Mitigating Cross-Domain Performance Degradation in Time-Series NIDS via LoRA

Ji-Hyun Choi , Seok-Won Hong , Hyeon-Jin Jung , Seok-Hwan Choi *

Posted Date: 22 May 2026

doi: 10.20944/preprints202605.1559.v1

Keywords: NIDS; LoRA; domain adaptation; time-series traffic analysis

 doi:10.20944/preprints202605.1559.v1

Article

Mitigating Cross-Domain Performance Degradation in Time-Series NIDS via LoRA

Ji-Hyun Choi [1], Seok-Won Hong [1], Hyeon-Jin Jung [1] and Seok-Hwan Choi [2,*]

[1] Dept of Computer Science, Yonsei University, Wonju 26493, Republic of Korea
[2] Dept of Software, Yonsei University, Wonju 26493, Republic of Korea
* Correspondence: sh.choi@yonsei.ac.kr

Abstract

Network intrusion detection systems (NIDS) play a crucial role in modern network environments where diverse and rapidly evolving traffic patterns are observed. Although deep learning-based NIDS have demonstrated strong performance within specific datasets, their effectiveness significantly degrades when applied to unseen network environments due to domain discrepancies. In this paper, we first experimentally demonstrate the performance degradation of time-series-based NIDS under cross-domain conditions using multiple benchmark datasets. Then, we propose a LoRA-based domain adaptation framework for time-series-based NIDS models. Instead of retraining the entire model, the proposed approach freezes the backbone network and applies low-rank updates to selected layers, enabling parameter-efficient adaptation to new domains. Experimental results show that the proposed method consistently improves cross-domain detection performance across multiple dataset combinations, particularly in terms of recall, while requiring only a small number of additional parameters.

Keywords: NIDS; LoRA; domain adaptation; time-series traffic analysis

1. Introduction

Network Intrusion Detection Systems (NIDS) have been actively studied as a core technology for securing modern infrastructure by identifying anomalous behaviors from network traffic [1–3]. However, modern network attacks are becoming increasingly difficult to distinguish based solely on the statistical features of individual packets or flows. In particular, many threats manifest incrementally through temporal traffic variations and behavioral patterns that accumulate over time. This creates security blind spots for conventional static analysis frameworks that treat traffic as isolated observations.

Consequently, modern NIDS are transitioning toward time-series-based approaches that model network traffic as continuous sequences. By modeling temporal dependencies between current observations and their past context, time-series-based NIDS can analyze the broader flow and historical context of malicious activities. This allows them to capture not only short-term statistical anomalies but also the continuity of actions and evolutionary patterns in traffic. As a result, time-series-based NIDS provide a more dynamic and sophisticated detection capability than static packet/flow-based and rule-based NIDS. To effectively learn such temporal dependencies, various neural network models have been adopted in time-series-based NIDS, including RNN, LSTM, and GRU [4–6]. These recurrent architectures retain information from previous time steps and model sequence patterns over time, enabling temporal representation of network traffic sequences.

However, these characteristics can also pose structural limitations for time-series-based NIDS. Since such systems often learn dataset-specific temporal patterns, their performance depends heavily on the traffic structure and protocol interactions observed during training. Consequently, their performance can degrade sharply when deployed in heterogeneous network environments that differ

 doi:10.20944/preprints202605.1559.v1

from the training domain. For example, a system optimized for enterprise networks often performs poorly in wireless or IoT environments, making it necessary to adapt the model to each new domain rather than relying on direct transfer alone. To address this issue, recent studies have explored various cross-domain adaptation approaches, including feature standardization across datasets, federated learning frameworks, and ensemble-based architectures. However, these approaches often require complex training procedures or large model architectures, resulting in inefficient deployment in resource-constrained environments.

In this paper, to demonstrate the performance degradation of time-series-based NIDS under heterogeneous network conditions, we first conduct a cross-domain performance analysis using multiple benchmark datasets. Specifically, we train RNN- and LSTM-based models on four benchmark datasets with distinct attack scenarios and feature sets [7]. By evaluating each pre-trained model on the remaining unseen datasets, we show that detection performance drops significantly when the model is exposed to unfamiliar traffic patterns and protocol interactions. To address this degradation, we then propose a LoRA-based cross-domain adaptation method that selectively adjusts the fully connected layers of time-series-based NIDS models. Instead of retraining the entire model, we freeze the backbone responsible for temporal feature extraction and apply LoRA modules only to the fully connected layers that determine classification decisions. This design enables efficient adaptation to new network environments by preserving temporal representations learned by the backbone while adjusting the classification boundaries to the target domain.

The main contributions of this paper are summarized as follows:

- **Analysis of cross-domain performance degradation**: We demonstrate that time-series-based NIDS suffer from substantial performance degradation when models trained on a source domain are evaluated on unseen target datasets with different traffic distributions.
- **LoRA-based cross-domain adaptation for time-series-based NIDS**: We propose a LoRA-based cross-domain adaptation method for time-series-based NIDS to mitigate performance degradation under domain shift through parameter-efficient adaptation without full model retraining. To the best of our knowledge, this is the first work that applies LoRA to time-series-based NIDS in a cross-domain adaptation setting.
- **Experimental validation across multiple network datasets**: From the experimental results using multiple network datasets, we show that the proposed LoRA-based adaptation method consistently improves cross-domain detection performance while maintaining parameter efficiency.

The rest of the paper is organized as follows: In Section 2, we provide preliminaries on time-series based NIDS and LoRA, followed by a review of related works on domain adaptation. In Section 3, we describe the data preprocessing procedure and alignment process, then we analyze cross-domain performance degradation across different datasets. In Section 4, we present a LoRA-based domain adaptation method. In Section 5 we provide the implementation details and evaluate the effectiveness of the proposed method through cross-domain experiments. Finally, we conclude the paper in Section 6.

2. Preliminaries and Related Work

2.1. Time-Series-Based NIDS

Time-series-based NIDS models network traffic as time-ordered input sequences to capture temporal correlations and dynamic variability beyond individual packet analysis. Specifically, traffic statistics are segmented by a sliding window into fixed-length sequences of length T, denoted as $X = \{x_1, x_2, \ldots, x_T\}$, where $x_t \in \mathbb{R}^d$. Here, x_t represents the feature vector at time t, and d denotes the feature dimension. These sequences are then processed by recurrent architectures, such as RNN and LSTM, that model sequential dependencies [8]. Typically, such architectures maintain temporal context by updating the hidden state using the current input and the previous hidden state, i.e., $h_t = f(x_t, h_{t-1})$, where $f(\cdot)$ denotes the recurrent operation. A classifier is then applied to the last hidden state h_T or to an aggregated output over $\{h_t\}_{t=1}^{T}$. Through this process, intrusion detection

 doi:10.20944/preprints202605.1559.v1

is formulated as a binary or multi-class classification problem that determines whether the input sequence is normal or malicious.

Previous studies have shown that recurrent architectures are effective for modeling temporal patterns in network traffic [8]. For example, Yin et al. proposed a deep learning-based intrusion detection system (RNN-IDS) that utilizes recurrent neural networks to model network traffic as sequential data [4]. Unlike traditional machine learning approaches, the RNN architecture introduces recurrent connections that enable the model to retain information from previous time steps. This allows the model to capture temporal dependencies in network traffic and improves intrusion detection performance in both binary and multi-class classification tasks. Kim et al. proposed an LSTM-based intrusion detection system to address the limitations of conventional RNNs in learning long-term dependencies in network traffic [5]. By mitigating the gradient-related issues in standard RNNs, the model incorporates LSTM cells with input, forget, and output gates, allowing it to selectively retain and propagate relevant temporal information. As a result, the model can effectively capture sequential patterns in network traffic and enhance detection performance across various attack types.

2.2. LoRA

LoRA (Low-Rank Adaptation) is a representative parameter-efficient fine-tuning (PEFT) technique designed to mitigate the inefficiency of full-parameter fine-tuning [9], [10]. The core idea is to freeze a pre-trained weight matrix W_0 and approximate task-specific updates using the product of two small low-rank matrices, B and A. Specifically, the adapted weight is expressed as:

$$W = W_0 + BA, \tag{1}$$

where only the low-rank matrices A and B are updated during training. This design is motivated by the observation that task-specific adaptation can be achieved with a limited number of parameter adjustments, without updating the entire parameter space. As a result, LoRA reduces the number of trainable parameters and the associated training overhead, leading to lower memory consumption and faster training. Moreover, since the backbone parameters remain shared and only the LoRA modules need to be adapted or replaced for each target domain, LoRA enables efficient domain-specific adaptation. During inference, LoRA operates by merging the low-rank correction term into the pre-trained weights, thereby introducing negligible overhead. These properties make LoRA well suited to cross-domain adaptation in NIDS, where models must be efficiently adapted to diverse and evolving network environments.

2.3. Related Work

Most studies on NIDS evaluate models by splitting a single dataset into training and testing sets. In such settings, the training and testing data belong to the same domain. However, this approach has limitations in assessing whether a model can maintain reliable detection performance in new network environments that are not represented during training. Even within the same domain, temporal changes in traffic distribution can degrade the performance of NIDS trained on historical data. These limitations motivate the need for effective cross-domain adaptation methods that can transfer intrusion detection models to new network environments. According to prior research, AI-based NIDS tends to show a significant decline in detection performance when evaluated on data outside the training domain. Cantone et al. demonstrated that existing machine learning-based NIDS suffer from limited cross-dataset generalization due to dataset heterogeneity [11]. This result highlights the need for methods that can adapt intrusion detection models to new environments more effectively. Existing efforts to address this problem have mainly focused on feature standardization, federated learning, and ensemble-based modeling. For example, Sarhan et al. applied a standardized feature set based on NetFlow and CICFlowMeter across multiple NIDS datasets [12]. They showed that a common feature set can improve both cross-dataset detection performance and explainability in machine learning-based NIDS. In addition, de Carvalho Bertoli et al. proposed a stacked structure combining

 doi:10.20944/preprints202605.1559.v1

unsupervised Autoencoders (AE) and Energy Flow Classifiers, trained via Federated Learning (FL) [13]. Their approach achieved improved cross-domain intrusion detection performance without requiring direct data sharing. Furthermore, Amin et al. proposed a stacked ensemble structure combining heterogeneous deep learning models to enhance detection performance across different network domains [14]. Their results showed that ensemble-based architectures can improve cross-domain intrusion detection performance compared to single-model methods.

3. Cross-Domain Performance Analysis

In this section, we experimentally analyze the cross-domain performance degradation of time-series-based NIDS.

3.1. Data Alignment and Preprocessing

3.1.1. Datasets

In this paper, we selected four datasets with diverse attack scenarios and feature configurations: KDD99 [15], [16], UNSW_NB15 [17], [18], CICIDS2017 [19], and AWID3 [20], [21]. These datasets are chosen because they span heterogeneous traffic environments, attack scenarios, and feature spaces, thereby providing an appropriate benchmark for analyzing cross-domain performance degradation in time-series-based NIDS. The characteristics of each dataset are summarized as follows:

- **KDD99** (Stolfo et al., 1999): A classic benchmark based on the 1998 DARPA evaluation data. It consists of 41 features and a label, and its attacks are categorized into four major groups comprising 22 specific attack types. A key characteristic of KDD99 is the distributional discrepancy between its training and test sets. Specifically, the test set includes 14 novel attack types that do not appear in the training set.
- **UNSW_NB15** (Moustafa et al., 2015): It was developed at the UNSW Canberra Cyber Range Lab to address the limitations of legacy datasets. It includes hybrid traffic that combines real-world normal activities with synthetically generated attack behaviors. The dataset consists of 49 features and nine attack classes, and it reflects diverse and sophisticated modern threats compared to traditional datasets.
- **CICIDS2017** (Sharafaldin et al., 2017): It was created by the Canadian Institute for Cybersecurity and includes recent attack scenarios collected in a realistic environment. It contains 79 features and 14 attack classes, and the traffic was collected under diverse operating systems and network settings.
- **AWID3** (Chatzoglou et al., 2021): It is a state-of-the-art wireless NIDS dataset that extends the previous AWID2 corpus. It captures diverse attacks in an IEEE 802.1X EAP environment across more than 10 different client devices and cloud-based applications. The dataset contains 254 features manually extracted from both MAC and application layers, which enables detailed analysis of wireless network intrusions.

3.1.2. Data Preprocessing

In this section, we describe a unified preprocessing pipeline that is applied to all datasets before training to ensure consistent feature-space alignment and reproducibility in cross-domain experiments. This step is particularly important because cross-domain evaluation is highly sensitive to inconsistencies in feature definitions and data formats across datasets.

First, we standardized the feature schema by removing unnecessary whitespace from column names and labels. This step prevents semantically identical features from being treated as distinct variables and ensures uniform feature representation across datasets. Next, we corrected invalid numerical values in the raw traffic features. Since negative values were treated as invalid observations in our preprocessing setting, we replaced all negative entries with 0. In addition, after converting infinity (Inf) values to NaN, we removed columns containing NaN values and duplicate records to avoid numerical instability during training. Then, we encoded categorical features into integer

 doi:10.20944/preprints202605.1559.v1

representations using LabelEncoder. This avoids the dimensionality increase caused by one-hot encoding and helps maintain a consistent feature representation across heterogeneous datasets. Finally, we normalized numerical features to the range of [0,1] for stable optimization and balanced feature scaling. Here, we excluded integer-encoded categorical attributes and labels from normalization. For labeling, we adopted a binary labeling scheme by mapping normal traffic to 0 and all attack traffic to 1. We also stored the feature schema and preprocessing parameters as metadata so that the same preprocessing rules could be applied consistently across all source and target domains.

3.1.3. Synchronization Strategies

Since the datasets used in this study were collected from different environments and protocols, their features and formats are not fully aligned. This heterogeneity makes it difficult to train and evaluate time-series-based NIDS using a consistent input structure. Thus, we defined the dataset used to train the backbone model as the reference dataset and used its feature schema as the standard input structure. Based on the reference feature schema, we established feature mapping rules to align all other target datasets to the same input structure. Specifically, we first identified features that share the same semantics but are represented under different column names. We then classified these features into high-level groups based on protocols and network behaviors, and manually mapped columns that serve the same functional role in traffic representation. For features without a clear match, the corresponding inputs were zero-filled to preserve the reference input structure. This strategy maintains dimensional consistency without introducing additional handcrafted values. Finally, we rearranged the features of each target dataset to match the column order of the reference dataset. As a result, all source- and target-domain experiments were conducted under the same input structure and feature order, enabling consistent cross-domain analysis and adaptation.

3.2. Model Training Strategy

In this paper, we used RNN- and LSTM-based models as representative backbone architectures for time-series-based NIDS. In total, we trained eight backbone models using RNN- and LSTM-based architectures that take network flow sequences as input. RNN-based models capture sequential temporal dependencies, whereas LSTM-based models further capture long-term dependencies through their internal gating mechanisms. The input was represented as a three-dimensional time-series tensor with shape $(\text{batch\_size}, \text{sequence\_length}, \text{feature\_dim})$.

Each model was trained on a single source-domain dataset and evaluated on the remaining target datasets to analyze cross-domain performance degradation. In addition, we performed random undersampling at the sequence level on the majority (benign) class (0) so that its sample size matches that of the attack class (1), thereby mitigating the severe class imbalance during training. This strategy reduces the tendency of the models to be biased toward the majority class and enables a more consistent evaluation of detection performance.

Based on these settings, we conducted three independent trials for each backbone model. In total, we performed 24 backbone training runs (8 models $\times$ 3 trials) for the cross-domain performance analysis. These pre-trained models are subsequently used as backbone models for the cross-domain evaluation and LoRA-based adaptation described in the following sections.

3.3. Evaluation Metrics

To quantify the degradation of detection performance under cross-domain settings, we assess the models using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics are defined for the binary classification setting adopted in this study, where the positive class corresponds to attack traffic and the negative class corresponds to normal traffic. Specifically, True Positive (TP) denotes attack traffic correctly predicted as an attack, and True Negative (TN) denotes normal traffic correctly predicted as normal. False Positive (FP) denotes normal traffic incorrectly predicted as an attack, and False Negative (FN) denotes attack traffic incorrectly predicted as normal.

 doi:10.20944/preprints202605.1559.v1

Each evaluation metric is defined as follows. Accuracy is defined as the proportion of correctly classified samples among all predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2}$$

Accuracy provides useful information when the class distribution is relatively balanced. However, NIDS datasets are typically highly imbalanced, with normal traffic significantly outnumbering attack traffic. In this setting, a model can achieve high accuracy by predicting the majority (normal) class while failing to detect attacks. This limitation becomes more critical in cross-domain settings, where class distributions and decision boundaries may vary across datasets. Thus, accuracy alone is insufficient to evaluate detection performance in NIDS.

Precision is defined as the proportion of correctly detected attack traffic among all samples predicted as attacks:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{3}$$

Precision quantifies the reliability of attack predictions by measuring how often predicted attacks are truly attacks. A higher precision indicates fewer false alarms caused by normal traffic being misclassified as attacks. Thus, precision is important in NIDS for evaluating the false alarm behavior of the detection system.

Recall is defined as the proportion of actual attack traffic that is correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4}$$

Recall measures attack detection capability by quantifying the proportion of attack traffic that is successfully detected. A higher recall indicates fewer false negatives, meaning that fewer attacks are missed by the detection system. Thus, recall is critical in NIDS, where missed attacks can lead to significant security risks.

The F1-score is defined as the harmonic mean of precision and recall:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

By jointly considering precision and recall, the F1-score provides a balanced measure of detection performance. Thus, it is suitable for imbalanced settings such as NIDS datasets and is useful for assessing overall detection performance under cross-domain shifts.

3.4. Cross-Domain Performance Degradation

In this section, we experimentally evaluate the performance of RNN- and LSTM-based NIDS models on unseen target datasets and analyze performance degradation under cross-domain settings. Table 1 presents the detection performance of RNN- and LSTM-based models when trained on different source datasets and evaluated on both same-domain and cross-domain datasets. When training and evaluation are performed on the same domain, both models achieve consistently high performance across all datasets. For example, on the KDD99 dataset, both RNN and LSTM models achieve accuracy, precision, recall, and F1-score values exceeding 0.99. Similarly, for the other datasets, the models maintain high performance with average scores above 0.97.

However, when evaluated on unseen target datasets, a substantial degradation in detection performance is observed compared to the same-domain results. For example, when an RNN model trained on the UNSW_NB15 dataset is evaluated on the KDD99 dataset, the accuracy drops to 0.7615, indicating a moderate absolute performance level but a substantial decline relative to the same-domain setting.

 doi:10.20944/preprints202605.1559.v1

Table 1. Cross-Domain Performance of Base RNN/LSTM Models without LoRA

Model	Source dataset	Test dataset	Accuracy	Precision	Recall	F1-score
RNN	KDD99	KDD99	0.9962	0.9964	0.9933	0.9949
		UNSW_NB15	0.6113	0.3738	0.6113	0.4639
		CICIDS2017	0.2411	0.7736	0.2411	0.1972
		AWID3	0.8210	0.0239	0.0002	0.0003
	UNSW_NB15	UNSW_NB15	0.9616	0.9600	0.9592	0.9596
		KDD99	0.7615	0.6743	0.6684	0.6712
		CICIDS2017	0.8134	0.7009	0.8134	0.7481
		AWID3	0.8221	0.0075	0.0000	0.0000
	CICIDS2017	CICIDS2017	0.9786	0.9752	0.9792	0.9632
		KDD99	0.6838	0.6681	0.6838	0.6782
		UNSW_NB15	0.5310	0.4746	0.5310	0.4314
		AWID3	0.7366	0.0179	0.0089	0.0119
	AWID3	AWID3	0.9800	0.9739	0.9121	0.9420
		KDD99	0.7561	0.0000	0.0000	0.0000
		UNSW_NB15	0.5000	0.0000	0.0000	0.0000
		CICIDS2017	0.8311	0.0000	0.0000	0.0000
LSTM	KDD99	KDD99	0.9970	0.9978	0.9941	0.9959
		UNSW_NB15	0.6114	0.7624	0.6114	0.4640
		CICIDS2017	0.8224	0.7991	0.8224	0.8072
		AWID3	0.7873	0.0375	0.0079	0.0131
	UNSW_NB15	UNSW_NB15	0.9662	0.9639	0.9651	0.9645
		KDD99	0.2506	0.7490	0.2506	0.1108
		CICIDS2017	0.5666	0.7264	0.5666	0.6202
		AWID3	0.3183	0.1766	0.7730	0.2875
	CICIDS2017	CICIDS2017	0.9806	0.9823	0.9856	0.9810
		KDD99	0.6562	0.3050	0.3202	0.3124
		UNSW_NB15	0.6111	0.3737	0.6111	0.4638
		AWID3	0.7872	0.0001	0.0000	0.0000
	AWID3	AWID3	0.9906	0.9807	0.9665	0.9736
		KDD99	0.7561	0.0000	0.0000	0.0000
		UNSW_NB15	0.5000	0.0000	0.0000	0.0000
		CICIDS2017	0.5000	0.0000	0.0000	0.0000

Precision and recall also decrease to 0.6743 and 0.6684, respectively. Overall, cross-domain detection performance often declines to below 0.6, and in some cases approaches zero, indicating a severe degradation in attack detection capability. For example, when a model trained on the KDD99 dataset is evaluated on the CICIDS2017 dataset, both accuracy and recall decrease sharply to around 0.2411, indicating that the model fails to maintain effective detection performance.

In particular, recall shows the most critical degradation, indicating that the models frequently fail to detect actual attacks under cross-domain settings. This pattern is especially evident for the AWID3 dataset, which reflects wireless network environments. In this case, accuracy remains relatively high (above 0.7), whereas precision, recall, and F1-score drop significantly to below 0.01. This indicates that the model tends to classify most traffic as normal, thereby failing to detect actual attack traffic. This pattern further confirms that accuracy alone is insufficient for evaluating NIDS under cross-domain settings.

These results demonstrate that deep learning-based NIDS models are highly dependent on the characteristics of the source-domain training data, and their cross-domain detection performance degrades substantially when the network environment and traffic distribution change. These findings motivate the need for cross-domain adaptation methods that can recover detection performance under domain shift without requiring full model retraining.

 doi:10.20944/preprints202605.1559.v1

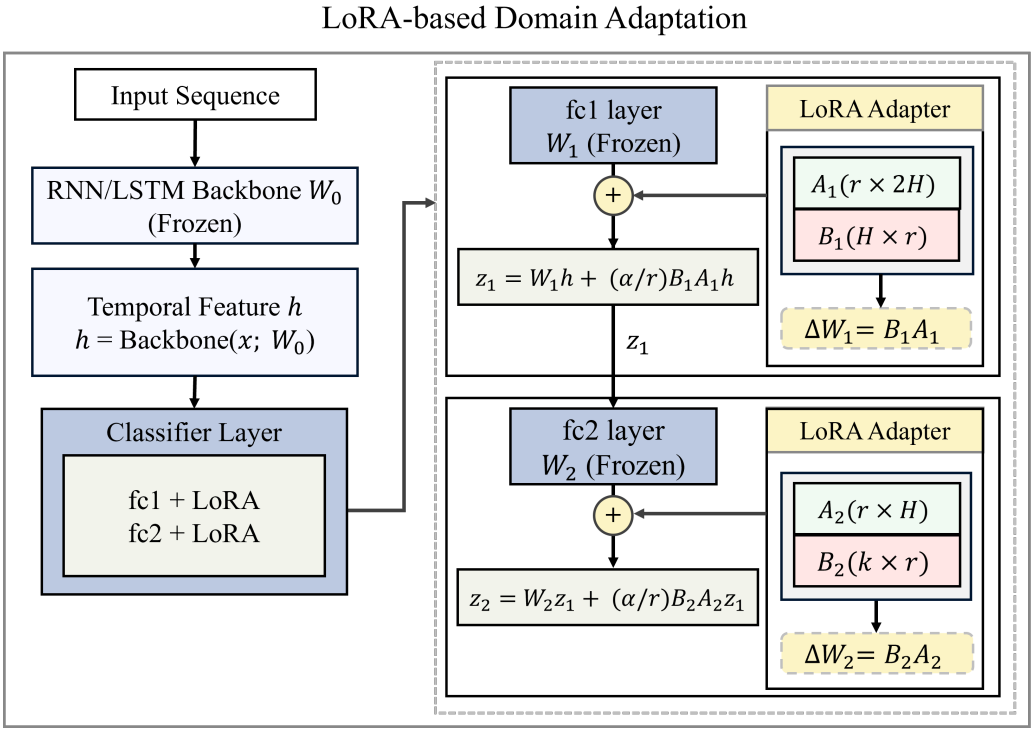

Figure 1. LoRA-based Domain Adaptation

Algorithm 1 Cross-Domain NIDS Training with LoRA Adaptation

Input: Source dataset D_r, Target dataset D_t, sequence length L, LoRA rank r, scaling factor α, epochs E
Output: Adapted model W^*
Initialization:

Load source-trained base model W_0 trained on D_r
Construct $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{val}}$ from D_t
Attach LoRA modules to selected layers
Initialize LoRA parameters θ_{lora}
Freeze all source-trained parameters in W_0
Initialize optimizer for θ_{lora}
$F1_{\text{best}} \leftarrow -\infty$

Procedure:

for epoch $= 1$ to E **do**
    Set model to training mode
    for each $(x, y) \in \mathcal{L}_{\text{train}}$ **do**
        Preprocess input x
        $\hat{y} = f(x; W_0, \theta_{\text{lora}})$
        $\ell = \text{CrossEntropy}(\hat{y}, y)$
        Backpropagate ℓ and update θ_{lora}
    end for
    Evaluate on $\mathcal{L}_{\text{val}}$
    if $F1_{\text{val}} > F1_{\text{best}}$ **then**
        $F1_{\text{best}} \leftarrow F1_{\text{val}}$
        $\theta^*_{\text{lora}} \leftarrow \theta_{\text{lora}}$
        $W^* \leftarrow \{W_0, \theta^*_{\text{lora}}\}$
    end if
end for
return W^*

 doi:10.20944/preprints202605.1559.v1

4. LoRA-based Domain Adaptation

In this paper, we formulate cross-domain adaptation as a low-rank adjustment of a source-trained model, rather than full retraining for each target domain. Thus, we apply a LoRA-based adaptation strategy to improve cross-domain detection performance under domain shift.

Specifically, the backbone layers of the source-trained RNN- and LSTM-based NIDS models are kept entirely frozen. LoRA modules are then applied only to the fully connected layers (fc1 and fc2) responsible for final classification decisions. These layers directly map the extracted temporal representations to the final intrusion labels and are therefore the most likely to require domain-specific adjustment. This design is motivated by the following considerations:

1. **Hierarchical Role Separation and Partial Cross-Domain Transferability:** The RNN and LSTM backbones extract time-series patterns and temporal dependencies from network flows. These temporal feature representations may capture partially reusable temporal patterns across domains, even though they are not fully invariant to domain shift.
2. **Domain Sensitivity of Classification Boundaries:** Conversely, distributional shifts between domains are often reflected in changes to the classification decision boundaries that map extracted features to benign or malicious labels. Therefore, rather than modifying the entire backbone, we consider it more effective to calibrate only the decision boundaries within a low-dimensional space at the classifier level.

When applying LoRA, the original weight matrix $W_{\text{fc},0} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ of a selected fully connected layer remains frozen. Here, d_{in} and d_{out} denote the input and output dimensions of the selected fully connected layer, respectively, and r denotes the LoRA rank. Instead, we introduce low-rank matrices $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ to define the weight update as follows:

$$W_{\text{fc}}^{*} = W_{\text{fc},0} + \Delta W = W_{\text{fc},0} + \frac{\alpha}{r} BA. \tag{6}$$

In this configuration, the trainable parameters are restricted to the low-rank matrices A and B. This approach shifts the adaptation process from optimization in the full parameter space to optimization in a lower-dimensional space, which improves parameter efficiency and can help reduce the risk of overfitting under domain shift. Under this setting, the source-trained backbone remains fixed, and only the low-rank adaptation parameters are optimized using target-domain data. In our experiments, the source-trained RNN- and LSTM-based NIDS models serve as fixed backbone models. We then apply feature mapping and LoRA-based low-rank adaptation to unseen target datasets. For adaptation, each target dataset is divided into two disjoint subsets, where 80% is used for adaptation and the remaining 20% is reserved for evaluation, ensuring that no overlap exists between adaptation and evaluation data. This pipeline enables efficient cross-domain adaptation without requiring full model retraining.

5. Experiments

5.1. Implementation Details

In this section, we describe the implementation details and training settings of the LoRA-based adaptation method introduced in Section 4. Consistent with the adaptation strategy described in Section 4, only the LoRA parameters A and B are updated during adaptation, while all backbone parameters remain frozen to maintain parameter efficiency. We applied LoRA modules to the fully connected layers of the model, whose the input-output dimensions of fc1 and fc2 layers are $(2H \times H)$ and $(H \times C)$, respectively. Here, $H = 256$ denotes the hidden size and $C = 2$ denotes the number of classes. We also configured LoRA with a rank of $r = 8$ and a scaling factor of $\alpha = 16$.

The backbone model consists of two-layer bidirectional recurrent architectures (RNN and LSTM), with a hidden size of 256 and a dropout rate of 0.2. We set the input sequence length to 10 and the batch size to 512, and trained the models using the AdamW optimizer [22] with a learning rate of 1×10^{-4}. To improve numerical stability during backpropagation, we applied gradient norm clipping

 doi:10.20944/preprints202605.1559.v1

with a threshold of 1.0 [23]. We also applied NaN-to-zero replacement and value clipping to all input tensors to reduce instability caused by missing values or extreme outliers during the feature mapping process. These preprocessing steps help prevent numerical divergence within the network.

All experiments were implemented using PyTorch and conducted on a workstation equipped with an NVIDIA GeForce RTX 5070 GPU with CUDA 12.8 support. This design allows us to manage lightweight LoRA adapters in a plug-in manner for each domain, enabling rapid and flexible domain adaptation across diverse network environments while reusing a single backbone model.

Table 2. TrainingConfiguration

Component	**Setting**
Backbone Model	2-layer Bidirectional RNN and LSTM
Hidden Size (H)	256
Dropout	0.2
Sequence Length	10
Batch Size	512
Optimizer	AdamW
Learning Rate	1×10^{-4}
Epochs	100–150
Gradient Clipping	Norm = 1.0
LoRA Rank (r)	8
LoRA Scaling (α)	16
Trainable Parameters	LoRA parameters only
Batch Normalization	Frozen
Preprocessing	NaN-to-zero replacement, value clipping
Framework	PyTorch
Hardware	NVIDIA GeForce RTX 5070 (CUDA 12.8)

5.2. Results and Analysis

In this section, we evaluate attack detection performance under cross-domain settings by applying LoRA-based adaptation across multiple source-target dataset pairs. We analyze performance in terms of accuracy, precision, recall, and F1-score, with particular emphasis on recall because it is a critical metric in NIDS. Since the primary objective of NIDS is to accurately detect malicious traffic, improvements in recall directly indicate enhanced detection capability under cross-domain settings.

Table 3 presents the cross-domain detection performance after LoRA-based adaptation. As shown in Table 3, applying LoRA-based adaptation leads to consistent improvements in cross-domain detection performance across most dataset combinations. In particular, the most pronounced gains are observed in recall, indicating improved attack detection under domain shift. Specifically, when the backbone model suffers from severe performance degradation, adaptation yields substantial recovery in detection performance. In such cases, adaptation not only improves performance but also restores effective attack detection capability. For example, as shown in Table 1, when a model trained on the KDD99 dataset is applied to the CICIDS2017 dataset, the backbone recall drops to 0.2411, indicating a significant degradation in attack detection capability. However, after adaptation (Table 3), recall increases to 0.8916, recovering to a level at which malicious traffic can be reliably detected. This demonstrates that, in certain domain combinations, the effect of adaptation goes beyond incremental improvement and results in substantial recovery of detection performance.

A more extreme pattern is observed for the AWID3 dataset, which represents a wireless network environment. In this case, the backbone model tends to classify most traffic as normal, resulting in near-zero recall, which indicates a near-complete failure in attack detection. After adaptation, recall increases to above 0.7, showing that the model can identify a substantial portion of previously

 doi:10.20944/preprints202605.1559.v1

undetected attack traffic. This indicates that the degradation caused by domain shift is not an inherent limitation of the model, but rather a problem that can be effectively mitigated through appropriate adaptation strategies.

Table 3. Cross-Domain Performance of RNN/LSTM Models with LoRA-Based Adaptation

Model	Source dataset	Test dataset	Accuracy	Precision	Recall	F1-score
RNN	KDD99	KDD99	0.9962	0.9964	0.9933	0.9949
		UNSW_NB15	0.6323	0.6870	0.6323	0.6341
		CICIDS2017	0.8916	0.8848	0.8916	0.8863
		AWID3	0.9550	0.9010	0.8393	0.8691
	UNSW_NB15	UNSW_NB15	0.9616	0.9600	0.9592	0.9596
		KDD99	0.9795	0.9805	0.9795	0.9797
		CICIDS2017	0.9118	0.9219	0.9118	0.9152
		AWID3	0.9139	0.7690	0.7375	0.7529
	CICIDS2017	CICIDS2017	0.9786	0.9752	0.9792	0.9632
		KDD99	0.9910	0.9919	0.9910	0.9909
		UNSW_NB15	0.7212	0.7378	0.7212	0.7106
		AWID3	0.9472	0.8667	0.8312	0.8485
	AWID3	AWID3	0.9800	0.9739	0.9121	0.9420
		KDD99	0.9543	0.9154	0.8952	0.9052
		UNSW_NB15	0.6239	0.5886	0.8231	0.6864
		CICIDS2017	0.8869	0.6704	0.6495	0.6598
LSTM	KDD99	KDD99	0.9970	0.9978	0.9941	0.9959
		UNSW_NB15	0.6666	0.7137	0.6666	0.6694
		CICIDS2017	0.8781	0.8683	0.8781	0.8701
		AWID3	0.9457	0.8245	0.8831	0.8528
	UNSW_NB15	UNSW_NB15	0.9662	0.9639	0.9651	0.9645
		KDD99	0.9884	0.9839	0.9884	0.9843
		CICIDS2017	0.9154	0.9191	0.9154	0.9169
		AWID3	0.9456	0.8379	0.8606	0.8491
	CICIDS2017	CICIDS2017	0.9806	0.9823	0.9856	0.9810
		KDD99	0.9876	0.9877	0.9876	0.9875
		UNSW_NB15	0.7797	0.8070	0.7797	0.7824
		AWID3	0.9552	0.8984	0.8435	0.8701
	AWID3	AWID3	0.9906	0.9807	0.9665	0.9736
		KDD99	0.9826	0.9725	0.9555	0.9639
		UNSW_NB15	0.6954	0.6696	0.7832	0.7220
		CICIDS2017	0.8835	0.8915	0.7684	0.8254

In addition, the following observations can be drawn from Table 3. First, from a model architecture perspective, both RNN and LSTM models exhibit similar performance improvement patterns after adaptation. In particular, recall consistently shows the most significant improvement across both models. However, the LSTM model generally showed more stable performance than the RNN model and achieved higher post-adaptation performance in most cases. This suggests that the stronger temporal modeling capability of LSTM may help capture complex sequential patterns more effectively, and that this property may be better leveraged under domain shift when combined with adaptation.

Second, in terms of domain discrepancy, the experimental results indicate that larger domain differences lead to greater performance gains after adaptation. When the datasets share similar network environments, such as wired network datasets, both models already exhibited moderate source-trained backbone performance, and the additional benefit of adaptation remains relatively limited. In contrast, when the domain difference is substantial, such as between wired and wireless environments, backbone performance drops significantly, and adaptation leads to substantial recovery.

 doi:10.20944/preprints202605.1559.v1

These observations suggest that domain discrepancy between datasets may have a greater impact on adaptation effectiveness than the choice of model architecture.

Overall, these results demonstrate that LoRA-based adaptation effectively restores degraded detection performance in cross-domain settings, particularly when the domain discrepancy is large. This suggests that, in our setting, addressing distribution mismatch between datasets may play a more critical role than modifying the model architecture itself. Furthermore, these findings highlight that domain adaptation techniques play an important role in achieving robust detection performance in real-world network environments, where diverse traffic patterns coexist.

5.3. Ablation Study

In this section, we conduct an ablation study to analyze performance changes according to LoRA hyperparameters, namely the scaling factor α and the rank r. Specifically, we consider $r \in \{4, 8, 16\}$ and $\alpha \in \{8, 16, 32\}$, and evaluate all combinations across multiple source-target dataset pairs. As shown in Figure 2, the effect of LoRA hyperparameters is dataset-dependent. Some source-target pairs show clear performance changes depending on r and α, whereas others exhibit only minor variations. This indicates that the optimal LoRA configuration for one source-target dataset pair does not necessarily generalize to all other domain shifts. In particular, higher-rank configurations such as $r = 16$ can achieve the highest score in some cases, but they do not consistently dominate across all settings.

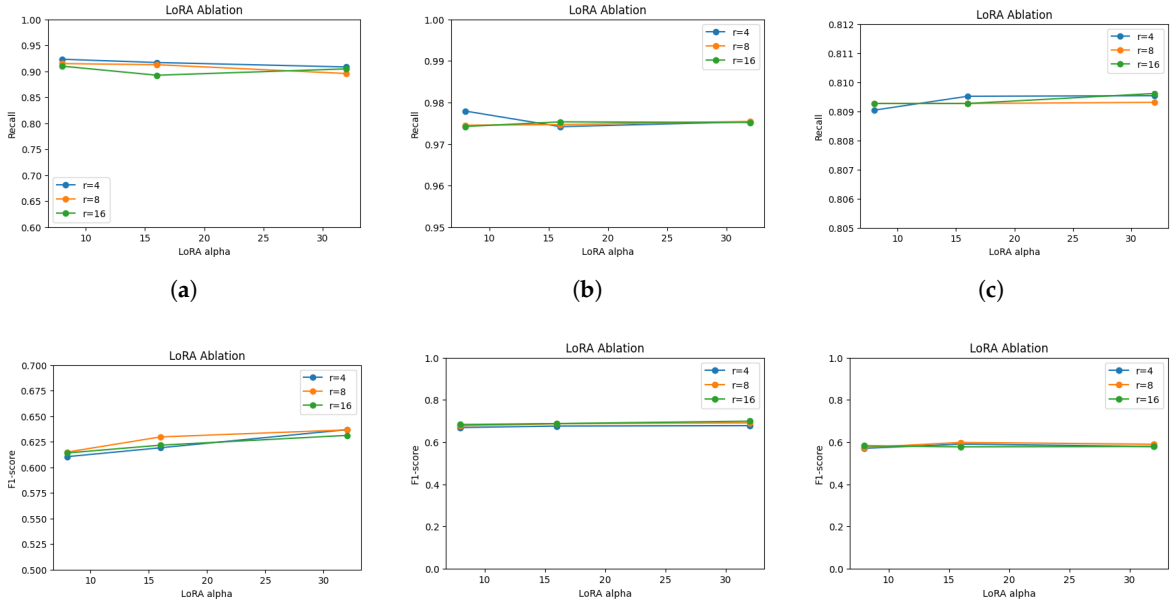

Figure 2. Ablation study results of LoRA hyperparameters across different cross-domain settings. (**a**) CICIDS2017 → AWID3 (Recall); (**b**) UNSW_NB15 → KDD99 (Recall); (**c**) AWID3 → KDD99 (Recall); (**d**) CICIDS2017 → UNSW_NB15 (F1-score); (**e**) UNSW_NB15 → CICIDS2017 (F1-score); (**f**) KDD99 → UNSW_NB15 (F1-score).

Therefore, rather than selecting the best-performing configuration for a single dataset pair, we seek a stable hyperparameter setting that performs consistently across diverse cross-domain conditions. Based on these observations, the choice of $r = 8$ and $\alpha = 16$ provides a practical trade-off between adaptation capacity, stability, and parameter efficiency.

Since the number of trainable LoRA parameters increases linearly with r, using $r = 8$ requires only half the trainable parameters of $r = 16$, while still achieving competitive performance across the evaluated source-target pairs. Moreover, $\alpha = 16$ corresponds to a moderate scaling factor of $\alpha = 2$, thereby avoiding excessively small or overly aggressive low-rank updates. Therefore, although $r = 8$, $\alpha = 16$ is not the absolute best configuration for every dataset pair, the ablation results support it as a stable and parameter-efficient default for general cross-domain adaptation.

 doi:10.20944/preprints202605.1559.v1

6. Conclusion

In this paper, we demonstrated that the performance of time-series-based NIDS models degrades in cross-domain environments. To address this issue, we proposed LoRA as a parameter-efficient adaptation method. Experimental results demonstrated that LoRA consistently improved performance across multiple datasets, with particularly notable gains in recall. These findings suggested that cross-domain generalization in NIDS was strongly influenced by domain characteristics, and that parameter-efficient adaptation methods such as LoRA could effectively mitigate this issue. However, the effectiveness of such approaches depended on the degree of domain discrepancy and dataset properties, indicating that adaptation alone may not fully resolve the generalization problem. In future work, we plan to further investigate whether adaptation alone is sufficient to fully address the generalization problem in cross-domain NIDS.

Author Contributions: Conceptualization, J.-H.C. and S.-H.C.; methodology, J.-H.C.; validation, J.-H.C., S.-W.H. and H.-J.J.; investigation, J.-H.C., S.-W.H. and H.-J.J.; resources, S.-H.C.; writing—original draft preparation, J.-H.C.; writing—review and editing, J.-H.C., S.-W.H., H.-J.J. and S.-H.C.; visualization, J.-H.C.; supervision, S.-H.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT)(IITP-2026-RS-2023-00259967, 90%) and the MSIT(Ministry of Science and ICT), Korea, under the National Program in Medical AI Semiconductor) supervised by the IITP(Institute of Information & Communications Technology Planning&Evaluation) in 2026(2024-0-0097, 10%).

Data Availability Statement: The datasets analyzed in this study are publicly available. The KDD99 dataset is available from the KDD Cup 1999 archive at https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. The UNSW_NB15 dataset is available from the Cyber Range Lab of UNSW Canberra at https://research.unsw.edu.au/projects/unsw-nb15-dataset. The CICIDS2017 dataset is available from the Canadian Institute for Cybersecurity at https://www.unb.ca/cic/datasets/ids-2017.html. The AWID3 dataset is available from Info-Sec-Lab, University of the Aegean at https://icsdweb.aegean.gr/awid/awid3. The processed data and source code used in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334.
2. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2018**, *2*, 41–50. https://doi.org/10.1109/TETCI.2017.2772792.
3. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials* **2016**, *18*, 1153–1176. https://doi.org/10.1109/COMST.2015.2494502.
4. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418.
5. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In *Proceedings of the International Conference on Platform Technology and Service (PlatCon)*, Jeju, Republic of Korea, 15–17 February 2016; pp. 1–5. https://doi.org/10.1109/PlatCon.2016.7456805.
6. Xu, C.; Shen, J.; Du, X.; Zhang, F. An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access* **2018**, *6*, 48697–48707. https://doi.org/10.1109/ACCESS.2018.2867564.
7. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Computers & Security* **2019**, *86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005.
8. Staudemeyer, R.C. Applying long short-term memory recurrent neural networks to intrusion detection. *South African Computer Journal* **2015**, *56*, 136–154. Available online: https://hdl.handle.net/10520/EJC173450 (accessed on 11 May 2026).

 doi:10.20944/preprints202605.1559.v1

9. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.
10. Houlsby, N.; Giurgiu, A.; Jastrzębski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
11. Cantone, M.; Marrocco, C.; Bria, A. Machine learning in network intrusion detection: A cross-dataset generalization study. *IEEE Access* **2024**, *12*, 144489–144508. https://doi.org/10.1109/ACCESS.2024.3472907.
12. Sarhan, M.; Layeghy, S.; Portmann, M. Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection. *Big Data Research* **2022**, *30*, 100359. https://doi.org/10.1016/j.bdr.2022.100359.
13. de Carvalho Bertoli, G.; Pereira Junior, L.A.; dos Santos, A.L.; Saotome, O. Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. *Computers & Security* **2023**, *127*, 103106. https://doi.org/10.1016/j.cose.2023.103106.
14. Amin, M.I.; Shen, M.; Ishak, M.K.; Manickam, S.; Karuppayah, S. Enhancing generalization of cross-domain intrusion detection: A heterogeneous deep stacked ensemble approach. *Connection Science* **2026**, *38*, 2599708. https://doi.org/10.1080/09540091.2025.2599708.
15. University of California, Irvine. KDD'99 Dataset. Available online: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 11 May 2026).
16. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528.
17. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *Proceedings of the Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 10–12 November 2015; pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942.
18. Moustafa, N.; Slay, J. The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective* **2016**, *25*, 18–31. https://doi.org/10.1080/19393555.2015.1125974.
19. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, Funchal, Portugal, 22–24 January 2018; pp. 108–116. https://doi.org/10.5220/0006639801080116.
20. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials* **2016**, *18*, 184–208. https://doi.org/10.1109/COMST.2015.2402161.
21. Chatzoglou, E.; Kambourakis, G.; Kolias, C. Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset. *IEEE Access* **2021**, *9*, 34188–34205. https://doi.org/10.1109/ACCESS.2021.3061609.
22. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.
23. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.