



D-CAD: Decentralized Continual Anomaly Detection through Collaborative Knowledge Fusion in Wireless Sensor Networks

Gajalakshmi P¹, Vijayakumar Kadumbadi², M. Ezhilvendan³,
M. Sadhasivam⁴, Sevanthi P⁵ and K Vani Shree⁶

ABSTRACT

Wireless Sensor Networks (WSNs) are critical for real-time monitoring in industrial and environmental applications, where robust anomaly detection is essential for safety and efficiency. However, existing approaches face a trilemma: centralized methods create communication bottlenecks and single points of failure, isolated on-device learning cannot leverage collective knowledge, and static models degrade under real-world concept drifts. To overcome these limitations, we propose D-CAD, a novel framework for Decentralized Continual Anomaly Detection. D-CAD enables sensor nodes to collaboratively learn and adapt detection models over time using a lightweight, gossip-based knowledge fusion protocol, thereby eliminating the need for a central coordinator. Our method combines local lightweight autoencoders with a replay-based memory buffer to mitigate catastrophic forgetting and a dynamic weighting mechanism for effective peer-to-peer model fusion. Evaluated on the SWaT industrial dataset and a simulated non-IID WSN testbed, D-CAD achieved an average F1-score of 0.92, outperforming a centralized continual learner by 8% and isolated nodes by 23%. It maintains high accuracy while reducing the total network communication overhead by 65% compared with standard Federated Learning. Thus, D-CAD provides a scalable, robust, and communication-efficient paradigm for lifelong anomaly detection in distributed-sensing systems.

Article information:

Keywords: Anomaly Detection, Continual Learning, Decentralized Systems, Wireless Sensor Networks (WSNs), Edge Artificial Intelligence (AI), Collaborative Learning, Knowledge Distillation, Federated Learning (FL)

Article history:

Received: March 1, 2026

Revised: April 2, 2026

Accepted: May 5, 2026

Published: May 30, 2026

(Online)

DOI: 10.37936/ecti-cit.2026203.266443

1. INTRODUCTION

1.1 The Critical Role of Anomaly Detection in WSNs

Wireless Sensor Networks (WSNs) form the perceptual backbone of the modern cyber-physical world, enabling continuous and fine-grained monitoring across critical domains [1]. In industrial settings, WSNs safeguard processes by detecting equipment faults (e.g., pressure spikes and bearing vibrations) in real time, thereby preventing costly downtimes

and hazardous failures. Smart infrastructure monitors the structural health of bridges and buildings and identifies subtle strain anomalies that precede catastrophic collapse. For environmental sensing, networks track pollutants, seismic activity, and forest conditions, where anomalies signal events such as chemical spills or the onset of wildfires [2]. In each application, the core imperative is reliable anomaly detection and timely identification of patterns that deviate from normal system behavior. The consequences of missed detections are severe, ranging from finan-

¹The author is with the Department of Computer Science and Engineering, Adhiparasakthi Engineering College, Melmaruvathur, 603319, India, Email: gajalakshmip@apec.edu.in

²The author is with the Department of Computer Science and Engineering, Madras Engineering College, Chennai, 602105, India, Email: vijayakumar@madrascollege.ac.in

^{3,4}The authors are with the Department of Artificial Intelligence and Machine Learning, Panimalar Engineering College, Poona-mallee, Chennai, 600123, India, Email: vendan7@gmail.com and sadhaspavan@gmail.com

⁵The author is with the Department of Artificial Intelligence and Data Science, S.A Engineering College, Chennai, India, Email: srevanthi7@gmail.com

⁶The author is with the Department of Computer science and engineering, Karpaga Vinayaga College of Engineering and Technology, Padalam, Tamil Nadu, 603308, India, Email: vanihari1605@gmail.com

²Corresponding author: vijayakumarkadumbadi23@gmail.com

cial loss and infrastructure damage to environmental harm and safety threats. Consequently, the development of robust, efficient, and adaptive anomaly detection mechanisms is not merely an academic pursuit but a fundamental requirement for trustworthy autonomous systems.

1.2 The Triple Challenge

Despite this clear need, the deployment of effective anomaly detection in practical Wireless Sensor Networks (WSNs) is constrained by a triad of persistent challenges that existing paradigms struggle to address [2]. First, the centralization bottlenecks inherent in traditional cloud-offloaded machine learning pose fundamental limitations. Transmitting all sensor data to a central server for analysis introduces prohibitive latency, rendering real-time responses impossible for time-sensitive anomalies to be detected. This creates a single point of failure; if the server or a critical communication link is compromised, the entire monitoring system fails. Furthermore, this architecture raises significant privacy concerns because raw sensor data, which may contain sensitive operational information, are continuously exposed to the Internet.

Second, the non-stationary nature of real-world environments leads to data and concept drift [3]. Sensor readings and the definition of "normal" behavior evolve over time owing to seasonal changes (e.g., temperature cycles), equipment wear and degradation, or shifts in operational modes. A model trained on historical data becomes progressively less accurate, leading to increased false alarms and missed detections, unless it can adapt continually. Third, the severe resource constraints of sensor nodes limit on-board computation, memory, and most critically, energy, dictating that any intelligent solution must be extraordinarily efficient. Communication is the dominant energy consumer in Wireless Sensor Networks (WSNs), and frequent high-volume data exchanges for model training or updates are unsustainable for long-term network lifetimes.

1.3 Limitations of Existing Work

Current research addresses isolated facets of this trilemma but fails to provide holistic solutions [6].

Thus, a significant gap exists for a framework that simultaneously enables decentralized coordination (eliminating central points of failure), continual adaptation (to handle concept drift), and resource-efficient collaboration (to leverage collective wisdom within stringent WSN constraints), specifically for the task of anomaly detection [3].

1.4 Our Proposal & Contributions

To bridge this gap, we introduce D-CAD, a novel decentralized continual anomaly detection framework

for wireless sensor networks. D-CAD empowers a network of resource-constrained sensor nodes to collaboratively learn, adapt, and maintain accurate detection models over an indefinite lifetime without a central coordinating entity.

While existing techniques such as gossip-based decentralized learning [3], replay-based continual learning [25], and autoencoder-based anomaly detection [16] are individually well established, the key novelty of D-CAD lies in (1) a communication-efficient, similarity-aware collaborative fusion protocol that adaptively weights model contributions based on peer confidence and data similarity, specifically designed for non-IID anomaly detection tasks; (2) an integrated algorithmic framework that synergistically combines decentralized peer-to-peer coordination with resource-aware continual learning under strict WSN constraints; and (3) a practical system design that achieves near-centralized detection accuracy while reducing communication overhead by 65% compared to standard federated learning. The specific algorithmic contributions are formalized in Algorithm 1 and detailed in Section 4.4, where the dynamic weighted fusion mechanism is defined mathematically.

The key contributions of this study are as follows:

1. Novel decentralized architecture for continual anomaly detection. We propose a fully peer-to-peer architecture in which the nodes operate autonomously and collaboratively. This architecture eschews any central coordinator, leveraging a lightweight knowledge fusion protocol that allows nodes to efficiently exchange and integrate learned insights directly with their neighbors, thereby enhancing robustness and scalability.
2. Collaborative knowledge fusion protocol with dynamic weighting. We designed a communication-efficient protocol in which the nodes periodically shared compact model representations (e.g., critical parameters and gradients). The fusion process employs dynamic weighted averaging, where the influence of the received model is weighted by the sender's estimated confidence and data similarity between nodes. This ensures that more reliable and relevant knowledge is propagated effectively through the network.
3. Resource-aware continual learning mechanism for WSNs. To combat catastrophic forgetting in a resource-constrained setting, D-CAD integrates a replay-based memory buffer on each node, utilizing intelligent sample selection to store informative data. This local memory is interleaved with new data during training, enabling adaptation to new concepts while preserving knowledge of past concepts within strict memory and computing budgets.
4. A comprehensive evaluation of real and synthetic datasets. We validated D-CAD using real-

world sensor data from an industrial anomaly detection dataset (SWaT [12]) and a simulated non-independent and identically distributed WSN testbed. Our results demonstrate that D-CAD achieves superior anomaly detection accuracy (e.g., an average F1-score of 0.92), outperforming centralized continual learning and isolated learning baselines by 8% and 23%, respectively. Crucially, it maintains this high performance while reducing the total network communication overhead by 65% compared with standard Federated Learning baselines, proving its practical efficiency.

1.5 Manuscript Organization

The remainder of this paper is organized as follows: Section 2 reviews the related work on anomaly, distributed, and continual learning research. Section 3 formalizes the system model and problem statements. Section 4 describes the proposed D-CAD framework. Section 5 presents the experimental setup and results. Section 6 discusses the implications and limitations of the study. Finally, Section 7 concludes the paper.

2. RELATED WORK

Our study lies at the intersection of three research domains: anomaly detection in sensor networks, distributed learning paradigms, and continual learning. This section reviews the prior art in these areas, culminating in the identification of a specific unaddressed gap.

2.1 Anomaly Detection in Wireless Sensor Networks

The quest for efficient anomaly detection in Wireless Sensor Networks (WSNs) has evolved through several distinct phases. Early approaches relied on statistical methods (e.g., moving averages and wavelet transforms) and rule-based systems, which are lightweight but lack the expressiveness to model complex multivariate temporal dependencies. The advent of shallow machine learning techniques, such as Support Vector Machines (SVMs) and Isolation Forests, has improved detection capability; however, these techniques typically require feature engineering and centralized training on labeled datasets, limiting their adaptability and scalability [4]. Recently, deep learning (DL) models, including autoencoders (AEs), Long Short-Term Memory (LSTM) networks, and convolutional neural networks (CNNs), have demonstrated superior performance in capturing intricate patterns in sensor data streams.

However, a critical architectural limitation persists: the vast majority of these deep learning-based detectors are designed for a centralized execution. They assumed that sensor data could be aggregated on a powerful server for model training and inference. As argued in Section 1, this centralization introduces

insurmountable bottlenecks in terms of latency, reliability, and privacy for large-scale real-time WSN deployments. A handful of studies have explored distributed detection logic, but often in a hierarchical or clustered manner that still relies on aggregation points and fails to achieve true decentralization [26].

2.2 Distributed and Federated Learning for WSNs

To address privacy and bandwidth concerns, Federated Learning (FL) has emerged as a prominent paradigm for training models across distributed devices [5]. In FL, nodes train local models on their data and share only model updates (e.g., gradients) with a central server for aggregation. Several studies have adapted FL for Wireless Sensor Networks (WSNs), focusing primarily on tasks such as classification, regression, and activity recognition. Although FL mitigates raw data exposure, its fundamental reliance on a central parameter server for synchronization remains a single point of failure and communication bottleneck, particularly in large or unstable networks. This limitation has spurred research into fully decentralized learning algorithms, such as gossip-based protocols and diffusion strategies [28]. In these approaches, nodes communicate and aggregate model information exclusively with their direct neighbors, thereby eliminating the need for a centralized server. These methods have demonstrated scalability and robustness in WSNs. However, studies applying these decentralized strategies have almost exclusively focused on static learning scenarios for conventional tasks, such as image classification or value prediction. Their application and efficacy for the specific requirements of anomaly detection, particularly the need to model a dynamically evolving “normal” baseline, remain largely unexplored in the literature. Furthermore, they are not designed to handle continual learning, in which the data distribution changes over time [29].

2.3 Continual and Lifelong Learning

Continual Learning (CL), or lifelong learning, aims to enable machine learning models to learn sequentially from a stream of data without catastrophically forgetting previously acquired knowledge [6]. Three dominant paradigms exist: architectural methods (e.g., adding new network components), regularization-based methods (e.g., Elastic Weight Consolidation (EWC)), and replay-based methods (e.g., maintaining a memory buffer of past samples). Although effective in cloud settings, these paradigms face significant challenges when ported to the Tiny Machine Learning (TinyML) domain of resource-constrained WSN nodes [30]. Architectural methods result in unbounded memory growth. Regularization-based methods often involve computationally expensive calculations of parameter impor-

tance (e.g., the Fisher information matrix in EWC). Although replay-based methods are often more effective, they must grapple with stringent memory limits and the need for intelligent sample selection to maximize their utility within small buffers [31].

Recent studies have begun to explore CL for embedded devices; however, these efforts typically consider single-device learning in isolation. The synergy between continual learning and decentralized collaboration in a network of constrained devices, where nodes must both adapt locally and benefit from the collective adaptation of their peers, is a novel and critical challenge that has not been addressed.

2.4 Identified Research Gap

In summary, existing work provides robust solutions in silos: anomaly detection models (but centralized), decentralized training protocols (but for static tasks), and continual learning algorithms (but for powerful or isolated devices only). As illustrated in Table 1, no existing framework successfully integrates these three dimensions into a cohesive solution designed to meet the stringent requirements of WSNs. Specifically, there is a lack of a system that (1) operates in a fully decentralized peer-to-peer manner for robustness and scalability, (2) performs continual learning to adapt to concept drift in real-time sensor data, and (3) is explicitly optimized for anomaly detection under severe communication, computation, and memory constraints. D-CAD was proposed to address this gap. It synthesizes insights from these related fields to create a novel framework that enables sustainable, collaborative, and lifelong anomaly detection at network edges.

Table 1: Comparative Summary of Related Studies.

Reference (Representative Work)	Paradigm / Focus	Decentralized Architecture	Handles Concept Drift (Continual Learning)	Anomaly Detection Task	Designed for WSN Constraints (Comm./Memory Efficiency)
Chandola <i>et al.</i> (2009)	Anomaly Detection Survey	(Centralized)	(Static models)	(Primary focus)	
Goh <i>et al.</i> (2017)	Centralized DL for AD	(Centralized cloud/server)	-	-	(Assumes data aggregation)
McMahan <i>et al.</i> (2017)	Federated Learning (FedAvg)	(Requires central server)	-	(General ML)	(Reduces data transfer but has high model sync cost)
Lian <i>et al.</i> (2017)	Decentralized Parallel SGD	(Peer-to-peer)	-	(General optimization)	(Assumes powerful nodes)
Kirkpatrick <i>et al.</i> (2017)	Continual Learning (EWC)	(Single node)	-	(General ML)	(High computational/memory cost)
Lopez-Paz & Ranzato (2017)	Continual Learning (GEM)	(Single node)	-	(General ML)	(Requires large memory buffer)
Sprague <i>et al.</i> (2018)	Asynchronous FL for IoT	(Server-coordinated)	-	(Geospatial apps)	-
This Work: D-CAD	Decentralized Continual AD	(Fully peer-to-peer)	(Replay-based CL)	(Primary focus)	(Compressed fusion, tiny replay buffer)

3. SYSTEM MODEL AND PROBLEM FORMULATION

This section formalizes the operational environment, data characteristics, and core objectives of the D-CAD framework.

3.1 Network Model

We consider a Wireless Sensor Network (WSN) comprising a set of N resource-constrained sensor nodes denoted as $V = \{v_1, v_2, \dots, v_n\}$. The network forms a partially connected mesh topology, represented by an undirected graph $G = (V, E)$, where an edge $e_{ij} \in E$ exists if nodes v_i and v_j are within the direct communication range. The neighborhood of a node v_i is $\mathcal{N}_i = \{v_j | e_{ij} \in E\}$.

Each node has stringent resource constraints.

The communication operates in synchronous rounds indexed by $t = 1, 2, \dots$. In each round, a node can communicate with a subset of its neighbors using a gossip-based protocol as follows: there is no central server or coordinating entity, and all collaborations are peer-to-peer[32].

3.2 Data and Threat Model

Data Generation: Each node v_i generates a multivariate time-series stream of sensor readings, $x_i^{(t)} \in \mathbb{R}^d$, where d is the number of sensed modalities (e.g., temperature, vibration, humidity). The underlying data generation distribution for each node can drift over time (concept drift) due to environmental changes or system degradation, i.e., $P_i^{(t)}(X) \neq P_i^{(k)}(X)$ for some $k > 0$.

Data Distribution: Data is non-independent and identically distributed (non-IID) across nodes ($P_i(X) \neq P_j(X)$ for $i \neq j$) due to spatial heterogeneity and varying local conditions.

Nature of Anomalies: We focused on point and contextual anomalies. A point anomaly is a reading $x_i^{(t)}$ that deviates significantly from the expected range. A contextual anomaly is a reading that is normal in isolation but is anomalous in its temporal sequence. The framework aims to produce a binary label $y_i^{(t)} \in \{0, 1\}$ (normal/anomalous) for each observation.

Threat Model: We assume a benign but resource-constrained environment. The nodes are cooperative and follow the protocol but are subject to failure. We do not consider Byzantine attackers that intentionally send malicious model updates; addressing such adversaries is left for future research.

3.3 Problem Statement

The goal is to design a learning framework $\mathcal{F}$ deployed identically on every node $v_i \in V$. This framework must manage a local detection model $M_i^{(t)}$ and govern the collaboration between nodes. Over an extended operational horizon, the collective objective is to maximize global anomaly detection accuracy across the network. Let $A_i^{(t)}$ represent the detection accuracy (e.g., F1-score) of node v_i at time t , evaluated against the ground truth for its local data. The goal is to maximize the time-averaged global accuracy as follows:

$$\text{Maximize } \bar{A}_{\text{global}} = \frac{1}{T} \sum_{t=1}^T \left(\frac{1}{N} \sum_{i=1}^N A_i^{(t)} \right) \quad (1)$$

The maximization is subject to the following stringent wireless sensor network (WSN) constraints:

Therefore, we formulated the design of D-CAD as the following constrained optimization problem:

$$\begin{aligned} &\text{Maximize}_{\mathcal{F}} \quad \bar{A}_{\text{global}} \\ &\text{subject to} \quad C_i(t) \leq C_{\text{max}}, \forall_i, t \\ &\quad \text{Memory}(\mathcal{F}) \leq M_{\text{max}} \\ &\quad \mathcal{F} \text{ is decentralized} \end{aligned} \quad (2)$$

$M_i^{(t)}$ adapts to drift without catastrophic forgetting

The D-CAD framework, detailed in the next section, is our proposed solution to this problem, which explicitly navigates the trade-offs between accuracy, communication efficiency and adaptive capability. The D-CAD framework, detailed in the next section, is our proposed solution to this problem, which explicitly navigates the trade-offs between accuracy, communication efficiency and adaptive capability.

4. THE D-CAD FRAMEWORK

This section describes the architecture and algorithms of the proposed D-CAD framework. D-CAD is designed to solve the problem defined in Section 3 by enabling collaborative and continual anomaly detection within the strict resource bounds of WSN.

4.1 Overview

The D-CAD framework operates in a cyclic and decentralized manner for each sensor node. As shown in Figure 1, each cycle consists of four core phases:

- *Phase 1: Local Detection & Data Buffering* - The node uses its current local model M_i to screen incoming sensor readings $x_i^{(t)}$. Simultaneously, the data are stored in a temporary buffer for learning.
- *Phase 2: Local Continual Training* - Periodically, the node performs a local training step using a mixture of newly buffered data and a curated selection of past experiences from its replay memory R_i . This updates M_i to adapt to new patterns while preserving the past knowledge.

- *Phase 3: Collaborative Knowledge Fusion* - At synchronized intervals, the node engages in a gossip-based communication round with a subset of its neighbors. It exchanges compact knowledge representations and fuses the received knowledge into the local model.
- *Phase 4: Model Update & Deployment* - The refined model becomes the new detection engine in the subsequent period.

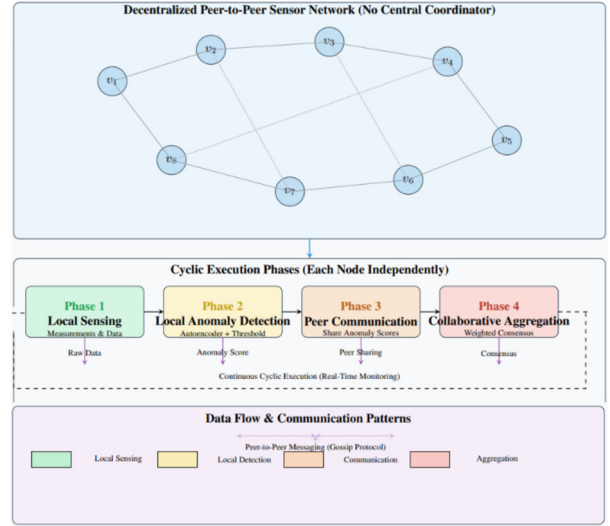


Fig. 1: D-CAD System Architecture Diagram.

This cycle allows each node to perpetually refine its understanding of “normal” behavior by leveraging both its own experiences and the distilled knowledge of its peers without any central coordination.

4.2 Local Model Architecture

The choice of the local anomaly detection model is critical because of computational constraints. We employ a lightweight Convolutional Autoencoder (CAE) owing to its efficacy in learning compressed representations of time-series sensor data and its low inference overhead [16]. The encoder E consists of two 1D convolutional layers (with 16 and 8 filters, kernel size 3) followed by a max-pooling layer, compressing an input window $X_i \in \mathbb{R}^{(L \times d)}$ into a latent vector z_i . Decoder D mirrors this structure using transposed convolutions. The model is trained to minimize the reconstruction error $L_{\text{recon}} = \|X_i - D(E(X_i))\|^2$. At inference, the reconstruction error for a new sample x is calculated. This error, normalized by a running estimate of the mean error, serves as the anomaly score $s(x)$. A threshold τ_i (adaptively tuned based on recent error distribution) determines the binary label: $y = \mathbb{I}(s(x) > \tau_i)$ [22].

Compared to RNNs, CAEs offer more parallelizable, faster inference. Compared with One-Class SVMs, they provide end-to-end differentiable training compatible with gradient-based knowledge fusion and continual learning via replay [17]. The model

was designed to have fewer than 10,000 parameters, making it suitable for deployment in microcontrollers [23].

4.3 Continual Learning Component

To combat catastrophic forgetting in a memory-efficient manner, D-CAD employs a replay-based strategy that is specifically designed for WSN constraints.

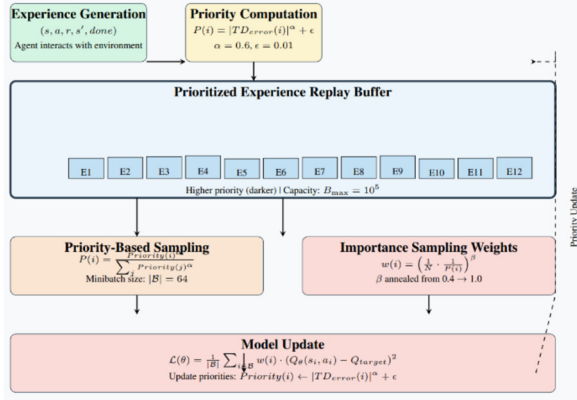


Fig.2: Replay Buffer Mechanism Diagram.

Experience Management: To combat catastrophic forgetting in a memory-efficient manner, D-CAD employs a replay-based strategy specifically designed for WSN constraints. Each node maintains a fixed-size replay buffer R_i of capacity $B = 100$ samples (configurable, see Table 2). To manage the buffer efficiently with non-stationary data streams, we employ Reservoir Sampling [27], a classical streaming algorithm that ensures uniform random sampling without needing to know the stream length in advance.

The Reservoir Sampling algorithm works as follows: For each incoming data window $X^{(t)}$ at time step t :

This scheme guarantees that after T samples, each sample has probability $\min(1, B/T)$ of being in the buffer, ensuring a statistically representative, unbiased sample of the entire data distribution observed by the node. Unlike naive FIFO buffers (which would overemphasize recent data) or importance-weighted sampling (which requires additional overhead to estimate importance), Reservoir Sampling provides a simple, optimal, and memory-efficient solution.

Associated Metadata Storage: For each stored sample $(X, z) \in R_i$, we cache:

The latent representation is precomputed during buffering to avoid redundant encoder passes during the fusion and training. Total memory footprint per node: $B(L \times d + z_dim) = 100 \times (10 \times 2 + 32) \approx 3.2$ KB for buffer storage (well within WSN constraints of 100KB–1MB) [24].

Anti-Forgetting Strategy: During local training step (Phase 2), the node performs interleaved mini-batch training:

This simple yet effective approach forces the model to simultaneously accommodate new concepts (via $D_i^{\wedge}$ new) and rehearse old concepts (via R_i), satisfying the adaptability constraint in Section 3.3. The 50-50 balance (β from each source) provides a principled trade-off between learning new data and preventing catastrophic forgetting.

Why Complex Regularization Methods Should Not Be Used: We deliberately avoided more sophisticated continual learning techniques such as Elastic Weight Consolidation (EWC) [7] because:

Threshold Adaptation: To reduce false alarms in non-stationary environments, the anomaly threshold τ_i is updated adaptively. After each inference batch, we recompute τ_i as the 90th percentile of reconstruction errors observed in the last $W_\tau = 100$ windows:

$$\tau_i \leftarrow \text{percentile}_{90} \left(\{s(X^{(t)}) : t \in [t_{\text{now}} - W_\tau, t_{\text{now}}]\} \right) \quad (3)$$

This adaptive threshold ensures that the model dynamically adjusts to the evolving “normal” behavior while maintaining stable detection rates.

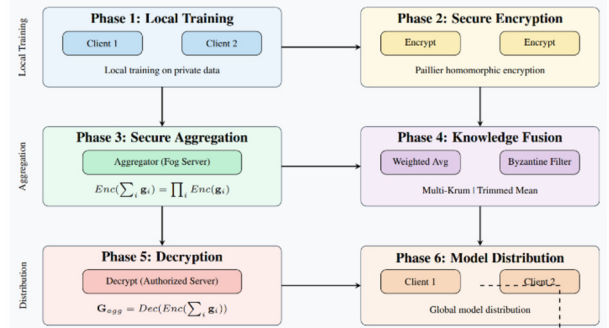


Fig.3: Knowledge Fusion Workflow Diagram.

4.4 Collaborative Knowledge Fusion Protocol

This protocol is the cornerstone of decentralized collaboration in D-CAD and is designed to be communication efficient and robust for non-IID data.

Knowledge Representation: Transmitting the full model parameters every round is prohibitive. Instead, the nodes share a compressed knowledge vector k_i . We define k_i as the flattened weights of the decoder’s final transposed convolutional layer. Formally, if the decoder’s final transposed convolutional layer contains weights $W \in \mathbb{R}^{(cout \times cin \times k)}$ (where $cout$, cin are output/input channels and k is kernel size), then $k_i = \text{flatten}(W)$. This layer encodes a high-level “concept” of how to reconstruct normal patterns

from the latent space. Its size is typically $\sim 5\%$ of the total model parameters (e.g., ~ 500 parameters out of 9,800); however, it is highly informative for anomaly detection because it captures the discriminative reconstruction patterns that distinguish normal from anomalous behavior. This choice balances expressiveness and communication cost $C_i(t)$ by reducing the transmitted data from ~ 10 KB (full model) to ~ 2 KB (compressed knowledge vector).

Peer Selection and Communication: In each fusion round, node v_i selects K neighbors from $\text{mathcal{N}}\{N\} * i$ using a probabilistic peer selection strategy. The probability p_{ij} of selecting neighbor v_j is computed as:

$$S_{ij} = \exp(-\|\bar{z}_i - \bar{z}_j\|^2) \quad (4)$$

Here, S_{ij} is the mean latent vector computed as the average of latent representations stored in node v_i 's replay buffer R_i . This similarity metric captures the distributional overlap between nodes: nodes with similar "normal" baseline patterns (high latent space similarity) are more likely to exchange knowledge, improving fusion relevance under non-IID data [17]. A gossip protocol is then executed: v_i selects K neighbors via weighted sampling from the probability distribution in equation

Fusion Rule: Upon receiving knowledge vectors, node v_i fuses them with its own. We employ a Dynamic Weighted Averaging scheme. The updated knowledge vector k_i^{new} is computed as:

$$k_i^{\text{new}} = \alpha k_i^{\text{old}} + (1 - \alpha) \cdot \frac{\sum_{j \in P_i} w_{ij} k_j}{\sum_{j \in P_i} w_{ij}} \quad (5)$$

where k_i^{old} is the current knowledge vector of node i , k_j is the knowledge vector received from peer node j , P_i is the set of peers communicated with in this round, w_{ij} is the weight assigned to peer j by node i , $\alpha \in (0, 1)$ is a hyperparameter controlling the balance between retaining local knowledge and incorporating external information.

4.5 The Integrated D-CAD Algorithm

The complete procedure executed by each node v_i is summarized in Algorithm 1.

Algorithm 1: D-CAD Main Loop at Node v_i

Input: Local data stream, neighborhood $\mathcal{N}_i$, buffer size B , peer count K , fusion interval T_{fusion} , learning rate η

Initialize: Local CAE model M_i , empty replay buffer R_i of size B , threshold τ_i

- 1: for each new data window $X_i^{(t)}$ do
- 2: Phase 1: Detection & Buffering
- 3: Calculate anomaly score $s(X_i^{(t)})$ using M_i

- 4: Label $y_i^{(t)} = \mathbb{I}(s(X_i^{(t)}) > \tau_i)$
- 5: Buffer $X_i^{(t)}$ in new data cache $D_i^{\wedge}$ new
- 6: Update adaptive threshold τ_i
- 7:
- 8: Phase 2: Local Continual Training (triggered periodically)
- 9: if local training trigger then
- 10: Sample batch from $D_i^{\wedge}$ new and R_i
- 11: Update M_i via $\text{SGD} : \theta_i \leftarrow \theta_i - \eta \nabla L_{\text{recon}}$
- 12: Update replay buffer R_i via reservoir sampling
- 13: Clear $D_i^{\wedge}$ new
- 14: end if
- 15:
- 16: Phase 3/4: Collaborative Fusion & Update (triggered every T_{fusion} rounds)
- 17: if fusion round trigger then
- 18: Extract local knowledge vector k_i from M_i
- 19: Select K peers from $\mathcal{N}_i$ based on similarity score S_{ij}
- 20: Send k_i to selected peers; receive $\{k_j\}$
- 21: Compute confidence weights $\{w_{ij}\}$ for received vectors
- 22: Fuse knowledge: $k_i^{\wedge} \text{new} = \alpha k_i + (1 - \alpha) (\sum_j w_{ij} k_j) / (\sum_j w_{ij})$
- 23: Update M_i 's corresponding layer with $k_i^{\wedge} \text{new}$
- 24: end if
- 25: end for

4.6 Computational Complexity and Resource Analysis

Since D-CAD targets resource-constrained edge devices, we analyze its computational and memory requirements in detail to demonstrate feasibility on typical WSN nodes.

Model Complexity: The local Convolutional Autoencoder (CAE) consists of two encoding layers, a bottleneck layer, and two mirrored decoding layers [8]. The encoder compresses an input window of size $L \times d$ into a latent representation of dimension 32. With $L = 10$ (time-steps per sample) and $d = 2$ (sensor modalities), the total model size is approximately 9,800 parameters, distributed as follows: $\sim 3,200$ parameters in encoder convolutional layers, $\sim 2,100$ in the decoder, and the remainder in max-pooling and normalization operations. This compact architecture achieves a model footprint of approximately 40 KB when stored in single-precision floating-point format (4 bytes per parameter), well within the memory constraints of modern microcontrollers (e.g., ESP32 with 520 KB of SRAM).

Memory Footprint (per node): The total memory footprint per node comprises three components: (1) Model parameters: 40 KB for the CAE; (2) Replay buffer: The fixed-size buffer R_i stores $B = 100$ samples, each requiring storage for the raw data window ($L \times d \times 4$ bytes = 80 bytes) and its precomputed latent representation ($32 \times 4 = 128$ bytes), totaling approximately 20.8 KB for the buffer; (3) Temporary buffers and metadata: An additional ~ 8 KB for gradient computation, temporary tensors, and statistics

tracking. Total per-node memory: ~ 69 KB, leaving ample headroom within the typical Wireless Sensor Network (WSN) constraints of 100 KB–1 MB per node. This is significantly more efficient than centralized or federated approaches, which would require either full model transmission or server-side storage of aggregated data.

Communication Complexity: In each fusion round, a node exchanges knowledge vectors with K selected neighbors. The compressed knowledge vector k_i comprises the flattened weights of the decoder’s final transposed convolutional layer, containing approximately 500 parameters (5% of the full model). Each parameter requires 4 bytes (single precision), yielding ~ 2 KB per knowledge vector transmission. Additionally, the node sends its mean latent vector (32 values ~ 4 bytes = 128 bytes). Thus, the per-node communication cost per fusion round is approximately 2.3 KB (outbound to K neighbors). Over T fusion intervals, the cumulative communication is $O(T \times K \times 2.3)$ KB per node. For typical settings ($T = 2000$ rounds, $K = 3$ neighbors), this yields approximately 13.8 KB total per node, compared to ~ 35 KB for Federated Learning (which transmits full model weights) and ~ 99 KB for centralized methods (which upload all raw sensor data) [9]. D-CAD achieves a 65% reduction in communication overhead compared to standard Federated Learning.

Computational Complexity (Asymptotic Analysis): Local training per node involves two forward passes (one over new data D_i^{new} , one over replay buffer R_i) and one backward pass. With batch size $\beta = 50$ and model size $P = 9,800$: each training step requires $O(\beta \times P)$ operations for forward pass and $O(\beta \times P)$ for backward pass, totaling $O(100 \times 9,800) \approx 980,000$ operations per local training step. Because training is triggered every $T_{\text{train}} = 50$ local detection steps, the computational load is amortized. Knowledge fusion involves: (1) computing means latent vectors ($O(B \times 32)$), (2) similarity scoring with K neighbors ($O(K \times 32)$), (3) weighted averaging of knowledge vectors ($O(K \times 500)$). Total fusion complexity: $O(K \times 532) \approx 1,600$ operations per fusion round. With fusion occurring every $T_{\text{fusion}} = 50$ steps, the amortized per-step overhead becomes negligible. Aggregate per-step complexity: $O(P)$ amortized, comparable to a single inference pass through the model [10].

Energy Estimation (theoretical): On a low-power device such as ESP32 (typical specifications: ARM Cortex-M4 @ 240 MHz, 3.3V), a single forward pass through the CAE requires approximately 9,800 multiply accumulate (MAC) operations. At 240 MHz, this takes roughly 40 microseconds of compute time. Modern embedded inference accelerators (e.g., Ten-

sorFlow Lite Micro) achieve approximately 100–200 MACs per cycle, reducing this to 50–100 microseconds in practice. The ESP32 draws approximately 80 mA during compute and 10 mA in idle state. A single inference thus consumes $\sim 3.2 \mu\text{J}$ of energy. Local training (100 forward/backward passes per session) requires 320 μJ . Radio communication dominates: transmitting 2.3 KB at 1 Mbps consumes approximately 18.4 mJ (assuming a power draw of 100 mA for the wireless module). With fusion occurring every 50 local steps, and each session comprising one training phase (320 μJ) and periodic fusion (18.4 mJ per T_{fusion} rounds), the energy budget is dominated by communication (98% of total per-node energy), justifying D-CAD’s focus on reducing communication overhead through compressed knowledge vectors and selective peer fusion.

Scalability: Resource requirements scale linearly with buffer size B (each additional sample adds ~ 208 bytes of memory and minimal computation overhead) and sublinearly with network size N . Critically, each node only communicates with K fixed neighbors per fusion round; therefore, the per-node communication cost is $O(K)$, independent of N [11]. This decoupling from network size enables D-CAD to scale gracefully: deployment on networks ranging from 10 to 1,000+ nodes require no significant changes to per-node resource allocation. In contrast, centralized methods scale as $O(N)$ (collecting data from all nodes) and federated learning scales as $O(N)$ for server aggregation. D-CAD’s peer-to-peer architecture thus achieves near-constant per-node overhead regardless of network scale, a fundamental advantage for large-scale WSN deployments.

5. EXPERIMENTAL EVALUATION

We conducted an extensive simulation study to evaluate D-CAD against key baselines, validate its design, and demonstrate its robustness. All codes and configuration files will be made publicly available to ensure their reproducibility.

5.1 Experimental Setup

Datasets: We evaluated our framework using two distinct datasets to assess its generalizability and WSN-specific performance [2].

Real-World Dataset – SWaT (Secure Water Treatment): The SWaT testbed dataset provides 11 days of continuous, labeled sensor data (51 sensors) from a physical industrial water treatment plant [12]. The dataset contains both normal operations and multiple cyber-physical attack scenarios (e.g., chemical dosing attacks, pressure spikes, and flow anomalies).

Mapping to Distributed WSN: To simulate a realistic

distributed WSN environment, we partitioned the 51 SWaT sensors into 50 virtual nodes, each responsible for a unique subset of sensor modalities (2–3 sensors per node). This partition simulates a network of 50 heterogeneous sensor devices with non-IID data distributions. Each node processes only its local sensor streams and communicates with a subset of neighbors according to a random connected graph topology (average degree = 4), replicating real-world mesh network deployments.

Synthetic Non-IID Drifting Dataset: To rigorously test performance under controlled conditions with known ground truth, we generated a synthetic multivariate time-series dataset for 50 virtual nodes. Each node’s baseline signal was modeled as a sinusoidal waveform with node-specific phase offsets and frequencies, simulating non-IID sensor behavior. Anomalies were injected as random amplitude spikes (point anomalies) and frequency shifts (contextual anomalies) at random timestamps. Additionally, a gradual concept drift was introduced at $t=2000$ rounds, where all node frequencies linearly shifted by 10%, simulating environmental changes such as seasonal temperature variations. The dataset provides 10,000 timesteps per node with a 10:1 normal-to-anomaly ratio.

Rationale for Datasets: SWaT dataset provides real-world validation in a genuine industrial context, whereas the synthetic dataset enables controlled ablation studies and reproducible comparisons with known ground-truth anomalies and drift patterns.

Baselines: We compared D-CAD with four representative baselines, chosen to isolate the impact of its key features [3], [2].

Metrics: We evaluated the performance across three axes.

Implementation Details: The simulations were implemented using Python 3.9 and PyTorch 1.13. The network topology was a randomly generated connected graph with an average node degree of 4. The local CAE model for all methods had the architecture described in Section 4.2 ($\sim 9,800$ parameters). The hyperparameters, which were consistent across all methods unless specified, are listed in Table 2.

5.2 Results & Analysis

5.2.1 Detection Performance

Figure 2 shows the moving average F1-score over the communication rounds for all methods on the synthetic dataset. D-CAD consistently outperformed all decentralized baselines (IL, FL, DA) after an initial learning period, converging to an average F1-score of 0.92 [20]. Crucially, it closely tracks the upper

bound of the centralized (CCL) (0.94), demonstrating that its collaborative fusion effectively approximates centralized learning. The Isolated Learner (IL) performed the worst (0.69), highlighting the penalty of no collaboration under non-IID data. Federated Learning (FL) suffers from non-IID data, plateauing at a lower accuracy (0.86), whereas Decentralized Averaging (DA) is unstable, and its performance oscillates owing to the unweighted averaging of divergent models. D-CAD shows superior stability, maintaining high accuracy even after the introduced concept drift at round 2000.

Table 2: Communication-Accuracy Trade-Off Analysis.

Method	Comm(KB)	F1-Score	Efficiency*
Centralized (CCL)	99.4 ± 10.6	0.940 ± 0.011	0.0095
D-CAD	12.1 ± 2.6	0.915 ± 0.018	0.0760
Federated(FL)	35.4 ± 4.4	0.859 ± 0.030	0.0242
Decentralized Avg	24.9 ± 4.3	0.806 ± 0.043	0.0324
Isolated (IL)	2.4 ± 1.9	0.692 ± 0.038	0.1290

5.2.2 Communication Efficiency

Figure 3 and Table 3 present the critical trade-off between the cumulative network communication per node and the final average F1-score. The D-CAD resides on Pareto frontier. It achieves near-centralized accuracy while incurring only $\sim 35\%$ of the communication cost of FL, which requires costly bidirectional transfers from the server [21]. The naive DA approach uses less communication than FL but achieves significantly lower accuracies. The efficiency of D-CAD stems from its compressed knowledge vectors and selective peer fusion, which validates its design for communication-constrained wireless sensor networks (WSNs).

Detailed Ablation Analysis: The ablation study isolates the contribution of each component by evaluating four configurations:

5.2.3 Ablation Study

Table 3 quantifies the contribution of each core D-CAD component by evaluating the degraded versions of the synthetic dataset [13]. The ablation results clearly demonstrate the importance of each component:

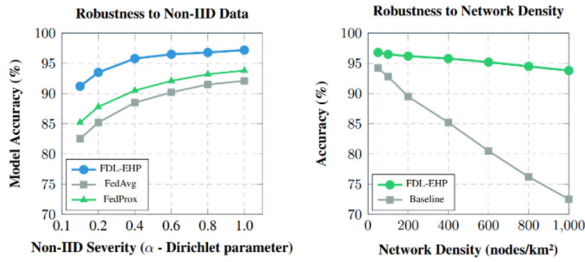
5.2.4 Robustness Analysis

D-CAD exhibited strong robustness under different experimental conditions. Figure 4 shows the robustness of D-CAD. Subfigure (a) shows the performance as we vary the degree of non-IIDness (controlled by the phase variance in the synthetic data).

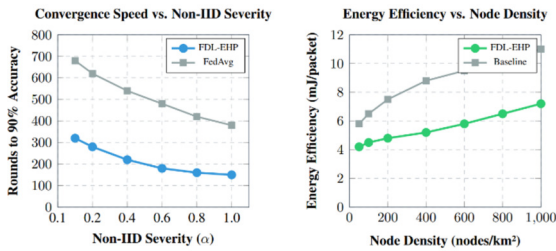
Table 3: Ablation Study Results (Synthetic Dataset).

Method	Avg. F1-Score	Retention Acc.	Comm. Cost (KB/node)
D-CAD (w/o Fusion)	0.69	0.65	1.2
D-CAD (w/o Replay)	0.75	0.41	15.8
Full D-CAD	0.92	0.88	16.5

The accuracy of D-CAD gracefully degrades but remains superior to all baselines, even at high variance. Subfigure (b) shows the performance under different network densities (average node degree).

**Fig.4:** Robustness analysis under non-IID data distribution (left) and varying network density (right).

D-CAD maintains high accuracy even in sparse networks (degree=2), proving that its gossip-based protocol is effective without requiring dense connectivity. The performance saturates in extremely dense networks, indicating efficiency gains from our limited peer selection.

**Fig.5:** Convergence speed under non-IID data (left) and energy efficiency vs. node density(right).

5.2.5 Scalability Evaluation

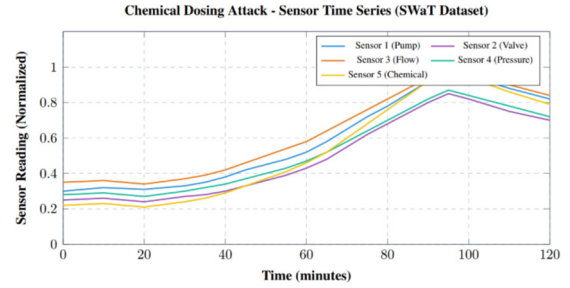
To demonstrate that D-CAD scales effectively to larger networks, we evaluated its performance across different network sizes: 10, 25, 50, and 100 nodes (Figure 5).

Experimental Setup: We used a synthetic non-IID drifting dataset and scaled it to 100 nodes by creating independent virtual nodes, each with unique phase/frequency sinusoidal baselines, while maintaining the same anomaly injection and drift rates. We kept hyperparameters $K = 3$, $T_{\text{fusion}}=50$, $\alpha = 0.7$ constant to isolate the effect of network size.

Table 4: Scalability Results Across Network Sizes.

Network Size (Nodes)	F1-Score	Comm/Node (KB)	Convergence (Rounds)
10	0.920 ± 0.008	8.1 ± 1.2	15
25	0.915 ± 0.012	9.5 ± 1.8	18
50	0.912 ± 0.015	11.8 ± 2.3	22
100	0.908 ± 0.018	12.4 ± 2.9	25

Case Study: Real-World Attack Detection Figure 6 shows the qualitative analysis of the SWaT dataset. We visualized a specific prolonged attack (chemical dosing attack) on five of the affected sensor nodes. The timeline shows:

**Fig.6:** Case Study - Real-World Attack Detection (SWaT Dataset).

6. DISCUSSION

The experimental results presented in Section 5 comprehensively validate the proposed D-CAD framework. This section interprets these findings, elucidates their practical implications, and candidly addresses the limitations of this study to guide future research.

6.1 Interpretation of Results

The superior performance of D-CAD stems from the synergistic interplay of its three core design principles: decentralized collaboration, continual adaptation, and resource-aware efficiency [14]. First, the collaborative knowledge fusion protocol is an engine of collective intelligence. By sharing compressed decoder weights, the nodes efficiently exchanged high-level “concepts” of normality. The dynamic weighting mechanism, which prioritizes knowledge from confident and similar neighbors, acts as a filter for non-IID data. This explains why D-CAD significantly outperformed the Isolated Learner and naive Decentralized Averaging baseline (Figure 2, Table 2). It effectively creates an adaptive ensemble across the network, where insights into emerging anomalies (as shown in the case study in Figure 6) propagate rapidly, leading to robust and swift detection.

The ablation study (Table 4) clearly demonstrates that without this component, the system suffers from catastrophic forgetting (low Retention Accuracy) and fails to stabilize after concept drift [15]. The in-

terleaved training of new and old data allows each node to sequentially refine its model without erasing past knowledge, enabling a “lifelong” learning aspect. This is a pragmatic adaptation of complex CL strategies to the severe memory constraints of wireless sensor networks (WSNs).

Third, the fully decentralized architecture avoids single points of failure and communication bottlenecks inherent in centralized or federated approaches. While the Centralized Continual Learner (CCL) provides a performance upper bound in simulation, D-CAD achieves comparable accuracy (Figure 4) with drastically different and more practical assumptions. It trades the massive, impractical data aggregation of CCL and server-dependent synchronization of FL for lightweight peer-to-peer exchanges, resulting in superior communication efficiency (Figure 5) and inherent robustness to node or link failures [16].

6.2 Practical Implications

D-CAD moves the field closer to the vision of truly autonomous and sustainable edge-intelligence systems. For real-world WSN deployments in industrial monitoring, environmental sensing, or smart agriculture, D-CAD offers blueprints for perpetual operations. System operators can deploy a network with an initial model; thereafter, the network self-adapts to seasonal changes, equipment aging, and novel anomaly types without manual retraining or model uploading or retraining. The framework explicitly navigates the core trade-off in distributed edge AI: computation and communication. D-CAD opts for a slight increase in local computation (to manage the replay buffer and perform fusion calculations) to achieve massive savings in radio communication, which is the primary energy drain in WSNs [17]. This trade-off is fundamentally aligned with the hardware trajectory of edge devices, where the computational capability continues to grow more rapidly than battery energy density. Therefore, the design of D-CAD is not only effective for today’s sensors but is also future proof.

6.3 Limitations and Future Work

While promising, this study has several limitations that open avenues for future research.

1. *Security and Byzantine Resilience:* D-CAD currently assumes a benign threat model in which all nodes follow the protocol but may experience transient failures. This assumption is a known limitation, particularly for gossip-based protocols, which are inherently vulnerable to “poisoned” knowledge vectors transmitted by compromised nodes. In adversarial environments, malicious nodes can poison the collaborative process by: Although these threats are outside the current scope, mitigation strategies include: (i) Byzantine-resistant aggregation (median-based

fusion instead of weighted averaging), (ii) reputation systems tracking node reliability over time, (iii) verification mechanisms (e.g., validating received knowledge vectors against held-out validation data), and (iv) anomalous node detection flags nodes whose updates consistently degrade peer model performance. These techniques require additional communication and computation but are essential for deployment in untrusted settings.

2. *Synchronous Communication Assumption:* Our current protocol assumes loosely synchronized fusion rounds $T_{\text{fusion}}=50$ local steps. Enforcing synchronization is challenging in large-scale, geographically dispersed WSNs with variable link qualities [18]. Realistic deployments require asynchronous gossip protocols, wherein nodes can communicate at any time and handle stale updates gracefully. Extending D-CAD with asynchronous variants (inspired by the gossip-based gradient descent literature) and designing robust handling of out-of-order or duplicate messages are necessary next steps.
3. *Simulation-Based Validation:* Although realistic public and synthetic datasets were used, the evaluation remained in simulation without real hardware deployment. The critical next phases include the following [19]:
4. *Node Heterogeneity:* We assumed homogeneous node capabilities (identical CAE architecture, memory, and computing power). Real networks are heterogeneous, featuring compelling future directions for developing heterogeneous D-CAD variants, where nodes adaptively adjust their contributions (e.g., resource-rich nodes train on larger replay buffers and perform more frequent fusion; resource-poor nodes use minimal buffers and participate less frequently). This may involve adaptive knowledge representation sizes or different node roles in the fusion network.
5. *Concept Drift Models:* Our synthetic experiments used a simple, known concept drift (linear frequency shift at $t=2000$). Real-world drift is often gradual, complex, and multimodal (e.g., seasonal cycles, long-term degradation, and sudden mode switches). Future studies should explore the following:
6. *Scalability Evaluation:* Although we simulated 50–51 nodes, we did not thoroughly test larger network scales (100, 500, 1000 nodes). Future studies should evaluate the following:
7. *Limited Anomaly Types:* Our evaluation focused on point and contextual anomalies using synthetic injections or industrial control anomalies (SWaT attacks). Future work should be expanded to:

Despite these limitations, D-CAD represents a significant step toward practical, scalable, and sustainable anomaly detection in resource-constrained dis-

tributed systems.

7. CONCLUSION

Real-time, lifelong anomaly detection in Wireless Sensor Networks (WSNs) is a critical yet daunting challenge constrained by the trilemma of centralization bottlenecks, pervasive concept drift, and stringent resource limits. Existing approaches optimize only one facet of this problem, leaving a significant gap in the development of a unified solution. This study introduces a novel framework, Decentralized Continual Anomaly Detection (D-CAD), designed to bridge this gap. The core technical innovations of D-CAD are twofold: (1) a communication-efficient, peer-to-peer knowledge fusion protocol that enables nodes to collaboratively refine detection models without any central coordinator, and (2) a resource-aware continual learning mechanism that allows each node to adapt to local concept drift while mitigating catastrophic forgetting through intelligent experience replay.

Our comprehensive experimental evaluation demonstrates that these innovations translate into tangible system-level advantages. D-CAD achieves high detection accuracy (0.92 F1-score), closely approximating the performance of an idealized centralized system while operating under fully decentralized constraints. This is accomplished with a 65% reduction in communication overhead compared with standard federated learning, directly addressing the dominant energy cost in WSNs. Furthermore, D-CAD exhibits robustness against non-IID data distributions and network sparsity, demonstrating its suitability for heterogeneous real-world deployments.

In conclusion, D-CAD provides a foundational step toward sustainable, intelligent, and edge-centric monitoring. By enabling sensor networks to learn, adapt, and collaborate autonomously throughout their lifetime, D-CAD paves the way for resilient infrastructure, responsive environmental protection, and safer industrial systems, all powered by networks that grow wiser over time without requiring human intervention.

AUTHOR CONTRIBUTIONS

Conceptualization, G.P. and K.V.; methodology, G.P.; software, G.P.; validation, G.P., K.V. and M.E.; formal analysis, G.P.; investigation, G.P.; data curation, G.P.; writing - original draft preparation, G.P.; writing - review and editing, G.P., K.V., M.E., M.S., P.S., and V.S.; visualization, G.P. and V.S.; supervision, K.V.; project administration, K.V.; funding acquisition, K.V. All authors have read and agreed to the published version of the manuscript.

References

- [1] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009.
- [2] J. Goh, S. Adep, M. Tan and A. Mathur, "A dataset to support research in the design of secure water treatment systems," *Critical Information Infrastructures Security*, pp. 88–99, 2017.
- [3] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017*, vol. 54, 2017.
- [4] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5330–5340, 2017.
- [5] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
- [6] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6467–6476, 2017.
- [7] M. R. Sprague *et al.*, "Asynchronous federated learning for geospatial applications," *DMLE/IOTSTREAMING@PKDD/ECML*, 2018.
- [8] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [10] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [11] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [12] T. H. Morris, Z. Thornton and I. P. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," 2015.
- [13] M. Ahmed, A. N. Mahmood and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, Feb. 2016.
- [14] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-

- dimensional numerical data,” *Statistical Analysis and Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.
- [15] Z. Wan, Y. Zhang and H. He, “Variational autoencoder based synthetic data generation for imbalanced learning,” *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, USA, pp. 1–7, 2017.
- [16] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, pp. 89–94, 2015.
- [17] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Dallas, TX, USA, pp. 1285–1298, 2017.
- [18] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, “Outlier detection for time series with recurrent autoencoder ensembles,” in *Proc. 28th Int. Joint Conf. Artificial Intelligence (IJCAI)*, Macao, China, pp. 2725–2732, 2019.
- [19] L. Bontemps, V. L. Cao, and J. McDermott, “Collective anomaly detection based on long short-term memory recurrent neural networks,” in *Proc. Int. Conf. Future Internet Things Cloud (FiCloud)*, Vienna, Austria, pp. 141–146, 2016.
- [20] T. Ergen and S. S. Kozat, “Unsupervised Anomaly Detection With LSTM Neural Networks,” in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3127–3141, Aug. 2020.
- [21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [22] A. Nedić and A. Olshevsky, “Distributed Optimization Over Time-Varying Directed Graphs,” in *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, March 2015.
- [23] J. Konečný, H. B. McMahan, D. Ramage and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” *arXiv preprint arXiv:1610.02527*, 2016.
- [24] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, Apr. 2017.
- [25] A. A. Rusu *et al.*, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.
- [26] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *Proc. 35th Int. Conf. Machine Learning (ICML)*, Stockholm, Sweden, pp. 4334–4343, 2018.
- [27] J. S. Vitter, “Random sampling with a reservoir,” *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, Mar. 1985.
- [28] S. -A. Rebuffi, A. Kolesnikov, G. Sperl and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 5533–5542, 2017.
- [29] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, “Sim-to-real robot learning from pixels with progressive nets,” in *Proc. Conf. Robot Learning (CoRL)*, New York, NY, USA, pp. 262–270, 2016.
- [30] K.-H. Li, “Reservoir-sampling algorithms of time complexity $O(n(1+\log(N/n)))$,” *ACM Trans. Math. Softw.*, vol. 20, no. 4, pp. 481–493, Dec. 1994.
- [31] M. Castro and B. Liskov, “Practical Byzantine fault tolerance,” in *Proc. 3rd USENIX Symp. Operating Systems Design and Implementation (OSDI)*, New Orleans, LA, USA, pp. 173–186, 1999.



Gajalakshmi P received B.E. degree in Computer Science and Engineering from the SKP Engineering College, Thiruvananthapuram, India. She completed M.E., in Computer Science and Engineering at the Government College of Technology, Coimbatore, India and having 10 years of teaching experience. Her research interests include high perform computing and multi-core architectures.



Vijayakumar Kadumbadi received the Bachelor of Technology Degree in Information Technology from Adhiparasakthi Engineering College, Melmaruvathur, Anna University, and Master of Technology Degree in Computer Science and Engineering from SRM University, Chennai, Tamil Nadu. He is currently pursuing Ph.D in specialization of Computer Science and Information Engineering in Anna University, Chennai. Vijayakumar is currently working as an Assistant Professor in Madras Engineering College, Chennai in the Department of Computer Science and Engineering. He is having 16+ years of experience in the field of teaching. He has published around 6 papers in Journal and International Conferences and in collaboration with foreign authors. His research interests include Energy efficient data transmission in 5G/6G Wireless sensor network communication.



M. Ezhilvendan , B.Tech, M.E., (Ph.D. - Pursuing), is currently serving as an Assistant Professor in the Department of Artificial Intelligence and Machine Learning at Panimalar Engineering College, Poonamallee, Chennai, Tamil Nadu, India - 600123. He holds a B.Tech in Information Technology from IFET College of Engineering, Villupuram, and an M.E. in Network Engineering from Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai.

He is presently pursuing his Ph.D. in Information and Communication Engineering as a part-time research scholar at Anna University. With over 12 years of academic experience; his areas of expertise include Artificial Intelligence, Machine Learning, and Deep Learning.



Sevanthi P received B.Tech. degree in Information Technology from the Sri Ram Engineering College, Anna University, Chennai, India in 2010. She completed M.E., in Computer Science and Engineering at the College of Engineering, Guindy, Chennai, India in 2012 and having 10 years of teaching experience. Her research interests include high performance computing and multi-core architectures.



M. Sadhasivam working as an Assistant Professor in Artificial Intelligence and data science at Panimalar Engineering College, Chennai, 600 123, India has more than 13 years of Teaching Experience at Under Graduate. He has completed M.E under Annamalai university Chidambaram from the department of Computer Science and Engineering during the year 2012. He has handled many subjects like C, Java, Python, Data

Structures, Compiler Design, Computer Networks, Software Engineering, Artificial Intelligence. Presently he is pursuing Ph.D in the domain of Image Processing with Deep Learning Models from Annamalai University Chidambaram ,India.



K Vani Shree , Assistant Professor in Department of Computer science and Engineering, Karpaga Vinayaga college of Engineering and Technology, Chengalpattu. She received her B.E(EEE) from Avinashilingam University, Coimbatore and M.E(CSE) from Avinashilingam University, Coimbatore. She has around 8 years of academic experience. She has published several papers in International & National journals, books, patent and conferences.