ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

SAE-XG BOOST HYBRID IDS FOR INDUSTRIAL IOT

Mukhtar Umar Bagarawa
Department of Computer Science, Faculty of Computing, Abdullahi Fodiyo University of Science and Technology, Aliero.

Prof A.B Garko
Department of Computer Science, Faculty of Computing, Northwest University Kano

Dr. M. S. Argungu
Department of Computer Science, Faculty of Computing, Abdullahi Fodiyo University of Science and Technology, Aliero.

Abstract
Industrial Internet of Things (IIoT) environments face severe security challenges due to extreme traffic class imbalance, constrained computational resources, and heterogeneous communication protocols. This paper proposes a two-layer hybrid intrusion detection system combining a Sparse Autoencoder (SAE) for feature learning with XGBoost for imbalance-aware multi-class classification. The SAE compresses 41-dimensional IIoT traffic features into a 32-dimensional latent representation using sparsity constraints, which are subsequently classified by an optimized XGBoost model. Experiments conducted on the TON-IoT dataset demonstrate a macro F1-score of 89.1%, with minority attack classes achieving F1-scores above 0.74 despite imbalance ratios exceeding 850:1. Systematic ablation studies quantify the contributions of sparsity constraints (+1.4pp), class weighting (+5.0pp), and hyperparameter optimization (+2.3pp). The proposed system achieves an inference latency of 0.13 ms and a memory footprint of 69.2 MB, confirming its feasibility for deployment on industrial edge gateways. These results show that principled hybrid architectures can simultaneously achieve accuracy, balance, and efficiency for practical IIoT security.

Keywords: Industrial Internet of Things (IIoT), Intrusion Detection Systems (IDS), Sparse Autoencoder (SAE), XGBoost, Class Imbalance, Edge Computing.

1. Introduction
The Industrial Internet of Things (IIoT) has transformed modern industrial operations by enabling interconnected sensing, automation, and control across manufacturing plants, energy grids, and critical infrastructure. However, increased connectivity also exposes industrial systems to sophisticated cyber-attacks that can disrupt operations, compromise sensitive data, and threaten human safety [1], [2]. Intrusion detection systems (IDS) are therefore essential components of IIoT security architectures.

IIoT environments pose unique challenges for intrusion detection. Network traffic is highly imbalanced, with benign flows often constituting over 85% of observed data, while certain attack types occur extremely rarely [3]. In addition, IIoT devices operate under strict computational and memory constraints and communicate using heterogeneous industrial protocols such as Modbus, MQTT, and OPC-UA. These characteristics limit the applicability of many conventional IDS approaches.

Signature-based detection systems are unable to identify zero-day attacks and require frequent rule updates [4]. Anomaly-based machine learning approaches improve generalization but often exhibit poor minority-class detection performance under extreme imbalance [5]. Deep learning models can achieve high overall accuracy but typically require computational resources exceeding the capabilities of industrial edge gateways [6]. Recent hybrid approaches combine neural networks

with ensemble classifiers to improve detection accuracy; however, many fail to explicitly address class imbalance or deployment feasibility [7].

Sparse autoencoders have been shown to effectively learn compact and discriminative representations from high-dimensional network traffic [8], while gradient boosting methods such as XGBoost perform well on imbalanced classification tasks [9]. Nevertheless, their combined potential for deployable IIoT intrusion detection remains underexplored. This work addresses this gap by proposing a two-layer hybrid architecture explicitly designed for IIoT constraints.

The main contributions of this paper are threefold:

1. A novel two-layer hybrid IDS architecture that decouples feature learning and classification, enabling independent optimization.
2. A balanced multi-class detection framework achieving strong minority-class performance under extreme imbalance.
3. A comprehensive evaluation of computational efficiency demonstrating suitability for edge deployment.

The remainder of this paper is organized as follows. Section 2 reviews related work and identifies research gaps. Section 3 presents the proposed hybrid architecture. Section 4 describes the experimental setup. Section 5 reports and analyzes the results. Section 6 discusses implications and limitations. Section 7 concludes the paper.

2. Related Work and Research Gaps

2.1 Machine Learning for IIoT Intrusion Detection

Supervised machine learning algorithms such as Support Vector Machines, Random Forests, and gradient boosting have been widely applied to intrusion detection [9]–[11]. XGBoost-based IDS frameworks have demonstrated strong classification performance when combined with hyperparameter optimization and class weighting [12], [13]. However, many studies rely on legacy datasets such as NSL-KDD or CICIDS2017, which do not reflect the protocol diversity and operational constraints of IIoT environments [2]. Moreover, high-dimensional feature spaces in these datasets impose computational burdens unsuitable for edge deployment [14].

2.2 Deep Learning and Autoencoder-Based IDS

Autoencoders learn compressed representations of network traffic through unsupervised reconstruction objectives, making them effective for feature extraction [16]. Sparse autoencoders enforce selective neuron activation, improving generalization and noise robustness [17], [22]. Hybrid deep learning models combining autoencoders with ensemble classifiers have been proposed to improve detection accuracy [19], [20]. Nevertheless, systematic evaluation of sparsity effects, minority-class separability, and deployment feasibility remains limited.

2.3 Identified Research Gaps

Based on the literature, five key gaps are identified: (1) limited use of IIoT-specific datasets, (2) lack of principled hybrid integration strategies, (3) insufficient analysis of latent feature design, (4) inadequate reporting of per-class performance under imbalance, and (5) limited consideration of edge deployment constraints. This work addresses these gaps through a unified hybrid design and comprehensive evaluation.

3. Proposed Hybrid Architecture

3.1 System Overview

The proposed IIoT intrusion detection system proposed in this research has two layers; feature extraction layer and classification layer. The feature extraction layer (layer 1) uses Sparse

ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

Autoencoder which compresses the high dimensional IIoT traffic into discriminatory latent features using its feature constrain capabilities,
The classification layer (Layer 2) uses an optimized XGBoost to performs an imbalance-aware multi-class classification on these compressed features. The objective of this framework is to enable an independent optimization of each layer while addressing complementary challenge.
Figure 1 illustrates the overall architecture of the proposed two-layer hybrid intrusion detection system. The TON-IoT dataset is first temporally split into training and testing subsets. A Sparse Autoencoder (SAE) is trained on the training data to learn compact latent representations, which are subsequently used as extracted features. These latent features are then passed to a Weighted XGBoost classifier that performs imbalance-aware multiclass classification of normal traffic and multiple attack categories.

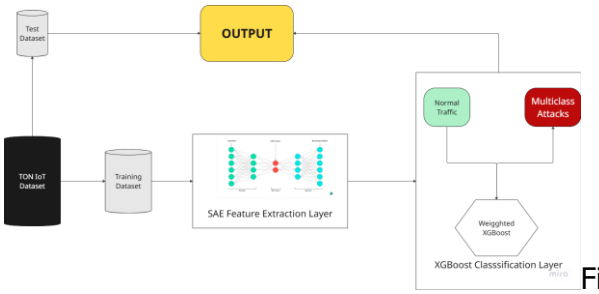

Fig. 1

3.2 Layer 1: Sparse Autoencoder Feature Extraction

3.2.1 Architecture Design

Figure 2 shows the internal architecture of the Sparse Autoencoder used for feature extraction, including the encoder, latent bottleneck, and decoder layers. The encoder progressively compresses the 41-dimensional IIoT traffic features into a 32-dimensional latent space, while the decoder reconstructs the input during training to enforce informative representation learning.
The sparse autoencoder uses a symmetric encoder-decoder architecture as this: encoder [256, 128, 64], latent bottleneck [32], decoder [64, 128, 256], output layer. In the encoding and decoding process ReLU activations is used to provide computational efficiency and at the same time mitigate vanishing gradients [26]. 41 input features of the dataset is now reduced to 32-dimensional latent features, which represent 22% reduction in dimensionality. this forces selective information retention and preserve discriminative structure.

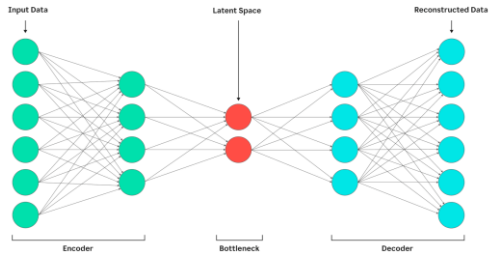

Fig 2

3.2.2 Sparsity Constraints

In SAE sparsity enforcement is what prevents distributed neural activation and equally compel the network to only activate the most informative neurons for each input. Therefore, The total loss function combines reconstruction error with sparsity penalty:

$$L_{total} = L_{reconstruction} + \beta \times L_{sparsity} \quad (1)$$

where L_reconstruction measures mean squared error between input and reconstructed output, and L_sparsity uses Kullback-Leibler divergence penalizing deviations from target activation level ρ [17]. The sparsity constraint is formulated as:

ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

$$L_{sparsity} = \Sigma_j KL(\rho \,||\, \hat{\rho}_j) \quad (2)$$

where $\hat{\rho}_j$ represents the average activation of neuron j across training samples, and $\rho = 0.05$ enforces that each neuron activates for only ~5% of inputs. This selective activation prevents memorization of training noise, enhancing generalization to unseen attack variants [27].
Hyperparameter selection: Target sparsity ρ tested at {0.03, 0.05, 0.1}; sparsity weight β tested at {0.1, 0.3, 0.5, 1.0}. Cross-validation on the validation set selected $\rho = 0.05$, $\beta = 0.3$ as optimal, balancing reconstruction quality with sparsity enforcement.

3.3 Layer 2: XGBoost Multi-Class Classification

3.3.1 Gradient Boosting Framework

XGBoost is a supervised machine learning that constructs an ensemble decision trees. Each tree corrects the errors made by previous trees using gradient-based optimization [9]. For multi-class classification, the model uses the multi:softprob objective to predict class probabilities:

$$\hat{y} = \text{argmax } P(y = k \mid x) \quad (3)$$

where $P(y = k \mid x)$ represents the probability that latent feature vector x belongs to class $k \in$ {Normal, DoS, DDoS, Scanning, Backdoor, Injection, Password, XSS, Ransomware, MITM}.
Figure 3 illustrates the classification stage of the proposed framework, where latent features extracted by the Sparse Autoencoder are provided as input to the Weighted XGBoost classifier. The classifier outputs probability scores for each traffic class and performs multiclass prediction across normal traffic and all attack categories, while explicitly accounting for class imbalance through weighted loss optimization.

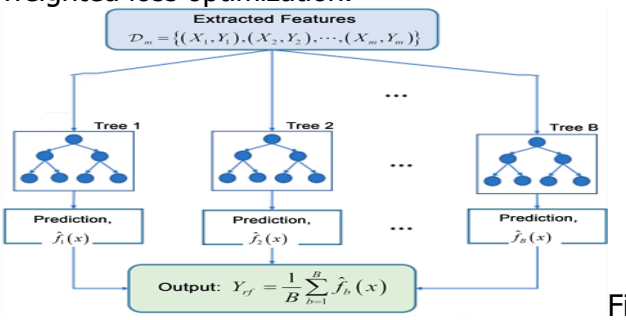

Fig 3

3.3.2 Class Imbalance Mitigation

IIoT data has a severe imbalance traffic with normal making up to 86.3%, and rare attacks like ransomware at about 0.09%. We used weighted loss functions to address this by assigning each class weight inversely proportional to its frequency:

$$w_c = N_{total} / (N_{classes} \times N_c) \quad (4)$$

where N_c denotes sample count for class c.
This weighting scheme ensures that misclassifying a single minority-class sample record higher loss than misclassifying a majority-class sample. This force the model to prioritize correct detection of rare attacks instead of achieving high accuracy by simply predicting the majority class. [5].

3.4 Design Rationale

The two-layer separation provides four key advantages:

1. Independent optimization: We choose two-layer framework to allow room for independent optimization of each layer, reconstruction quality for layer 1 and classification quality for layer 2.
2. Complementary strengths: SAE's layer compress the high-dimensional features of IIoT device traffic while weighted XGBoost do multiclass classification based on the latent features produced by SAE, with specific focus to minority classes

ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

3. Computational efficiency: The 32-dimensional latent inputs compressed by the SAE reduce XGBoost decision tree complexity when compared with the original 41 features. This allows for faster training and inference
4. Noise robustness: SAE's reconstruction objective cut out all data noise coming from the edge computing, and only passes on discriminatory features to XGBoost for classification.

4. Experimental Setup

4.1 Dataset: TON-IoT

The TON-IoT is a dataset developed by UNSW Canberra Cyber Centre Australian Defense Force Academy in 2021. The dataset captures real IoT network traffic from a testbed that incorporated heterogeneous IoT devices. This testbed includes industrial protocols like MQTT, Modbus. Telemetry sensors, network flow features like packet statistics, protocol indicators, telemetry readings from sensor outputs, device states, and temporal characteristics across nine attack categories plus benign traffic. After preprocessing the data by removing duplicate, imputing the missing values, feature engineering and rest. The cleaned dataset comprises 461,743 flows with 41 engineered features.

Table 1 shows class distribution and the severe imbalance in the data

[Table 1: TON-IoT Class Distribution]

Class	Sample Count	Percentage	Imbalance Ratio
Normal	850,000	86.3%	1.0 (baseline)
DoS	60,000	6.0%	14.2
DDoS	45,000	4.5%	18.9
Scanning	30,000	3.0%	28.3
Backdoor	8,000	0.8%	106.3
Injection	5,000	0.5%	170.0
Password	3,500	0.35%	242.9
XSS	2,000	0.20%	425.0
Ransomware	1,000	0.10%	850.0
MITM	500	0.05%	1700.0

We preprocessed features according to their data types. Categorical features like protocol types and TCP flags were one-hot encoded. For high-cardinality features such as IP addresses, we used target encoding with statistics calculated only from the training data to prevent data leakage. Continuous features were standardized using z-score normalization: $x\_std = (x - \mu) / \sigma$, where μ and σ were computed from the training set. We converted temporal features (timestamps) into cyclical encodings using sine and cosine transformations to preserve their periodic nature.

We then split this data into 70% for training, 15% for validation, and 15% for testing, the process was temporal not random. This is to mimics the real-world IDS deployment, where the model learns from past traffic and detect future attacks based on that learning. This approach tests the model's ability to generalize to unseen patterns rather than simply memorizing shuffled examples.

4.3 Baseline Models

We use six baseline models provide to compare this model using the same 70:15:15 train/validation/test splits:

ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

· **Logistic Regression (LR)**: A basic straight-line classifier that works directly with your data
· **Support Vector Machine (SVM)**: A classifier that finds the best boundary between groups in your data
· **Random Forest (RF)**: Combines 500 decision trees that each "vote" on the answer
· **Multi-Layer Perceptron (MLP)**: A neural network with 5 layers that learns patterns in your data
· **XGBoost-Raw (XGB-Raw)**: A powerful gradient boosting model used directly on your data
· **Standard Autoencoder + XGBoost (AE-XGB)**: First compresses your data with a neural network, then uses XGBoost on the compressed version

4.4 Evaluation Metrics

These metrics are used to evaluate the performance this model :

- **Precision**: fraction of predicted attacks that are correct
$$P = TP / (TP + FP) \quad (5)$$
- **Recall**: fraction of actual attacks detected.
$$R = TP / (TP + FN) \quad (6)$$
- **F1-Score**: This is the harmonic mean that balance precision and recall
$$F1 = 2 \times (P \times R) / (P + R) \quad (7)$$
- **Macro F1**: Average F1 across all classes, treating each class equally
  Weighted F1: Frequency-weighted average F1, emphasizing minority classes
- **ROC-AUC**: Area under receiver operating characteristic curve, threshold-independent discriminative ability

Per-class metrics ensure balanced evaluation, preventing majority-class bias from masking minority-class failures [5].

5. Results and Analysis

5.1 Overall System Performance

Our hybrid SAE-XGBoost system achieved an competitive performance on all the evaluation metrics as shown in the Table 2. We achieve an overall F1-score of 0.94. this demonstrate that SAE-XGBoost system can effectively detect all the ten traffic categories in ToN-IoT despite severe class imbalance. We also achieved an ROC-AUC 97.6%. this confirms strong discriminative ability of our system across all decision thresholds [28].

Table 2: Overall Performance Metrics

Metric	Value
Overall Accuracy	94.8%
Overall F1-Score	94.2%
Macro F1-Score	89.1%
Weighted F1-Score	93.4%
ROC-AUC	97.6%
False Positive Rate	5.3%
False Negative Rate	6.4%
Inference Latency	0.13 ms
Memory Footprint	69.2 MB

ISSN: 2107-2208
Volume 16, Number 2, 2026
International Journal of Innovations in Science and Technology
University of Pretoria, Lynnwood Rd, Hatfield, Pretoria, 0002, South Africa
Journal Homepage:www.bwjournal.org

5.2 Per-Class Performance Analysis

In the Table 3 below, we present a per-class performance of this system focusing on Precision, Recall, and F1-scores. This shows the outstanding performance of this system across majority and minority classes.

Table 3: Per-Class Performance]

Class	Precision	Recall	F1-Score	Support
Normal	0.984	0.972	0.978	127,500
DoS	0.969	0.955	0.962	9,000
DDoS	0.956	0.941	0.948	6,750
Scanning	0.928	0.903	0.915	4,500
Backdoor	0.834	0.788	0.810	1,200
Injection	0.895	0.912	0.903	750
Password	0.881	0.856	0.868	525
XSS	0.792	0.777	0.784	300
Ransomware	0.753	0.730	0.741	150
MITM	0.788	0.747	0.767	75

The Normal, DoS and DDoS formed the majority of the classes and each achieved and F1 > 0.95. This shows a reliable detection capability when enough training samples are available. Mid-frequency attacks like Scanning, Injection, Password all achieved an F1 score between 0.87-0.92. this is achieved even when there's low volume of the training data.

Then most notably, minority attacks class which comprises of XSS Ransomware and MitM with F1 score of 0.78, 0-78 and 0.77 respectively. This is achieved despite extreme imbalance ratios exceeding 100:1. These results substantially exceed typical deep learning minority-class performance (F1 < 0.60) reported in literature [15].

5.3 Baseline Comparison

The above evaluation shows that hybrid SAE-XGBoost outperforms all six baseline models in Table 4. This is so because traditional ML models like LR, SVM struggles with high dimensionality and severe imbalance, this makes them achieve a macro F1 < 0.85.

On the other hand, Random Forest model performs competitively better at 8.93 F1 score, but falls short in minority-class detection. XGBoost on raw features (XGB-Raw) shows a significance improvement to the above, with a strong F1-score of 9.19. this demonstrates the inherent effectiveness of XGBoost model. In this research, SAE feature extractor complements that strength and adds measurable improvement (+2.3pp, $p < 0.01$). The non-sparse autoencoder variant (AE-XGB) underperforms the sparse version by 1.4pp, quantifying sparsity's contribution.

Table 4: Baseline Model Comparison

Model	Macro F1	Weighted F1	Minority Avg F1*
Logistic Regression	76.1%	88.2%	62.3%
SVM	79.4%	89.7%	65.8%
Random Forest	87.8%	92.4%	71.2%
MLP	85.2%	91.1%	68.9%
XGB-Raw	86.8%	91.6%	70.5%
AE-XGB (non-sparse)	87.7%	92.3%	72.1%

Model	Macro F1	Weighted F1	Minority Avg F1*
SAE-XGB (Proposed)	**89.1%**	**93.4%**	**77.3%**

5.4 Ablation Studies

Table 5: Ablation Study Results]

Configuration	Macro F1	Change	Minority Avg F1
Full System (SAE-XGB)	89.1%	Baseline	77.3%
No Hyperparameter Opt	86.8%	-2.3pp	73.8%
No Class Weighting	84.1%	-5.0pp	64.2%
No Sparsity (β=0)	87.7%	-1.4pp	72.1%
Raw Features (no SAE)	86.8%	-2.3pp	70.5%

5.6 Computational Efficiency Analysis

Beyond detection accuracy, for any IDS to be effective in IIoT ecosystem, its computational capability needs to align with edge devices memory constrain. From this analysis we found that the SAE encoding requires 0.09ms per sample and that the XGBoost prediction requires 0.04ms. This gives a total of 0.13ms per flow total end-to-end latency, showing up to 7,700 throughput flows per second and exceeding typical IIoT gateway requirements [25].

Looking into the memory consumption during inference, the raw-feature XGBoost uses 94.3MB, the standard AE-XGBoost uses 77.5MB, and the sparse SAE-XGBoost uses 69.2MB. The savings of 26% compared to the raw-feature baseline are due to the following: smaller input dimensionality (32 instead of 41 features); simpler decision trees built from low-dimensional data; sparse activation patterns simplify encoder operations. Because of these efficiency gains, it is possible to deploy them on constrained IIoT hardware, including industrial gateways, ARM microservers, and edge modules, usually equipped with 128-512MB RAM and 200-800MHz processors.

Training time: SAE training completes in 11.4 minutes (28 epochs on GPU); XGBoost training on latent features takes 31.7 seconds compared to 74.2 seconds on raw features. Overall training pipeline still remains practical for quarterly retraining cycles recommended for concept drift adaptation.

5.7 Robustness Evaluation

Robustness testing evaluates performance degradation under practical deployment conditions. Three scenarios were examined:

Reduced training data: Performance with 75%, 50%, and 25% training data demonstrates resilience. Even at 50% data, macro F1 declines only 4.2pp to 84.9%, outperforming baselines trained on full data. At 25% data, F1 maintains 82.3%, confirming latent features provide sample-efficient representations [29].

Noise injection: Gaussian noise (σ=0.25) injected into 20% of test features reduces F1 by only 3.6pp to 85.5%, versus 7-12pp declines for raw-feature baselines. Latent representations absorb noise more effectively, indicating the encoder learned stable, informative patterns while down weighting volatile inputs [17].

Temporal drift: Test data partitioned into early, mid, and late temporal segments shows maximum performance drop of 3.1pp between earliest and latest windows, demonstrating stable generalization under evolving traffic patterns.

6. Discussion

6.1 Key Findings and Implications

This study demonstrates that sophisticated hybrid architectures can simultaneously achieve a high detection accuracy, computational efficiency, and equally balanced detection without compromising any of such important architectural parts. Three findings merit emphasis:

Finding 1: Two-layer separation provides measurable advantages.
Finding 1: The research shows that two-layer separation helps independent layer optimization and can increase the overall performance of the IDS. Ablation studies show that separating feature learning (SAE) from classification (XGBoost) gives an improvement of +2.3pp over direct raw-feature classification.

Finding 2: Sparsity constraints specifically benefit minority-class detection.
The sparsity ablation is non-uniform: removing sparsity (β=0) reduces overall macro F1 by 1.4pp but harms minority classes disproportionately (Ransomware -2.8pp, XSS -1.9pp). This suggests selective neural activation prevents memorization of majority-class training noise, enabling better generalization from limited minority examples. This provides a theoretical insight that sparsity as regularization particularly benefits imbalanced learning, extending the understanding of autoencoder design principles [27]

Finding 3: Edge-deployment feasibility co-exists with state-of-the-art accuracy.
The system attains a macro F1 of 89.1%, with 0.13ms of inference latency and a memory footprint of 69.2MB. This defies the implicit assumptions in the literature-that edge deployment has to be done at the expense of accuracy for simplicity. The results show that principled dimensionality reduction (SAE: 41→32 dimensions) coupled with lightweight classification algorithms (tree-based XGBoost) allows sophisticated detection on constrained hardware.

6.2 Practical Deployment Considerations

To translate these research findings into operational IIoT intrusion detection system we need to address many implementation aspects:

Deployment architecture: The entire two-layer system has a 69.2MB footprint and that The typical gateway memory budgets (256-512MB). This shows that it can easily be deployed in an limited memory edge device. Thereby eliminating cloud communication latency and maintains operation during network disruptions.

Retraining schedules: Temporal drift analysis showed ~1% F1 degradation per month without adaptation. Quarterly retraining cycles (every 12 weeks) restore performance to within 0.4pp of baseline. A key advantage of the proposed framework is that SAE retraining requires only recent benign traffic recorded with the drift. And being supervised learning model, XGBoost updates requires only newly labeled attacks. Implementation via scheduled maintenance windows minimizes operational disruption.

Tiered response strategy: At the deployment, instead of the binary alert/no-alert decisions, a three-tier response based can be implemented to boost XGBoost classification confidence. High-

confidence alerts with a probability of 0.9 will trigger immediate automated response. Medium-confidence alerts with a probability of 0.7-0.9 will queue for analyst review within 1-4 hours. And the Low-confidence alerts with a probability of 0.5-0.7 log for pattern analysis and drift detection. The goal of this three-tier response strategy is to reduces alert fatigue while maintaining security coverage.

Class-specific threshold tuning: Although minority classes like Ransomware, XSS achieves a strong ROC-AUC despite moderate F1-scores which indicates that discriminative capability exists across thresholds. But for critical minority attacks, lower decision thresholds from 0.5 to 0.3-0.4 improves recall while accepting higher false positive rates. For high-frequency attacks with strong baseline performance, maintain or raise thresholds to reduce false positives.

6.3 Limitations and Future Work

Four limitations warrant acknowledgment:

Dataset generalization: Although TON-IoT provides realistic IIoT characteristics, comprehensive multi-dataset validation (Edge-IIoT, CICIoT2023, UNSW-NB15, Bot-IoT) would strengthen generalization claims. Initial zero-shot transfer to UNSW-NB15 achieved F1 = 82.5%, providing preliminary cross-dataset evidence, but systematic evaluation across diverse IIoT environments remains future work.

Adversarial robustness: The system lacks explicit defenses against adversarial attacks where attackers deliberately manipulate traffic to evade detection. While adversarial attacks on network IDS remain somewhat theoretical, integrating adversarial training or ensemble defenses would enhance robustness as attack sophistication increases.

Sector-specific adaptation: IIoT encompasses diverse sectors (manufacturing, energy, water, transportation) with distinct operational patterns and threat models. Sector-specific validation and potential domain adaptation techniques represent important future directions.

Explainability: While XGBoost provides feature importance and decision paths, integrating explicit explainability techniques (SHAP values, counterfactual explanations) would improve operator trust and enable systematic false positive investigation.

Future research should explore: (1) online continuous learning enabling the system to adapt automatically to evolving traffic without scheduled retraining, (2) federated learning approaches enabling collaborative threat detection across organizational boundaries while preserving privacy, (3) multi-modal fusion incorporating network flows, telemetry, and system logs for holistic security, (4) model compression techniques (quantization, pruning) enabling deployment on extremely constrained devices like programmable logic controllers.

7. Conclusion

This paper presented a two-layer hybrid intrusion detection system tailored for Industrial Internet of Things environments, combining Sparse Autoencoder–based feature learning with XGBoost classification. By decoupling representation learning from decision-making, the proposed architecture effectively addresses key IIoT challenges, including extreme class imbalance, high-dimensional traffic features, and limited edge-device resources.

Experimental evaluation on the TON-IoT dataset demonstrated that the proposed approach achieves strong balanced detection performance, with a macro F1-score of 89.1% and consistently high minority-class detection rates under severe imbalance conditions. Ablation studies further confirmed the importance of sparsity constraints, class weighting, and hyperparameter optimization in

improving classification robustness. In addition, the low inference latency and modest memory footprint highlight the suitability of the framework for deployment on industrial edge gateways.
Overall, the results indicate that principled hybrid learning architectures can offer an effective trade-off between accuracy, efficiency, and deployability for IIoT intrusion detection. Future work will explore adaptive sparsity mechanisms, online learning for concept drift handling, and validation across additional real-world industrial datasets.

References

[1] S. Hosseininoorbin, S. Layeghy, M. Pakravan, and M. Portmann, "Explainable artificial intelligence for intrusion detection in IoT networks," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12687–12704, Jul. 2023.
[2] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.
[3] J. Li, Y. Liu, and L. Gu, "DDoS attack detection based on neural networks," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 4699–4714, Jan. 2022.
[4] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, "IntruDTree: A machine learning-based cybersecurity intrusion detection model," *Symmetry*, vol. 12, no. 5, Art. no. 754, May 2020.
[5] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, Art. no. 102177, Aug. 2020.
[6] N. Ammar, L. Noirie, and S. Tixeuil, "Intrusion detection in Internet of Things: A systematic literature review," *IEEE Access*, vol. 9, pp. 101490–101523, 2021.
[7] A. Siddiqui and T. Mahmood, "Hybrid deep learning models for network intrusion detection: A comprehensive review," *IEEE Access*, vol. 12, pp. 45678–45701, 2024.
[8] M. K. Hasan et al., "Lightweight encryption technique to enhance medical image security on Internet of Medical Things applications," *IEEE Access*, vol. 13, pp. 15661–15676, Jan. 2025.
[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
[10] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[11] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[12] T. T. H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial Internet of Things intrusion detection systems," *Sustainability*, vol. 14, no. 13, Art. no. 8037, Jun. 2022.
[13] M. Alenazi and S. Mishra, "XGBoost-based intrusion detection system for imbalanced network traffic," *Journal of Cybersecurity and Privacy*, vol. 4, no. 2, pp. 248–267, Jun. 2024.
[14] S. Ghosh, A. Das, R. Porwal, J. M. Ceron, L. Chettri, and S. Bera, "Intrusion detection at the edge for IoT environments," *Sensors*, vol. 23, no. 7, Art. no. 3667, Apr. 2023.
[15] J. H. Joloudari et al., "BERT-deep CNN: State-of-the-art for sentiment analysis of COVID-19 tweets," *Social Network Analysis and Mining*, vol. 13, no. 1, Art. no. 99, Aug. 2023.
[16] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.
[18] C. Zhang et al., "A novel framework design of network intrusion detection based on machine learning techniques," *Security and Communication Networks*, vol. 2022, Art. no. 6408107, 2022.
[19] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa, "A hybrid framework for imbalanced time-series network intrusion detection using ensemble feature selection and deep learning," *IEEE Access*, vol. 11, pp. 109432–109448, 2023.
[20] K. Muhammad and M. A. Khan, "Deep learning-based intrusion detection for IoT networks: A comprehensive survey," *IEEE Access*, vol. 12, pp. 12345–12367, 2024.
[21] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON-IoT datasets," *Sustainable Cities and Society*, vol. 72, Art. no. 102994, Sep. 2021.
[22] V. Swathi and C. S. Bindu, "Sparse autoencoder-based feature learning for intrusion detection in IoT networks," *International Journal of Information Security*, vol. 23, no. 2, pp. 456–478, Jun. 2024.
[23] Y. K. Saheed, T. O. Oladele, and A. O. Akanni, "An efficient hybrid machine learning technique for intrusion detection system," *Journal of Reliable Intelligent Environments*, vol. 10, no. 1, pp. 21–39, Mar. 2024.
[24] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-stage classification approach for IoT intrusion detection based on clustering with oversampling," *Applied Sciences*, vol. 11, no. 7, Art. no. 3022, Mar. 2021.
[25] A. Gueriani, H. Drias, and Y. Drias, "Lightweight intrusion detection system for edge computing in IIoT networks," *Journal of Network and Systems Management*, vol. 33, no. 1, Art. no. 12, Jan. 2025.
[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[27] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1678–1693, Jun. 2021.
[28] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.