

RESEARCH ARTICLE

Self-Supervised One-Class Tabular Deep Learning for Intrusion Detection in Artificial Intelligence of Things Systems

Thavavel Vaiyapuri^{1,*} , Karthiyayini Murugesan²  and Yassine Bouteraa¹

¹ College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Saudi Arabia

² Department of Information Technology, Thiagarajar College of Engineering, India

Abstract: The emergence of the artificial intelligence of things (AIoT) has led to increasingly sophisticated cyberattacks. For many of these threats, labeled training data are scarce or unavailable, which increases the need for label-free intrusion detection systems. To address this issue, this study proposes a self-supervised one-class TabNet framework, termed SS-OC-TabNet, for intrusion detection in tabular network traffic. The framework combines self-supervised representation learning and one-class compactness learning within a two-stage training process. In the first stage, masked-feature reconstruction is used to learn the feature structure of normal traffic from unlabeled data. In the second stage, a hypersphere compactness objective is applied to make the latent representations of normal traffic more compact for anomaly detection. TabNet is used as the backbone to support sequential feature selection, which also improves interpretability. The proposed model was evaluated on the NSL-KDD dataset and further validated through comparison with related works on the ToN-IoT dataset, achieving an accuracy of 93.20% with a false alarm rate of 2.13%. Further, the interpretability results show which traffic features influence the model's decisions. The compact design and low computational complexity of the proposed framework also support energy-efficient deployment in resource-constrained AIoT environments. These results indicate that SS-OC-TabNet is a viable tool for label-free intrusion detection in AIoT environments.

Keywords: cybersecurity, self-supervised learning, one-class learning, representation learning, energy-efficient AIoT

1. Introduction

The exponential increase in artificial intelligence of things (AIoT) devices has expanded the cyberattack surface. Consequently, intrusions have become more frequent and more complex [1]. This trend is shifting the intrusion detection system (IDS) design from rule-based and statistical IDSs to deep learning (DL) models, as they have proven to capture richer representations of network behavior [2–4]. However, the success of the DL models depends on the availability of large labeled training datasets [5]. In real-world deployment, in particular with zero-day or polymorphic attacks, collecting curated labeled datasets is extremely difficult. Therefore, there is growing demand for IDS models that can detect intrusions without large labeled datasets and can learn directly from unlabeled traffic.

In this context, one-class learning (OCL) has gained momentum [6, 7]. It trains the model only on normal traffic. Thereby, the model learns normal traffic and detects deviation as malicious. In real-world deployments, normal traffic evolves due to changes in user behavior, protocol updates, and dynamic operational conditions. Consequently, OCL methods like autoencoder (AE)-based

methods generally create unstable latent embeddings. This hinders their capacity to generalize to unknown attack patterns [7].

In parallel, self-supervised learning (SSL) has shown to learn informative representations without relying on labeled supervision [8, 9]. SSL frameworks employ auxiliary pretext tasks, such as masked-feature reconstruction or contrastive instance prediction, to encourage a model to capture intrinsic correlations within the data. In this manner, SSL can learn semantically meaningful and invariant features that support better generalization. However, on its own, SSL is primarily representation-oriented rather than decision-oriented. It aims to model the internal structure of the data by capturing relationships among features, without explicitly specifying what distinguishes normal behavior from abnormal behavior. Consequently, although SSL-pretrained models can produce coherent latent spaces, they do not inherently define a principled decision boundary for intrusion detection.

This limitation motivates integrating self-supervised representation learning with one-class compactness optimization [10, 11]. SSL enhances the descriptive power of the model with the aid of robust structural embedding learned from unlabeled traffic, while the one-class objective confines normal behavior to a compact subset of the latent space. Therefore, the two paradigms (SSL and OCL) are complementary when used together: SSL alleviates the representation collapse common in OCL, and OCL helps resolve

*Corresponding author: Thavavel Vaiyapuri, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Saudi Arabia. Email: t.thangam@psau.edu.sa

the boundary ambiguity common in SSL [12, 13]. Together, the paradigm provides a unified approach where the model can first determine the manifold of normal behavior and subsequently compress that manifold into a discriminative boundary. Therefore, it improves representation quality and anomaly sensitivity.

This research proposes a new label-free intrusion detection framework named SS-OC-TabNet for dynamic networks. The proposed framework integrates SSL and OCL within a two-stage training design for tabular intrusion data. In the first stage, TabNet is trained with a masked-feature reconstruction task to learn the intrinsic relationships of normal traffic without using attack labels, thereby producing informative latent representations. In the second stage, the same pretrained encoder is refined using a hypersphere compactness objective so that normal samples are clustered more tightly in the latent space. In this way, the proposed framework differs from conventional Deep Support Vector Data Description (SVDD)-style methods, which mainly emphasize compactness learning, by first establishing structured representations through self-supervision and then refining them for one-class discrimination. At the same time, it also differs from generic self-supervised anomaly detection methods that improve representation quality without explicitly forming a compact normality region for detection. Moreover, by adopting TabNet as the backbone, the framework is better suited to tabular network traffic and provides sequential sparse feature selection that improves interpretability. In addition, the compact architecture of the proposed framework supports low-latency and energy-aware deployment in resource-constrained edge AIoT environments. The primary contributions of this paper include:

- 1) A two-stage self-supervised one-class framework called SS-OC-TabNet was presented for label-free intrusion detection in tabular network traffic.
- 2) The framework first uses masked-feature reconstruction loss to learn latent representations of normal traffic. Then, the framework utilizes a hypersphere compactness loss to make these representations more compact.
- 3) The framework utilizes a joint optimization strategy to combine reconstruction and compactness loss to make the normal traffic representations both structured and more compact for intrusion detection.
- 4) The framework adopts TabNet as the backbone encoder to support sequential sparse feature selection for enhanced model interpretability and improved performance for tabular intrusion data.
- 5) Experiments on benchmark datasets validated the effectiveness of the proposed framework for discrimination among latent representations, intrusion detection capabilities, and false positive reductions.

2. Literature Review

The most recent literature on AIoT intrusion detection used a variety of learning-based approaches. Early literature is based on supervised learning, where models are trained using labeled attack data. Examples include adversarially augmented ensemble learning for handling class imbalance [14], time-series models for IoT traffic analysis [15], attention-based architectures for feature-focused detection [16], and DL frameworks for threat intelligence in industrial IoT [17]. Other studies explored feature selection and representation learning methods to improve intrusion detection [18]. These studies achieved strong performance in supervised intrusion detection settings. However, their dependence on labeled attack data restricts their effectiveness in zero-day scenarios.

Apart from these supervised models, some studies introduced mixed intrusion detection pipelines that combine classification components with anomaly detection components. For example, cascaded IDS models combined conventional Machine Learning (ML) with one-class support vector machines (OC-SVM) to reduce false alarms and improve detection [19]. These methods broaden the detection pipeline beyond standard classification, but they still do not fully address label scarcity in dynamic attack environments.

To reduce reliance on labeled data, researchers have explored semi-supervised and adaptive intrusion detection methods. In Reference [20], feature selection and lightweight models were used for resource-constrained environments. These methods improve flexibility and reduce annotation requirements to some extent; however, they still depend on partial supervision or predefined learning strategies and do not fully capture the underlying structure of normal traffic.

Later, unsupervised and OCL approaches were developed to model normal behavior directly from unlabeled data. For example, in Reference [21], AEs were combined with OC-SVM to learn representations for intrusion detection. Generative and adversarial models have also been proposed to learn latent representations and detect deviations from normal patterns [22]. In addition, open-set and reinforcement learning-based methods have been explored for detecting unknown attacks in industrial IoT environments [23]. In Reference [24], OC-SVM combined with Gaussian mixture models was used for anomaly detection. A recent survey highlights the increasing use of one-class methods in industrial anomaly detection tasks [6, 25]. Even though these approaches reduce dependency on labeled data, they often suffer from unstable training, limited feature representation, and lack of built-in interpretability.

Beyond these approaches, hybrid and paradigm-fusion learning frameworks that combine different modeling approaches to enhance generalization and robustness have gained the researchers' attention in recent literature. These frameworks integrate data-driven learning with structured or mechanism-based constraints, allowing models to learn from data while also reflecting the underlying behavior of the system. For example, physics-informed neural networks incorporate domain-specific physical laws into neural network training, ensuring consistency with known system dynamics while learning from data [26]. Similarly, hybrid models [27] that combine constraints with data-driven learning have been used in complex systems, specifically when the available data are limited or noisy.

Collectively, the existing body of IDS research has gradually shifted toward less label-dependent learning paradigms including semi-supervised, unsupervised, and one-class approaches. Despite this transition, several challenges exist. Current methods in the literature typically do not utilize self-supervision to represent feature relationships or combine representation learning with compact latent modeling. Furthermore, interpretability is usually assessed via post hoc analysis rather than being inherently included within each model especially for tabular intrusion data. To address these shortcomings, this study introduces a new unified framework called SS-OC-TabNet that combines the feature representation learning and one-class compactness learning of normal traffic within the TabNet architecture with built-in support for interpretability in tabular intrusion detection.

3. Proposed SS-OC-TabNet Model for IDS

The proposed SS-OC-TabNet framework is a label-free intrusion detection framework that combines self-supervised representation learning and one-class compactness optimization

within a unified architecture. The self-supervised component learns informative representations of normal network traffic, while the compactness component refines them by encouraging normal samples to cluster in the latent space. As a result, the framework supports anomaly detection without attack labels while improving representation quality and interpretability. Figure 1 illustrates the overall workflow of the SS-OC-TabNet framework, which consists of three sequential stages. In the first stage, the encoder is trained to capture relationships among features in network traffic data using a masked-feature reconstruction task. In the second stage, the encoder is further refined using a hypersphere-based compactness objective, encouraging the latent representations of normal traffic to form a tightly clustered region. In the final stage, unseen traffic samples are encoded into the latent space, and their distance from the learned normal center is used as an anomaly score for detection. These stages in the unified framework form a coherent, label-free, and interpretable intrusion detection pipeline. TabNet is employed as the backbone encoder of the proposed framework because its sequential attention mechanism is well suited to learning feature dependencies in tabular network traffic while promoting sparse feature selection [28, 29]. This enables the framework to focus on the most informative traffic attributes, suppress less relevant features, and provide greater transparency in the prediction process. Such properties are especially valuable in security-critical and resource-constrained environments. In addition, its compatibility with structured high-dimensional data makes it suitable for scalable real-time intrusion detection [30].

3.1. Representation learning

The first stage of the proposed SS-OC-TabNet framework learns feature representations from normal training samples only. In this phase, the encoder is trained in a self-supervised manner using unlabeled benign network traffic, without using attack labels. Restricting the training process to normal samples allows the model to learn the intrinsic feature dependencies of benign

traffic patterns, which provides a stable initialization for the subsequent anomaly detection stage. This pretraining also improves generalization and reduces the risk of representation collapse that may occur in unsupervised learning settings [7].

To realize the SSL objective, a masked-feature reconstruction task is adopted [12, 13]. For each input vector $x_i \in \mathbb{R}^d$, a subset of features is randomly masked, and the model reconstructs the masked features from the remaining unmasked attributes. This encourages the encoder to capture relationships among tabular features and to form informative latent representations of normal traffic. The reconstruction loss is defined as [9]

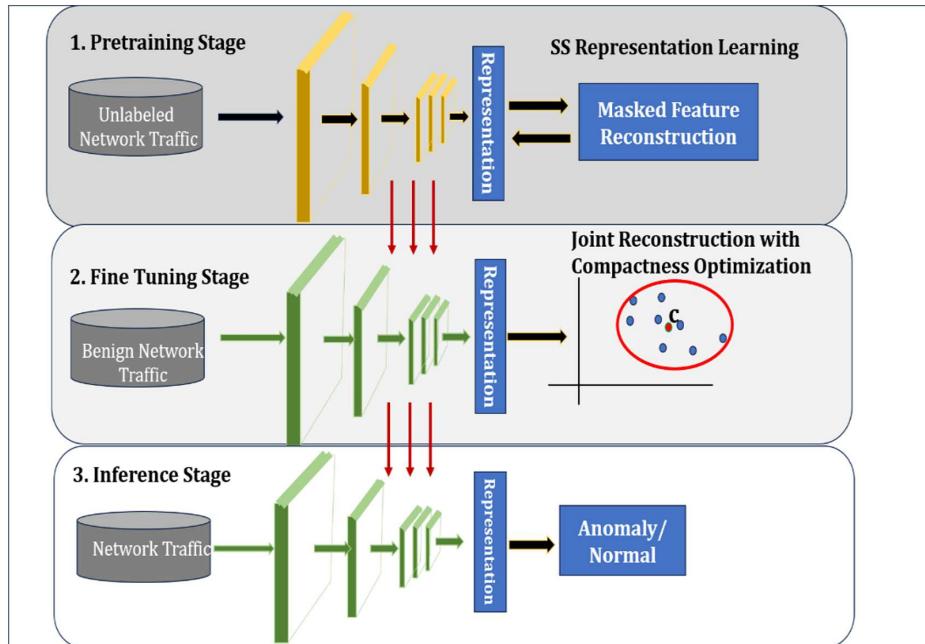
$$L_{SSL} = \frac{1}{N} \sum_{i=1}^N \frac{1}{|M_i|} \left\| x_i^{(M_i)} - \hat{x}_i^{(M_i)} \right\|_2^2 \quad (1)$$

where $M_i \subset \{1, \dots, d\}$ denotes the set of masked-feature indices for the i^{th} sample, $x_i^{(M_i)}$ and $\hat{x}_i^{(M_i)}$ denote the true and reconstructed masked features, respectively, and N is the total number of normal training samples. Minimizing L_{SSL} encourages the encoder to learn local and global correlations among normal traffic attributes. In addition, TabNet's attentive feature selection mechanism emphasizes the most informative unmasked features [29], which helps improve sparsity and interpretability in the learned representations. After convergence, the encoder parameters are denoted by θ , which define the mapping $f_\theta(x)$ from an input sample x to its latent representation $z = f_\theta(x)$. The pre-trained encoder obtained in this stage serves as the foundation for subsequent one-class compactness optimization [11, 31].

3.2. Compactness optimization

The second stage of the proposed SS-OC-TabNet framework fine-tunes the pretrained encoder using the same normal training samples only in order to learn a compact distribution of benign traffic in the latent space. Since only normal samples are used, the encoder is encouraged to map benign traffic representations close to one another, thereby forming a compact normality region that

Figure 1
The proposed SS-OC-TabNet model workflow for intrusion detection in AIoT systems



can later be used to distinguish normal and anomalous behavior [6, 32]. The compactness objective is formulated with respect to a hypersphere center c , where c is defined as the mean of the latent embeddings of the normal training samples. The optimization objective minimizes the mean squared distance between each latent representation and this common center, as follows [21, 33]

$$L_{OC} = \frac{1}{N} \sum_{i=1}^N \|f_{\theta}(x_i) - c\|_2^2, \quad (2)$$

where N denotes the number of normal training samples. This objective can be interpreted as minimizing the intra-class variance of normal samples in the latent space, thereby restricting the support of the learned normal distribution and forming a compact representation of benign behavior. Minimizing L_{OC} enforces intra-class compactness in the latent space by encouraging normal samples to cluster around the center c . Unlike conventional Deep SVDD, which directly imposes compactness on latent representations [34], the proposed framework applies this objective after a self-supervised pretraining stage.

During fine-tuning, the reconstruction objective from the self-supervised pretraining stage is retained and optimized jointly with the compactness objective. This allows the encoder to preserve the informative feature relationships learned during reconstruction while further shaping the latent space so that benign traffic becomes more tightly grouped. The overall fine-tuning objective is therefore defined as [21]

$$L_{\text{joint}} = L_{\text{SSL}} + \lambda_{OC} L_{OC} \quad (3)$$

where L_{SSL} is the masked-feature reconstruction loss defined in the previous subsection, L_{OC} is the one-class compactness loss, and λ_{OC} is a weighting parameter that controls the contribution of the compactness term. The joint optimization of these two objectives allows the model to preserve meaningful feature structure learned during self-supervision while simultaneously reducing the spread of normal samples in the latent space. From a generalization perspective, this combination improves robustness to new unseen attacks as the reconstruction component captures feature relationships in normal traffic and the compactness loss enforces a compact normal region for anomaly detection. As a result, anomalous samples can be more effectively identified during inference based on their deviation from the learned normal region in the latent space.

3.3. Inference mechanism

After the training process, the learned encoder $f_{\theta}(x)$ with the compactness center c is utilized to test the new network traffic. Inference uses the learned encoder to transform a test instance x into its latent space and then measures how much this instance deviates from the learned center c , to determine whether it behaves normally. The anomaly score is determined by the Euclidean distance between the encoded test instance and the center c [7, 13],

$$S(x) = \|f_{\theta}(x) - c\|_2, \quad (4)$$

where $\|\cdot\|_2$ denotes the Euclidean norm. A detection threshold τ is found using the validation dataset to separate normal and anomalous traffic samples. The classification rule is defined as below:

$$\hat{y}(x) = \begin{cases} 1 & \text{if } S(x) > \tau \text{ (anomalous),} \\ 0 & \text{if } S(x) \leq \tau \text{ (normal).} \end{cases} \quad (5)$$

Here, $\hat{y}(x)$ represents the predicted class label. Samples with anomaly scores exceeding τ are identified as intrusions, whereas those below the threshold τ are considered benign. The inference procedure requires only one pass of the encoder, thus providing fast inference and allowing for deployment in real time or other resource-constrained environments. The ability of the model's decision-making process to be explained due to the use of compact representations and attention-based feature selection is another benefit of the approach.

4. Experimental Setup

4.1. Dataset description

The NSL-KDD dataset derived from the benchmark dataset, KDD Cup 1999, was used to evaluate the proposed framework [20, 35]. In this version, duplicate and redundant records from the original KDD dataset were removed to provide a more reliable benchmark for intrusion detection. Each record represents a unique TCP/IP network connection and contains 41 traffic features, including both basic connection attributes and statistical properties of network behavior. In addition, the ToN-IoT dataset [36] was considered to assess the framework on a more recent IoT-oriented intrusion detection benchmark. The ToN-IoT dataset contains telemetry and network traffic data collected from heterogeneous IoT and IIoT environments and reflects modern attack scenarios in connected systems. The data distribution is given in Table 1.

As shown in Table 1, the NSL-KDD training set contains 67,343 normal and 58,630 attack samples, while the testing set contains 9,710 normal and 12,833 attack samples. For ToN-IoT, the training set contains 150,000 records, comprising 41,167 normal and 108,833 attack samples, whereas the testing set contains 50,000 records, including 36,336 normal and 13,664 attack samples. For both datasets, the intrusion detection task was defined as a binary classification problem consisting of normal and attack classes. This binary setting allows the model to learn normal traffic representations and detect deviations as anomalies. It is therefore appropriate for one-class and self-supervised intrusion detection.

Table 1
Data distribution for the study datasets

	Class	Training set	Testing set
NSL-KDD	Normal	67343	9710
	Attack	58630	12833
ToN-IoT	Normal	41167	36336
	Attack	108833	13664

4.2. Model architecture

The proposed SS-OC-TabNet was developed based on an encoder-decoder architecture for self-supervised representation learning. Table 2 summarizes the network configuration for the proposed model by including input and output dimensions with activation functions for each layer. In this table, n is the number of input features, and d is the batch size [10].

The encoder is implemented using a TabNet block that performs feature transformation and attentive masking across multiple sequential decision steps. The block produces a transformed feature representation of dimension 32 along with

Table 2
Network architecture of the proposed model

Layer	Output shape	Activation	# Param
Input	(n, d)	—	0
TabNet block*	(n, 32)	ReLU	12,891
Latent code	(n, 16)	ReLU	528
Dense	(n, 64)	ReLU	1,088
Dense	(n, 64)	ReLU	4,160
Output	(n, d)	Linear	2,665

Note: * The TabNet block includes feature transformer and attentive mask modules, producing three attention masks of shape (n, d).

attention masks over the input features, enabling the model to emphasize informative attributes while suppressing less relevant ones [11]. The obtained feature representation is passed through a linear bottleneck layer with 16 neurons to obtain a compact latent vector.

The decoder reconstructs the input from the latent vector using two fully connected layers with ReLU activation, followed by a final linear layer that restores the original feature dimension (n, d). As shown in Table 2, the architecture maintains the input dimensionality while learning a compact and informative latent representation for anomaly detection.

4.3. Model training setup

All experiments were conducted in the Google Colab Pro environment using a Tesla T4 GPU with 16 GB VRAM, 12 GB RAM, and Python 3.10. The TabNet-based models were implemented in TensorFlow 2.15 using Keras [8]. The training procedure consisted of two stages. In the first stage, the reconstruction-based pretraining model was trained for 50 epochs with a batch size of 128 using the Adam optimizer. In the second stage, the pretrained model was fine-tuned for 15 additional epochs using a joint reconstruction and one-class compactness objective. For fine-tuning, the Adam optimizer was also used with a learning rate of 1×10^{-3} . The one-class loss weight was set to $\lambda_{OC} = 0.01$. Model weights were initialized using Glorot uniform initialization. To improve training stability and limit overfitting, a ReduceLROnPlateau scheduler and early stopping with a patience of 5 epochs were applied. More importantly, during both stages of training, random seeds were fixed to ensure reproducibility.

5. Results and Discussion

This section describes the experiments designed to evaluate the proposed SS-OC-TabNet framework from multiple perspectives and discusses the results to demonstrate its effectiveness for label-free intrusion detection. The analysis focuses on the model's learning stability, detection capability, robustness under different evaluation settings, and interpretability. In addition, the results are compared with related methods to provide a clearer understanding of the strengths of the proposed framework.

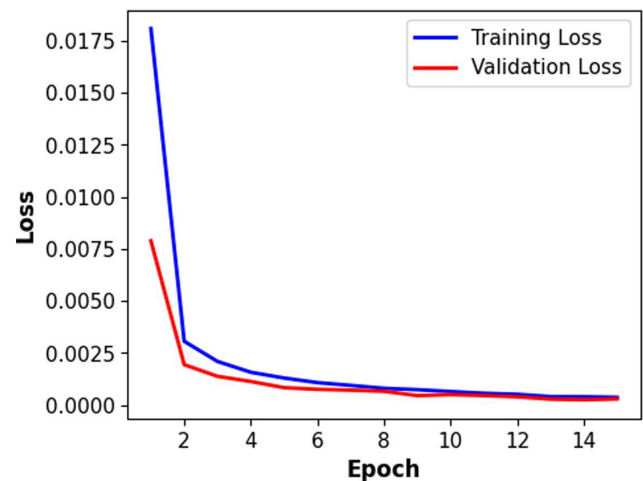
5.1. Model generalization ability analysis

The robustness and stability of the proposed SS-OC-TabNet framework are evaluated by analyzing its training behavior and reconstruction performance.

5.1.1. Learning curve analysis

Learning curves are used to assess training stability and generalization behavior. The trends in training and validation loss provide insight into convergence and potential overfitting [7]. For one-class models such as SS-OC-TabNet, stable learning behavior is important for accurately modeling normal patterns. Figure 2 shows the training and validation loss over 15 epochs. The observation clearly indicates that the two curves follow a similar downward trend throughout training, with no clear divergence. Both losses decrease rapidly in the initial epochs and gradually stabilize as training progresses, showing only minor variation. This behavior indicates stable learning without evident overfitting. The close agreement between the two curves also suggests that the model learns representative patterns of normal traffic without evident overfitting, which is important for reliable intrusion detection.

Figure 2
Learning curve of the proposed model



5.1.2. Reconstruction ability analysis

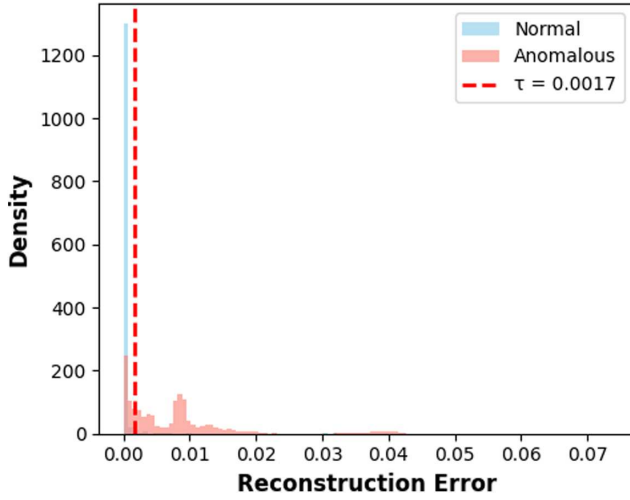
Reconstruction error (RE) analysis is used to evaluate the ability of the model to distinguish normal data from anomalous patterns [7, 21]. In a one-class setting, normal samples are expected to have low RE, while anomalous samples typically produce higher errors due to their deviation from learned patterns.

Figure 3 shows the distribution of REs for normal and anomalous samples. Normal traffic (blue) is concentrated near zero, indicating accurate reconstruction of benign patterns. On the other hand, anomalous samples (red) exhibit a wider distribution with higher error values. The dashed red line ($\tau = 0.0017$) denotes the decision threshold used to distinguish normal and anomalous samples. Most normal samples lie below this threshold, and anomalous samples lie above this threshold. This separation indicates that the model can effectively distinguish abnormal patterns based on RE.

5.2. Structural ablation analysis

In this section, the contribution of each component of the proposed framework is evaluated using a structural ablation analysis under a controlled experimental setting. All variants were evaluated using the optimal configuration described in the experimental setup, with the same encoder architecture, training settings, and hyperparameter values retained throughout. Therefore, the performance differences observed across the variants reflect the contribution of the individual components.

Figure 3
RE distribution of the proposed model



5.2.1. Performance evaluation

Table 3 presents the results of the structural ablation analysis. The unsupervised TabNet model (U-TabNet) serves as the baseline. The U-TabNet + RE variant refers to the baseline with the representation learning component, and the U-TabNet + OC variant refers to the baseline with the OC compactness component. The proposed model includes both components. In addition, the supervised TabNet model (S-TabNet) is included for comparison. Relative to the baseline U-TabNet, the U-TabNet + RE variant increases the accuracy from 0.7017 to 0.8313, the detection rate (DR) from 0.5222 to 0.7569, improves the F1-score from 0.6641 to 0.8363, and reduces the false alarm rate (FAR) from 0.0514 to 0.0278. This reveals that the U-TabNet + RE variant enhances feature quality and improves the separation between normal and anomalous traffic.

Table 3
Performance evaluation results of the proposed model

Variants	ACC	DR	FAR	F1
U-TabNet	0.7017	0.5222	0.0514	0.7569
U-TabNet + OC	0.8296	0.7217	0.0278	0.8282
U-TabNet + RE	0.8313	0.7569	0.0278	0.8363
S-TabNet	0.8707	0.7913	0.0515	0.8546
Proposed	0.8589	0.8074	0.0140	0.8767

A similar improvement trend is observed for the U-TabNet + OC variant. In this case, the accuracy increases from 0.7017 to 0.8296, the DR rises from 0.5222 to 0.7217, the F1-score improves from 0.6641 to 0.8282, and the FAR decreases from 0.0514 to 0.0278. This suggests that the one-class compactness component contributes to a more discriminative feature space by encouraging normal traffic patterns to remain more concentrated, which improves their separation from anomalous samples. The proposed model, which combines both components, achieves the best overall performance among the unsupervised variants, with an accuracy of 85.89%, DR of 80.74%, FAR of 1.40%, and F1-score of 87.67%. These results indicate that these two components are more effective when used jointly, with the representation learning

component improving feature quality and the OC compactness component further regularizing the learned space. As such, their combination leads to stronger discrimination capability and substantially fewer false alarms than either variant alone.

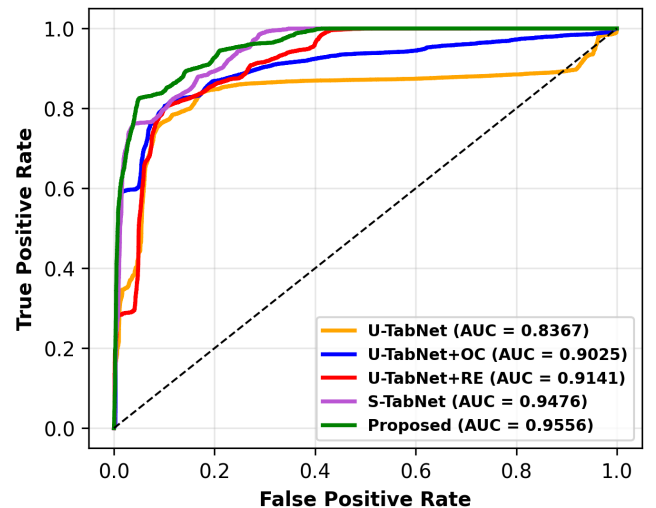
Overall, the proposed model outperforms both U-TabNet + RE and U-TabNet + OC, achieving the best performance among the unsupervised variants. However, when compared with the S-TabNet model, it shows slightly lower accuracy, but it achieves a higher F1 with lower FAR. This characteristic is particularly advantageous in IDS, where excessive false alarms reduce operational effectiveness. Moreover, the proposed SS-OC-TabNet model avoids the need for labeled attack data.

5.2.2. ROC curve analysis

The receiver operating characteristic (ROC) curve assesses classifier performance by plotting the true positive rate against the false positive rate at different decision thresholds. The area under the curve (AUC) assesses the overall discrimination ability of a classifier, with higher values indicating better separation between normal and anomalous traffic [8, 32].

Figure 4 illustrates the ROC curves of all compared models. The proposed model shows a steep increase at low false positive rates and also remains close to the upper-left corner of the ROC plot. This confirms that it can detect anomalous traffic more effectively across different thresholds. S-TabNet also shows a similar pattern. However, the other unsupervised models, U-TabNet + RE and U-TabNet + OC remain below these two curves with U-TabNet showing the weakest curve overall, especially as the false positive rate increases. The same trend is also observed in the AUC values, with the proposed model gaining the highest AUC at 0.9556 among the unsupervised variants. S-TabNet achieves an AUC of 0.9476, while U-TabNet + RE, U-TabNet + OC, and U-TabNet obtain 0.9141, 0.9025, and 0.8367, respectively. This ordering is consistent with the structural ablation results discussed in the previous subsection and confirms that both the representation learning and OC compactness components improve the discrimination capability of the baseline model. Overall, the results indicate that the proposed model achieves the most effective separation between normal and anomalous traffic and offers the best balance between detection performance and false alarms among the unsupervised variants.

Figure 4
ROC analysis results of the proposed model



5.2.3. Latent space behavior analysis

To better understand the role of each component in shaping the latent space, the distributions of reconstruction score, compactness score, and anomaly score are analyzed for normal and anomalous samples. Figure 5(a) shows the reconstruction score distribution obtained from the reconstruction component. Normal samples exhibit lower RE, while anomalous samples produce higher errors. This indicates that the representation learning is effective in capturing the normal traffic behavior. However, the overlap between the two distributions suggests that reconstruction alone does not provide a clear boundary for separating normal and anomalous samples.

Figure 5(b) presents the compactness score distribution from the compactness component. Normal samples are more concentrated around the latent center, whereas anomalous samples are more dispersed. While this improves the organization of the latent space, the separation is still limited because compactness is applied without sufficiently rich feature representations.

Figure 5(c) shows the anomaly score distribution obtained from the proposed model. In this case, the separation between normal and anomalous samples becomes more distinct, with less overlap than that observed in the reconstruction-based and compactness-based variants. Here, the reconstruction component helps the proposed model learn the normal traffic behavior, while the compactness component helps to pull the normal samples close together in the latent space. Thus, these two components, when combined in the proposed model, contribute in different ways to give a more discriminative anomaly score. On the whole,

the latent behavior analysis results are in line with the performance and ROC analyses discussed in earlier subsections, validating that the proposed model outperforms the variants built with only one of the two components.

5.2.4. Statistical significance analysis

To further validate the contribution of each component, a statistical significance analysis was performed using McNemar's test. Figure 6 presents the odds ratios (OR) with confidence intervals for pairwise comparisons between different model variants. An OR greater than 1 indicates that the first model in the comparison makes fewer errors than the second. The comparisons between the proposed model and its ablated variants show consistent improvement. In particular, the proposed model demonstrates clear gains over the unsupervised variants, U-TabNet + RE and U-TabNet + OC, which indicates that combining representation learning with OC compactness leads to fewer misclassifications than using either component independently.

The improvement is even more pronounced when compared with the baseline U-TabNet, where a substantially higher OR is observed. In contrast, the comparison between U-TabNet + RE and U-TabNet + OC indicates that U-TabNet + RE has an advantage over U-TabNet + OC, but the difference is smaller than the gain achieved by the proposed model over either single-component variant. This shows that each component is useful individually, while their combination provides the strongest discrimination performance. The confidence intervals for most comparisons remain above 1, indicating that the observed

Figure 5

Latent space behavior analysis of the proposed model in terms of (a) reconstruction, (b) compactness, and (c) anomaly scores

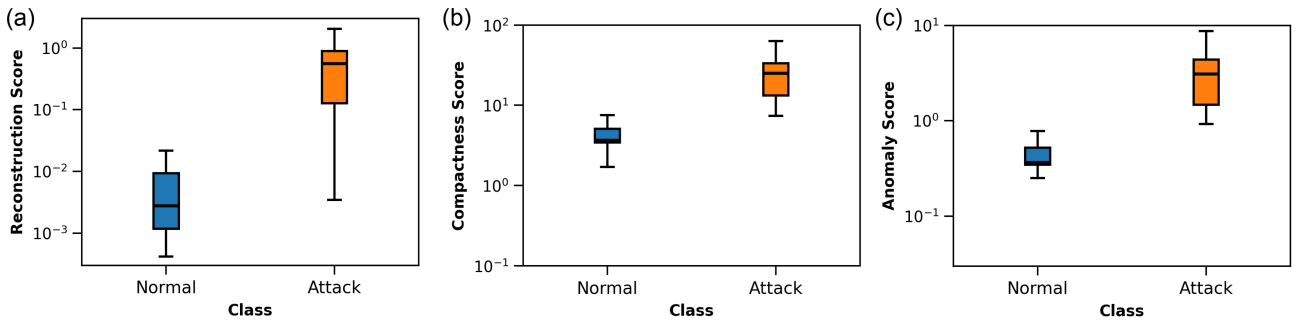
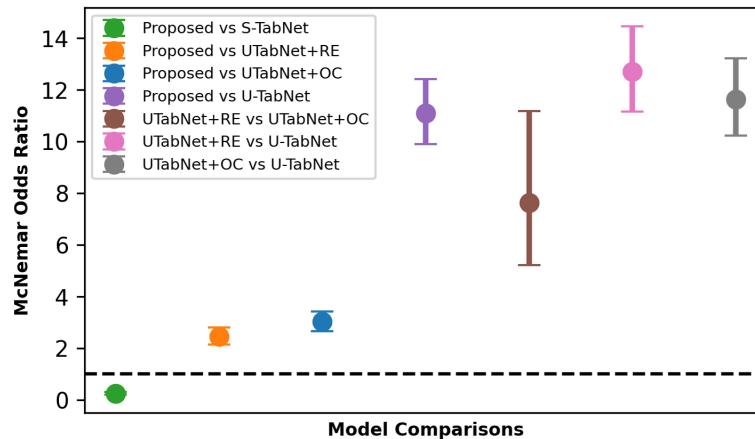


Figure 6

Statistical analysis results of the proposed model



improvements are statistically significant rather than due to random variation. Overall, these results provide statistical evidence that the integration of representation learning and OC compactness produces a complementary effect, leading to more reliable and consistent intrusion detection performance.

5.3. Model classification behavior analysis

This subsection analyzes the classification behavior of the proposed model using confusion matrices and t-distributed stochastic neighbour embedding (t-SNE) visualizations. The analysis highlights class separation and misclassification patterns in comparison with the supervised baseline, S-TabNet. Figure 7(a) depicts the confusion matrix of S-TabNet. Although the model correctly classifies many samples, a noticeable number of misclassifications are observed, particularly for attack samples being predicted as normal. This indicates that while S-TabNet achieves strong performance, its ability to distinguish certain attack patterns from normal traffic remains limited.

Figure 7(b) presents the confusion matrix of the proposed model. The number of correctly classified normal (9266) and attack (10362) samples is higher, while the number of misclassifications is reduced compared to S-TabNet. In particular, fewer normal samples are misclassified as attacks, indicating better control of FAR. However, some attack samples are still classified as normal, suggesting that certain attack patterns are difficult to distinguish from normal traffic. Subsequently, t-SNE is used to project the high-dimensional feature representations into a two-dimensional space for visualization. This helps illustrate how well normal and anomalous samples are separated and where misclassified samples are located. As shown in Figure 7(c) and (d), the t-SNE plots depict the classification behavior of the S-TabNet and the proposed model, respectively. In S-TabNet, the normal

and attack samples are less distinctly separated, and the misclassified samples appear in several regions of the feature space. In the proposed model, the normal samples are grouped more compactly, while the attack samples are positioned farther away from them. The remaining misclassified samples are mostly found in the transition region between the two groups, showing that errors mainly occur for samples that are harder to distinguish. Overall, compared with S-TabNet, the proposed model shows better classification behavior, with improved feature-space separation and fewer misclassifications. This confirms that the learned representations of the proposed model better distinguish between normal and anomalous traffic, resulting in more reliable intrusion detection.

5.4. Model interpretability analysis

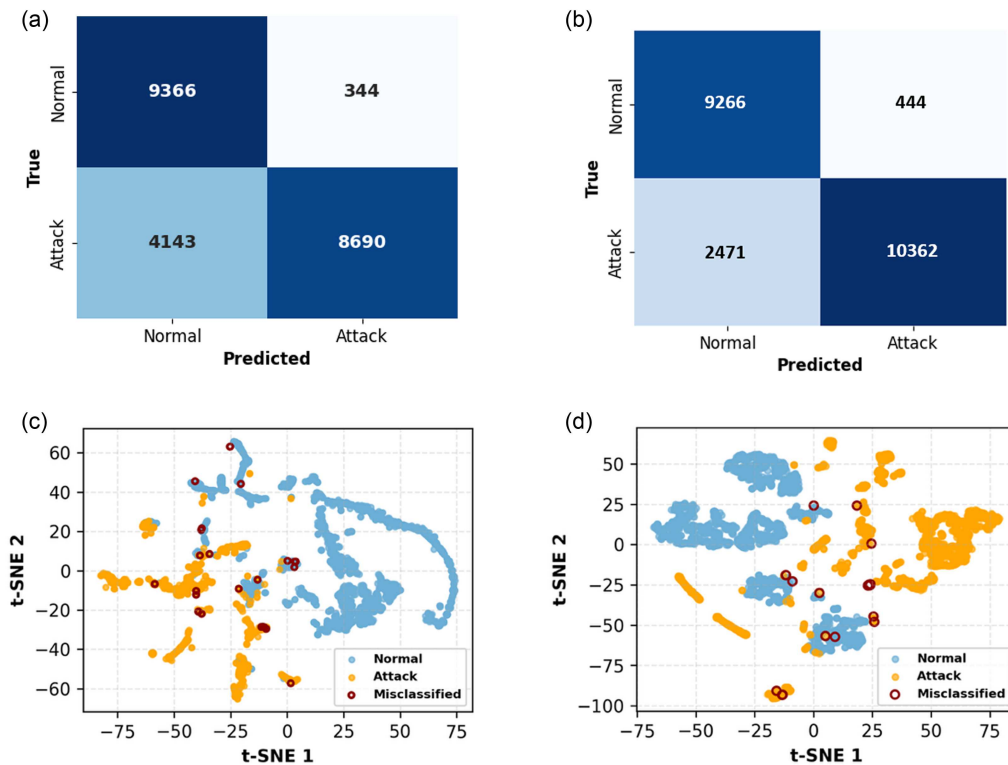
Model interpretability is important for building trust and improving transparency in IDSs, especially in security-critical environments. SHapley Additive exPlanations (SHAP) are utilized to compare the interpretability of the S-TabNet and the proposed model. Global SHAP analysis is used to examine overall feature importance, whereas local SHAP analysis explains feature contributions for individual predictions. The analysis shows which features contribute most to the predictions of each model and how the decision patterns differ between them.

5.4.1. Global summary plot

Global feature importance is evaluated using mean absolute SHAP values to compare the decision behavior of S-TabNet and the proposed SS-OC-TabNet model. This analysis identifies the features that consistently influence model predictions and highlights how the two models differ in their use of traffic features

Figure 7

Classification behavior analysis results. (a) S-TabNet confusion matrix, (b) proposed model confusion matrix, (c) S-TabNet t-SNE plot, and (d) proposed model t-SNE plot



[29, 37, 38]. Figure 8(a) presents the global SHAP feature importance of the supervised S-TabNet model. The highest contributions come from *protocol_type*, *error_rate*, and *service*, indicating that the model mainly relies on protocol and error-related features. The host-based features such as *dst_host_srv_count* and *dst_host_same_srv_rate* also contribute, but with lower importance. This indicates that the model relies mainly on features that separate attack and normal traffic based on the training data used for supervision.

In contrast, Figure 8(b) presents the global SHAP feature importance for the proposed SS-OC-TabNet model. The model assigns higher importance to features that describe traffic behavior, such as *same_srv_rate*, *error_rate*, *srv_error_rate*, and *count*. Furthermore, it is evident that the host-level features such as *dst_host_srv_count* and *dst_host_same_srv_rate* contribute more strongly to the predictions. This reveals that the proposed model utilizes more the features that describe general traffic behavior than the features related to specific attack types. Overall, the two models follow different feature-use patterns. S-TabNet uses more on protocol- and error-related features, whereas the proposed model uses general traffic features. Hence, it is evident that the proposed model identifies malicious traffic by examining how it differs from regular traffic patterns, rather than by depending mainly on features linked to predefined attack classes.

5.4.2. Local force plot

To examine how individual predictions are formed, SHAP force plots are used to provide local, instance-level explanations

for both S-TabNet and the proposed SS-OC-TabNet model. These plots are utilized to examine how individual features contribute to the prediction for a given traffic instance. Features in red push the prediction toward the anomalous class, while features in blue push it toward the normal class [25, 30].

Figure 9(a) presents a representative local explanation from S-TabNet. In this example, the prediction is mainly influenced by *service*, *flag*, *count*, and *srv_count*, which increase the output score. In contrast, features such as *dst_host_same_srv_rate* and *num_outbound_cmds* reduce the score and partially offset this effect. This suggests that the supervised model bases its decision mainly on a small set of influential features. Figure 9(b) presents the local SHAP explanation for the proposed SS-OC-TabNet model. Here, several traffic-related features contribute to the prediction, including *count*, *srv_error_rate*, *dst_host_count*, and *error_rate*. The prediction is influenced by the combined contribution of several host-level and error-related features, rather than by a single dominant feature. This is consistent with the global SHAP results, where the proposed model gives importance to features that describe general traffic behavior. This indicates that the model identifies anomalous traffic by examining general traffic features. Overall, the local SHAP results indicate that S-TabNet makes its prediction mainly from a small number of highly influential features, whereas the proposed SS-OC-TabNet model uses several features related to normal traffic behavior. This agrees with the global SHAP results, where the proposed model places more importance on behavior-related traffic features.

Figure 8
Global summary plot comparing (a) S-TabNet and (b) proposed model for intrusion detection

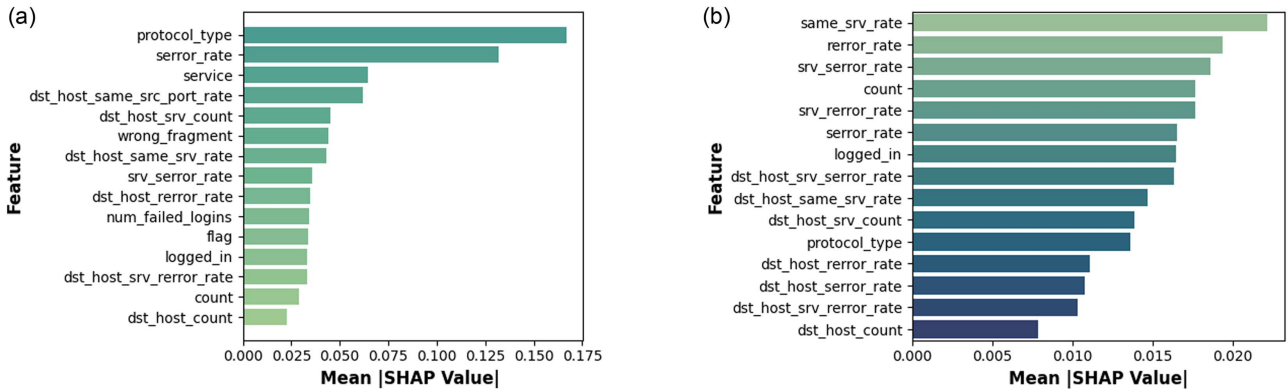
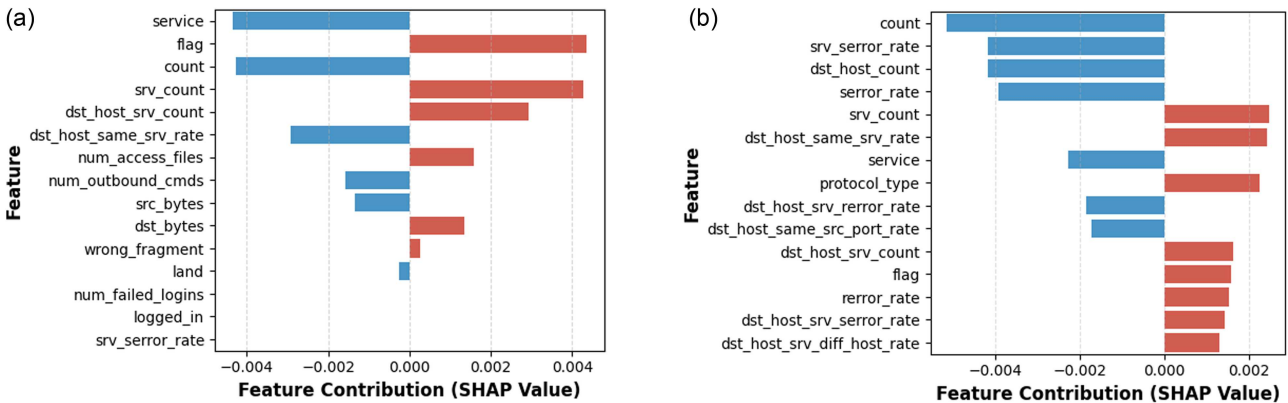


Figure 9
Local force plot comparing (a) S-TabNet and proposed (b) model for intrusion detection



5.4.3. Feature mask importance analysis plot

To understand how the proposed model selects and uses features during decision-making, the feature masks learned at different decision steps are analyzed. TabNet employs a sequential attention mechanism, where different features are emphasized at each step. Figure 10 shows the feature importance across decision steps in the proposed model. In Step 2, features such as *protocol_type*, *srv_count*, and *error_rate* have higher importance, indicating their role in the initial stage of the decision process. In Step 3, features such as *duration*, *land*, *dst_bytes*, and *flag* become more important, indicating that different traffic features are used in the later stage of the decision process. This change across decision steps indicates that the proposed SS-OC-TabNet model does not depend on a static set of features but instead improves its decision by focusing on different features at each stage.

5.5. Deployment feasibility analysis

The feasibility of deploying the proposed model is assessed through its parameter complexity, memory footprint, and inference latency. These aspects determine the applicability of the proposed model in real-time intrusion detection scenarios with constrained resources. As shown in Table 4, the proposed model contains 21,332 parameters, including 20,948 trainable and 384 non-trainable parameters, which indicates a compact design. The model has an estimated size of 0.081 MB, while the saved file size is 0.144 MB. The difference is attributed to the additional structural and serialization information stored with the saved model file. Despite this difference, the storage footprint remains low, making the proposed model suitable for memory-constrained edge environments.

Table 4
Deployment feasibility metrics of the proposed model

Category	Metric	Value
Model complexity	Total parameters	21,332
	Trainable parameters	20,948
	Non-trainable parameters	384
Memory footprint	Model size (MB)	0.081
	Saved model size (MB)	0.144
Inference efficiency	Inference time/sample (ms)	0.263

Inference analysis results in Table 4 show that the model requires 5.93 s in total, with an average processing time of 0.263 ms per sample. This suggests that each traffic sample can be analyzed with very low delay, and this supports its use in time-sensitive intrusion detection settings. The small model size and low inference time show that the proposed framework is suitable for deployment in edge AIoT environments with limited memory and computing resources. Furthermore, the reduced model size and efficient execution support scalable and energy-aware deployment across distributed monitoring nodes without significant overhead.

5.6. Comparison with related works

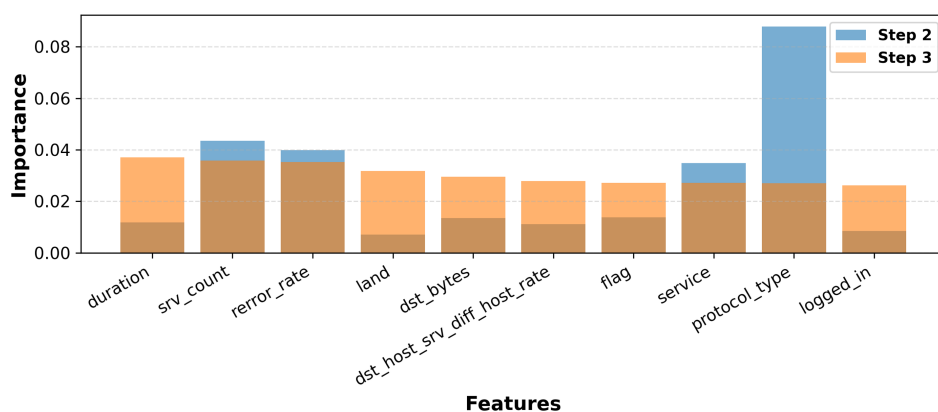
To further assess the proposed model in a realistic intrusion detection scenario, it was evaluated on the ToN-IoT dataset, and its performance was compared with several recent studies as summarized in Table 5. The proposed model shows an accuracy of 93.20%, precision of 93.20%, a recall of 93.71%, and an F1-score of 93.45%, with a FAR of 2.13%, demonstrating reliable detection performance with few false alarms.

Table 5
Comparison with related works on the ToN-IoT dataset

Studies	ACC	PRE	DR	FAR	F1
[15]	99.76	99.76	99.76		99.76
[16]	99.68	99.67	99.68		99.62
[17]	98.52	98.88	98.85		98.46
[18]	98.60	98.35	98.65		98.50
[39]	98.59	96.91	97.38	1.01	97.15
[6]	88.00	85.00	81.00	9.20	83.00
Proposed	93.20	93.20	93.71	2.13	93.45

As shown in Table 5, the methods in References [15–18, 39] achieve higher classification performance than the proposed model. These methods are based on supervised learning, and their performance depends on the availability of labeled training data. In zero-day intrusion detection, such a requirement is restrictive because newly emerging attacks do not have labeled samples during training. A more suitable comparison is with Reference [6], which also considers a one-class detection setting. In this case, the proposed model outperforms [6] in accuracy, precision, recall, and F1-score and lowers the FAR from 9.20 to 2.13. These results

Figure 10
Feature mask importance plot across decision steps in the proposed model



highlight the effectiveness of the proposed model in minimizing false alarms and improving the reliability of intrusion detection.

From the above findings, the proposed model can be considered effective for anomaly-oriented intrusion detection. Although supervised methods report higher classification scores, their performance depends on labeled training data. In contrast, the proposed model outperforms other one-class methods, particularly in terms of FAR. Hence, these findings suggest that the proposed model can provide competitive detection performance in a more practical zero-day setting.

6. Conclusion

This study proposed a self-supervised one-class framework named SS-OC-TabNet to address the problem of limited labeled data in intrusion detection for AIoT environments. The proposed framework combines self-supervised representation learning with one-class compactness optimization within a unified TabNet-based architecture, allowing normal traffic patterns to be learned effectively without the need for labeled attack data.

The extensive experimental evaluation demonstrated that the proposed model achieved strong intrusion detection performance with low false alarms compared to other unsupervised approaches while maintaining performance comparable to supervised methods. The ablation, ROC, latent space, and interpretability analyses further demonstrated that using the two learning stages together improved the distinction between normal and anomalous traffic and increased the interpretability of the detection process. In summary, these findings indicate that the proposed SS-OC-TabNet provides an effective and interpretable solution for intrusion detection in AIoT environments where labeled attack data are limited or unavailable. The compact model size and low inference latency also indicate its potential for energy-efficient deployment in edge AIoT environments.

The future scope of this work includes evaluating the proposed framework on more diverse and realistic datasets to further examine its generalization ability. Another direction for future work is to study multicenter compactness objectives for complex normal traffic distributions and drift-aware adaptation strategies for evolving traffic conditions. A further scope of this work lies in validating the proposed framework on edge-based platforms to assess its practical suitability with respect to processing efficiency, energy demand, and scalability in AIoT applications.

Funding Support

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/37756).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The KDD CUP 99 datasets that support the findings of this study are openly available at <https://doi.org/10.1109/CISDA.2009.5356528>, reference number [35] and the Network TON_IoT datasets are openly available at <https://doi.org/10.1016/j.scs.2021.102994>, reference number [36].

Author Contribution Statement

Thavavel Vaiyapuri: Conceptualization, Methodology, Formal analysis, Writing – original draft, Supervision, Funding acquisition. **Karthiyayini Murugesan:** Methodology, Software, Resources, Writing – original draft. **Yassine Bouteraa:** Software, Validation, Formal analysis, Investigation, Resources, Writing – review & editing, Visualization, Project administration.

References

- [1] Jamshidi, S., Nikanjam, A., Nafi, K. W., Khomh, F., & Rasta, R. (2025). Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review. *Internet of Things*, 31, 101531. <https://doi.org/10.1016/j.iot.2025.101531>
- [2] Ali, Z., Tiberti, W., Marotta, A., & Cassioli, D. (2025). Deep transfer learning for intrusion detection in edge computing scenarios. *IEEE Internet of Things Journal*, 12(21), 44882–44896. <https://doi.org/10.1109/JIOT.2025.3597892>
- [3] Albalwy, F., & Almohaimeed, M. (2025). Advancing artificial intelligence of things security: Integrating feature selection and deep learning for real-time intrusion detection. *Systems*, 13(4), 231. <https://doi.org/10.3390/systems13040231>
- [4] Jacob, S. L., & Habibullah, P. S. (2025). A systematic analysis and review on intrusion detection systems using machine learning and deep learning algorithms. *Journal of Computational and Cognitive Engineering*, 4(2), 108–120. <https://doi.org/10.47852/bonviewJCCE42023249>
- [5] Laddi, M., Allagi, S., Rachh, R., Sambrekar, K., & Athanikar, S. (2025). An advanced cyber security model using federated machine learning approach for intrusion detection in networks. *Journal of Computational and Cognitive Engineering*, 4(2), 223–235. <https://doi.org/10.47852/bonviewJCCE42023751>
- [6] Paolini, D., Dini, P., Soldaini, E., & Saponara, S. (2025). One-class anomaly detection for industrial applications: A comparative survey and experimental study. *Computers*, 14(7), 281. <https://doi.org/10.3390/computers14070281>
- [7] Vaiyapuri, T., & Binbusayyis, A. (2020). Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: A comparative evaluation. *PeerJ Computer Science*, 6, e327. <https://doi.org/10.7717/peerj-cs.327>
- [8] Kumar, P. J., & Subbu, N. (2025). BSA-SGRU-A novel deep learning framework for alleviate the multiple attacks in IoT-Cloud environment. *Journal of Computational and Cognitive Engineering*, 4(3), 309–318. <https://doi.org/10.47852/bonviewJCCE42023061>
- [9] Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., & Tao, D. (2024). A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), 9052–9071. <https://doi.org/10.1109/TPAMI.2024.3415112>
- [10] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2023). Self-supervised learning: Generative or

- contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1), 857–876. <https://doi.org/10.1109/TKDE.2021.3090866>
- [11] Ju, H., Lee, D., Kang, S., & Yu, H. (2022). Mitigating viewpoint sensitivity of self-supervised one-class classifiers. *Information Sciences*, 611, 225–242. <https://doi.org/10.1016/j.ins.2022.08.042>
- [12] Yang, Z., Huang, Y., Nazeer, F., Zi, Y., Valentino, G., Li, C., . . . , & Huang, H. (2023). A novel fault detection method for rotating machinery based on self-supervised contrastive representations. *Computers in Industry*, 147, 103878. <https://doi.org/10.1016/j.compind.2023.103878>
- [13] Chatterjee, S., Göksel, O., Sladoje, N., & Lindblad, J. (2025). Detection of extremely sparse key instances in whole slide cytology images via self-supervised one-class representation learning. In *Pattern Recognition: 27th International Conference*, 408–421. https://doi.org/10.1007/978-3-031-78398-2_27
- [14] Ren, H., Tang, Y., Ren, S., Wang, R., & Dong, W. (2025). ADHS-EL: Dynamic ensemble learning with adversarial augmentation for accurate and robust network intrusion detection. *Journal of King Saud University Computer and Information Sciences*, 37(1), 7. <https://doi.org/10.1007/s44443-025-00015-4>
- [15] bin, Hanafiah, I. M., Pang, Y. H., Zarakovitis, C., Lim, H. S., Skordoulis, D., Nikolakakou, C. D., Hiew, F. S., . . . Hiew, F. S. (2026). TSM-NIDS: A time-series mixer-based intrusion detection system for IoT networks. *MethodsX*, 16, 103885. <https://doi.org/10.1016/j.mex.2026.103885>
- [16] Elwhishi, A., Younis, A. A., & Akhonzada, A. (2026). Attention-enhanced hybrid architecture for efficient intrusion detection in industrial IoT. *IEEE Open Journal of the Communications Society*, 7, 1330–1347. <https://doi.org/10.1109/OJCOMS.2026.3661768>
- [17] Malik, J., Akhonzada, A., Al-Shamayleh, A. S., Zeadally, S., & Almogren, A. (2025). Hybrid deep learning based threat intelligence framework for Industrial IoT systems. *Journal of Industrial Information Integration*, 45, 100846. <https://doi.org/10.1016/j.jii.2025.100846>
- [18] Dharini, N., Janani, V. S., & Katiravan, J. (2026). Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach. *Scientific Reports*, 16(1), 7763. <https://doi.org/10.1038/s41598-026-37834-y>
- [19] Ahamed, M. K. U., & Karim, A. (2025). Cascaded intrusion detection system using machine learning. *Systems and Soft Computing*, 7, 200182. <https://doi.org/10.1016/j.sasc.2024.200182>
- [20] Anagha, A. S., Thomas, C., & Balakrishnan, N. (2025). Optimized intrusion predictions through feature selection methods. *Computers & Security*, 157, 104541. <https://doi.org/10.1016/j.cose.2025.104541>
- [21] Binbusayyis, A., & Vaiyapuri, T. (2021). Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. *Applied Intelligence*, 51(10), 7094–7108. <https://doi.org/10.1007/s10489-021-02205-9>
- [22] Li, Z., Wang, P., & Wang, Z. (2024). FlowGANAnomaly: Flow-based anomaly network intrusion detection with adversarial learning. *Chinese Journal of Electronics*, 33(1), 58–71. <https://doi.org/10.23919/cje.2022.00.173>
- [23] Yu, S., Zhai, R., Shen, Y., Wu, G., Zhang, H., Yu, S., & Shen, S. (2024). Deep Q-network-based open-set intrusion detection solution for industrial Internet of Things. *IEEE Internet of Things Journal*, 11(7), 12536–12550. <https://doi.org/10.1109/JIOT.2023.3333903>
- [24] Wang, C., Sun, Y., Lv, S., Wang, C., Liu, H., & Wang, B. (2023). Intrusion detection system based on one-class support vector machine and Gaussian mixture model. *Electronics*, 12(4), 930. <https://doi.org/10.3390/electronics12040930>
- [25] Vaiyapuri, T., & Murugesan, K. (2026). Unsupervised one-class learning for IoT intrusion detection: A review of methods and prospects. *International Journal of Computer Networks and Applications*, 13(1), 179–199. <https://doi.org/10.22247/ijcna/2026/10>
- [26] Valderrama, D. F., Gaggero, G. B., Ferro, G., Mokarim, A., Robba, M., Girdinio, P., & Marchese, M. (2026). An online intrusion detection system for photovoltaic generators through physics-based neural networks. *Electric Power Systems Research*, 253, 112528. <https://doi.org/10.1016/j.epr.2025.112528>
- [27] Suprabhath, K. S., Machina, V. S. P., Madichetty, S., & Mishra, S. (2025). Securing DC microgrids against cyber-attacks: Hybrid physics-informed neural network control strategy with real-time implementation. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 13(5), 6292–6304. <https://doi.org/10.1109/JESTPE.2025.3574175>
- [28] Al-Harbi, O. (2025). Enhancing intrusion detection through anomaly detection and integrated deep learning with Tab-Transformer. *International Journal of Sensor Networks*, 48(3), 166–177. <https://doi.org/10.1504/IJSNET.2025.147642>
- [29] Fares, I. A., & Abd Elaziz, M. (2025). Explainable TabNet transformer-based on Google Vizier optimizer for anomaly intrusion detection system. *Knowledge-Based Systems*, 316, 113351. <https://doi.org/10.1016/j.knosys.2025.113351>
- [30] Naseer, M., Ullah, F., Saeed, S., Algarni, F., & Zhao, Y. (2025). Explainable TabNet ensemble model for identification of obfuscated URLs with features selection to ensure secure web browsing. *Scientific Reports*, 15(1), 9496. <https://doi.org/10.1038/s41598-025-93286-w>
- [31] Uddin, M. A., Aryal, S., Bouadjenek, M. R., Al-Hawawreh, M., & Talukder, M. A. (2025). A dual-tier adaptive one-class classification IDS for emerging cyberthreats. *Computer Communications*, 229, 108006. <https://doi.org/10.1016/j.comcom.2024.108006>
- [32] Hu, B., Bi, Y., Zhi, M., Zhang, K., Yan, F., Zhang, Q., & Liu, Z. (2022). A deep one-class intrusion detection scheme in software-defined industrial networks. *IEEE Transactions on Industrial Informatics*, 18(6), 4286–4296. <https://doi.org/10.1109/TII.2021.3133300>
- [33] Yang, Q., Chen, Y.-A., & Shi, Y. (2025). Multi-view cluster structure guided one-class BLS-autoencoder for intrusion detection. *Applied Sciences*, 15(14), 8094. <https://doi.org/10.3390/app15148094>
- [34] Algarni, A., Ahmad, Z., & Ala'anzy, M. A. (2024). An edge computing-based and threat behavior-aware smart prioritization framework for cybersecurity intrusion detection and prevention of IEDs in smart grids with integration of modified LGBM and one class-SVM models. *IEEE Access*, 12, 104948–104963. <https://doi.org/10.1109/ACCESS.2024.3435564>
- [35] Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6. <https://doi.org/10.1109/CISDA.2009.5356528>

- [36] Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72, 102994. <https://doi.org/10.1016/j.scs.2021.102994>
- [37] Hermosilla, P., Díaz, M., Berrios, S., & Allende-Cid, H. (2025). Use of explainable artificial intelligence for analyzing and explaining intrusion detection systems. *Computers*, 14(5), 160. <https://doi.org/10.3390/computers14050160>
- [38] Arreche, O., Guntur, T. R., Roberts, J. W., & Abdallah, M. (2024). E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection. *IEEE Access*, 12, 23954–23988. <https://doi.org/10.1109/ACCESS.2024.3365140>
- [39] Duan, G., Lv, H., Wang, H., Feng, G., & Li, X. (2024). Practical cyberattack detection with continuous temporal graph in dynamic network system. *IEEE Transactions on Information Forensics and Security*, 19, 4851–4864. <https://doi.org/10.1109/TIFS.2024.3385321>

How to Cite: Vaiyapuri, T., Murugesan, K., & Bouteraa, Y. (2026). Self-Supervised One-Class Tabular Deep Learning for Intrusion Detection in Artificial Intelligence of Things Systems. *Journal of Computational and Cognitive Engineering*. <https://doi.org/10.47852/bonviewJCCE62027931>