



**University of
Sunderland**

Wang, Qian, shang, jiahui, wang, weihao, Cheng, Yongqiang and zhang, bing (2026) A semi-supervised intrusion detection algorithm using mutual-information maximization adversarial network and pseudo-labeling. Cluster Computing, 29 (845). ISSN 1386-7857

Version: AAM

Available under license: Creative Commons: Attribution 4.0

Downloaded from: <https://sure.sunderland.ac.uk/id/eprint/20800/>

Which version am I reading (articles, book chapters and books)

Pre-print (or submitted version): the version of an article or chapter before any peer-review. Can be submitted to a publisher or added to a pre-print server.

AAM: Author Accepted Manuscript - The version of the article formally accepted for publication after the peer-review process. It includes the changes made by the author(s) in line with peer-reviewers' comments. It has not undergone the publisher's formatting, type-setting and copyediting.

VOR: Version of Record - The version published on the publisher's website which includes formatting, typesetting and copyediting.

Usage guidelines

Please refer to the usage guidelines at <http://sure.sunderland.ac.uk/policies.html> or alternatively contact sure@sunderland.ac.uk.

A semi-supervised intrusion detection algorithm using mutual-information maximization adversarial network and pseudo-labeling

Qian Wang^{a*}, Jiahui Shang^{a†}, Weihao Huang^{a†},
Yongqiang Cheng^{b†}, Bing Zhang^{a†}

^{a*} School of Information Science and Engineering, Yanshan University,
Qinhuangdao, 066004, China.

^{b*} School of computer science, University of Sunderland, Sunderland,
SR1 3SD, UK.

*Corresponding author(s). E-mail(s): wangqian@ysu.edu.cn;

Contributing authors: liuhan01@stumail.ysu.edu.cn;

1805648338@stumail.ysu.edu.cn; yongqiang.cheng@sunderland.ac.uk;

bingzhang@ysu.edu.cn;

[†]These authors contributed equally to this work.

Abstract

The massive amounts of network traffic data needs significant costs to be labeled, and due to the different data distribution, there is always a performance degradation of the model trained by public datasets for real applications. In order to solve the scarcity of labeled data and domain shift problem, this paper proposes a semi-supervised domain adaptive intrusion detection algorithm based on generative adversarial network with maximum mutual information and pseudo-label selection. Firstly, the improved LR-SMOTE (Limit Radius Synthetic Minority Oversampling Technique) data enhancement method based on multi-samples and random smoothing means is proposed to enhance the minority class data in the source domain to avoid overfitting of majority classes when the distribution is aligned. Secondly, a domain-invariant feature extraction method based on generative adversarial network with maximum mutual information is proposed. By maximizing the mutual information between the original features and domain-invariant features of the source and target domains, the integrated semantic information is fully retained, thereby the domain-invariant features equivalent to the original features are extracted. Finally, in order to make full use of the unlabeled data in the target domain, Uncertainty-aware Pseudo-label Selection (UPS)

method is used to train the domain-invariant feature classifier OSVM (One-class Support Vector Machine) in a semi-supervised manner. Experiments on network datasets and IoT (Internet of Things) datasets demonstrate the effectiveness of proposed algorithm in cross-domain intrusion detection scenarios.

Keywords: Intrusion detection, Domain adaptation, Domain-invariant features, Maximum mutual information, Pseudo label selection

1 Introduction

With the continuous development of the Internet, network security issues have become increasingly prominent. Traditional intrusion detection systems (IDS) typically rely on known threat characteristics or abnormal behavior to identify potential attacks, and perform poorly when dealing with unknown domains or unknown types of attacks. Existing IDS usually utilizes machine learning (ML) or deep learning (DL) techniques to correlate features of internet traffic, identify patterns in the collected data, and detect abnormal traffic corresponding to attacks. ML/DL based intrusion detection methods can perform well with enough labeled data on benchmark datasets. However, in the real world, labeling massive network traffic data is time-consuming and laborious, and there is a considerable difference in the distribution of traffic features between real-world datasets and benchmark datasets (domain shift) [1], the performance of traditional IDS is severely degraded [2] especially when applies to cross-domain scenarios. Therefore, it is not feasible to directly apply the model trained on the benchmark dataset to the real environment.

Semi-supervised learning can combine labeled samples and unlabeled samples to guide the learning of model, thereby overcoming the problem of scarce labeled data. Park et al. [3] proposed a semi-supervised learning method based on adversarial representation teaching, which combines labeled sample-guided representation teaching and adversarial noise-based consistency regularization (CR), enabling the teacher model with adversarial attack robustness to guide the student model to form a high-density distribution in the representation space, thereby maximizing the performance of semi-supervised learning. However, the semi-supervised model is prone to performance degradation due to domain drift, which is difficult to adapt to cross-domain intrusion detection scenarios with significant changes.

Domain adaptation can overcome the scarcity of labeled data and solve the domain shift problem. In the field of intrusion detection, many domain adaptive methods have been proposed, among which the domain adversary approach has been widely explored, involving aligning source and target domain features to extract domain-invariant representations [4]. The domain adversary method consists of three steps: (1) The source domain classifier is used to assist the source domain feature extractor to extract domain-invariant features. (2) The features extracted by the feature extractor of the target domain are aligned to the domain-invariant features. (3) The domain-invariant feature classifier is trained by using the domain-invariant feature of source domain and the label of source domain. Although some intrusion detection algorithms

like Domain-Invariant Network Intrusion Detection System(DI-NIDS) [5] and Malware Detection Adversarial Domain Adaptation framework(MD-ADA) [6] use unsupervised domain adaptive method to project original features into domain-invariant space, and use adversarial network to align the distribution of source domain and target domain, both of them still have limitations.

Firstly, the issues arising from data imbalance are often overlooked in both distribution alignment and classification tasks. In the context of intrusion detection, data imbalance—typically manifested as a disproportionately small number of malicious samples compared to benign ones—has been recognized as a critical challenge that significantly degrades classifier performance. This problem becomes even more pronounced under domain adaptation scenarios, where the alignment of feature distributions between source and target domains is a center issue. When the data from the target domain is severely imbalanced, the alignment process may inadvertently bias the learned domain-invariant representations towards the dominant class—usually benign traffic. As a consequence, the model may overfit to the benign class during the distribution matching phase, which leads to suboptimal feature representations for the minority (malicious) class and ultimately compromises the classification accuracy. Therefore, it is essential to consider and mitigate the influence of class imbalance throughout both the domain alignment and the subsequent classification stages.

Secondly, although DANN (Domain Adversarial Neural Networks) can align the feature distributions of the source and target domains to a certain extent and extract domain-invariant representations, it suffers from the semantic degradation of these features. The adversarial training strategy focuses primarily on reducing domain divergence rather than preserving class-level semantics. Furthermore, the adversarial objective of the domain discriminator conflicts with the classification objective of the task-specific classifier. This conflict can cause unstable gradients during training and hinder the classifier from effectively learning discriminative boundaries, especially when the target domain data is limited or exhibits complex distributions. Consequently, the classifier may not receive sufficient optimization, which ultimately leads to reduced detection performance on real-world traffic data.

Thirdly, under the assumption that the target domain data is unlabeled, methods such as DANN attempt to align source and target domain features through adversarial training, aiming to extract domain-invariant representations. However, aligning feature distributions alone does not guarantee good classification performance in the target domain. First, residual domain discrepancies may still exist after alignment, and such differences can impair the classifier’s generalization ability when applied to target domain samples. Second, since the feature alignment is performed in an unsupervised manner without leveraging label information from the target domain, there is no direct supervision to ensure that the aligned features possess sufficient class-discriminative structure. As a result, the domain-invariant features may not effectively separate different categories in the target domain, leading to degraded performance. These limitations indicate that relying solely on distribution alignment is insufficient for robust cross-domain classification, and highlight the need for incorporating additional

strategies—such as pseudo-label refinement, category-level alignment, or semantic consistency constraints—to better guide the learning of discriminative features in the absence of target domain labels.

To address the above problems, this paper proposes a semi-supervised domain adaptive intrusion detection algorithm based on generative adversarial network with maximum mutual information and pseudo-label selection. Firstly, an improved Limit Radius SMOTE(LR-SMOTE) [7] data enhancement method based on multiple samples and random smoothing averages is proposed to balance the data, which enhances the distribution fitting capability by introducing random center distance means and random distribution means. Secondly, to cope with distribution discrepancies and the scarcity of labeled data in the target domain, we adopt a domain adaptation approach. By leveraging a domain adversarial neural network, we align the source and target domain features in a shared domain-invariant space. To address the semantic information loss in traditional domain adversarial methods, a feature extraction strategy is proposed which combines deep mutual information maximization with adversarial learning. This approach maximizes the mutual information between the original and domain-invariant features to preserve semantic completeness while still aligning cross-domain representations. Finally, to improve classification performance under limited supervision, the classification task is reformulated as a semi-supervised problem. Uncertainty-aware Pseudo-label Selection(UPS) [8] is applied in conjunction with an OSVM classifier to perform semi-supervised learning on the extracted domain-invariant features, enabling more accurate intrusion detection in the target domain.

The paper is organized as follows. Section 2 presents the related work. Section 3 provides the proposed algorithm. Section 4 carries out the experiments. Section 5 is the conclusion.

2 Related Work

Existing intrusion detection methods can be divided into single-domain and cross-domain. Single-domain intrusion detection methods assume that the training data and test data have the same distribution, whereas cross-domain intrusion detection methods assume that the two distributions differ significantly, considering as the source domain and the target domain. For single-domain, Liu et al. [9] proposed an anomaly detection method based on Variational AutoEncoder(VAE) that decouples benign features and attack features in a semi-supervised manner by maximizing mutual information, then an adaptive filter is used to discard noisy samples and the intermediate features are denoised. Ding et al. [10] proposed Multi-frequency Intrusion Detection Network (MF-Net) for intrusion detection in internet traffic data. MF-Net regards internet traffic patterns as superpositions of sequential data with various frequencies and is capable of recognizing the multi-frequency nature of network traffic data. W. Al-Yaseen et al. [11] proposed a wrapper feature selection method based on Differential Evolution (DE) and Extreme Learning Machine (ELM) to enhance the performance and reduce the processing time of the Intrusion Detection System (IDS). Lopez-Martin

et al. [12] and Xiao et al. [13] proposed a traditional graph embedding method for network intrusion detection. The network traffic is converted into a graph based on the source and destination IPs and ports, and the graph embedding technique DeepWalk is applied for network intrusion detection. These methods only consider single-domain intrusion detection performance, the performance degrades severely due to the significant distribution differences between training data and test data, especially for the cross-domain scenario.

Domain adaptive method achieves the transfer effect by reducing the distribution difference between different domains. The source and target domain features can be mapped into the same feature space for domain alignment by extracting domain-invariant features. Recently, domain adaptation methods combined with deep networks have become increasingly popular. Ngo et al. [14] proposed a semi-supervised domain adaptive framework that combines Convolutional Neural Network (CNN) and Vision Transformer (ViT) models. It dynamically exchanges knowledge between the ViT model trained on the labeled source domain and the CNN model trained on a small number of labeled target samples by proposing the Dual Dynamic Consistency Regularization (D2CR) method. Thereby enhancing the generalization ability for unlabeled target data. Ngo et al. [15] proposed a novel semi-supervised domain adaptive method to address the issue of scarce labeled samples in the target domain. They trained two models respectively on the labeled source samples and a small number of labeled target samples, and through the collaborative work between the models, fully utilized the information of the unlabeled target data. Moreover, adversarial domain adaptation are widely used because it can automatically align the feature distributions of the source and target domains through unsupervised or semi-supervised adversarial training, achieving better cross-domain generalization. Therefore, this paper primarily introduces related work based on adversarial domain adaptation.

Inspired by Generative Adversarial Network (GAN), adversarial domain adaptation methods typically train a domain discriminator alongside the feature learning network through an adversarial mechanism. The feature learning network plays a role similar to but not the same as the generator in GANs. It is to learn the domain-invariant features rather than generate novel data, with the goal of confusing the domain discriminator. Domain-Adversarial Neural Network (DANN) [16] uses adversarial networks for domain adaptation and learns the domain-invariant features by adding an adversarially trained domain discriminator after the feature extractor. Adversarial domain adaptation [17] learns a domain adapter through adversarial training, mapping the source domain data to the target domain data distribution. It utilizes separately trained source and target domain feature extractors and an adversarially trained discriminator to distinguish between source and target domain features. Multi-adversarial domain adaptation [18] is a multi-domain adaptation method that introduces multiple domain classifiers and adversarial training strategies, enabling the model to adapt to the differences between multiple source and target domains simultaneously. Conditional domain adversarial network [19] introduces the concept of conditional adversarial training to minimize the differences between the source and target domains, thereby enhancing the generalization capability across different domains. Adversarial-learned loss for domain adaptation [20] introduces an adversarial

learning loss function, allowing the model to better learn feature representations that are insensitive to the differences between the source and target domains. In summary, these methods reduce the differences between the source and target domains through adversarial training, thereby improving the performance of cross-domain tasks. However, these methods only achieve global alignment of the feature distributions between the source and target domains, and overlooking the semantic integrity.

In the field of intrusion detection, Layeghy et al. [5] proposed DI-NIDS, which utilizes adversarial domain adaptation techniques to extract domain-invariant features from source and target domains and combines One-class SVM (OSVM) for anomaly detection. Ma et al. [21] proposed an adversarial domain adaptation method combining a dual-domain pairing strategy, designing a data augmentation algorithm and a lightweight feature extraction network, and incorporating an improved Adversarial Discriminative Domain Adaptation (ADDA) [17] model to construct a new loss function and domain discriminator. Li et al. [22] proposed a multi-source refined adversarial domain adaptation model for IoT intrusion detection, aiming to address cross-domain and intrusion class inconsistencies under limited sample conditions. Zhang et al. [23] proposed a semi-supervised domain adversarial training framework to adapt to new attacks occurring at different times, solving the issue of inconsistent training and testing data distributions in smart grids. Ajayi et al. [24] proposed a domain adaptation method to enhance the performance of host-based intrusion detection systems. Singla et al. [25] proposed a method using adversarial domain adaptation techniques to train a deep learning model for network intrusion detection, which aims to address the problem of training deep learning classification models with scarce labeled data by transferring knowledge from existing datasets. However, all the above methods do not consider the class imbalance problem during feature extraction and distribution alignment, and the integrity of the semantic information is lost.

3 Proposed algorithm

In order to improve the performance of adversarial domain adaptive intrusion detection, this paper proposes a semi-supervised domain adaptive intrusion detection algorithm based on generative adversarial network with maximum mutual information and pseudo-label selection. The overall framework of the proposed algorithm is shown in Figure 1. The proposed algorithm consists of three parts. Firstly, an improved LR-SMOTE oversampling algorithm based on multiple samples and random smoothing means is applied to the minority class data for data enhancement. Then, a source domain classifier is trained to assist the source domain feature extractor in extracting domain-invariant features. At the same time, the mutual information between the original features and the domain-invariant features is maximized to ensure the integrity of semantic information, and the extracted domain-invariant features are used for training the adversarial network. While aligning distributions, the mutual information between the input and output of the target domain feature extractor is maximized to ensure the integrity of the semantic information. Finally, in order to make full use of the target domain data, a domain-invariant feature classifier is trained in a semi-supervised manner by performing pseudo-label selection on the target domain data.

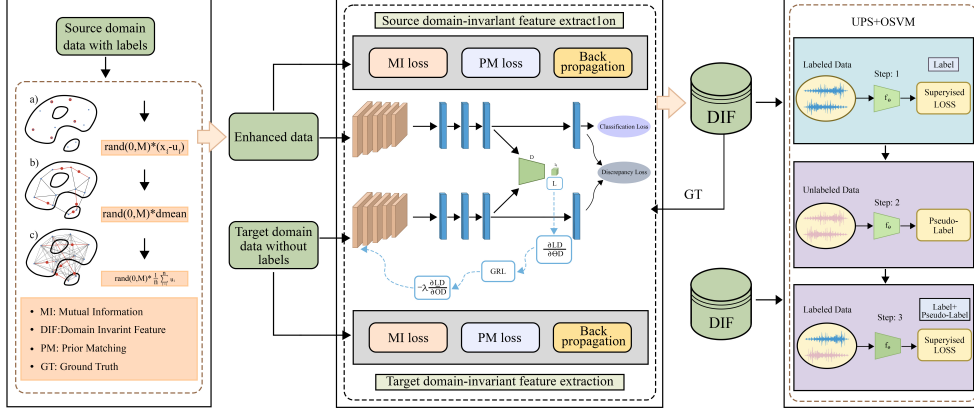


Fig. 1: The framework of proposed algorithm

3.1 Improved LR-SMOTE based on multiple samples and random smoothing mean

In traditional SMOTE, new samples are generated only between two samples, which can easily lead to overfitting. Therefore, LR-SMOTE calculates the sample center of the minority class, and enhances the data by adding a random center distance to the minority class samples, as shown in Equation 1.

$$x_{new} = u_i + rand(0, M) \cdot (x_c - u_i) \quad (1)$$

where x_{new} is the newly generated sample, u_i is a sample of minority class, x_c is the center of the minority class samples, $M = d_{mean}/d$, d_{mean} is the average distance of all samples from the sample center, and d is the distance of a sample from the sample center.

However, this still generates new data based on two points (if the sample center is considered as an existing sample). Although the sample generation range dynamically changes according to M , it cannot solve the overfitting problem and data distribution fitting problem. Therefore, random center distance means d_r and random distribution means u_r are added to the fitting process to address the above issues, where $d_r = rand(0, M) \cdot d_{mean}$, $u_r = rand(0, M) \cdot u_{mean}$, u_{mean} is the sample mean. By introducing d_r , the dependency on the number of samples for generating new samples is increased, making it no longer dependent on two samples but on all minority class samples for generation. By introducing u_r , data distribution information is incorporated into the generation process, allowing the generated samples to fit the corresponding data distribution. The final data generation process is shown in Equation 2.

$$\begin{aligned} x_{new} = & u_i + rand(0, M) \cdot (x_i - u_i) \\ & + rand(0, M) \cdot d_{mean} \\ & + rand(0, M) \cdot u_{mean} \end{aligned} \quad (2)$$

3.2 Domain-invariant feature extraction with integrated semantic information

Traditional domain adversarial networks leverage the idea of generative adversarial training to learn a domain discriminator, which aligns the feature distributions of the target and source domains in a shared domain-invariant space, thereby extracting domain-invariant features. However, the features obtained through this approach often lack complete semantic information. To address this issue, this paper proposes a domain-invariant feature extraction method based on deep mutual information maximization and domain adversarial learning. By maximizing the mutual information between the original features and the extracted domain-invariant features during the feature extraction process, the proposed method preserves richer semantic content, thereby enhancing the performance of the subsequent classification model.

3.2.1 Feature extractor

Both the original features of the source domain and the target domain require a feature extractor to map them to the domain-invariant space. Assuming that the dimensions of the original features and the domain-invariant features are u and v respectively, the feature extractor $E(x) : R^u \rightarrow R^v$ can be defined as shown in Equation 3.

$$E(x) = \text{sigmoid}(Wx + b) \quad (3)$$

where W represents the weights and b represents the biases of the model, $X \in R^{n \times u}$ denotes the input space of the feature extractor, consisting of n samples with u dimensions each. $X \sim \mathbb{P}$, which indicates the distribution of the random variable X is $\mathbb{P}$ over the input space. The output space of the feature extractor, i.e., the domain-invariant features, is denoted as $Z \in R^{n \times v}$, $Z \sim \mathbb{U}_{\mathbb{P}}$, where $\mathbb{U}_{\mathbb{P}}$ denotes the distribution of the domain-invariant feature Z .

3.2.2 Mutual information estimation

In information theory, mutual information is commonly used to measure the correlation or dependency between two random variables. Given two random variables X and Z , their mutual information is expressed as Equation 4, which indicates the extent to which observing one random variable reduces the uncertainty about the other random variable.

$$I(X; Z) = \sum_{x \in X} \sum_{z \in Z} p(x, z) \log \frac{p(x, z)}{p(x)p(z)} \quad (4)$$

where $p(x, z)$ is the joint probability distribution of the random variables X and Z , $p(x)$ and $p(z)$ are the marginal probability distribution of the random variables X and Z respectively. Mutual information is strongly positively correlated with variable dependency. A higher mutual information indicates a stronger correlation between the random variables X and Z , and vice versa. The variables X and Z are independent if the mutual information is 0.

However, the calculation of mutual information is challenging in continuous and high-dimensional data environments, because the data is always scarce in high-dimensional spaces, which leads to the difficulty to accurately estimate the joint and marginal distributions. According to the research by Hjelm et al. [26], the problem of maximizing mutual information between two random variables can be transformed into maximizing a lower bound of mutual information, thus addressing the difficulty of mutual information estimation. The lower bound of mutual information between two random variables X and Y can be expressed using the Donsker-Varadhan representation based on KL divergence, as shown in Equation 5.

$$\begin{aligned} I(X; Z) &= D_{KL}(\mathbb{J} \parallel \mathbb{M}) \\ &\geq \mathbb{E}_{\mathbb{J}}[T_w(x, z)] - \log(\mathbb{E}_{\mathbb{M}}[\exp(T_w(x, z))]) \end{aligned} \quad (5)$$

where $\mathbb{J}$ represents the joint distribution of X and Z , $\mathbb{M}$ represents the product of their marginal distributions. $T_w : R^u \times R^v \rightarrow R^1$ represents the feature discriminator with parameters w , which is used to distinguish whether a feature x is the original feature of a domain-invariant feature z . According to Equation 5, the contrastive learning approach is employed to maximize the lower bound of mutual information. Each domain-invariant feature z is combined with the corresponding original feature x to form a positive sample pair and with other original features to form negative sample pairs. All sample pairs are input into the feature discriminator to determine whether x and z are paired. Aiming to find the optimal parameters w, φ to maximize the lower bound of mutual information. Here w represents the parameters of the feature discriminator, and φ represents the parameters of the feature extractor. The loss function for maximizing the mutual information between input and output is shown in Equation 6.

$$\begin{aligned} MIloss &= \min_{w, \varphi} - \left(\frac{1}{n} \sum_{i=1}^n T_w(x_+^i, z^i) \right. \\ &\quad \left. - \log \frac{1}{n} \sum_{i=1}^n \exp(T_w(x_-^i, z^i)) \right) \end{aligned} \quad (6)$$

where, $MIloss$ means mutual information loss, x_+^i is the positive sample of z^i , i.e. $z^i = E(x_+^i)$, x_-^i is the randomly selected negative sample.

If the extracted domain-invariant features obey a prior distribution $\mathbb{V}$, it is helpful to make the encoding space more regular and decouples the features. Therefore, in order to match a prior distribution, a distribution distance discriminator $D_\phi : R^v \rightarrow R^1$ is trained to estimate and minimize the difference between $\mathbb{V}$ and $\mathbb{U}_{\mathbb{P}}$ denoted as $D(\mathbb{V} \parallel \mathbb{U}_{\mathbb{P}})$. The prior matching loss function is shown in Equation 7.

$$\begin{aligned} PMloss &= \min_{\varphi} \max_{\phi} D(\mathbb{V} \parallel \mathbb{U}_{\mathbb{P}}) \\ &= \mathbb{E}_{\mathbb{V}}[\log D_\phi(p)] + \mathbb{E}_{\mathbb{P}}[\log(1 - D_\phi(E(x)))] \end{aligned} \quad (7)$$

where $PMloss$ represents prior matching loss, $\mathbb{V}$ is the prior distribution that the domain-invariant feature obeys, $\mathbb{P}$ represents the empirical distribution of the original feature, $\mathbb{U}_{\mathbb{P}}$ represents the distribution of the domain-invariant feature Z , φ represents

the parameters of feature extractor E , D_ϕ represents the discriminator with parameters ϕ , which is used to identify whether the domain-invariant feature matches the prior distribution.

Finally, the loss of the feature extractor can be defined as Equation 8.

$$loss_E = MIloss + PMloss \quad (8)$$

$MIloss$ represents the loss of the feature extractor maximizing the lower bound of mutual information, and $PMloss$ represents the loss of the feature extractor matching the prior features.

3.2.3 Domain-invariant features extraction of source domain

The purpose of training the feature extractor in the source domain is to provide the ground truth domain-invariant features for the target domain. In order to extract useful features, traditional methods use the classifier of the source domain to assist in training the feature extractor, this only retain the class information of all the original features, resulting in the loss of domain information and integrated semantics, which will further affect the classification performance. In order to retain both class information and domain information, this paper proposes to maximize the mutual information between the original features and the domain-invariant features while correcting the class information during source domain training to ensure the integrity of the semantic information.

In this phase, the source domain data is used to pre-train the source domain feature extractor E^S and classifier C_α . Given the source domain input space $X^S \in R^{n \times u}$ and its corresponding labels $Y^S \in R^{n \times 1}$, a source domain feature extractor is trained to convert the original features into domain-invariant features $Z^S \in R^{n \times v}$, and an auxiliary classifier $C_\alpha : R^u \rightarrow R^1$ is trained to preserve the class information. While correcting the class information, the mutual information is maximized to preserve the integrated semantics. The loss function of this phase can be defined as Equation 9.

$$SourceDomainLoss = loss_{E^S} + loss_{C_\alpha} \quad (9)$$

where $loss_{E^S}$ represents the loss of the source domain feature extractor, $loss_{C_\alpha}$ represents the loss of the classifier, which is defined as Equation 10.

$$loss_{C_\alpha} = CrossEntropy(y, y_{pred}) \quad (10)$$

where y_{pred} represents the label predicted by the classifier.

The pseudo-code is shown in Algorithm 1.

3.2.4 Domain-invariant feature extraction of target domain

In this phase, the source domain-feature extractor is used to assist the domain discriminator for distribution alignment, and GAN is used to align the domain-invariant features of the target domain and the source domain. GAN consists of a generator G and a discriminator D , G is used to simulate the real distribution of data and D is used to judge the authenticity of the simulated data. The goal of the generator is

Algorithm 1 Domain-invariant features extraction of source domain

Input: X^S = feature of source domain

Y^S = label of source domain

Output: E^S = trained domain-invariant feature extractor of source domain

Z^S = domain-invariant feature of source domain

```
1:  $X^S, Y^S \leftarrow$  Data enhancement using Improved LR-SMOTE
2: Initialize source domain feature extractor  $E^S$ 
3: for each  $i = 1 \dots epoch$  do
4:   for  $x, y$  in  $X^S, Y^S$  do
5:      $z \leftarrow E^S(x)$  {Transform  $x$  to domain-invariant feature  $z$ }
6:      $loss_{E^S} \leftarrow \text{MILoss}(x, z) + \text{PMLoss}(z)$  {Calculate feature extraction loss based
       on Equation 8}
7:      $y_{pred} \leftarrow C_\alpha(z)$  {Predict labels using auxiliary classifier}
8:      $loss_{C_\alpha} \leftarrow \text{CrossEntropy}(y, y_{pred})$  {Calculate classifier loss}
9:      $SourceDomainLoss \leftarrow loss_{E^S} + loss_{C_\alpha}$ 
10:    Backward  $SourceDomainLoss$ 
11:    Update the parameters of  $E^S$  and  $C_\alpha$ 
12:   end for
13: end for
14: return  $E^S, E^S(X^S)$ 
```

to generate data as real as possible to deceive the discriminator, and the discriminator is to discriminate as correctly as possible. The objective function of GAN can be expressed as Equation 11.

$$\min_G \max_D V(D, G) = \mathbb{E}_{Z \sim P_{data}(z)} [\log D(z)] + \mathbb{E}_{X \sim P_X(x)} [\log(1 - D(G(x)))] \quad (11)$$

where $Z \sim P_{data}(z)$ represents the ground truth data used to correct the discriminator, $X \sim P_X(x)$ represents the input space of the generator which is the original features of the target domain. And the data used to correct the discriminator is the domain-invariant features of the source domain.

The features finally generated by the feature extractor of the target domain should have a distribution approximately the same as that of the source domain features to achieve the purpose of generating domain-invariant features. Therefore, the feature extractor $E^T : R^u \rightarrow R^v$ of the target domain is used to generate simulated data to deceive the discriminator. The domain information can be retained by the domain discriminator, while the class information of the features may be lost, and the integrity of the semantic information is not fully considered. In order to retain both the class information and domain information, the mutual information between the original features and the domain-invariant features are maximized to fully ensure the correlation between the original features and the domain-invariant features, so as to achieve the semantic integrity. The mutual information between the input and output of the feature extractor is maximized while discriminating the domain information with the

discriminator. The target domain extractor and the discriminator should be optimized as Equation 12.

$$\begin{aligned} \text{TargetDomainLoss} = \min_{\varphi} \max_{\sigma} & \frac{1}{n} \sum_{i=1}^n \log(D_{\sigma}(E^S(x_i^S))) \\ & + \frac{1}{m} \sum_{i=1}^m \log(1 - D_{\sigma}(E^T(x_i^T))) \\ & + \text{loss}_{E^T} \end{aligned} \quad (12)$$

where, E^S is the feature extractor of the source domain, which is used to provide the discriminator with distribution features. E^T is the feature extractor of the target domain, φ is the parameter of the target domain feature extractor, D_{σ} represents the domain discriminator with parameter σ , $X^S \in R^{n \times u}$ is the original feature of the source domain, $X^T \in R^{m \times u}$ is the original feature of the target domain, loss_{E^T} is the loss function of target domain extractor.

The pseudo-code is shown in Algorithm 2.

Algorithm 2 Domain-invariant feature extraction of target domain

Input: X^S = feature of source domain
 X^T = feature of target domain
 E^S = trained domain-invariant feature extractor of source domain
Output: E^T = trained domain-invariant feature extractor of target domain
 Z^T = domain-invariant feature of target domain

- 1: Initialize target domain feature extractor E^T
- 2: Initialize domain discriminator D
- 3: **for** each $i = 1 \dots \text{epoch}$ **do**
- 4: **for** x^t, x^s in X^T, X^S **do**
- 5: $z^s \leftarrow E^S(x^s)$ {Transform x^s to domain-invariant feature z^s }
- 6: $z^t \leftarrow E^T(x^t)$ {Transform x^t to domain-invariant feature z^t }
- 7: $\text{loss}_{E^T} \leftarrow \text{MILoss}(x^t, z^t) + \text{PMLoss}(z^t)$ {Calculate feature extraction loss based on Equation 8}
- 8: $\text{loss}_D \leftarrow \text{DomainAlignmentLoss}(x^t, x^s)$ {Calculate discriminator loss based on Equation 11}
- 9: $\text{TargetDomainLoss} \leftarrow \text{loss}_{E^T} + \text{loss}_D$
- 10: Backward TargetDomainLoss
- 11: Update the parameters of E^T and D
- 12: **end for**
- 13: **end for**
- 14: return $E^T, E^T(X^T)$

In summary, the algorithm for extracting domain-invariant features with integrated semantic information is as follows.

Step 1: Use the improved LR-SMOTE oversampling algorithm based on multi-samples and smoothed mean to enhance the source domain data.

Step 2: Train the source domain feature extractor E^S and source domain classifier C_α , and maximize $I(X^S; Z^S)$ at the same time to retain domain information, where X^S is the original feature of the source domain, Z^S is the domain-invariant feature of the source domain.

Step 3: Keep the parameters of E^S unchanged, use the adversarial network to train the target domain feature extractor E^D and domain discriminator D_σ , maximize $I(X^D; Z^D)$ at the same time to retain class information while aligning the distribution.

3.3 Semi-supervised domain-invariant feature classification method based on UPS and OSVM

Under the assumption of this paper, the target domain data is unlabeled. Although methods based on DANN align the features of the source and target domains through adversarial training to extract "domain-invariant features," mere distribution alignment does not guarantee good performance of the classifier on the target domain. This limitation arises for the following reasons: (1) There are still distribution discrepancies between the source and target domains. Feature alignment alone may not fully compensate for the classifier's performance degradation in the target domain. (2) The feature alignment is unsupervised and lacks direct supervision from the target domain's class information, making it difficult to ensure that the aligned features possess good class discriminability in the target domain.

To further improve the model's performance and allow the target domain data to better contribute to the classifier's training, this paper reformulates the problem as a semi-supervised learning problem. Specifically, the domain-invariant features extracted by DANN from both the source and target domains are treated as unlabeled data in a semi-supervised learning setting, with the labels from the source domain being considered as labels for a subset of the data. To address this, we propose a classifier training algorithm based on Uncertainty-aware Pseudo-label Selection (UPS) in the semi-supervised scenario. UPS selects high-confidence pseudo-labels by quantifying the uncertainty of the model's predictions, effectively reducing pseudo-label noise and improving the quality of the pseudo-labels, thus fully exploiting the potential of unlabeled data.

Furthermore, the One-Class Support Vector Machine (OSVM) is employed as the classifier in this semi-supervised setup. OSVM excels in scenarios where there is a lack of labeled data, as it focuses on identifying the boundary that best separates the majority class (benign traffic) from outliers (malicious traffic) in an unlabeled setting. OSVM is particularly advantageous in intrusion detection tasks because it is robust to noise and can effectively distinguish between normal and anomalous behaviors, making it a suitable choice for handling the challenges posed by the limited labeled data in the target domain.

3.3.1 Uncertainty-aware Pseudo-label Selection

Uncertainty-aware Pseudo-label Selection (UPS) is a semi-supervised learning method that selects pseudo labels by evaluating the uncertainty of the pseudo-label to ensure the high-confidence predictions, improves the learning effect of the model on unlabeled data and enhances the overall performance of the classifier. UPS selects pseudo labels with high certainty based on the relationship between confidence and uncertainty of pseudo labels. The algorithm execution process is as follows.

Step 1: Train the model f_θ on the labeled D_L , θ is the parameter of classifier.

Step 2: Generate pseudo labels according to $y_c^{(i)} = [p_c^{(i)} \geq \gamma]$ for all D_U . D_U represents dataset without labels, $\gamma \in (0, 1)$ represents the threshold, $p_c^{(i)}$ represents the model's predicted probability of the c -th class of the i -th sample, $y_c^{(i)}$ is the predicted pseudo label of the c -th class of the i -th sample.

Step 3: According to Equation 13, pseudo labels with high confidence are selected for the next round of training.

$$g_c^{(i)} = [u(p_c^{(i)}) \leq \kappa_p] \cdot [p_c^{(i)} \geq \tau_p] + [u(p_c^{(i)}) \leq \kappa_n] \cdot [p_c^{(i)} \leq \tau_n] \quad (13)$$

where $u(p)$ is the uncertainty of $p^{(i)}$, κ_p and κ_n are the uncertainty thresholds of positive and negative pseudo labels respectively, τ_p and τ_n represent the thresholds of positive pseudo labels and negative pseudo labels, $g_c^{(i)}$ represents the confidence of the c -th class of the i -th sample, which indicates whether this pseudo label is participated in next training.

Step 4: Use the dataset D_L and the pseudo-labeled data selected in Step 3. Repeat Steps 2 and 3 iteratively until convergence is observed.

3.3.2 One-class Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm widely used in classification and regression tasks. The basic principle of SVM is to maximize the interval between data points of different classes by finding an optimal separation hyperplane in high-dimensional space. One-class support vector machine (OSVM) is a special type of support vector machine, which is mainly used for anomaly detection and data distribution estimation. Different from traditional SVM, its basic idea is to surround the training data by constructing a hyperplane or boundary so that most training samples are contained in the boundary, and whether the test samples are in the boundary determines whether they are outliers. By learning the distribution of benign data, OSVM can effectively identify abnormal data that deviates from the normal distribution, which has a good effect on outlier detection.

3.3.3 Semi-supervised training method based on UPS and OSVM

In order to make full use of the domain-invariant feature of the target domain, this paper combines UPS to use the pseudo-label of the target domain to train the domain-invariant feature classifier OSVM in a semi-supervised way. The execution process is as follows.

Step 1: Input the original features of the source domain X^S into the trained source domain feature extractor to obtain the domain-invariant features Z^S , and input the target domain original features X^T into trained target domain feature extractor to obtain the domain-invariant features Z^T . It should be noted that the domain-invariant features of the source domain are labeled with classes.

Step 2: Train the domain-invariant feature classifier (OSVM) f_θ on Z^S .

Step 3: Generate pseudo labels according to $y_c^{(i)} = [p_c^{(i)} \geq \gamma]$ for all Z^T .

Step 4: Perform confidence detection on each generated pseudo label according to Equation 13, and discard pseudo labels with low confidence.

Step 5: Use the pseudo labels to train the new domain-invariant feature classifier f_θ , and repeat Steps 2-5 until the algorithm converges.

The pseudo-code is shown in Algorithm 3.

Algorithm 3 Uncertainty-aware Pseudo-label Selection

Input: Z^S = domain-invariant feature of source domain

Y^S = label of source domain

Z^T = domain-invariant feature of target domain

Output: well-trained domain-invariant feature classifier f_θ

- 1: Train a OSVM classifier $f_{\theta,0}$ using the samples from Z^S
 - 2: **for** each $i = 1 \dots epoch$ **do** {Repeat until convergence}
 - 3: Pseudo-label Z^T using $f_{\theta,i-1}$ {Equation 13}
 - 4: $Z_{selected} \leftarrow$ Select pseudo-labels using UPS {Equation 13}
 - 5: $\tilde{Z} \leftarrow Z_{selected} \cup Z^S$
 - 6: Initialize new classifier $f_{\theta,i}$
 - 7: Train $f_{\theta,i}$ using the samples from $\tilde{Z}$
 - 8: $f_\theta \leftarrow f_{\theta,i}$
 - 9: **end for**
 - 10: return f_θ
-

4 Experiments

The experimental environment is a 64-bit Ubuntu operating system. The CPU is Intel(R) Xeon(R) Gold 6133 CPU @ 2.50GHz, the memory size is 128GB, the GPU is GeForce RTX 4090, and the GPU memory is 24GB. The hyper-parameters are shown in Table 1. Where Batch size is the number of samples in one training iteration, τ_p and τ_n represent the thresholds of positive and negative pseudo labels, κ_p and κ_n are the uncertainty thresholds of positive and negative pseudo labels respectively, kernel defines how data is mapped to a high-dimensional space.

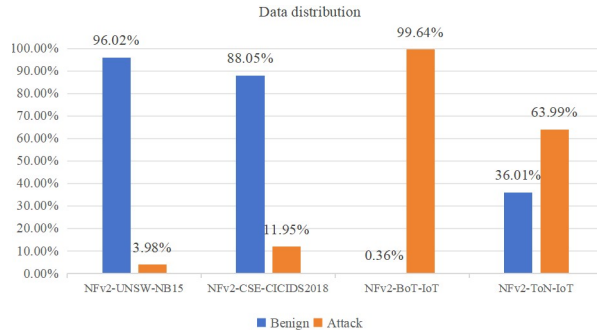
4.1 Datasets

Four NIDS benchmark datasets [27] are used to evaluate the proposed algorithm, namely NFv2-UNSW-NB15 (Network Flow Version 2 of University of New South

Table 1: Hyper parameters used in the training process.

Parameter	Value
epoch	1024
Feature extractor layers (S/T)	[64, 64, 64]
Auxiliary classifier layers	[64, 32, 16, 8, 4, 2]
Feature discriminator layers	[64, 32, 16, 8, 4, 2]
Distribution distance discriminator layers	[64, 32, 16, 8, 4, 2]
Domain discriminator layers	[64, 32, 16, 8, 4, 2]
Learning rate	0.001
Batch size	128
τ_p	0.7
τ_n	0.05
κ_p	0.05
κ_n	0.005
kernel	rbf

Wales Network Benchmark) , NFv2-CSE-CIC-IDS2018 (Network Flow Version 2 of Communications Security Establishment & the Canadian Institute for Cybersecurity) , NFv2-BoT-IoT (Network Flow Version 2 of BotNet Internet of Things) , and NFv2-ToN-IoT (Network Flow Version 2 of The Open Network Internet of Things) , corresponding to the network flow versions of UNSW-NB15 [28], CSE-CIC-IDS2018 [29], BoT-IoT [30], and ToN-IoT [31]. The four datasets are collected in completely different network environments and represent different network traffic distributions, which are suitable for cross-domain experimental verification. NFv2-UNSW-NB15 and NFv2-CSE-CIC-IDS2018 are the network traffic in the traditional network environment, and NFv2-BoT-IoT, and NFv2-ToN-IoT are the traffic characteristics of the Internet of Things. This paper focuses on the binary classification, so all the traffic are classified as Benign and Attack. Figure 2 shows the distribution of the datasets, where there is a serious class imbalance.

**Fig. 2:** Data distribution of the datasets

4.2 Evaluation indicators and data preprocessing

4.2.1 Evaluation indicators

In order to comprehensively evaluate the proposed algorithm, Precision, Recall, F1-Score, and FPR (False Positive Rate) are used as the indicators, which are based the confusion matrix as shown in Table 2.

Table 2: Confusion matrix

	Positive (Benign)	Negative (Attack)
Positive (Benign)	TP	FN
Negative (Attack)	FP	TN

TP (True Positive) indicates that the data is Benign and predicted to be Benign, TN (True Negative) indicates that the data is Attack and predicted to be Attack, FN (False Negative) indicates that the data is Benign but predicted to be Attack, and FP (False Positive) indicates that the data is Attack but predicted to be Benign.

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + recall} \quad (16)$$

$$FPR = \frac{FP}{FP + TN} \quad (17)$$

In addition, in cross-domain experiments, Degradation is employed to measure the performance reduction when transferring from a single-domain experiment on a certain dataset to a cross-domain experiment using the same dataset as the target domain. In this work, the Degradation value is defined as the difference between the F1-Score in the single-domain scenario and the F1-Score in the cross-domain scenario:

$$Degradation = F1_{\text{single}} - F1_{\text{cross}}. \quad (18)$$

A smaller Degradation indicates better cross-domain generalization capability.

4.2.2 Data preprocessing

The useless features such as IPADDR, PORT are removed, as well as the samples with missing values and infinite values. In order to make the data features within the same scale range, the data is standardized as shown in Equation 19. Where x_{max} and x_{min} are the maximum and minimum values of feature x .

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (19)$$

4.3 Experimental results

To verify the effectiveness of the proposed algorithm, three group experiments are designed. Firstly, a single-domain experiments on the NFv2-UNSW-NB15 and NFv2-CSE-CICIDS2018 datasets are carried out and compared with other algorithms including domain adaptive algorithms. The single-domain experiments is to provide a baseline for cross-domain evaluation. Then, the four datasets are divided into two groups to evaluate the cross-domain performance, NFv2-UNSW-NB15 and NFv2-CSE-CICIDS 2018 in one group, and the other two datasets in another group. Each dataset served as the source domain and then the target domain. Finally, the ablation experiments are carried out to verify the effectiveness of proposed algorithm. In the comparative experiments with other algorithms, only F1-Score is used which can relatively comprehensively measure the algorithm performance because it's a harmonic average of Precision and Recall.

4.3.1 Single-domain experiment

In order to provide a benchmark for cross-domain experiments, single-domain experiments are conducted on NFv2-UNSW-NB15 and NFv2-CSE-CIC-IDS2018. In these experiments, the traing process and the test process are on the same dataset. Training data is regarded as the source domain data, and the test data is regarded as the target domain data. The algorithms OSVM, Random-Forest, Isolation-Forest, Extra Tree, DANN, and DI-NIDS [5] are used for comparison.

Table 3: F1-Score of single-domain experiments on NFv2-UNSW-NB15 and NFv2-CSE-CIC-IDS2018

Algorithm	UNSW-NB15	IDS2018
OSVM	98.28%	92.97%
Random Forest	92.17%	95.44%
Isolation Forest	87.43%	91.02%
Extra Tree	91.73%	84.62%
LSTM NN	92.82%	90.17%
DANN	93.38%	97.81%
DI-NIDS	98.68%	93.23%
Proposed method	97.71%	97.82%

IDS2018: CSE-CIC-IDS2018 dataset.

Table 3 shows the single-domain performance of the proposed algorithm and other algorithms. It can be seen that the single-domain detection performance of each algorithm is not significantly different. The proposed algorithm reaches the highest 97.82%

on the NFv2-CSE-CICIDS2018 dataset, and reaches 97.71% on NFv2-UNSW-NB15 dataset, which is only 0.97% lower than the best one. The main reason for the insignificant performance difference of each algorithm is that the training dataset and the test dataset are from the same dataset, that is, the distribution of the source domain and the target domain are quite similar, which makes it impossible to show the advantage of cross-domain distribution alignment in this paper.

4.3.2 Cross-domain experiments

In cross-domain experiments, the degradation refers to the performance reduction from a single-domain experiment on a certain dataset and to the cross-domain experiment using the dataset as the target domain, which can reflect the generalization ability of the algorithms, thus the degradation indicator is used in the cross-domain experiments to better reflect the cross-domain capability and generalization performance.

Table 4 and Table 5 show the results of the first group of cross-domain experiments. It can be seen that the performance of the proposed algorithm is 92.84% in the first cross-domain scenario and 94.16% in the second cross-domain scenario, both of which are the best. In addition, the degradations are the lowest as 4.98% and 3.55% respectively, which fully demonstrates the generalization ability of the proposed algorithm. It can be seen that the traditional machine learning or deep learning algorithms, such as OSVM, Random-Forest, Isolation-Forest, LSTM NN, and Extra Tree, have low average performances because they assume that the training data and test data are of the same distribution without the cross-domain consideration. The F1-score and the degradation of proposed algorithm achieves the best, which are higher than the DANN algorithm and DI-NIDS algorithm based on domain adversary, because both of the two algorithms do not retain the integrated semantic information equivalent to the original features, and the proposed algorithm maximizing the mutual information between the domain-invariant features and the original features, taking into account both the class information and domain information, making full use of the target domain data, thus achieves the best classification performance.

Table 4: Cross-domain performance from CSE-CIC-IDS2018 to UNSW-NB15

Algorithm	F1-Score	Degradation
OSVM	86.15%	6.79%
Random Forest	0.84%	94.60%
Isolation Forest	26.96%	64.06%
LSTM NN	9.63%	80.54%
Extra Tree	0.57%	84.05%
DANN	17.31%	80.50%
DI-NIDS	85.79%	7.44%
Proposed method	92.84%	4.98%

Table 5: Cross-domain performance from UNSW-NB15 to CSE-CIC-IDS2018

Algorithm	F1-Score	Degradation
OSVM	15.74%	82.54%
Random Forest	7.70%	84.47%
Isolation Forest	32.80%	54.63%
LSTM NN	14.20%	78.62%
Extra Tree	17.47%	74.26%
DANN	61.94%	31.44%
DI-NIDS	93.29%	5.39%
Proposed method	94.16%	3.55%

Table 6: The cross-domain performance of the proposed algorithm on NFv2-BoT-IoT and NFv2-ToN-IoT

Training dataset	Test dataset	Precision	Recall	F1-Score	FPR	Degradation
NFv2-BoT-IoT	NFv2-BoT-IoT	96.13%	95.02%	95.57%	1.98%	3.44%
	NFv2-ToN-IoT	93.17%	91.31%	92.23%	3.16%	
NFv2-ToN-IoT	NFv2-ToN-IoT	97.76%	99.21%	98.45%	1.18%	6.59%
	NFv2-BoT-IoT	90.97%	92.75%	91.86%	2.77%	

Table 7: Cross-domain F1-Score performance of different classifiers. Extra-Tree(ET), Random-Forest(RF)

Target	Source	ET	RF	Feed Foward	LSTM	OSVM
NFv2-BoT-IoT	NFv2-CSE-CICIDS2018	64.15%	62.91%	54.74%	49.35%	71.83%
NFv2-CSE-CICIDS2018	NFv2-BoT-IoT	22.32%	22.32%	22.32%	28.15%	56.86%
NFv2-ToN-IoT	NFv2-UNSW-NB15	73.39%	19.91%	3.82%	81.40%	83.44%
NFv2-UNSW-NB15	NFv2-ToN-IoT	59.82%	56.25%	68.91%	44.10%	74.85%

In order to further verify the effectiveness of the algorithm, NFv2-BoT-IoT and NFv2-ToN-IoT datasets are used for the cross-domain experiments. It can be seen from Table 6 that the F1-Scores of the proposed algorithm in the two cross-domain scenarios are 92.23% and 91.86%, respectively, and the degradation is 3.44% and 6.59%, respectively. Specifically, the multiple indicators Precision, Recall and F1-score of the proposed algorithm are also high, and the FPR and degradation is low, which fully demonstrates that the proposed algorithm in this paper successfully aligns the source domain and the target domain, learns the correct domain-invariant features, and can accurately detect abnormal traffic in the network.

4.3.3 Effectiveness of OSVM

One-Class Support Vector Machine (OSVM) is a semi-supervised learning approach particularly suited for anomaly detection and domain generalization tasks. Unlike traditional classifiers such as Extra Trees (ET), Random Forest (RF), Feedforward Neural Networks, and LSTM, OSVM requires only positive samples during training. It offers strong generalization across domains, is computationally efficient, and performs well in high-dimensional feature spaces, making it ideal for scenarios with limited or unlabeled data.

Table 7 demonstrate the superiority of OSVM in cross-domain detection tasks. In all four source-target pairs, OSVM consistently outperforms other models. Notably, it achieves 71.83% accuracy when transferring from NFv2-CSE-CICIDS2018 to NFv2-BoT-IoT and 56.86% in the reverse direction, while other models perform significantly worse. These findings underscore OSVM’s robustness and effectiveness in handling distribution shifts and identifying outliers in unseen environments.

4.3.4 Ablation experiments

In order to verify components of the proposed algorithm, four ablation experiments are conducted. The first ablation experiment is to verify the effectiveness of data enhancement, the second ablation experiment is to verify the effectiveness of adversarial network, and the third ablation experiment is to verify the effectiveness of uncertain pseudo-label selection method for classification. The fourth ablation experiment is the complete algorithm of this paper. The experimental configuration is shown in Table 8.

Table 8: Configuration of the four ablation experiments

Experiment	Data enhancement	MI maximization	Pseudo label selection
1	No	Yes	Yes
2	Yes	No	Yes
3	Yes	Yes	No
4	Yes	Yes	Yes

The cross-domain ablation experiment with source domain NFv2-CSE-CICIDS2018 and target domain NFv2-BoT-IoT are carried out in the same experimental environment to verify the impact of data enhancement, mutual information maximization, and pseudo-label semi-supervised training methods on the overall performance of the proposed algorithm as shown in Table 9. There is no data enhancement in Experiment 1, and the performances are close to 50% due to the serious class imbalance of the data, which further affects the classification process and distribution alignment process. From Experiment 2, it can be seen that without maximizing the mutual information between the original features and the domain-invariant features, the integrated semantics of the domain-invariant features are not retained, resulting

in the loss of class information in the target domain and also the performance degradation. In Experiment 3, the domain-invariant feature classifier is only trained by the domain-invariant features of the source domain, and the domain-invariant features of the target domain are not involved. Since the two distributions could not be completely aligned, the classification performance is poor.

To verify the training efficiency of the algorithm proposed in this paper, the convergence rate of the training loss of the model after ablation was compared in this paper. In this paper, the definition of the convergence point is as follows. When the loss change of the model for five consecutive epochs starting from the p -th epoch of training iteration is less than 0.01, then p is called the convergence point. The calculated training loss curve is shown in Figure 3.

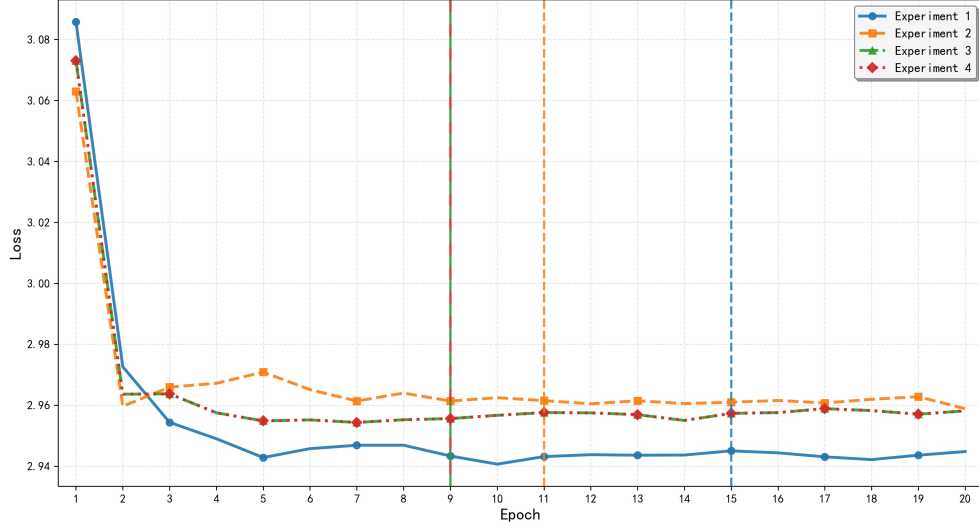


Fig. 3: The comparison of training loss convergence rate

As can be seen from Figure 3, the convergence rate of Experiment 1 is the slowest. This is because the class imbalance of the data leads to the loss being prone to oscillation, which makes the model training difficult to converge. After adding mutual information maximization, the convergence rate has been relatively improved. This is because mutual information maximization can effectively retain the integrated semantics of domain-invariant features and the class information in the target domain, thereby further enhancing the learning ability of the model and enabling the model training to converge more quickly. After adding data enhancement and mutual information maximization, the training convergence rate of the model has been further improved. Since the pseudo-label selection does not participate in the model training, the curves of Experiment 3 and Experiment 4 completely overlap.

PCA (Principal Components Analysis) can be used for data dimensionality reduction to see why the data imbalance affects the classification performance. Figure 4

Table 9: Ablation experiment comparison from NFv2-CSE-CICIDS2018 to NFv2-BoT-IoT

Exp.	Precision	Recall	F1-Score	FDR
1	50.17%	55.49%	52.69%	44.43%
2	25.18%	49.58%	33.39%	32.13%
3	74.27%	81.34%	77.64%	13.21%
4	91.61%	88.21%	89.87%	5.52%

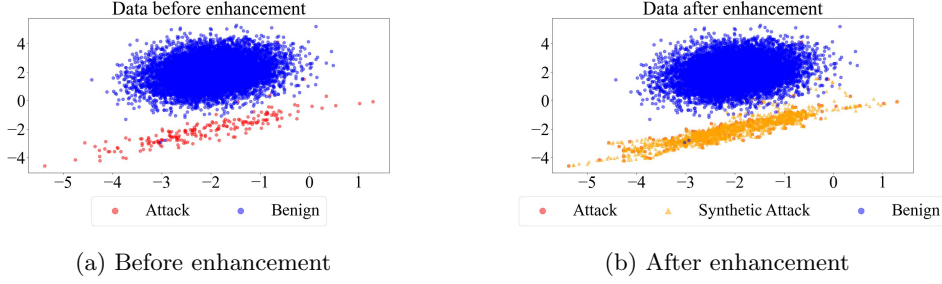


Fig. 4: Data distribution before and after data enhancement

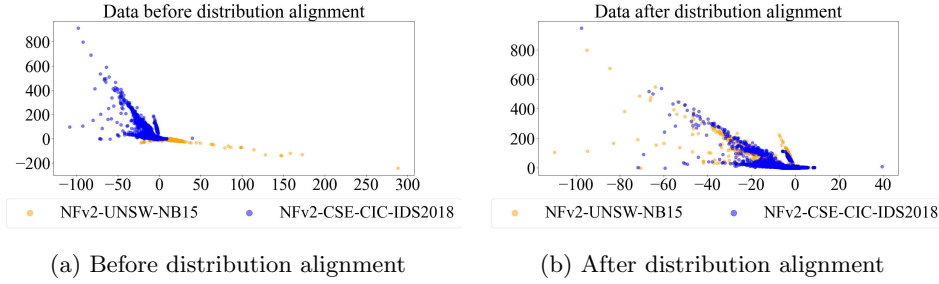


Fig. 5: Data distribution before and after distribution alignment

shows the two-dimensional data distribution of the NFv2-UNSW-NB15 dataset before and after data enhancement. It can be seen that the minority class data is improved by data enhancement, and the data distribution of the minority class is effectively preserved. Figure 5 shows the data distribution of the NFv2-UNSW-NB15 dataset and the NFv2-CSE-CIC-IDS2018 dataset before and after distribution alignment after dimensionality reduction. It can be seen that before distribution alignment, the distribution of the two data sets rarely overlaps, but after distribution alignment, the two sets of data are effectively aligned.

5 Conclusion

Aiming at the problems of scarce labeled data and domain distribution differences in intrusion detection, this paper proposes a semi-supervised intrusion detection algorithm based on mutual information maximization adversarial network and pseudo-label selection. The problem of data imbalance is solved by the improved LR-SMOTE data enhancement algorithm based on multi-sample random smoothed mean. Furthermore, to avoid the loss of semantic information when extracting domain-invariant features, a new method based on mutual information maximization adversarial networks is proposed. Subsequently, a pseudo-label selection method is adopted to train the domain-invariant feature classifier OSVM in a semi-supervised manner, enabling it to better adapt to the target domain data. Experimental results show that the proposed algorithm performs exceptionally well in both single-domain and cross-domain scenarios, with ablation experiments further validating its effectiveness.

The algorithm proposed in this paper can be further optimized in terms of computational overhead and generalization ability of unknown attack recognition. In the future, more efficient mutual information estimation methods will be studied, and methods such as graph neural network or reinforcement learning will be combined to further improve the adaptability and robustness of the model in complex network environments.

Acknowledgments

This work is supported by the S&T Program of Hebei [grant number 236Z0304G]. The authors are grateful to valuable comments and suggestions of the reviewers.

Funding

This work was supported by the S&T Program of Hebei (No. 236Z0304G).

Conflict of interest

The authors declare that they have no conflict of interest.

Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Qian Wang, Jiahui Shang and Weihao Huang. The first draft of the manuscript was written by Qian Wang and Jiahui Shang, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability

The datasets used and analyzed during the current study available from the corresponding author on reasonable request.

Ethics approval

Not applicable.

References

- [1] Layeghy, S., Gallagher, M., Portmann, M.: Benchmarking the benchmark—comparing synthetic and real-world network ids datasets. *Journal of Information Security and Applications* **80**, 103689 (2024)
- [2] Wang, Q., Fan, M., Wu, Z., Yu, H., Cheng, Y., Zhang, B.: A domain adaptive iot intrusion detection algorithm based on aec-gat feature extraction and joint domain adversary. *IEEE Transactions on Industrial Informatics* **22**(3), 1850–1860 (2026) <https://doi.org/10.1109/TII.2025.3631964>
- [3] Park, J.H., Kim, J.H., Ngo, B.H., Kwon, J.E., Cho, S.I.: Adversarial representation teaching with perturbation-agnostic student-teacher structure for semi-supervised learning. *Applied Intelligence* **53**, 26797–26809 (2023) <https://doi.org/10.1007/s10489-023-04950-5>
- [4] Pan, X., Chen, H., Wang, W., Su, X.: Adversarial domain adaptation based on contrastive learning for bearings fault diagnosis. *Simulation Modelling Practice and Theory* **139**, 103058 (2025) <https://doi.org/10.1016/j.simpat.2024.103058>
- [5] Layeghy, S., Baktashmotlagh, M., Portmann, M.: Di-nids: Domain invariant network intrusion detection system. *Knowledge-Based Systems* **273**, 110626 (2023)
- [6] Bhardwaj, S., Li, A.S., Dave, M., Bertino, E.: Overcoming the lack of labeled data: Training malware detection models using adversarial domain adaptation. *Computers & Security* **140**, 103769 (2024)
- [7] Liang, X., Jiang, A., Li, T., Xue, Y., Wang, G.: Lr-smote—an improved unbalanced data set oversampling based on k-means and svm. *Knowledge-Based Systems* **196**, 105845 (2020)
- [8] Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In: *International Conference on Learning Representations*, pp. 1–20 (2021)
- [9] Liu, S., Tian, M.: Mutual information maximization for semi-supervised anomaly detection. *Knowledge-Based Systems* **284**, 111196 (2024)
- [10] Ding, Z., Zhong, G., Qin, X., Li, Q., Fan, Z., Deng, Z., Ling, X., Xiang, W.: Mf-net: Multi-frequency intrusion detection network for internet traffic data. *Pattern Recognition* **146**, 109999 (2024)

- [11] Al-Yaseen, W.L., Idrees, A.K., Almasoudy, F.H.: Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognition* **132**, 108912 (2022)
- [12] Lopez-Martin, M., Carro, B., Arribas, J.I., Sanchez-Esguevillas, A.: Network intrusion detection with a novel hierarchy of distances between embeddings of hash ip addresses. *Knowledge-based systems* **219**, 106887 (2021)
- [13] Xiao, Q., Liu, J., Wang, Q., Jiang, Z., Wang, X., Yao, Y.: Towards network anomaly detection using graph embedding. In: *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part IV* 20, pp. 156–169 (2020). Springer
- [14] Ngo, B.H., Lam, B.T., Nguyen, T.H., Dinh, Q.V., Choi, T.J.: Dual dynamic consistency regularization for semi-supervised domain adaptation. *IEEE Access* **12**, 36267–36279 (2024) <https://doi.org/10.1109/ACCESS.2024.3374105>
- [15] Ngo, B.H., Kim, J.H., Chae, Y.J., Cho, S.I.: Multi-view collaborative learning for semi-supervised domain adaptation. *IEEE Access* **9**, 166488–166501 (2021) <https://doi.org/10.1109/ACCESS.2021.3136567>
- [16] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. *Journal of machine learning research* **17**(59), 1–35 (2016)
- [17] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176 (2017)
- [18] Zhang, Z., Zhong, Z., Li, Z., Mao, W., Cui, Y.: Failure mechanism-driven multi-adversarial domain transfer learning for rolling bearing fault diagnosis. *Results in Engineering* **27**, 106165 (2025) <https://doi.org/10.1016/j.rineng.2025.106165>
- [19] Deng, M., Zhou, D., Ao, J., Xu, X., Li, Z.: Bearing fault diagnosis of variable working conditions based on conditional domain adversarial-joint maximum mean discrepancy. *The International Journal of Advanced Manufacturing Technology* **136**(11), 5043–5060 (2025)
- [20] Chen, M., Zhao, S., Liu, H., Cai, D.: Adversarial-learned loss for domain adaptation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3521–3528 (2020)
- [21] Ma, W., Liu, R., Li, K., Yan, S., Guo, J.: An adversarial domain adaptation approach combining dual domain pairing strategy for iot intrusion detection under few-shot samples. *Information Sciences* **629**, 719–745 (2023)

- [22] Li, K., Ma, W., Duan, H., Xie, H.: Multi-source refined adversarial domain adaptation with transfer complementarity infusion for iot intrusion detection under limited samples. *Expert Systems with Applications*, 124352 (2024)
- [23] Zhang, Y., Yan, J.: Semi-supervised domain-adversarial training for intrusion detection against false data injection in the smart grid. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–7 (2020). IEEE
- [24] Ajayi, O., Gangopadhyay, A.: Dahid: Domain adaptive host-based intrusion detection. In: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 467–472 (2021). IEEE
- [25] Singla, A., Bertino, E., Verma, D.: 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, pp. 127–140 (2020)
- [26] Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: International Conference on Learning Representations, pp. 1–24 (2018)
- [27] Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. *Mobile networks and applications*, 1–14 (2022)
- [28] Moustafa, N., Slay, J.: 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), pp. 1–6 (2015). IEEE
- [29] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., *et al.*: Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **1**, 108–116 (2018)
- [30] Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* **100**, 779–796 (2019)
- [31] Moustafa, N.: Ton-iot datasets (2019). doi: 10.21227/fesz-dm97. URL <http://dx.doi.org/10.21227/fesz-dm97> (2019)