



Article

Deep Learning-Enabled Heterogeneous Transfer Learning for Improved Network Attack Detection in Internal Networks

Gang Wang , Dong Liu, Chunrui Zhang * and Teng Hu 

Institute of Computer Application, China Academy of Engineering Physics, Mianyang 621000, China; gary.wang@caep.cn (G.W.); liudong@caep.cn (D.L.); mailhuteng@gmail.com (T.H.)

* Correspondence: zhangcr@caep.cn

Featured Application: This work has potential usages in cyber-attack detection in air-gapped internal networks that lack sufficient labeled data samples to build detection models for network attack activities.

Abstract: Cybersecurity faces constant challenges from increasingly sophisticated network attacks. Recent research shows machine learning can improve attack detection by training models on large labeled datasets. However, obtaining sufficient labeled data is difficult for internal networks. We propose a deep transfer learning model to learn common knowledge from domains with different features and distributions. The model has two feature projection networks to transform heterogeneous features into a common space, and a classification network then predicts transformed features into labels. To align probability distributions for two domains, maximum mean discrepancy (MMD) is used to compute distribution distance alongside classification loss. Though the target domain only has a few labeled samples, unlabeled samples are adequate for computing MMD to align unconditional distributions. In addition, we apply a soft classification scheme on unlabeled data to compute MMD over classes to further align conditional distributions. Experiments between NSL-KDD, UNSW-NB15, and CICIDS2017 validate that the method substantially improves cross-domain network attack detection accuracy.

Keywords: cybersecurity; attack detection; deep learning; heterogeneous transfer learning



Citation: Wang, G.; Liu, D.; Zhang, C.; Hu, T. Deep Learning-Enabled Heterogeneous Transfer Learning for Improved Network Attack Detection in Internal Networks. *Appl. Sci.* **2023**, *13*, 12033. <https://doi.org/10.3390/app132112033>

Academic Editor: Luis Javier Garcia Villalba

Received: 24 September 2023

Revised: 23 October 2023

Accepted: 2 November 2023

Published: 4 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the ubiquitous application of internet and mobile communications [1], network attackers have more opportunities to compromise devices and applications for sabotaging the infrastructure or stealing valuable data [2]. In addition, due to system and application vulnerabilities, attack methods have been evolving to be more and more sophisticated, which poses an unprecedented challenge to the field of cybersecurity. The ability to detect and respond to these novel and evolving threats in real time has become critical for safeguarding sensitive information and ensuring the integrity of digital assets [3,4].

The realm of network attack detection has been the subject of exhaustive research efforts, yielding a plethora of innovative approaches tailored to the task of identifying and categorizing malicious activities with remarkable precision. Within this landscape, traditional techniques embedded within intrusion detection systems (IDS), including signature-based and anomaly-based methods, have played a pivotal role in shedding light on well-established attack patterns, thus contributing significantly to the field's knowledge base [5–7]. However, these time-honored methods are not without their limitations, particularly when confronted with the challenge of recognizing previously unseen or novel attack strategies. One notable drawback of these conventional approaches arises from their reliance on static signature databases or normal patterns. However, signature-based

methods require ongoing database curation, which places a substantial burden on human experts, demanding their vigilant efforts in identifying, analyzing, and cataloging new attack signatures. To maintain a high level of accuracy in attack detection for anomaly-based methods, these normal patterns must be constantly updated to encompass emerging normal behaviors with threats and evolving attacks. Moreover, the process of updating the signature database introduces an inherent time delay, which can hinder the timely detection of emerging threats and compromise network security [8]. These challenges underscore the critical need for more adaptive and proactive approaches in the ever-evolving landscape of network attack detection.

To address this ever-growing concern, researchers have turned to machine learning techniques to extract features and build models [7,9,10]. Machine learning, particularly deep learning, has demonstrated its potential in learning intricate patterns and features from data, making it an attractive approach for network security applications [11,12]. Deep learning has established itself as a formidable force, showcasing impressive achievements across a spectrum of domains that encompass computer vision, natural language processing, and speech recognition. In network detection, deep learning can be used for representation learning to automatically discover the features needed for detection, as the collected data features do not directly reflect use behaviors in networks. It can also be used to learn complicated user behavior sequences with a recurrent neural network (RNN) and the recently popular Transformer and learning relationship and interactions between entities in networks for anomaly detection with a graph neural network (GNN) [11]. The construction of robust and accurate classification models typically relies on substantial amounts of labeled data to effectively capture the intricacies of network attacks. However, in the context of internal networks, acquiring sufficient labeled data for training presents a significant challenge due to the sensitive nature of the data and the inherent difficulty in obtaining real-world attack samples [13]. As a result, the model training procedure always receives insufficient data samples from outdated datasets, which significantly degrades the performance of machine learning algorithms.

To overcome this limitation, researchers have turned their attention to transfer learning as a viable solution [14–19]. Transfer learning leverages knowledge from source domains with labeled data and applies it to target domains where labeled samples may be scarce or entirely absent. This approach has shown promise in network attack detection tasks, as it facilitates the extraction of relevant knowledge from external networks to improve the detection capabilities of models operating in internal network environments.

In the context of an internal network, where the availability of labeled data is notably scarce, the endeavor of constructing a robust prediction model for the detection of network attacks presents a formidable challenge. Given this predicament, it becomes imperative to explore avenues for model enhancement, and one such approach involves the integration of data gleaned from external networks, leveraging the principles of transfer learning. However, the introduction of data from external sources brings forth a significant caveat: the inherent diversity among networks often results in disparities within the feature space characterizing their respective data collections.

The main hurdle in applying transfer learning for network attack detection lies in addressing the heterogeneity in feature spaces and probability distributions between source and target domains. Given the potential for variations in communication network types, service categories, and data acquisition techniques across the two domains under consideration, it is conceivable that the feature spaces collected from these networks could exhibit disparities. These distinctions may arise due to fundamental distinctions in the architecture, protocols, and operational objectives inherent to each network, all of which influence the types of data collected and the resulting feature representations. Consequently, these disparities in feature spaces pose a fundamental challenge when attempting to align distributions and extract valuable insights from datasets of these distinct network domains. Specifically, the crux of the matter is how to identify an intermediary, universally applicable data representation capable of bridging these discrepancies across disparate

feature spaces. In addition to the disparate feature spaces, an equally critical objective is to align the probability distributions inherent in datasets originating from corresponding networks, similar to most domain adaptation work [20,21]. Recent studies have highlighted the importance of aligning these distributions to ensure the effective transfer of knowledge and improve model generalization. However, few studies have focused on deep learning-enabled heterogeneous transfer learning [14], which can overcome these challenges by learning common knowledge from domains with different feature spaces.

In this paper, we propose a novel deep learning-enabled heterogeneous transfer learning model tailored explicitly for network attack detection in internal networks. In network detection, deep learning can be used for representation learning to automatically discover the features needed for detection, as the collected data features do not directly reflect use behaviors in networks [22]. By applying transfer learning, we try to align the probability distribution of the source and target domains so that the data from the source and target domain can be used in the same model without concept drifting [21,23]. The main contribution of this article is summarized as follows:

- Two feature projection networks are built for the source and target domains, transforming heterogeneous feature data into a shared, unified feature space. By learning domain-specific representations, our model effectively mitigates feature space heterogeneity and establishes a foundation for seamless knowledge transfer.
- We employ the maximum mean discrepancy (MMD) technique [24] along with the classification loss as the optimization objective for the model so that it forces the alignment of probability distributions between domains during model training. One notable advantage of our proposed model is that MMD computation can leverage the samples' unconditional distribution by utilizing the vast number of unlabeled samples in the target domain, which is common for collected datasets in internal networks.
- Additionally, we apply soft classification to the unlabeled data, using the classification sub-network to compute MMD over classes, thereby aiming to align conditional distributions between domains more effectively [20].
- To validate the effectiveness and generalizability of our approach, we conduct multiple transfer learning tasks between diverse datasets, including the widely used NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets [25].

Through rigorous experimentation, we demonstrate substantial improvement in cross-domain attack detection accuracy in various learning scenarios, validating the efficacy of our proposed method. As the proposed method eliminated the requirement for massive labeled data in the target network by transferring knowledge from heterogeneous source networks, it lays a good foundation for the application of deep transfer learning in internal network attack detection.

The remainder of this paper is organized as follows: Section 2 provides an overview of related works in the fields of network attack detection, transfer learning, and deep learning techniques. Section 3 details the methodology and architecture of our proposed model. Section 4 presents the experimental setup and evaluation results and discusses the findings and analyzes the performance of the model. Finally, Section 5 concludes the paper with a summary of contributions and highlights potential future research directions.

2. Related Work

In this section, we provide an overview of the related works in the fields of network attack detection with machine learning, deep learning, and transfer learning techniques. We briefly review existing studies that have attempted to address the challenges of network attack detection in internal networks and those that have explored transfer learning to improve model performance in the presence of limited labeled data.

2.1. Machine Learning for Network Attack Detection

In response to the limitations inherent in signature-based approaches, the research community has increasingly embraced the application of machine learning techniques to

analyze system logs and traffic data. The overarching goal of this approach is to construct a robust prediction model, which can subsequently be employed to effectively differentiate and classify instances of attacks from normal network behaviors [26].

Within the realm of machine learning-based network attack detection, particular attention has been accorded to supervised learning algorithms, owing to their demonstrated capacity for achieving high accuracy. The supervised methods rely on labeled data, employing them for rigorous training to fine-tune their predictive capabilities, ultimately facilitating the accurate detection of network attacks [27]. In Ref. [28], the authors eliminated highly correlated features and evaluated three algorithms, i.e., SVM, artificial neural network (ANN), and AdaBoost with decision tree, on the preprocessed dataset. In particular, the AdaBoost model uses decision trees as the weak learner and updates weights using the AdaBoost algorithm. Comparative analysis shows the AdaBoost model outperforms previous methods such as ANN and SVM.

As deep learning techniques gradually became popular in recent years, they were also applied to network intrusion detection. In Ref. [29], the authors propose to reconstruct the traffic data logs as two-dimensional image features and then apply CNN and CNN-LSTM separately on image data to perform network intrusion detection. The results are better than previous IDS methods, which verify the efficacy of the adoption of CNN.

However, it is essential to acknowledge a fundamental challenge that looms over the adoption of these machine learning-based methods. The performance of these algorithms is intrinsically linked to the availability of expansive and meticulously labeled datasets, a resource that tends to be in short supply within the complex and dynamic landscape of real-world internal network environments. This scarcity of large-scale, accurately labeled datasets underscores a significant hurdle that researchers and practitioners must grapple with as they strive to deploy effective machine learning solutions for network attack detection in practical settings. In Ref. [30], the author proposed an unsupervised deep learning approach for insider threat detection from system logs. They trained deep neural network (DNN) and recurrent neural network (RNN) models to learn normal user behavior and detect anomalies. The models are trained in an online fashion on streaming log data so that they can adapt to changing user patterns. The model output anomaly scores in the 96th percentile, and anomalies can be explained by decomposing the score into contributions from individual features, which reduces analyst workload significantly.

2.2. Transfer Learning

Supervised techniques necessitate a substantial amount of labeled data, while they demand significant labor and time when gathering data within an organization's internal network. Furthermore, because cyberattacks exhibit diverse patterns, the network behavior distribution fluctuates, rendering pre-built models ineffective, and thus it requires repeatedly retraining models with fresh labeled data. To overcome the scarcity of labeled data in the target domain, transfer learning has emerged as an effective approach. Transfer learning aims to transfer knowledge from a source domain with abundant labeled data to a target domain with limited labeled data [31,32].

Transfer learning approaches are categorized into three classes based on the nature of knowledge transfer: instance-based, model-based, and feature-based. In the realm of instance-based methods, the objective is to harness the potential of data samples from a source domain to enhance the learning process in a target domain. A notable illustration of this approach is the TrAdaBoost framework, which employs a small volume of fresh data to selectively filter out outdated data distributions [33]. This is achieved through iterative updates of sample weights, guided by the predictive errors of a basic learner. Model-based methods, on the other hand, focus on extracting deep learning model parameters that can be effectively shared across different domains. In the realm of feature-based methods, the goal is to identify a common latent feature space where the mapped samples from each domain exhibit closely aligned probability distributions. Bukhari et al. explore this approach by selecting covariate invariant features between training and testing datasets

and subsequently employing linear discriminant analysis (LDA) for dimensionality reduction [34]. Meanwhile, Pan et al. propose embedding data samples into a reproducing kernel Hilbert space (RKHS) while minimizing the maximum mean discrepancy (MMD) between domains [35]. This process leads to the derivation of a kernel matrix, which, in turn, yields low-dimensional representations of data samples using kernel-PCA. Moreover, the authors also proposed the transfer component analysis (TCA) method that takes a unified approach to kernel learning, aiming to attain a low-rank representation by minimizing distribution disparities while preserving data variance [36]. This technique offers a comprehensive means of knowledge transfer by combining distribution distance minimization with data variance preservation.

2.3. Deep Learning for Transfer Learning

Within the specific realm of transfer learning, the spotlight has intensely shone upon deep learning techniques as a means to harness knowledge acquired in one domain and effectively apply it to another. This pursuit has driven extensive research endeavors, seeking to unlock the potential of deep learning in facilitating knowledge transfer between related domains. The motivation lies in the realization that the complex and hierarchical representations learned by deep neural networks can be instrumental in unraveling the intricacies of diverse domains, thus expanding the horizons of what is attainable in the field of transfer learning.

One popular approach is to use deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), as feature extractors. These networks learn hierarchical representations of data, which can be fine-tuned for specific tasks in the target domain [37]. One remarkable example of model-based transfer learning is demonstrated by Long et al., who introduce adaptive layers into deep neural networks (DNNs) constructed from source domain data [21]. Subsequently, these adaptive layers are trained using target domain data. The key insight here lies in sharing the structure of the DNN and its weight parameters across domains, thus extracting common knowledge to enhance learning efficiency in the target domain.

Some studies on transfer learning have focused on domain adaptation methods, which attempt to align the source and target domains by reducing domain shifts. Pre-trained models, such as deep neural networks trained on large-scale datasets such as ImageNet, have been fine-tuned for specific tasks in the target domain, enabling efficient knowledge transfer. Domain adversarial neural networks (DANN) [38] and discrepancy-based domain adaptation [39] are examples of approaches that learn domain-invariant features and align distributions across domains. Other works have explored the use of pre-trained models for transfer learning [37]. While these studies have demonstrated the potential of deep learning in transfer learning, few have explored its application in the context of heterogeneous transfer learning for network attack detection in internal networks.

2.4. Transfer Learning in Network Attack Detection

Despite notable advancements in transfer learning techniques for network attack detection, there remains a pressing need to confront the intricacies associated with feature space disparities and distribution heterogeneity, particularly within the confines of internal networks. Researchers have embarked on a journey to explore an array of transfer learning methodologies, all geared toward bolstering the effectiveness of network attack detection in various scenarios.

One notable endeavor in this arena was undertaken by Zhao et al., who introduced the concept of HeMap (heterogeneous mapping [40]) into the realm of network attack detection [15]. Expanding upon this idea, they devised a strategy involving the pre-clustering of data samples across different domains. The primary aim was to mitigate the influence of mismatched samples. However, a critical limitation emerged in the form of cross-domain distance computation, which relied upon a heterogeneous feature space transformation facilitated by principal component analysis (PCA). Regrettably, this approach failed to accu-

rately reflect the genuine distance between cross-domain samples, ultimately culminating in suboptimal results.

In a parallel vein, previous work conducted by us introduced an alternative approach. In this method, the linear projection was employed for its computational simplicity to transform heterogeneous data into a shared latent space [41]. This transformation was followed by the application of maximum mean discrepancy (MMD) to align the distribution patterns across domains. While this approach offered certain advantages, such as computational efficiency, its overall performance remained constrained due to its reliance on linear projection, which could not fully capture the nuances inherent in the complex relationships within the data. More advanced techniques for finding an effective shared feature space could help overcome heterogeneity issues. In addition, distribution alignment methods tailored for network traffic data characteristics could better match distributions. Exploring nonlinear projections and more domain-specific alignment metrics are promising directions for improving transfer learning in this application.

3. System Design and Methods

In light of the existing research gaps, our proposed model addresses the feature space and distribution heterogeneity by utilizing deep learning-enabled heterogeneous transfer learning.

To illustrate the deep transfer learning model, we first define the notations for datasets. A transfer learning domain is defined as a dataset with its probability distribution. Thus, the source domain is defined as

$$\mathcal{D}_s = \{\mathcal{X}_s, P_s(\mathcal{X}_s)\}, \quad (1)$$

where the dataset \mathcal{X}_s is a d_s -dimensional feature with C classes and P_s is the associated probability distribution of the dataset. The dataset consists of n_s data samples, which is denoted as

$$\mathcal{X}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}, x_i^s \in \mathbb{R}^{d_s}, y_i^s \in \{0, 1, \dots, C-1\}. \quad (2)$$

Particularly, we consider binary a classification task, wherein the class label is set to $\{0, 1\}$.

Similarly, the target domain is defined as

$$\mathcal{D}_t = \{\mathcal{X}_t, P_t(\mathcal{X}_t)\}, \quad (3)$$

where the dataset \mathcal{X}_t is a d_t -dimensional feature with C classes and P_t is the associated probability distribution of the dataset. The dimension of feature space might be different from the source domain, i.e., $d_t \neq d_s$, which hinders the direct alignment of probability distribution of two domains. Even if the dimension sizes might be the same, there is still a large chance that the original features have different meaning in two domains. In order to reflect the scarcity of labeled data in the target domain, the data samples in the target domain dataset are further divided into a labeled subset \mathcal{X}_L with n_l samples and an unlabeled subset \mathcal{X}_U with n_u samples. Note that $n_l \ll n_u$, meaning the majority of the target domain data are unlabeled. Thus, the target data samples are denoted as

$$\begin{aligned} \mathcal{X} &= \mathcal{X}_l \cup \mathcal{X}_u \\ &= \{(x_i^t, y_i^t)\}_{i=1}^{n_l} \cup \{(x_i^t, y_i^t)\}_{i=n_l+1}^{n_l+n_u}, x_i^t \in \mathbb{R}^{d_t}, y_i^t \in \{0, 1, \dots, C-1\}. \end{aligned} \quad (4)$$

The proposed method only tries to address the heterogeneous transfer learning task with heterogeneous feature spaces, but the learning task (classes for the data) should be the same cross-domain. That is to say, the two datasets should share the same labels (as we denoted above, both datasets have labels of C classes). The heterogeneous transfer learning task with both heterogeneous feature space and learning task is out of the scope of this article and should be investigated in future research.

To effectively tackle the inherent challenge of lacking labeled training data, our approach makes the best of labeled datasets from the external internet. At the same time, we incorporate a small, yet invaluable, portion of labeled data along with the bulk of unlabeled data available from the internal network. The incorporation of the two heterogeneous

sources of datasets is realized by devising a deep neural network model and formulating the problem of network attack detection as a binary classification task. Particularly, by employing feature projection networks for each data source, the heterogeneous feature spaces are transformed into a common latent space. Thus, the transfer of attack detection knowledge from external networks to internal networks is possible via minimization of maximum mean discrepancy (MMD).

3.1. Network Architecture Design

To address the problems of feature space heterogeneity and probability distribution misalignment, we introduce the deep learning-enabled heterogeneous transfer learning framework, depicted in Figure 1.

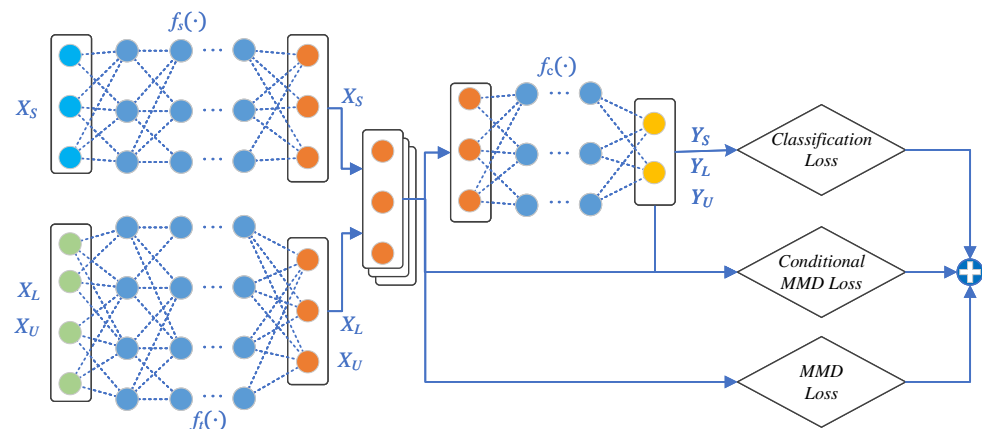


Figure 1. Deep network architecture for heterogeneous transfer learning

Due to the inherent complexity of real-world data, linear projections or shallow networks may be inadequate for capturing the intricate relationships within the data. Thus, we adopt a distinct approach by employing separate deep networks $f_s(\cdot)$, $f_t(\cdot)$ to facilitate the conversion of data from both the source and target domains into a shared feature space, leveraging the enhanced capacity for nonlinear feature transformation offered by deep neural networks.

Then, the transformed labeled data stemming from both the source and target domains serve as input for the classification network $f_c(\cdot)$. This ensures that the resulting model is robust and versatile, capable of effectively discerning patterns and making predictions on data originating from the target domain, a crucial requirement for successful transfer learning. Simultaneously, on the other side of this paradigm, the transformed data play a pivotal role in the calculation of the maximum mean discrepancy (MMD). This statistical metric serves as a vital tool for aligning the probability distributions across the two domains. By reducing distribution discrepancies, MMD facilitates the integration of unlabeled data from the target domain into the classification task, further enhancing the model's capability to make informed predictions based on this previously untapped data source. This dual-pronged approach capitalizes on the strengths of deep neural networks and statistical alignment techniques to optimize the utility of data from both domains in the context of the classification task.

3.1.1. Feature Projection Networks

In order to effectively manage the inherent heterogeneity present within the feature spaces of both the source and target domains, we devised a comprehensive strategy involving the creation of two distinct feature projection networks, $f_s(\cdot)$ and $f_t(\cdot)$, one meticulously tailored to each domain's unique characteristics. Each network's input is drawn from the corresponding dataset, i.e.,

$$\mathbf{X}_s = [x_1^s, x_1^s, \dots, x_{n_s}^s], \tag{5}$$

$$\begin{aligned} \mathbf{X}_t &= [\mathbf{X}_l; \mathbf{X}_u] \\ &= [x_1^t, x_2^t, \dots, x_{n_l}^t, x_{n_l+1}^t, \dots, x_{n_l+n_u}^t], \end{aligned} \tag{6}$$

and the projected output is given by

$$\tilde{\mathbf{X}}_s = f_s(\mathbf{X}_s), \tag{7}$$

$$\tilde{\mathbf{X}}_t = [\tilde{\mathbf{X}}_l; \tilde{\mathbf{X}}_u] = f_t(\mathbf{X}_t) = f_t([\tilde{\mathbf{X}}_l; \tilde{\mathbf{X}}_u]). \tag{8}$$

Then, the labeled projected data $[\tilde{\mathbf{X}}_s; \tilde{\mathbf{X}}_l]$ together with their corresponding labels $\tilde{\mathbf{Y}} = [\mathbf{Y}_s; \mathbf{Y}_l]$ are sent to the classification network, where

$$\mathbf{Y}_s = [y_1^s, y_2^s, \dots, y_{n_s}^s], \tag{9}$$

$$\mathbf{Y}_l = [y_1^l, y_2^l, \dots, y_{n_l}^l]. \tag{10}$$

At the same time, all the transformed data are sent to compute the maximum mean discrepancy (MMD) in order to align the probability distribution of the project data from two domains.

These dedicated networks assume the pivotal role of not only acquiring domain-specific representations but also orchestrating the transformation of input feature data into a unified and shared feature space. This concerted effort is strategically engineered to alleviate the potentially detrimental effects stemming from differences in feature spaces, thereby safeguarding the model’s overall performance. Each of these feature projection networks is thoughtfully structured, comprising a cascade of fully connected layers, each with its own set of learnable parameters. This hierarchical arrangement empowers the networks to progressively acquire increasingly abstract and discriminative representations of the input data, ensuring that the nuances and subtleties of the feature space peculiar to each domain are effectively captured.

This meticulous process yields the output of the feature projection networks, which manifests as the transformed feature data. These transformed data find their home in the shared and harmonized feature space, where they seamlessly coexist with their counterparts from the other domain. This pivotal transformation effectively bridges the feature space gaps between the source and target domains, laying the foundation for the smooth and effective transfer of knowledge and insights across domains.

3.1.2. Classification Network

The transformed features from the feature projection networks are fed into the classification network, $f_c(\cdot)$, which is responsible for predicting the corresponding labels. The classification network plays a crucial role in extracting meaningful patterns from the transformed features and making accurate predictions. It consists of several fully connected layers and a soft-max layer, with cross-entropy loss as the classification loss function. The classification loss is formulated as

$$L_c[\tilde{\mathbf{Y}}, f_c(\tilde{\mathbf{X}})] = \frac{1}{n_s + n_l} \mathcal{L}(\tilde{\mathbf{Y}}, f_c(\tilde{\mathbf{X}})), \tag{11}$$

where $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_s; \tilde{\mathbf{X}}_l]$ consists of the transformed labeled data samples from two domains, $\tilde{\mathbf{Y}} = [\mathbf{Y}_s; \mathbf{Y}_l]$ is the corresponding labels, $f_c(\cdot)$ is the classification network, and $\mathcal{L}(\cdot)$ is the cross-entropy loss function. By using the transformed features, rather than the original data, as input to the classification network, the model can benefit from the aligned feature representations and generalize better in the target domain with limited labeled data.

3.1.3. Distribution Alignment

To further mitigate the challenges posed by distribution heterogeneity between the source and target domains, we have incorporated a crucial mechanism: maximum mean

discrepancy (MMD). This strategic addition is designed to effectively align the probability distributions of data in both domains, thus enhancing the model’s performance in a transfer learning context. To delve into the specifics, our approach involves the computation of MMD, which serves as a non-parametric metric for quantifying the dissimilarity between probability distributions. This measure plays a pivotal role in quantifying the extent of divergence or alignment between the distributions of the transformed data originating from the source and target domains.

What sets our approach apart is the utilization of not only labeled samples but also the substantial pool of unlabeled data instances found within the target domain. This innovative strategy allows us to harness the wealth of unlabeled data for the purpose of computing cross-domain MMD, effectively leveraging a vast and previously untapped resource. During the training process, a key objective is to minimize the MMD value systematically. This optimization criterion serves as a guiding principle for the model, compelling it to actively align the probability distributions characterizing the two domains. Through this alignment process, the classification model is primed to generalize effectively, demonstrating robust performance when applied to unlabeled target data samples. This harmonization of distributions across domains serves as a critical bridge that allows the model to transfer knowledge seamlessly and adapt successfully to the intricacies of the target domain.

The MMD is calculated as the distance of two centroids corresponding to two datasets, which is expressed as:

$$Q_m [\tilde{X}_s, \tilde{X}_t] = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \tilde{x}_i^s - \frac{1}{n_t} \sum_{i=1}^{n_t} \tilde{x}_i^t \right\|^2, \tag{12}$$

where $\tilde{x}_i^s \in \tilde{X}_s$ and $\tilde{x}_i^t \in \tilde{X}_t$ are the transformed samples output by the corresponding projection networks.

To more effectively align distribution, conditional distributions between domains can also be aligned via minimizing centroid distance between corresponding classes across domains. Though we only know a few labeled data in the target domain, we can apply a pseudo classification to the unlabeled samples by reusing the classification network $f_c(\cdot)$. By using the pseudo classification for the unlabeled data, we compute the MMD over classes, i.e.,

$$Q_c [\tilde{X}_s, \tilde{X}_t] = \sum_{k=1}^C \left\| \frac{1}{n_s^k} \sum_{i=1}^{n_s^k} \tilde{x}_{k,i}^s - \frac{\sum_{i=1}^{n_t^k} \tilde{x}_{k,i}^l + \sum_{i=1}^{n_u^k} \tilde{x}_{k,i}^u}{n_t^k + n_u^k} \right\|^2, \tag{13}$$

where $\tilde{x}_{k,i}^s \in \tilde{X}_s$, $\tilde{x}_{k,i}^l \in \tilde{X}_l$ are the labeled projected data samples belonging to the k th class of the source and target domain, while $\tilde{x}_{k,i}^u \in \tilde{X}_u$ is the data sample of the k th pseudo class of the target domain. Correspondingly, n_s^k , n_t^k , and n_u^k are the number of samples of the k th class. This approach helps to capture underlying similarities and differences between the classes, contributing to the overall improvement in transfer learning performance.

Instead of assigning hard labels according to the pseudo classification result, soft labeling assigns probability distributions over classes to each unlabeled sample, which results in a more stable iteration process and avoids negative transfer. At the initial stage, the untrained classification network makes random guesses for unlabeled data, thus the minimization of conditional distributions distance takes little effect. As the iteration proceeds, the accuracy of the classification network will improve for unlabeled samples, thus boosting the minimization of conditional distributions. This is why choosing the soft labeling scheme is better than hard labeling. Furthermore, we can introduce a weight for the soft labels, and the weight increases with the iteration procedure, i.e., $w_r = \frac{r}{R}$, where R is the total number of iteration stage and r is the current stage number. Introducing the soft

labeling and iteration weight for conditional distribution distance, the MMD over classes can be rewritten as

$$Q_c[\tilde{\mathbf{X}}_s, \tilde{\mathbf{X}}_t] = \sum_{k=1}^C \left\| \frac{1}{n_s^k} \sum_{i=1}^{n_s^k} \tilde{\mathbf{x}}_{k,i}^s - \frac{\sum_{i=1}^{n_t^k} \tilde{\mathbf{x}}_{k,i}^l + \sum_{i=1}^{n_u^k} w_r \hat{y}_{k,i}^u \tilde{\mathbf{x}}_i^u}{n_t^k + \sum_{i=1}^{n_u^k} w_r \hat{y}_{k,i}^u} \right\|^2. \quad (14)$$

3.1.4. The Optimization Objective of the Transfer Learning Network

The overall loss function of our proposed model consists of the classification loss and the MMD-based distribution alignment loss (including MMD loss for both unconditional and condition distributions). The optimization objective is to jointly minimize the classification loss of labeled data and minimize the distribution distance in terms of MMD. Therefore, the optimization objective can be expressed as:

$$\min_{f_s, f_t, f_c} L_c[\mathbf{Y}, f_c(\tilde{\mathbf{X}})] + \alpha(Q_m[\tilde{\mathbf{X}}_s, \tilde{\mathbf{X}}_t] + Q_c[\tilde{\mathbf{X}}_s, \tilde{\mathbf{X}}_t]) \quad (15)$$

where α is a coefficient to adjust the relative importance of classification accuracy of labeled data and the distribution alignment across domains. During the training procedure with stochastic gradient descent-based methods, we iteratively update parameters of the feature projection networks and the classification network.

4. Performance Evaluation

In this section, we describe the experimental setup and evaluation process used to assess the effectiveness of our proposed deep learning-enabled heterogeneous transfer learning model for network attack detection. We conduct multiple transfer learning tasks between diverse datasets and present the results to validate the model's performance in various learning scenarios.

4.1. Datasets

We perform our experiments on three widely used and publicly available network intrusion detection datasets: NSL-KDD, UNSW-NB15, and CIC-IDS2017.

1. NSL-KDD (NSL-KDD Cup'99 Dataset) is a widely used dataset in the field of network intrusion detection and security. It is an improved version of the original KDD Cup '99 datasets, designed to address some shortcomings in the latter, such as redundancy and unrealistic traffic patterns. NSL-KDD contains a large collection of network traffic data, including both normal and various types of malicious activities (i.e., DoS, Probe, R2L, and U2R attacks), making it a valuable resource for training and evaluating intrusion detection systems.
2. UNSW-NB15 (University of New South Wales Network-Based 15) consists of network traffic data captured in a controlled environment, simulating a real network. This dataset includes a diverse range of attack types (contains nine types of attacks, including Brute Force, DoS, and Web Attacks, etc.) and normal traffic, providing a realistic representation of network security challenges for developing and testing intrusion detection systems.
3. CIC-IDS2017 (Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset 2017) is a comprehensive dataset that includes various attack scenarios, such as DoS, DDoS, and Port Scans. It offers a wide variety of network traffic scenarios, including both benign and malicious traffic, across different network protocols. This dataset is particularly valuable for researchers and practitioners working on cybersecurity, as it helps in the development and assessment of effective intrusion detection and prevention mechanisms.

We summarize the main attack types in each dataset in Table 1 so that we can choose similar attack types from two datasets to simulate the cross-domain transfer learning task.

Table 1. Summary of the attack types in each dataset.

Dataset	Attack Types	Description
NSL-KDD	DoS	Involves overwhelming a network or system to disrupt its services.
	Probe	Attackers attempt to gather information about the target network without direct exploitation.
	U2R	Attackers exploit vulnerabilities to gain unauthorized access and escalate privileges.
	R2L	Attackers attempt to connect to a local system remotely without proper credentials.
UNSW-NB15	Fuzzers	aimed at testing software vulnerabilities through unexpected inputs.
	Analysis	Techniques for gathering information about target systems.
	Backdoors	Unauthorized access methods left by attackers.
	DoS	Flooding a system to disrupt its services.
	Exploits	Attacks exploiting known vulnerabilities.
	Generic	General or unspecified attacks.
	Reconnaissance	Preparing for future attacks by gathering information.
CIC-IDS2017	Shellcode	Malicious code for executing arbitrary commands.
	DoS	Flooding the target with traffic to disrupt services.
	PortScan	Scanning target ports to find potential vulnerabilities.
	DDoS	Distributed denial of service attacks from multiple sources.
	Patator	Brute-force attacks against SSH and FTP services.
	Web Attack	Attacks targeting web applications and services.
	Botnet	Activities related to botnets, including command and control traffic.
	Infiltration	Unauthorized access and data exfiltration attempts.

4.2. Transfer Learning Tasks

We implement our proposed model using the PyTorch deep learning framework. We perform two groups of transfer learning tasks to demonstrate the efficacy of our model in different learning task settings and compare with related transfer learning methods.

4.2.1. UNSW-NB15 to CIC-IDS2017 Transfer Learning

In order to demonstrate the efficacy of the proposed transfer learning framework, we apply it to multiple scenarios with two datasets. Specifically, we use the UNSW-NB15 dataset as the source domain for training and transfer the knowledge to the CIC-IDS2017 dataset as the target domain with different attack scenarios. Through a grid search procedure for tuning hyper-parameters, we decide to set the number of hidden layers of the feature transform network for source and target to 2 and 3, respectively, set the dimension of common feature space to 128, and set the number of hidden layers of the classification network to 4, and ReLU is used as the activation function for all hidden layers. The training process includes four epochs.

According to the attack characteristics summarized in Table 1, we divide data samples of similar attack types for source and target datasets into several groups to simulate multi-scenario cross-domain transfer learning. In each group of source and group dataset, data labeled with normal/benign are negative samples, while others are positive samples. The transfer learning settings are summarized in Table 2.

Table 2. Heterogeneous transfer learning data configuration for UNSW-NB15 to CIC-IDS2017.

#	Source	Target	Description
1	Normal, Fuzzers	BENIGN, Web Attack, Brute Force, FTP-Patator, SSH-Patator	accessing target system via brute-force manner
2	Normal, DoS	BENIGN, DoS {Hulk, GoldenEye, slowloris, Slowhttptest}	denial of service attack
3	Normal, Recon, Analysis	BENIGN, PortScan	retrieve information about target system
4	Normal, Generic	BENIGN, Bot, Web Attack XSS, Web Attack SQL Injection, Infiltration	other attack types cannot be categorized

We employ several standard evaluation metrics for binary classification tasks, including accuracy, accuracy, recall, and F1-score. We present the results of our experiments in Table 3, including the performance metrics for each transfer learning task (identified by the TaskID (#)).

Table 3. Performance metrics for transfer learning from UNSW-NB15 to CIC-IDS2017.

TaskID	Accuracy	Precision	Recall	F1-Score
1	0.987	0.465	0.938	0.622
2	0.983	0.457	0.969	0.621
3	0.985	0.431	0.984	0.599
4	0.998	0.981	0.994	0.988

In various transfer learning scenarios, the prediction accuracy is consistently high, i.e., achieving 98%, and the recall achieves as high as 93%. On the other hand, the precision and F1-score only achieve medium rate (above 43% and 59%, respectively). As the performance metrics are defined as

$$\text{Recall} = \frac{TP}{TP + FN'} \quad (16)$$

$$\text{Precision} = \frac{TP}{TP + FP'} \quad (17)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}'} \quad (18)$$

we can conclude that the method works well for attack detection, i.e., detecting the true positive data samples, while there is a relatively high possibility to mis-detect normal data samples as attack instances (the false positive rate). The reason why the false positive rate is high might be that the dataset is extremely unbalanced, that is to say, the communication sessions in those datasets comprise only a minor portion of attack instances. To resolve this issue, we may try to up-sample the attack data instances to reduce the imbalance. In addition, we can try to increase the weight of attack instances during the training stage.

During tuning of the hyper-parameters, we compared the results of using several variants of ReLU activation functions, i.e., Leaky ReLU, Parametric ReLU, and ELU, to determine if the model is influenced by “dead neuron” due to negative input. The results are show in Figure 2.

Though Leaky ReLU, Parametric ReLU, and ELU introduce some mechanism to eliminate zero gradients for negative input, from the results, we can see that ReLU does not degrade the model’s performance. Hence, in this experiment, it is safe to use ReLU as an activation function. However, for a wider-range application of the learning model, using those modified ReLU variants usually achieves better stability in terms of the model’s performance.

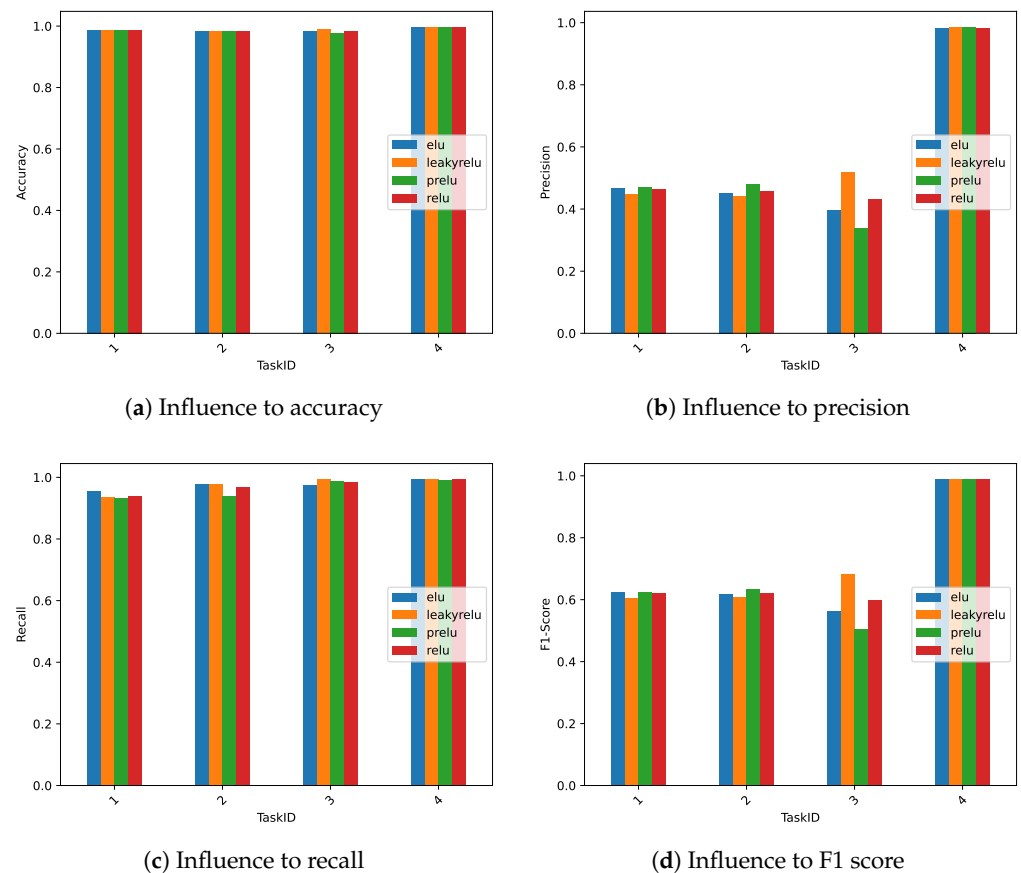


Figure 2. Influence of ReLU activation functions to model training.

4.2.2. NSL-KDD to UNSW-NB15 Transfer Learning

In this task, we train the model on the NSL-KDD dataset (source domain) with abundant labeled samples and transfer the knowledge to the UNSW-NB15 dataset (target domain) with limited labeled data. Different from the previous experiments, as the datasets change, due to a relatively small dataset of NSL-KDD, we only configure one hidden layer for the source and target feature transform networks, and the classification network has three hidden layers. The dimension of common feature space, i.e., the number of units of output layer in the feature transform networks, is configured to 256. The training process includes three epochs, and each epoch contains multiple iterations, which depend on the batch size (set to 1024) and how much data we have in the dataset.

We compare the proposed method with several methods that are mentioned in Ref. [41] to validate the superiority of our proposed method (denoted as *dhetl*), including:

- The *hemap* method, which employs linear projection to transform the diverse source and target feature space into a shared latent space, concurrently minimizing projection errors and sample distances across different domains.
- The *hetl* [15], which is similar to *hemap* but includes clustering target data before each iteration.
- The *base* approach, which entails the direct training of the source domain while subsequently applying predictions to target domain data. This is accomplished by orchestrating the transformation of both source and target data into a shared feature space through principal component analysis (PCA).
- The *hemmd* method [41], which is similar to *hemap* but minimizes cross-domain distribution distance with measurement of MMD.

In total, seven transfer learning tasks are constructed. In each task, data samples belong to normal and an attack class are selected to represent source and target domain

data from the NSL-KDD dataset and UNSW-NB15 datasets. To compare with existing methods, we employ several standard evaluation metrics for binary classification tasks, including accuracy. The results are shown in Table 4.

Table 4. The accuracy of cross-domain network attack detection.

Source ↓ Target	DoS ↓ DoS	DoS ↓ Fuzzers	DoS ↓ Generic	Probing ↓ Analysis	Probing ↓ Fuzzers	Probing ↓ Reconn	R2L ↓ Exploits
hemap	0.773	0.757	0.784	0.744	0.734	0.720	0.800
het1	0.701	0.693	0.693	0.699	0.696	0.700	0.695
base	0.846	0.532	0.914	0.725	0.585	0.654	0.617
hemmd	0.945	0.956	0.587	0.898	0.814	0.814	0.878
dhet1	0.990	0.989	0.995	0.995	0.989	0.989	0.981

From the results, we can see that the proposed *dhet1* has the highest prediction accuracy in the given transfer learning scenarios. Except for *dhet1*, *hemmd* has the highest accuracy compared with other methods, which has been analyzed in [41]. Compared with *hemmd*, the main improvement of *dhet1* is attributed to the nonlinear projection of feature spaces to common space, which is more expressive than the linear projection in *hemmd*. In addition, *dhet1* trains the network by optimizing the classification loss, which is directly related to the learning task, while *hemap* is not since it optimizes projection loss. Furthermore, the compared methods only utilize partial data from the dataset to optimize their model, while the proposed method uses all available data to train a model. Therefore, we have verified that the proposed method outperformed other methods.

5. Conclusions

In this paper, we have proposed a deep learning-enabled heterogeneous transfer learning model for network attack detection in internal networks. Through feature transformation and the training procedure to minimize classification loss and align probability distribution, we finally obtain a model that achieves the highest detection accuracy among compared methods.

We also find, though it can achieve high detection accuracy for attack instances, the mis-detection rate for normal instances is still at a moderate level. The reason might be that both the source and target datasets are highly imbalanced. Hence, in future work, we need to work hard for a solution to address this imbalance to further enhance the performance of the proposed deep learning-enabled heterogeneous transfer learning model.

Author Contributions: Conceptualization, G.W. and C.Z.; methodology, G.W.; validation, D.L.; writing—original draft preparation, G.W.; writing—review and editing, T.H.; funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (grant number 2021YFB3302105).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
CIC-IDS2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017 dataset
CNN	Convolutional Neural Network
DANN	Domain Adversarial Neural Network
DDoS	Distributed Denial of Service
DoS	Denial of Service
GNN	Graph Neural Network
IDS	Intrusion Detection Systems
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
MMD	Maximum Mean Discrepancy
NSL-KDD	NSL-KDD Cup'99 Dataset
PCA	Principal Component Analysis
R2L	Root to Local attacks
ReLU	Rectified Linear Activation Function
RKHS	Reproducing Kernel Hilbert Space
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TCA	Transfer Component Analysis
UNSW-NB15	University of New South Wales Network-Based 15 dataset
U2R	User to Root attack

References

1. Cisco. *Cisco Annual Internet Report (2018–2023) White Paper*; Techreport; Cisco Systems: San Jose, CA, USA, 2020.
2. Homoliak, I.; Toffalini, F.; Guarnizo, J.; Elovici, Y.; Ochoa, M. Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–40. [[CrossRef](#)]
3. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [[CrossRef](#)]
4. Liu, L.; De Vel, O.; Han, Q.L.; Zhang, J.; Xiang, Y. Detecting and preventing cyber insider threats: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1397–1417. [[CrossRef](#)]
5. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [[CrossRef](#)]
6. Erlacher, F.; Dressler, F. FIXIDS: A high-speed signature-based flow intrusion detection system. In Proceedings of the NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–8.
7. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics* **2020**, *9*, 1177. [[CrossRef](#)]
8. Kim, H.A.; Karp, B. Autograph: Toward Automated, Distributed Worm Signature Detection. In Proceedings of the USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004; Volume 286.
9. Sommer, R.; Paxson, V. Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316.
10. Kim, G.; Lee, S.; Kim, S. A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [[CrossRef](#)]
11. Chalapathy, R.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407.
12. Zhou, C.; Paffenroth, R.C. Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, USA, 13–17 August 2017; pp. 665–674.
13. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.K.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 104650–104675. [[CrossRef](#)]
14. Kheddar, H.; Himeur, Y.; Awad, A.I. Deep Transfer Learning Applications in Intrusion Detection Systems: A Comprehensive Review. *arXiv* **2023**, arXiv: 2304.10550.
15. Zhao, J.; Shetty, S.; Pan, J.W.; Kamhoua, C.; Kwiat, K. Transfer learning for detecting unknown network attacks. *EURASIP J. Inf. Secur.* **2019**, *2019*, 1. [[CrossRef](#)]
16. Xu, Y.; Liu, Z.; Li, Y.; Zheng, Y.; Hou, H.; Gao, M.; Song, Y.; Xin, Y. Intrusion detection based on fusing deep neural networks and transfer learning. In Proceedings of the Digital TV and Wireless Multimedia Communication: 16th International Forum, IFTC 2019, Shanghai, China, 19–20 September 2019; pp. 212–223.

17. Masum, M.; Shahriar, H. TL-nid: Deep neural network with transfer learning for network intrusion detection. In Proceedings of the 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 8–10 December 2020; pp. 1–7.
18. Mahdavi, E.; Fanian, A.; Mirzaei, A.; Taghiyarrenani, Z. ITL-IDS: Incremental transfer learning for intrusion detection systems. *Knowl.-Based Syst.* **2022**, *253*, 109542. [[CrossRef](#)]
19. Pawlicki, M.; Kozik, R.; Choraś, M. Towards Deployment Shift Inhibition Through Transfer Learning in Network Intrusion Detection. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–6.
20. Yao, Y.; Zhang, Y.; Li, X.; Ye, Y. Heterogeneous domain adaptation via soft transfer network. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1578–1586.
21. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2208–2217.
22. Yuan, S.; Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Comput. Secur.* **2021**, *104*, 102221. [[CrossRef](#)]
23. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 97–105.
24. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
25. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Comput. Secur.* **2019**, *86*, 147–167. [[CrossRef](#)]
26. Li, J.; Qu, Y.; Chao, F.; Shum, H.P.; Ho, E.S.; Yang, L. Machine learning algorithms for network intrusion detection. *AI Cybersecur.* **2019**, *151*, 151–179.
27. Belavagi, M.C.; Muniyal, B. Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Comput. Sci.* **2016**, *89*, 117–123. [[CrossRef](#)]
28. Ahmad, I.; Ul Haq, Q.E.; Imran, M.; Alassafi, M.O.; AlGhamdi, R.A. An efficient network intrusion detection and classification system. *Mathematics* **2022**, *10*, 530. [[CrossRef](#)]
29. Zainel, H.; Koçak, C. LAN intrusion detection using convolutional neural networks. *Appl. Sci.* **2022**, *12*, 6645. [[CrossRef](#)]
30. Tuor, A.; Kaplan, S.; Hutchinson, B.; Nichols, N.; Robinson, S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *arXiv* **2017**, arXiv:1710.00811.
31. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [[CrossRef](#)]
32. Day, O.; Khoshgoftaar, T.M. A survey on heterogeneous transfer learning. *J. Big Data* **2017**, *4*, 1–42. [[CrossRef](#)]
33. Dai, W.; Yang, Q.; Xue, G.R.; Yu, Y. Boosting for Transfer Learning. In Proceedings of the 24th International Conference on Machine Learning, Corvallis OR, USA, 20–24 June 2007; pp. 193–200.
34. Bukhari, M.; Bajwa, K.B.; Gillani, S.; Maqsood, M.; Durrani, M.Y.; Mehmood, I.; Ugail, H.; Rho, S. An efficient gait recognition method for known and unknown covariate conditions. *IEEE Access* **2020**, *9*, 6465–6477. [[CrossRef](#)]
35. Pan, S.J.; Kwok, J.T.; Yang, Q. Transfer learning via dimensionality reduction. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; Volume 8; pp. 677–682.
36. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2010**, *22*, 199–210. [[CrossRef](#)] [[PubMed](#)]
37. Marcelino, P. Transfer learning from pre-trained models. *Towards Data Sci.* **2018**, *10*, 23.
38. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
39. Saito, K.; Watanabe, K.; Ushiku, Y.; Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3723–3732.
40. Shi, X.; Liu, Q.; Fan, W.; Yu, P.S. Transfer across completely different feature spaces via spectral embedding. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 906–918. [[CrossRef](#)]
41. Zhang, C.; Wang, G.; Wang, S.; Zhan, D.; Yin, M. Cross-domain network attack detection enabled by heterogeneous transfer learning. *Comput. Netw.* **2023**, *227*, 109692. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.