Diagnostic Interpretation

The corrected results indicate that routing, rather than weakness of the fog detector itself, is the dominant source of system-level degradation. This conclusion follows from the gap between the forwarded-fog diagnostics in Table 2 and the final end-to-end metrics in Table 1. In both routing variants, the fog detector performs substantially better on the retained subset than the full system does on the complete timeline. Appendix A Fig. A1 visualizes this gap directly, while Appendix A Table A1 summarizes the route composition of the triage stress test. This phenomenon is consistent with prior observations in hierarchical filtering systems, where early-stage pruning may distort downstream decision distributions. Accordingly, the main contribution of the revised evaluation is not a claim of uniform superiority over fog-only inference, but a calibrated analysis of how routing changes workload, delay, alert burden, and retained anomaly evidence.

We further conduct a post-hoc oracle-routing diagnostic to isolate routing-induced information loss. The oracle forwards all positive/event-containing windows and then fills the remaining forwarding budget with normal windows so that its keep rate matches the learned routing policy. This diagnostic is not an implementable deployment policy because it uses ground-truth labels, and it should not be interpreted as showing that oracle routing improves detection delay. Because oracle routing forwards all event-containing windows using ground-truth labels, its metrics should be interpreted strictly as an upper-bound diagnostic rather than a realizable deployment result. Indeed, oracle routing has the same mean delay as fog-only inference (93.6 h), while the learned routing point has a shorter mean delay of 75.6 h because high edge-response windows trigger earlier alerts. This behavior is expected because oracle routing preserves all anomaly evidence and therefore removes the early edge-side suppression that shortens delay in the learned routing configuration. The diagnostic instead separates retained evidence from learned routing behavior: learned routing retains event coverage of 0.901, implying an information-loss rate of 0.099, whereas the oracle retains event coverage of 1.000 under the same keep-rate budget. The learned-oracle gap in Appendix A Table A4 and Fig. A2 therefore quantifies the portion of degradation attributable to routing-induced evidence loss.

We clarify that the term “drift” in our experimental setup refers to injected gradual anomaly patterns rather than distributional concept drift in the data-generating process. The current work does not explicitly evaluate adaptation under distributional shifts over time. Instead, the proposed framework assumes a relatively stable calibration regime, where thresholds (τ_e, τ_f) are fixed after validation. Extending the framework to handle true concept drift, including dynamic recalibration or online adaptation, remains an important direction for future work [32].

7 Conclusion

This paper presented a hierarchical edge–fog anomaly detection framework for electric vehicle (EV) charging infrastructures, designed to balance detection accuracy and system efficiency. By integrating lightweight edge-level filtering with a fog-level TCN-based detector, the proposed approach reduces fog processing load while enabling configurable anomaly-detection behavior under operational constraints.

Under end-to-end evaluation on real-world EV charging load data, the hierarchical framework is best interpreted as a configurable systems operating point rather than as a uniformly superior detector. The representative binary-routing configuration reduces fog workload by 34.9% and shortens average detection delay from 93.6 to 75.6 h relative to fog-only inference, but it also increases false alarms per day and lowers F1-score. Sensitivity analysis over routing thresholds reveals a consistent workload–performance frontier, while routing diagnostics show that information loss induced by filtering is the primary cause of the end-to-end performance shift.

These findings indicate that incorporating edge intelligence into large-scale EV charging monitoring is practical, but that the routing policy must be calibrated with explicit attention to retained anomaly evidence. This work reframes hierarchical anomaly detection not as a model-level improvement, but as a system-level design problem centered on controllable trade-offs.

Future work will explore adaptive thresholding and online learning, including concept-drift-aware recalibration, to improve robustness under evolving load patterns [32]. In addition, extending the proposed framework to multi-site charging networks and integrating domain-specific constraints remain promising directions for real-world deployment.

Acknowledgement: Not applicable.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request. The source code used in this study is available via Zenodo (doi:10.5281/zenodo.19928369).

Ethics Approval: Not applicable. This study did not involve human or animal subjects.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A Supporting Routing Diagnostics

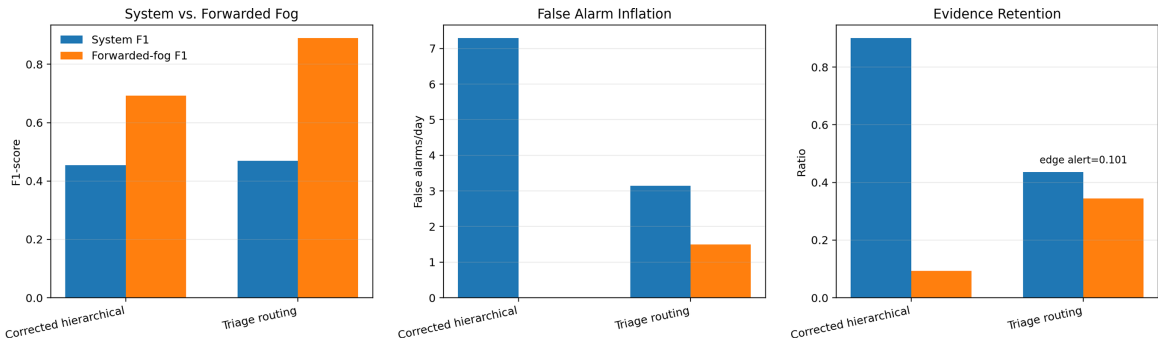

Figure A1: Comparison between end-to-end system metrics and forwarded-subset fog diagnostics for the corrected hierarchical and triage routing variants. The gap between system F1 and forwarded-fog F1 indicates that routing-induced evidence loss, rather than fog-model weakness alone, drives the final performance degradation.

Table A1: Detailed routing statistics for the triage stress-test configuration on the fixed test split. Count columns are identical across the five seeds.

Method	Total	Fog Forward	Drop	Edge Alert	Fwd. Pos. Count	Filt. Pos. Count	Pos. Frac. in Fwd.	Pos. Frac. in Filt.	Event Fwd. Cov.
Triage routing	553	257	240	56	79	102	0.307 ± 0.000	0.345 ± 0.000	0.436 ± 0.000

Table A2: Paired statistical tests between fog-only TCN-AE and the corrected hierarchical operating point over five matched seeds. Differences are computed as corrected hierarchical minus fog-only. For fog keep rate and load reduction, the paired difference is identical across seeds because the fixed routing mask yields the same forwarded-window count on the fixed test split.

Metric	Fog-Only Mean	Corrected Hierarchical Mean	Mean Difference (95% CI)	Paired p-Value
F1-score	0.547	0.455	−0.092 [−0.231, 0.047]	0.139
Avg. delay (h)	93.6	75.6	−18.0 [−39.8, 3.8]	0.084
False alarms/day	0.88	7.29	6.41 [5.59, 7.23]	<0.001
Fog keep rate	1.000	0.651	−0.349 [−0.349, −0.349]	<0.001
Load reduction (%)	0.0	34.9	34.9 [34.9, 34.9]	<0.001

Table A3: Operating-point candidates from the binary-routing threshold sweep. Pareto efficiency is assessed with respect to average detection delay and false alarms per day; event forwarded coverage, fog workload, and F1-score are used as secondary selection criteria.

Candidate	γ	Selected	F1-Score	Avg. Delay (h)	False Alarms/day	Event Fwd. Cov.
$\gamma = 0.05$	0.05	Yes	0.455	75.6	7.29	0.901
$\gamma = 0.06$	0.06	No	0.434	85.2	6.81	1.000
$\gamma = 0.07$	0.07	No	0.226	99.5	6.03	1.000
$\gamma = 0.08$	0.08	No	0.343	93.8	5.26	1.000
$\gamma = 0.09$	0.09	No	0.379	96.5	3.03	1.000

Table A4: Post-hoc oracle routing diagnostic. Oracle routing is a controlled upper-bound diagnostic that forwards all positive windows while matching the learned routing keep-rate budget; it is not an implementable deployment policy. Because oracle routing uses ground-truth event labels for forwarding, its metrics should be interpreted only as an upper-bound diagnostic reference and not as a realizable deployment result.

Method	F1-Score	PR-AUC	Avg. Delay (h)	False Alarms/day	Fog Keep Rate	Event Fwd. Cov.	Info. Loss
Fog-only TCN-AE	0.547	0.791	93.6	0.88	1.000	1.000	0.000
Learned routing	0.455	0.318	75.6	7.29	0.651	0.901	0.099
Oracle routing	0.593	0.999	93.6	0.00	0.651	1.000	0.000

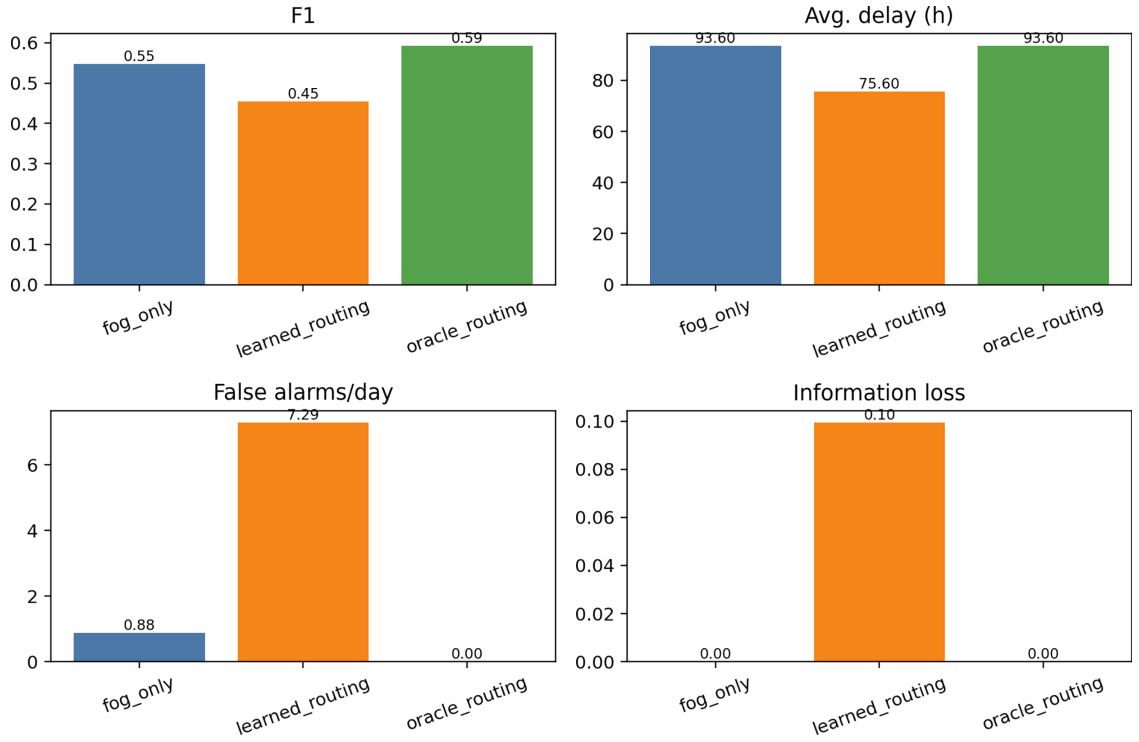

Figure A2: Oracle-routing diagnostic compared with fog-only and learned routing. The oracle is used to separate routing-induced information loss from fog-detector capacity; it is not used to claim improved detection delay. The near-perfect oracle metrics arise from label-aware forwarding and should therefore be interpreted only as a diagnostic upper bound for retained evidence analysis.

References

1. Kern D, Krauß C, Hollick M. Detection of anomalies in electric vehicle charging sessions. In: Proceedings of the 39th Annual Computer Security Applications Conference; 2023 Dec 4–8; Austin, TX, USA. p. 298–309. doi:10.1145/3627106.3627127.
2. Izquierdo Gómez P, Barragan Moreno A, Lin J, Dragičević T. Data-driven thermal modelling for anomaly detection in electric vehicle charging stations. In: 2022 IEEE Transportation Electrification Conference and Expo (ITEC); 2022 Jun 15–17; Anaheim, CA, USA. doi:10.1109/ITEC53557.2022.9813767.
3. Mitikiri SB, Srinivas VL, Pal M. Anomaly detection of adversarial cyber attacks on electric vehicle charging stations. e-Prime—Adv Electr Eng Electron Energy. 2025;11(15):100911. doi:10.1016/j.prime.2025.100911.
4. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv. 2009;41(3):1–72. doi:10.1145/1541880.1541882.
5. Blázquez-García A, Conde A, Mori U, Lozano JA. A review on outlier/anomaly detection in time series data. ACM Comput Surv. 2021;54(3):1–33. doi:10.1145/3444690.
6. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining; 2008 Dec 15–19; Pisa, Italy. p. 413–22. doi:10.1109/ICDM.2008.17.
7. Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl. 2016;60(1):19–31. doi:10.1016/j.jnca.2015.11.016.
8. Adhikari D, Jiang W, Zhan J, Rawat DB, Bhattarai A. Recent advances in anomaly detection in Internet of Things: status, challenges, and perspectives. Comput Sci Rev. 2024;54(4):100665. doi:10.1016/j.cosrev.2024.100665.
9. Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv:1607.00148. 2016.

10. Munir M, Siddiqui SA, Dengel A, Ahmed S. DeepAnT: a deep learning approach for unsupervised anomaly detection in time series. IEEE Access. 2019;7:1991–2005.
11. Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271. 2018.
12. Tuli S, Casale G, Jennings NR. TranAD: deep transformer networks for anomaly detection in multivariate time series data. Proc VLDB Endow. 2022;15(6):1201–14. doi:10.14778/3514061.3514067.
13. Xu J, Wu H, Wang J, Long M. Anomaly Transformer: time series anomaly detection with association discrepancy. arXiv:2110.02642. 2022.
14. Little RJA, Rubin DB. Statistical analysis with missing data. 3rd ed. Hoboken, NJ, USA: John Wiley & Sons; 2019.
15. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR. Pattern classification with missing data: a review. Neural Comput Appl. 2010;19(2):263–82. doi:10.1007/s00521-009-0295-6.
16. Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent neural networks for multivariate time series with missing values. Sci Rep. 2018;8(1):6085. doi:10.1038/s41598-018-24271-9.
17. Cao W, Wang D, Li J, Zhou H, Li L, Li Y. BRITS: bidirectional recurrent imputation for time series. Adv Neural Inf Process Syst. 2018;31:6776–86.
18. Xiao F, Fan J. Unsupervised anomaly detection in the presence of missing values. Adv Neural Inf Process Syst. 2024;37:138130–62. doi:10.52202/079017-4385.
19. Wang J, Du W, Yang Y, Qian L, Cao W, Zhang K, et al. Deep learning for multivariate time series imputation: a survey. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence; 2025 Aug 16–22; Montreal, QC, Canada. doi:10.24963/ijcai.2025/1187.
20. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing; 2012 Aug 17; Helsinki, Finland. p. 13–6.
21. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. IEEE Internet Things J. 2016;3(5):637–46.
22. Chiang M, Zhang T. Fog and IoT: an overview of research opportunities. IEEE Internet Things J. 2016;3(6):854–64.
23. Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30–9. doi:10.1109/mc.2017.9.
24. Ngo MV, Luo T, Chaouchi H, Quek TQS. Contextual-bandit anomaly detection for IoT data in distributed hierarchical edge computing. In: 2020 IEEE 40th International Conference on Distributed Computing Systems; 2020 Nov 29–Dec 1; Singapore. p. 1227–30. doi:10.1109/ICDCS47774.2020.00191.
25. Arias MB, Bae S. Electric vehicle charging demand forecasting model based on big data technologies. Appl Energy. 2016;183(2):327–39. doi:10.1016/j.apenergy.2016.08.080.
26. Xydas E, Marmaras C, Cipcigan LM, Jenkins N, Carroll S, Barker M. A data-driven approach for characterising the charging demand of electric vehicles: a UK case study. Appl Energy. 2016;162(3):763–71. doi:10.1016/j.apenergy.2015.10.151.
27. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data; 2000 May 16–18; Dallas, TX, USA. p. 93–104. doi:10.1145/335191.335388.
28. Salehi M, Leckie C, Bezdek JC, Vaithianathan T, Zhang X. Fast memory efficient local outlier detection in data streams. IEEE Trans Knowl Data Eng. 2016;28(12):3246–60. doi:10.1109/TKDE.2016.2597833.
29. Zhao H, Wang Y, Duan J, Huang C, Cao D, Tong Y, et al. Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining; 2020 Nov 17–20; Sorrento, Italy. p. 841–50. doi:10.1109/ICDM50108.2020.00093.
30. Li E, Zeng L, Zhou Z, Chen X. Edge AI: on-demand accelerating deep neural network inference via edge computing. IEEE Trans Wirel Commun. 2020;19(1):447–57. doi:10.1109/TWC.2019.2946140.
31. Sun Y, Chen T, Nguyen QVH, Yin H. TinyAD: memory-efficient anomaly detection for time-series data in industrial IoT. IEEE Trans Ind Inform. 2024;20(1):824–34. doi:10.1109/TII.2023.3254668.
32. Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G. Learning under concept drift: a review. IEEE Trans Knowl Data Eng. 2019;31(12):2346–63. doi:10.1109/tkde.2018.2876857.