Enhanced Adversarial Domain Adaptation for Intrusion Detection Systems

Ines Guerziz[1], Zakaria Abou El Houda[2], and Long Bao Le[3]
[1,2,3]Institut National de la Recherche Scientifique
Centre Énergie, Matériaux et Télécommunications (INRS-EMT), Québec, Canada
ines.guerziz@inrs.ca, zakaria.abouelhouda@inrs.ca, long.le@inrs.ca

***Abstract*—The increasing sophistication of cyber threats demands robust and adaptive Intrusion Detection Systems (IDS) capable of generalizing across diverse network environments. However, traditional AI-driven IDS models suffer from performance degradation when deployed in unseen domains due to domain shift discrepancies in data distributions caused by varying network configurations, attack patterns, or data collection methods. While unsupervised domain adaptation has recently been applied to address domain shift, its use in IDS remains limited and often lacks adaptation to the unique challenges of network data. To bridge this gap, we propose an Enhanced Adversarial Domain Adaptation (E-ADDA) Framework for IDS, designed to align feature representations between source and target domains, enhancing model generalizability. Our framework is rigorously evaluated on three publicly available IDS datasets, demonstrating significant improvements in key metrics such as accuracy, F1 score, and loss compared to existing domain adaptation methods. The results highlight the viability of adversarial domain adaptation in improving IDS resilience against zero-day attacks and evolving threats, offering a promising direction for real-world cybersecurity applications.**

***Index Terms*—Domain Adaptation; Intrusion Detection System; Cybersecurity.**

I. Introduction

As interconnected systems grow, so does the risk of cyber-attacks, making Intrusion Detection Systems (IDS) essential for detecting malicious network activity. While traditional IDS models [1] perform well in controlled environments, they often fail in real-world settings due to domain shift between training and deployment data.

Domain shift arises due to differences in network topology, protocol usage, traffic volume, feature spaces, or labeling conventions across datasets. Consequently, IDS models trained on one dataset (e.g., NSL-KDD [2]) tend to perform poorly when tested on others (e.g., CSE-CIC-IDS2018 [3] or UNSW-NB15 [4]), despite capturing similar attack behaviors. This limitation restricts the practicality of deploying pre-trained IDS models across environments, retraining for each deployment environment demands substantial labeled target data, a resource that is often scarce, expensive, or impractical to obtain in operational settings.

To address domain shift, Unsupervised Domain Adaptation (UDA) transfers knowledge from labeled source data to unlabeled target domains by learning domain-invariant features. Among UDA methods, adversarial domain adaptation [5] has shown strong performance, originating from vision-based models like Domain Adversarial Neural Network (DANN) by Ganin *et al.* [6] and Adversarial Discriminative Domain Adaptation (ADDA) by Tzeng *et al.* [7], which align cross-domain features via adversarial training. These foundational techniques have since been extended to intrusion detection. Singla *et al.* [8] applied a GAN-based conditional domain adaptation model to map representations between the NSL-KDD and UNSW-NB15 datasets. Layeghy *et al.* [9] proposed DI-NIDS, which leverages adversarial training and a one-class SVM for anomaly detection in NetFlow-formatted traffic from NFv2-CIC-2018 and NFv2-UNSW-NB15. Alami *et al.* [10] used DANN with deep packet-based features extracted using NFStream, employing GRL and focal loss to manage class imbalance across USTC-TFC2016, CIC-IDS2017, and CUPID. Xue *et al.* [11] explored heterogeneous domain adaptation using hierarchical autoencoders with Maximum Mean Discrepancy (MMD) and entropy minimization across CIC-IDS2017, Kitsune, and IoTID20. Xu *et al.* [12] introduced the NAEF framework, combining nonlinear explicit feature augmentation with transformer attention mechanisms for NSL-KDD to CIC-IDS2017 adaptation. For 5G network scenarios, Kim *et al.* [13] proposed a Stacked Denoising Autoencoder (SDA)-based model enhanced with GRL and Proxy A distance minimization for adaptation from NSL-KDD to UNSW-NB15 under high-class imbalance.

Despite these advances, most of the of the existing UDA-based IDS methods face several limitations. Many rely on architectures originally designed for image or sequential data, which are not well-suited for sparse, high-dimensional tabular network traffic. Others fail to account for class imbalance, feature constraints, or uncertainty, factors that are critical in practical cybersecurity deployments. Moreover, few works explore modular or ensemble-based designs that can offer both robustness and interpretability.

To address these gaps, we propose an Enhanced Adversarial Discriminative Domain Adaptation (E-ADDA) framework specifically designed for tabular network intrusion detection. Building upon the classical ADDA structure, we evaluate our framework on three benchmark IDS datasets: NSL-KDD, UNSW-NB15, and CSE-CIC-IDS2018.

The main contributions of this paper are summarized as follows:

- We propose a UDA framework tailored for tabular IDS,

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

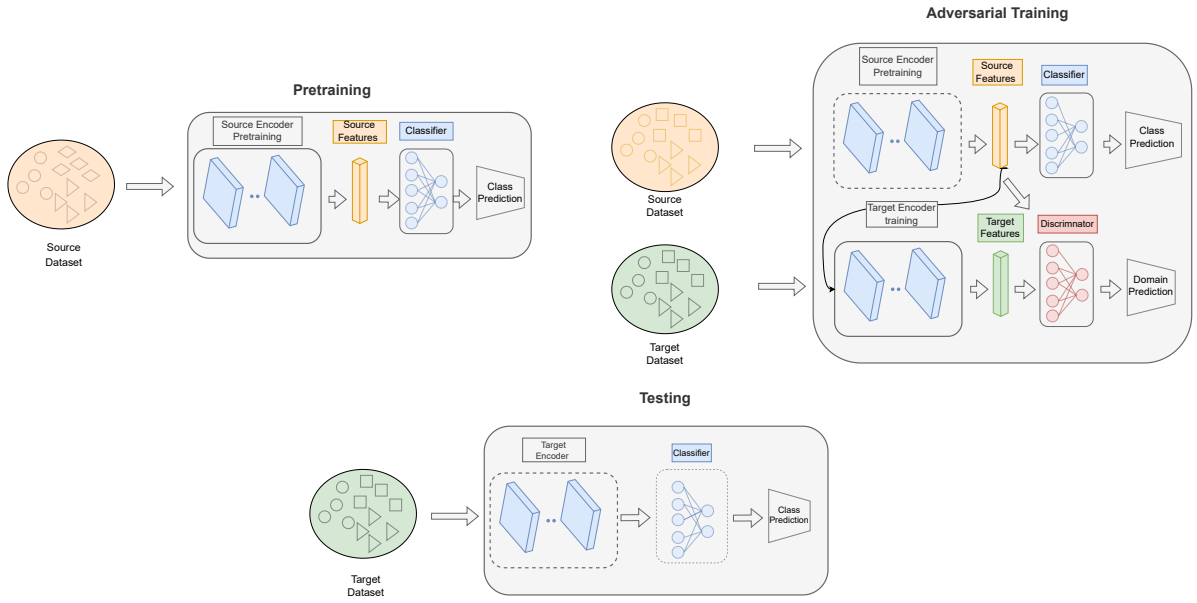


Fig. 1. An overview of our proposed Enhanced Adversarial Discriminative Domain Adaptation (E-ADDA) approach.

addressing domain shift and class imbalance across heterogeneous network environments.

- Our method leverages adversarial training, domain-specific augmentation, and ensemble learning with uncertainty modeling to enhance robustness and generalization.
- We conducted extensive cross-domain experiments on three real-world IDS datasets, showing consistent improvements over state-of-the-art domain adaptation methods.

The remainder of this paper is organized as follows. We introduce the proposed framework and its methodologies in Section II. Section III covers the experimental setup and results, and Section IV concludes the paper.

II. System Model

We propose an Enhanced Adversarial Discriminative Domain Adaptation (E-ADDA) framework that aligns source and target data distributions in a shared latent space without requiring any supervised feedback from the target domain. Building upon the classical ADDA architecture, our method incorporates deep tabular representation learning, ensemble-based uncertainty estimation, and dual-domain augmentation, organized into four key components.

A. Architecture Overview

The architecture comprises several key components. The Source Encoder is a deep Tabular ResNet designed to learn rich feature representations from the labeled source domain. The Target Encoder is initialized as a copy of the source encoder, with additional domain-specific layers that are optimized during the adversarial training phase. The Discriminator acts as a binary classifier, distinguishing whether a given feature originates from the source or the target domain. Finally, a Classifier Ensemble is employed on top of the encoder outputs to perform intrusion classification and estimate prediction uncertainty.

Each encoder block comprises stacked residual blocks featuring GELU activations, dropout, and batch normalization. The discriminator incorporates a gradient penalty to stabilize training and enforce Lipschitz continuity, while the classifier is built with layer normalization and dropout for regularization.

B. Training Procedure

The training process is divided into three distinct phases. First, during the **Source Pretraining** phase, the source encoder and classifier are jointly trained using labeled source data. To address class imbalance, Focal Loss [14] is employed, which emphasizes harder examples and reduces the influence of majority classes. This enables the encoder to learn class-discriminative representations effectively.

Next, in the **Adversarial Adaptation** phase, the trained source encoder is cloned to initialize the target encoder. As detailed in Algorithm 1, this phase proceeds adversarially: the discriminator attempts to distinguish between source features (which are fixed) and target features (which are being learned). The target encoder is optimized to fool the discriminator by producing target features that resemble source features. Additionally, dual-domain mixup [15] is applied at both the global (feature) and local (region) levels to regularize the domain boundaries and enhance generalization. A gradient penalty term is also introduced to stabilize the training of the discriminator.

Finally, in the **Ensemble Prediction and Uncertainty Estimation** phase, a committee of classifiers evaluates each

input sample during inference. To estimate epistemic uncertainty without access to labeled target data, we employ Monte Carlo Dropout (MCD) [16], which enhances robustness to domain shift and improves the reliability of predictions, an essential requirement in security-sensitive intrusion detection scenarios.

Algorithm 1: Enhanced Adversarial Domain Adaptation (E-ADDA)

Input: Source data $\mathcal{D}_s$, Target data $\mathcal{D}_t$, Learning rate η, Batch size d, Steps T
Output: Trained target encoder E_t and classifier C
Initialize source encoder E_s, target encoder $E_t \leftarrow E_s$, classifier C, and discriminator D
Pretrain E_s and C on $\mathcal{D}_s$ using Focal Loss
for $t \leftarrow 1$ **to** T **do**
  Sample batch x_s from $\mathcal{D}_s$ and x_t from $\mathcal{D}_t$;
  Compute source features $f_s = E_s(x_s)$ and target features $f_t = E_t(x_t)$;
  Discriminator Update:
  Compute domain loss:

$$\mathcal{L}_{\text{domain}} = -\frac{1}{d}\sum_{i=1}^{d} \log D(f_s^i) + \log(1 - D(f_t^i))$$

  Compute gradient penalty:

$$\mathcal{L}_{\text{gp}} = \lambda \cdot \mathbb{E}_{\hat{x}} \left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2$$

  Update discriminator D by minimizing:

$$\mathcal{L}_{\text{domain}} + \mathcal{L}_{\text{gp}}$$

  Target Encoder Update:
  Apply dual-domain mixup to (x_s, x_t) and recompute target features;
  Compute classification loss using Enhanced Focal Loss:

$$\mathcal{L}_{\text{class}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

  Update target encoder E_t by minimizing:

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{domain}} + \mathcal{L}_{\text{class}}$$

end
return E_t, C

C. Loss Functions

Three core loss functions are integrated into our Enhanced ADDA framework to support effective domain adaptation and robust classification.

The **Classification Loss** is implemented using the Focal Loss, which addresses class imbalance by emphasizing difficult-to-classify examples. It is defined as:

$$\mathcal{L}_{\text{class}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{1}$$

where α_t is a class-specific weighting factor, γ controls the focusing strength, and p_t represents the predicted probability for the true class label.

The **Domain Loss** is formulated as a binary cross-entropy loss to train the discriminator in distinguishing between source and target domain features. It is given by:

$$\begin{aligned} \mathcal{L}_{\text{domain}} = & -\mathbb{E}_{x_s \sim S} \left[\log D(E_s(x_s)) \right] \\ & -\mathbb{E}_{x_t \sim T} \left[\log \left(1 - D(E_t(x_t)) \right) \right] \end{aligned} \tag{2}$$

where D is the domain discriminator, and E_s, E_t are the encoders corresponding to the source and target domains, respectively.

To improve training stability and enforce the Lipschitz constraint, a **Gradient Penalty** is applied. This regularization term is defined as:

$$\mathcal{L}_{\text{gp}} = \lambda \cdot \mathbb{E}_{\hat{x} \sim \mathcal{I}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right] \tag{3}$$

where $\hat{x}$ denotes a randomly sampled point obtained by linearly interpolating between encoded source and target features, and λ is the penalty coefficient.

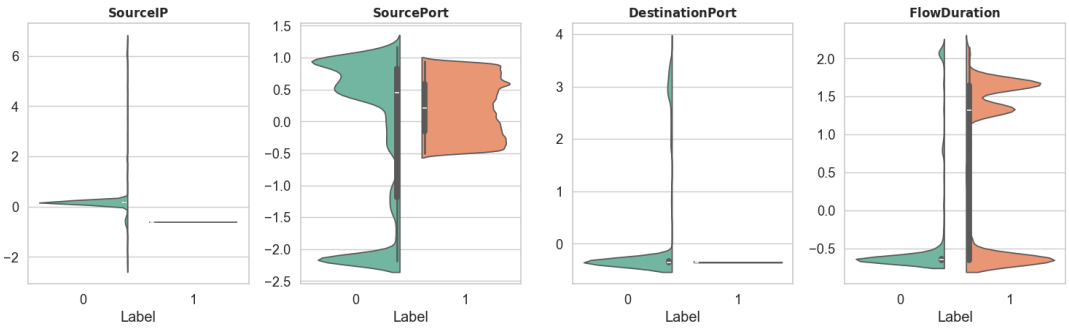
III. Experiment Details

A. System Setup

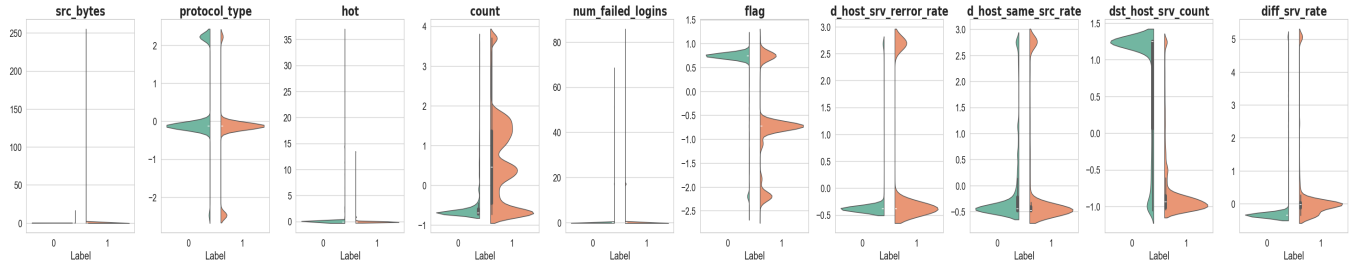
All experiments were performed on a Windows machine featuring a 13th Gen Intel® Core™ i7-13620H processor (2.40 GHz), 454 GB of RAM, and an Nvidia GeForce RTX 4050 GPU. The models were developed using PyTorch. The TabularResNet-based encoders comprise three residual blocks with 256 hidden units each, employing ReLU activations and dropout rates of 0.2 and 0.3 for regularization. The training process utilized the AdamW optimizer, with learning rates set to 0.001 for the source encoder, 0.0005 for the target encoder, and 0.0001 for the domain discriminator. A batch size of 1024 was used across all training phases.

B. Datasets

To evaluate the effectiveness and generalization capabilities of the proposed E-ADDA framework, we conduct experiments on three widely used publicly available intrusion detection datasets: NSL-KDD, UNSW-NB15, and CSE-CIC-IDS2018. These datasets represent diverse network environments, attack categories, and traffic characteristics, making them well-suited for cross-domain evaluation in domain adaptation research. NSL-KDD is a refined version of the original KDD'99 dataset, addressing issues of redundancy and imbalance. It contains various categories of attacks such as DoS, Probe, U2R, and R2L, making it a standard benchmark in IDS research. UNSW-NB15 includes modern attack types and realistic network traffic generated using the IXIA PerfectStorm tool. It covers a wide range of threats including Exploits, Fuzzers, Shellcode, and Worms, providing a more contemporary representation of cyberattacks. CSE-CIC-IDS2018 (also referred to as CIC-IDS2018) is a comprehensive dataset that captures benign and malicious traffic scenarios over multiple days, including brute force attacks,



(a)



(b)

Fig. 2. Violin plots showing the distribution of selected features in the source dataset (NSL-KDD).(a) The top 4 features from the source dataset. (b) The Top 10 features from the source dataset.

botnets, and infiltration attempts. It offers high diversity and volume, simulating real-world network behavior across enterprise settings.

Since the original datasets were designed for multiclass classification, we reformulate the task as binary classification by aggregating all attack types into a single "attack" class. This transformation simplifies the problem and allows us to focus on distinguishing between benign traffic and any form of malicious activity. Each of these datasets poses unique challenges in terms of feature space alignment, label imbalance, and distribution shift, making them ideal candidates for evaluating the transferability of IDS models across domains.

C. *Data Pre-Processing*

1) Data Standardization: Given a source domain $\mathcal{D}_s = (\mathbf{X}_s, \mathbf{y}_s)$ and target domain $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ represents the feature matrix (n samples with d dimensions) and $\mathbf{y}$ contains the corresponding labels, we perform the following standardization steps:

Z-Score Normalization: For each continuous feature column $\mathbf{x}_j \in \mathbf{X}$, where $j \in 1, \ldots, d$, we apply Z-score normalization:

$$\tilde{\mathbf{x}}_j = \frac{\mathbf{x}_j - \mu_j}{\sigma_j} \tag{4}$$

Here, $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ denotes the empirical mean of feature j, and $\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}$ represents the corresponding standard deviation. This transformation ensures that all continuous features are centered at zero mean and scaled to unit variance, enhancing the numerical stability of subsequent learning procedures.

Categorical Feature Encoding: For categorical features $C \subset \mathbf{X}$ with cardinality $|C| = m$, we employ integer encoding:

$$\phi : C \to \mathbb{Z}^+$$
$$\phi(c_i) = k \quad \text{if and only if } c_i \text{ is the } k\text{-th unique category}$$

where $k \in \{1, ..., K\}$ and K is the total number of unique categories. This injective mapping preserves the categorical relationships while converting symbolic values to numerical representations compatible with our machine learning framework.

2) Label Space Unification: To enable binary intrusion detection across heterogeneous datasets, we unify all dataset-specific label variants into a standardized binary label space $\mathcal{Y} = 0, 1$ via a surjective mapping:

$$f : \mathcal{L} \to \mathcal{Y} \quad \text{where} \quad f(y_i) = \begin{cases} 0 & \text{if } y_i \in \mathcal{B} \text{ (benign)} \\ 1 & \text{if } y_i \in \mathcal{A} \text{ (attack)} \end{cases} \tag{5}$$

where $\mathcal{L}$ denotes the original label space, which varies across datasets. The subset $\mathcal{B}$ includes labels associated with benign or normal behavior (e.g., benign, normal, legitimate, 0, etc.),

while $\mathcal{A}$ comprises attack-related labels (e.g., attack, malicious, DoS, BruteForce, 1, etc.). This normalization facilitates a unified binary classification objective across domains.

3) Feature Selection: To reduce dimensionality and identify the most discriminative features for intrusion detection, we perform feature selection prior to the domain adaptation process using the embedded feature importance mechanism of XGBoost [17]. Importantly, this selection is conducted exclusively on the source domain, where labeled data is available.For each feature $\mathbf{x}_j \in \mathbf{X}$, we calculate an importance score G_j based on the aggregated second-order gradients of the loss function across the ensemble:

$$G_j = \frac{1}{T} \sum_{t=1}^{T} \sum_{i \in I_t} \frac{\partial^2 \mathcal{L}}{\partial x_j^2} \tag{6}$$

where T denotes the total number of decision trees, I_t represents the set of instances traversing tree t, and $\mathcal{L}$ is the objective loss function used by XGBoost. The term $\frac{\partial^2 \mathcal{L}}{\partial x_j^2}$ captures the second-order gradient of the loss for feature j, which serves as a proxy for its predictive contribution. Features with higher scores are retained for downstream learning tasks.

D. *Description of experiments*

To evaluate the effectiveness of our proposed domain adaptation framework, we first perform feature selection using XGBoost prior to adaptation. Importantly, feature selection is conducted only on the labeled source domain, ensuring that no information from the target domain, including its labels, influences the selection process. Specifically, we train an XGBoost classifier on the source dataset, compute gain-based feature importance scores, and retain the top 4 and top 10 features. The smaller subset of 4 features allows us to evaluate model performance in feature-constrained scenarios that reflect practical limitations in real-world deployments. The 10-feature subset, by contrast, provides a richer representation to explore the impact of higher-dimensional input spaces. These selected source domain features are then used throughout the domain adaptation process. To visualize their discriminative power, we present violin plots showing their value distributions across binary class labels (benign vs. attack) within the source domain, as shown in Figure 2.

For the domain adaptation experiments, we adopt a source-to-target evaluation strategy. The source domain is typically NSL-KDD, as it has been widely used in prior work, while the target domain varies across different experiments. For instance, we evaluate the transfer from NSL-KDD to UNSW-NB15 and from CSE-CIC-IDS2018 to UNSW-NB15, which allows us to test the model's performance in adapting across different network environments with diverse attack patterns. These datasets represent distinct network configurations and data formats, making them ideal for domain adaptation research. By using both the top 4 and top 10 selected features, we conduct a detailed performance comparison against state-of-the-art domain adaptation techniques, ensuring a thorough evaluation under varied conditions.

E. *Performance Evaluation*

The adversarial training process is evaluated through the discriminator and target encoder loss curves as shown in Figure 3. The discriminator loss (Figure 3a) shows convergence across all experimental configurations, indicating successful alignment of feature distributions between the source (e.g., NSL-KDD, CSE-CIC-IDS2018) and target (UNSW-NB15) domains. Notably, the adaptation with 10 features achieves faster stabilization compared to the 4 features setup, suggesting that higher-dimensional feature spaces facilitate more robust domain alignment. Conversely, the target encoder loss (Figure 3b) demonstrates consistent minimization, reflecting effective adversarial training where the target encoder learns to deceive the discriminator while preserving discriminative features for intrusion detection.

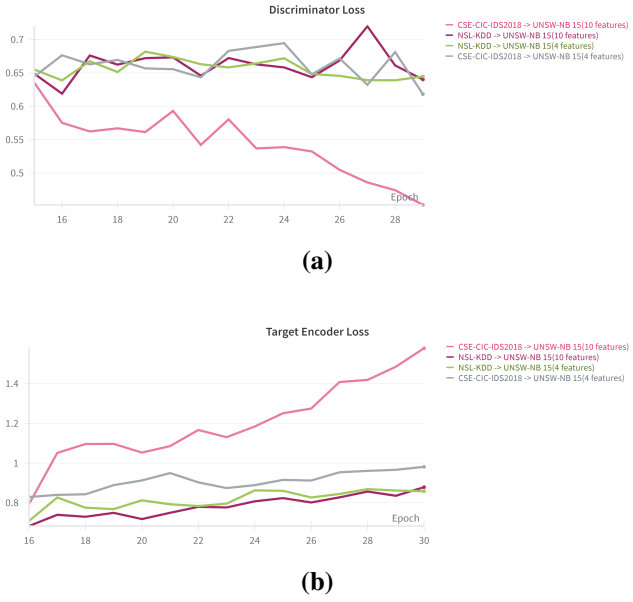


Fig. 3. Loss curves during adversarial training. (a) Target domain loss, (b) Discriminator loss.

The quantitative results in Tables I and II reveal critical insights about the E-ADDA framework's generalization capabilities across heterogeneous network environments.

For the NSL-KDD $\rightarrow$ UNSW-NB15 scenario (Table I), our 10-feature Enhanced ADDA model achieves 99.02% accuracy, outperforming previous works such as Singla et al. [8] (92.00%) and Kim et al. [13] (84.55%). This 7.02–14.47% improvement underscores the model's ability to maintain discriminative feature alignment across domains, particularly in legacy-to-modern network adaptation.

The 4-feature variant maintains competitive performance (85.14% accuracy), showing robustness even with constrained feature spaces. The 13.88% gap between 4- and 10-

TABLE I
PERFORMANCE COMPARISON OF DOMAIN ADAPTATION METHODS
(NSL-KDD → UNSW-NB15)

Reference	Method	Accuracy (%)	F1-Score (%)
Singla et al. [8]	CNN-LSTM with DA	92.00	91.00
Kim et al. [13]	DA for 5G Security	84.55	85.37
Our Work	Enhanced ADDA (10 features)	**99.02**	**99.01**
	Enhanced ADDA (4 features)	85.14	84.85

TABLE II
PERFORMANCE COMPARISON OF DOMAIN ADAPTATION METHODS
(CSE-CIC-IDS2018 → UNSW-NB15)

Reference	Method	Accuracy (%)	F1-Score (%)
Layeghy et al. [9]	DANN	—	17.31
Layeghy et al. [9]	DI-NIDS (Domain Invariant)	—	85.79
Our Work	Enhanced ADDA (10 features)	84.97	84.62
	Enhanced ADDA (4 features)	**99.40**	**99.40**

feature models highlights a trade-off between interpretability and detection fidelity

In the CSE-CIC-IDS2018 → UNSW-NB15 scenario (Table II), the 4-feature configuration unexpectedly achieves 99.40% accuracy, surpassing both DI-NIDS (85.79%) and our 10-feature model (84.97%). This result suggests that feature quality may be more important than quantity, as the top 4 selected features might inherently capture domain-invariant traffic patterns between these modern datasets, which contain IoT and cloud attack vectors. Additionally, the higher-dimensional features in the 10-feature model could introduce noisy, dataset-specific artifacts. Our adversarial training likely suppresses such interference more effectively in low-dimensional spaces, leading to the improved performance observed with the 4-feature configuration. The framework's worst-case performance (84.62% F1 for CSE-CIC-IDS2018 → UNSW-NB15 with 10 features) still exceeds baseline DANN's 17.31%, which shows reliability under distributional shifts.

IV. CONCLUSION

This paper proposed E-ADDA, a novel adversarial domain adaptation framework that significantly enhances cross-domain intrusion detection, including detection of zero-day attacks through domain-invariant feature learning. Our key innovation lies in aligning feature distributions between source and target domains while addressing class imbalance and uncertainty through focal loss and ensemble learning. The framework achieved state-of-the-art performance with 99.02% accuracy (NSL-KDD → UNSW-NB15) and 99.40% accuracy (CSE-CIC-IDS2018 → UNSW-NB15), outperforming existing methods by up to 15%, while demonstrating strong generalization to unseen attack patterns. As future work, we plan to consider self-supervised adaptation to eliminate labeled data dependency and extend the framework for dynamic threat environments. E-ADDA provides a practical solution for deploying robust intrusion detection across diverse networks, advancing the field toward adaptive cybersecurity systems capable of defending against both known and emerging zero-day threats.

REFERENCES

[1] T. Sowmya and E. M. Anita, "A comprehensive review of ai based intrusion detection system," *Measurement: Sensors*, vol. 28, p. 100827, 2023.

[2] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[3] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[4] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[5] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*, R. Stahlbock, G. M. Weiss, M. Abou-Nasr, C.-Y. Yang, H. R. Arabnia, and L. Deligiannidis, Eds. Cham: Springer International Publishing, 2021, pp. 877–894.

[6] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 1180–1189.

[7] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2962–2971.

[8] A. Singla, E. Bertino, and D. Verma, "Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 127–140.

[9] S. Layeghy, M. Baktashmotlagh, and M. Portmann, "Di-nids: Domain invariant network intrusion detection system," *Know.-Based Syst.*, vol. 273, no. C, Aug. 2023.

[10] H. Alami, M. J. Idrissi, A. El Mahdaouy, A. Bouayad, Z. Yartaoui, and I. Berrada, "Investigating domain adaptation for network intrusion detection," in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2023, pp. 1–7.

[11] B. Xue, H. Zhao, and W. Yao, "Deep transfer learning for iot intrusion detection," in *2022 3rd International Conference on Computing, Networks and Internet of Things (CNIOT)*, 2022, pp. 88–94.

[12] X. Yu, Y. Lu, F. Jiang, Q. Hu, J. Du, and D. Gong, "A cross-domain intrusion detection method based on nonlinear augmented explicit features," *IEEE Transactions on Network and Service Management*, vol. 22, no. 1, pp. 187–197, 2025.

[13] H.-J. Kim, J. Lee, C. Park, and J.-G. Park, "Network anomaly detection based on domain adaptation for 5g network security," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 2022, pp. 976–980.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[15] Y. Wu, D. Inkpen, and A. El-Roby, "Dual mixup regularized learning for adversarial domain adaptation," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 540–555.

[16] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794.