Abstract—In this article, we propose a joint semantic transfer network (JSTN) toward effective intrusion detection (ID) for large-scale scarcely labeled Internet of Things (IoT) domain. As a multisource heterogeneous domain adaptation (MS-HDA) method, the JSTN integrates a knowledge-rich network intrusion (NI) domain and another small-scale IoT intrusion (II) domain as source domains and preserves intrinsic semantic properties to assist target II domain ID. The JSTN jointly transfers the following three semantics to learn a domain-invariant and discriminative feature representation. The scenario semantic endows source NI and II domains with characteristics from each other to ease the knowledge transfer process via a confused domain discriminator and categorical distribution knowledge preservation. It also reduces the source–target discrepancy to make the shared feature space domain invariant. Meanwhile, the weighted implicit semantic transfer boosts discriminability via a fine-grained knowledge preservation, which transfers the source categorical distribution to the target domain. The source–target divergence guides the importance weighting during knowledge preservation to reflect the degree of knowledge learning. Additionally, the hierarchical explicit semantic alignment performs centroid-level and representative-level alignment with the help of a geometric similarity-aware pseudo-label refiner, which exploits the value of the unlabeled target II domain and explicitly aligns feature representations from a global and local perspective in a concentrated manner. Comprehensive experiments on various tasks verify the superiority of the JSTN against state-of-the-art comparing methods, on average a 10.3% of accuracy boost is achieved. The statistical soundness of each constituting component and the computational efficiency is also verified.

Index Terms—Domain adaptation (DA), heterogeneity, Internet of Things (IoT), intrusion detection (ID), semantic transfer.

I. INTRODUCTION

As the Internet of Things (IoT) devices become more ubiquitous in our daily life [1], [2], [3], [4], they have transformed various fields, such as healthcare [5], [6], and public transport [7] into a smart space. However, IoT infrastructures are usually formed by resource-limited devices with infrequent security maintenance effort from their vendors [8], which poses security threats for malicious attacks to take advantage of IoT security flaws and perform intrusions which harm the underlying IoT infrastructures [9], [10] and the applications they support. Therefore, a robust intrusion detection system (IDS) [11] is crucial to effectively detect these malicious intrusions faced by IoT infrastructures.

With the rapid development of machine learning (ML) and deep learning (DL) techniques, recently, several DL-based IDSs become popular. For instance, Anthi et al. [12] analyzed the IDS performance of several supervised methods under the smart home IoT scenario, such as naive Bayes classifier, support vector machine, etc. The results verified the intrusion detection (ID) effectiveness of these methods. However, these methods highly depend on a vast amount of fully labeled data, which is expensive to collect and laborious to annotate. This is particularly difficult for IoT intrusion (II) detection, since data generated by IoT devices usually involves user privacy issues [13], [14], which hinder the publication of II detection data. Besides, these ML models are less capable of handling newly emerged intrusion types due to the shortage of annotated data. Considering that ID data for IoT is expensive to collect and seldom available, several domain adaptation (DA) approaches were proposed to transfer the rich knowledge from the network intrusion (NI) domain to facilitate the ID for label-scarce IoT domains. Since the NI data is relatively richer than II domains [15], these DA approaches treated the NI as the source domain, and the II as the target domain. As the network and IoT share several common attack types, by mapping both domains into a common feature subspace, these DA approaches can transfer the enriched NI knowledge to assist ID in the target IoT domain. For instance, Vu et al. [16] utilized two autoencoders as feature extractors for source and target domains and minimized the maximum mean discrepancy (MMD) between their bottleneck layers to achieve knowledge transfer (KT). However, previous DA-based ID models usually produced coarse-grained alignment. They aligned the source and target domains into a common feature subspace by brute force without transferring intrinsic semantic properties, which may result in instances from different categories being
confounded together and therefore hurt the discriminability of learned features.

To address the limitations of coarse-grained DA-based ID models and facilitate better transferability, in this article, we propose a joint semantic transfer network (JSTN) which leverages the intrinsic semantic knowledge between domains to facilitate a more fine-grained KT. Considering that there is a huge domain gap between the source NI and target II domains due to heterogeneities, such as different feature representations, different distributions, etc., the effectiveness of direct DA may be hindered. Hence, to ease the adaptation process, we utilize another labeled II domain as the source domain, which is smaller in scale than both the source NI and target II domains, to partially mask the heterogeneities. All domains have their own domain encoders to map instances into a common feature subspace, and a domain discriminator is confused to shorten the source NI–source II divergence and the source–target divergence. The predicted categorical distribution knowledge is also transferred between source domains for better discriminability. By introducing this auxiliary small-scale source II domain, it can equip the heterogeneous source NI domain with the semantics of IoT scenarios. By drawing the network and IoT intrusion scenarios closer and letting them become similar to partially mask scenario heterogeneities, the source NI and II domains form a holistic source domain with rich intrusion knowledge and IoT scenario characteristics, which can therefore benefit the source–target KT performed later.

Additionally, to overcome the category confounding caused by the coarse-grained feature alignment, we propose a weighted implicit semantic transfer, which preserves the correlation knowledge between categories from the source to the target domain. It is intuitive that the same class from either source or target domain should share a relatively similar categorical distribution. During the weighted implicit semantic transfer, the knowledge from the source NI and II domains is weighted based on their divergence with the target domain to dynamically emphasize varied source domain importance which reflects the degree of knowledge learning. The weighted implicit semantic alignment can effectively enhance the discriminability of the learned feature.

Given that the majority of target II domain instances are unlabeled, exploring unlabeled target data is beneficial during DA [1], [17], especially when there are huge heterogeneities present between domains. Therefore, we propose a hierarchical explicit semantic alignment from centroid-level and representative-level. The centroid-level alignment matches each category between the source domain, the target domain, and the combination of source and target domains from a global centroid perspective. Considering that only utilizing the global centroid-level alignment may hurt the concentration of aligned features, we also leverage the representative-level alignment, which performs a class-wise representative selection and minimizes the pairwise divergence between class-wise representatives from source and target domains. Hence, it boosts the concentration of aligned features yielded by the semantic alignment from a local perspective without causing a heavy computational burden. To fully excavate the potentials of unlabeled target II data during hierarchical explicit semantic transfer and avoid the misleading direct pseudo-label assignment [18], [19], a pseudo-label refiner (PLR) is leveraged to assign unlabeled target II instances with pseudo-labels via an ensemble approach. It will investigate the geometric similarity (GS) between each unlabeled target instance and the centroid of labeled instances of each category and regard the most geometrically similar category as the geometric label. Then, the PLR will only assign pseudo-label to an instance if the geometric label agrees with the prediction yielded by the shared classifier. Assisted by the more accurate PLR, the hierarchical explicit semantic alignment with a global and local perspective can explicitly minimize domain divergence in a concentrated manner and promote discriminability.

Ultimately, by jointly utilizing these semantics, the JSTN model can robustly transfer enriched knowledge from the knowledge-rich NI domain and a small-scale II domain to facilitate more accurate ID of the scarcely labeled target II domain and hence secure the IoT infrastructures.

In summary, the contributions of this article are as follows.

1) We utilize the joint semantic transfer to leverage the enriched knowledge of the NI domain with the help of an auxiliary small-scale II domain to facilitate more accurate ID of the large-scale scarcely labeled target II domain.

2) We propose a novel JSTN that explores and excavates the semantic transfer to achieve a more effective intrusion KT despite significant heterogeneities present between NI and II domains.

3) We conduct comprehensive experiments of several tasks on five well-known ID data sets and demonstrate the effectiveness of the JSTN algorithm, exceeding state-of-the-art comparing methods.

The remainder of this article is organized as follows. Section II summarizes related works on signature-based, ML-based, and DA-based ID approaches, explains their limitations, and reveals our research opportunities. Section III presents the model and the architecture of the JSTN method. The details of the proposed JSTN method are presented in Section IV. The experimental setup, results, and insight analyses are given in Section V. Section VI concludes this article.

II. RELATED WORK

A. Signature-Based Intrusion Detection

As a popular research direction, several signature-based ID methods have been proposed. They maintained a set of signatures or rules of malicious attacks and performed ID by matching incoming network traffic with these predefined attack patterns. Zhang et al. [20] proposed a preventive measure specifically targeting Distributed Denial-of-Service (DDoS) attacks on IoT devices. It kept track of the content of incoming requests. If requests from a node show a pattern, e.g., similar meaningless content being repeatedly sent, the preventive measure will flag the corresponding sender as malicious and refuse its future requests subsequently. Dietz et al. [21] proactively performed an automatic scan of neighboring IoT devices for potential vulnerabilities such as using default credential
settings. Once a vulnerable IoT device is detected, it will be temporarily isolated since these IoT devices suffer from a higher chance to be compromised and be manipulated as part of the malicious Botnet. Jun and Chi [22] utilized the complex event processing (CEP) technique, a technique to filter and process real-time events. The CEP required a predefined rule pattern repository which contains rules and patterns of common IoT security violations. Summerville et al. [23] presented a lightweight deep packet anomaly detection strategy via efficient bit-pattern matching. The patterns of the payload contents were studied, and the n-gram matching algorithm was leveraged to find pattern matching in an efficient manner.

Although these previous signature-based ID methods can produce satisfying results, they require substantial expert knowledge to build the pattern repository as the working foundation. The expert knowledge is usually laborious to acquire, barely thorough and complete, and is unable to tackle newly emerged attack types if the pattern repository is not updated on a frequent basis. Hence, it leaves rooms for other research directions.

B. Machine Learning-Based Intrusion Detection

ML and DL techniques can also be applied to tackle ID for IoT scenarios. Shukla [24] presented several new ID methods based on K-means clustering, decision tree, and an ensemble of these two classical ML algorithms. The proposed approach is lightweight and is capable of accurately detecting wormhole attacks targeting IoT under a 6LoWPAN network environment. Anthi et al. [12] focused on the ID of smart home IoT devices. Several popular classifiers, such as naive Bayes, support vector machines, etc., were evaluated to detect four mean network attack categories on a realistic testbed. Ge et al. [25], McDermott et al. [26], and Meidan et al. [27] all focused on leveraging DL-based methods. A feedforward neural network, bidirectional-LSTM recurrent neural network, and a deep autoencoder were constructed to perform ID for IoT devices, respectively, and demonstrated satisfying outcomes.

However, these ML and DL-based methods require a large-scale labeled data set, which is expensive and laborious to acquire. Some data sets become out-of-date quickly as IoT devices and attacks keep evolving, which hinders the effectiveness of these methods. Therefore, it naturally leads to the DA-based (DA) methods [28], [29], which performs KT to facilitate the ID of data-scarce IoT spaces.

C. Domain Adaptation and Its Application in Intrusion Detection

Heterogeneous DA: HDA transfers knowledge from a knowledge-rich domain to facilitate learning in a similar but knowledge-scarce target domain. The source and target domains present heterogeneities. For instance, intrusion data from the network and IoT domain can have different types of devices that work under different environments, using different feature sets, and follow different distributions, etc. Several research efforts have been presented to address the HDA problem with a specific focus on the feature-level. Wang and Mahadevan [30] utilized manifold alignment (DAMA) to construct mappings to project source and target data to a latent space while preserving the label topology. Hoffman et al. [31] presented max-margin domain transforms (MMDTs) to simultaneously learn the feature projection and the classifier. Chen et al. [32] proposed the transfer neural tree (TNT) algorithm with stochastic pruning to perform feature transformation and enhance prediction accuracy. Yao et al. [33] proposed the discriminative distribution alignment (DDA) that incorporated several losses, such as cross-entropy loss (DDAC) and squared loss (DDAS) to improve the data separability during alignment. However, these methods mainly focused on the feature-level information, none of them leveraged the intrinsic semantic correlations contained in the predicted distributions, which may result in confounded attack types and confused predictions if work on II detection.

Besides, some research efforts tackled the HDA problem by explicitly enforcing domain alignment. Tsai et al. [18] presented cross-domain landmark selection (CDLS) to learn a domain-invariant feature subspace for HDA via cross-domain landmarks. To jointly match the marginal and class-conditional distributions, Hsieh et al. [19] presented the generalized joint distribution adaptation (G-JDA) method and confirmed its effectiveness. To circumvent the negative effect brought by falsely assigned pseudo-labels, Yao et al. [34] proposed the soft transfer network (STN) which utilized soft labels during alignment. However, some of these methods directly used the predicted label as pseudo-label, which will cause severe negative transfer due to falsely assigned pseudo-labels. These wrong labels will mislead the model, the model will then accumulate more wrong labels, which forms a negative loop. Although methods, such as STN, attempted to avoid the negative transfer incurred by wrongly assigned pseudo-labels, they failed to consider the intrinsic geometric semantic contained in the feature space, which can effectively guide the pseudo-label assignment and boost the pseudo-label confidence.

Finally, considering that utilizing two source domains may enhance the KT process, Yao et al. [35] proposed a conditional weighting adversarial network (CWAN) to address the multisource HDA problem. However, it did not verify the effectiveness of multisource DA method on ID tasks, which left a void to be filled. Hence, it did not attempt the idea of scenario semantic to be used between network and IoT domains either and also lacked the joint consideration of semantic transfer.

DA-Based ID: The capability of the DA to transfer knowledge and facilitate robust learning in the target domain makes it a perfect choice for ID. Vu et al. [16] trained two autoencoders for a label rich and a label scarce IoT domain separately and bridged the gaps between the bottleneck layers of these two autoencoders by minimizing the MMD. Hu et al. [36] proposed a deep subdomain adaptation network with attention mechanism (DSAN-AT), which utilized the local MMD to boost the prediction accuracy and an attention mechanism to prevent overly long convergence time.

To circumvent the labor-intensive data set collection process, Ning et al. [37] proposed a KT ConvLaddernet to work under a semi-supervised setting, i.e., transfer knowledge from a small-scale source domain to facilitate ID of the target domain.
Although previous DA-based methods have been applied to perform ID, they failed to jointly consider the implicit categorical and explicit distance semantics during KT, which may hinder their effectiveness. Besides, these DA-based methods did not realize that utilizing an NI domain plus a small-scale II domain can boost the intrusion performance of a large-scale scarcely labeled target IoT domain. Hence, they left the potential of scenario semantic untouched.

III. MODEL AND JSTN ARCHITECTURE

In this section, we will mainly present the problem setting, followed by the architecture of the JSTN algorithm.

A. Model Preliminary

The JSTN works under a semi-supervised setting. More specifically, it involves a source NI domain that is defined as follows:

\[ D_{SN} = \{X_{SN}, Y_{SN}\} = \{(x_{SNi}, y_{SNi})\} \]

\[ x_{SNi} \in \mathbb{R}^{d_{SN}}, y_{SNi} \in [1, K], i \in [1, n_{SN}] \]  \hspace{1cm} (1)

where the source NI domain contains \( n_{SN} \) instances with their corresponding label, each instance is a \( d_{SN} \)-dimensional vector, and each label is within a total of \( K \) categories. Similarly, the small-scale source II domain is defined in a similar way as follows:

\[ D_{SI} = \{X_{SI}, Y_{SI}\} = \{(x_{SIi}, y_{SIi})\} \]

\[ x_{SIi} \in \mathbb{R}^{d_{SI}}, y_{SIi} \in [1, K], i \in [1, n_{SI}], n_{SI} < n_{SN} \]  \hspace{1cm} (2)

Note that the amount of instances in the source II domain is smaller than the source NI domain due to the data scarcity of IoT domains. Together, both the source NI domain \( D_{SN} \) and the source II domain \( D_{SI} \) form the source domain \( D_{S} = \{X_{S}, Y_{S}\} = D_{SN} \cup D_{SI}, n_{S} = n_{SN} + n_{SI} \). Under the semi-supervised setting, the target II domain is scarcely labeled and is defined as follows:

\[ D_{TL} = \{X_{TL}, Y_{TL}\} = \{(x_{TLi}, y_{TLi})\} \]

\[ D_{TU} = \{X_{TU}\} = \{(x_{TUj})\}, D_{T} = D_{TL} \cup D_{TU} \]

\[ x_{TLi}, x_{TUj} \in \mathbb{R}^{d_{T}}, y_{TLi} \in [1, K], i \in [1, n_{TL}], j \in [1, n_{TU}] \]

\[ n_{T} = n_{TL} + n_{TU}, n_{TL} \ll n_{TU} \]  \hspace{1cm} (3)

where only a small amount of target II data is labeled, i.e., \( n_{TL} \ll n_{TU} \). The source NI domain, source II domain, and the target II domain present heterogeneities as they come from distinct feature spaces, i.e., \( d_{SN} \neq d_{SI} \neq d_{T} \). All notations used in this article, and their corresponding interpretations, are presented in the Appendix to ease understanding.

B. JSTN Model Architecture

The architecture of the JSTN model is illustrated in Fig. 1. For each domain, a feature encoder \( E \) is utilized to map the original feature into a shared common feature subspace with dimension \( d_{C} \). The feature encoder \( E \) is defined as follows:

\[ f(x_i) = \begin{cases} 
E_{SN}(x_i), & \text{if } x_i \in X_{SN} \\
E_{SI}(x_i), & \text{if } x_i \in X_{SI} \\
E_{T}(x_i), & \text{if } x_i \in X_{T} = X_{TL} \cup X_{TU} 
\end{cases} \]

\[ f(x_i) \in \mathbb{R}^{d_{C}} \]  \hspace{1cm} (4)

Instead of aligning heterogeneous domains into a common feature subspace via brute-force and impair the feature discriminability, we apply a joint semantic transfer strategy to achieve a more fine-grained KT. Specifically, the scenario semantic transfer partially masks heterogeneities between source NI and II domains by confusing the domain discriminator \( D \) to produce domain-invariant common feature subspace. Meanwhile, the categorical distribution knowledge is also transferred between source domains. Additionally, the weighted implicit semantic transfer is used to transfer the correlation relationships between category distributions so that the category distribution semantic will be preserved by the
target and different categories will not be confounded mistakenly during transfer. The knowledge from source domains is weighted based on their divergence with the target domain to adaptively emphasize varied source importances which reflect the degree of knowledge learning. Moreover, the hierarchical explicit semantic alignment is utilized to explicitly minimize the gap between instances of the same category from different domains via a global centroid-level alignment and a local representative-level alignment to increase discriminability. To fully explore the potentials of unlabeled target II domain instances while avoiding negative transfer caused by wrongly assigned pseudo-labels, a PLR with an ensemble mechanism is used to leverage the GS information to enhance the pseudo-label accuracy. Finally, the labeled data will provide the supervision for training via a globally shared classifier. The ultimate goal of the model is to use the trained shared classifier to work on the common feature subspace, so that the prediction accuracy of the unlabeled target II data is maximized.

IV. JSTN ALGORITHM

This section focuses on the detailed mechanisms of three JSTN constituting semantics, with their advantages explained in detail. We then present the overall optimization objective.

A. Scenario Semantic Transfer

1) Auxiliary Source II Domain: The NI data is rich in scale and intrusion knowledge, while the II data have rich IoT scenario characteristics. However, significant heterogeneities are present between network and II domains as illustrated in Fig. 2. For instance, the NI data are usually captured from servers in data centers, while II data comes from resource-constrained IoT infrastructures. Their diverse device types and working environment lead to heterogeneities, such as different set of features, different feature dimensions, follow different distributions, etc. On the other hand, although there are also heterogeneities between different II domains, however, the gap between II domains is smaller than the gap between NI and II domains. For example, although a fridge temperature monitor and a parcel GPS tracker have different functionalities, they usually work under similar network conditions compared with servers in top-tier data centers, they may utilize the same IoT network protocol that is different from servers, and hence the similarity between different II domains are higher than between NI and II domains. If we directly transfer the knowledge from the NI domain to the II domain via DA to facilitate ID, the huge domain gap between NI and II is still likely to hinder the effectiveness of KT as in Fig. 2. Since it not only needs to ensure fine-grained KT, but also needs to tackle significant divergences caused by different network protocol usages between domains, etc.

However, if the source NI domain can be endowed with the IoT scenario semantic by using even a small amount of II data that is not from the target II domain due to target data scarcity, the gap between NI and II domains can be bridged more effectively, which can therefore benefit the source-target KT performed later, as indicated in Fig. 2. Hence, to endow the NI domain with the characteristics of IoT II domains, we use a small amount of II data from another IoT domain.

2) Scenario Semantic Transfer via Domain Discriminator: The JSTN performs scenario semantic transfer via a domain discriminator $D$. When training the domain discriminator $D$, data instances from both the source NI and source II domains will be labeled as 1, while the target II instances are labeled with 0. By confusing the domain discriminator, the source NI and source II instances will be fused, so that the NI instances are equipped with IoT scenario semantic. When the domain discriminator $D$ is confused to distinguish the domain origin of instances, it promotes a domain-invariant common feature subspace to be learned, which can benefit positive transfer. In the JSTN model, the domain discriminator is a neural network with a single layer that performs binary classification task with the loss defined as follows:

$$L_{SSD} = \frac{1}{n_{SN} + n_{SI}} \sum_{x_i \in X_{SN} \cup X_{SI}} \log(D(f(x_i))) + \frac{1}{n_{TL} + n_{TU}} \sum_{x_i \in X_{TL} \cup X_{TU}} (1 - \log(D(f(x_i)))) \tag{5}$$

The domain encoders $E_{SN}$, $E_{SI}$, and $E_T$ will try to confuse the discriminator $D$ while the discriminator tries to stay unconfused. The common feature subspace yielded by domain encoders will become domain invariant when this minimax game reaches an equilibrium.

3) Scenario Semantic Transfer via Distribution Matching: Besides, the source NI domain should also equip the source II domain with rich intrusion knowledge via the predicted categorical distribution KT. Although during KT, the source NI and II domains present heterogeneities, however, it is reasonable that instances from the same category should possess similar predicted categorical distribution, irrespective of which domain they come from. Using objects as the example, a PC monitor should be highly similar to other PC monitors, relatively similar to TV screens, and less likely to be similar to a bicycle or an orange, irrespective of its domain origin. This preservation of categorical correlation applies for ID as well. Hence, transferring this distribution correlation knowledge will promote a more fine-grained feature alignment between domains in the common feature subspace, as indicated in Fig. 3. Besides, it avoids mistaken category confounding, especially at the category boundaries in the learned feature space that tend to be misinterpreted, since the categorical distribution can only be matched when the categories between domain align in the common feature subspace, as illustrated in Fig. 3. Mathematically, the source NI domain average probabilistic output of each category $k$ serves as the
teacher to transfer the distribution correlation to the source II domain and is defined as follows:

\[
q^{(k)} = \frac{1}{|X_{SN}^{(k)}|} \sum_{x_i \in X_{SN}^{(k)}} \text{softmax} \left( \frac{C(f(x_i)))}{T_1} \right)
\]  

where \(C(f(x_i)))\) is the logit produced by the shared classifier, \(X_{SN}^{(k)}\) represents the set of source NI instances belonging to the \(k\)th category, \(|\cdot|\) denotes the number of instances, and \(T_1\) is a temperature hyperparameter that can smooth or sharpen the categorical distribution during the semantic transfer. Similarly, the average probabilistic output of each category \(k\) of the source II domain is defined as follows:

\[
p^{(k)} = \frac{1}{|X_{SI}^{(k)}|} \sum_{x_i \in X_{SI}^{(k)}} \text{softmax} \left( \frac{C(f(x_i)))}{T_1} \right).
\]  

The distribution correlation knowledge is transferred from the source NI domain to the source II domain by minimizing the divergence between \(q^{(k)}\) and \(p^{(k)}\) via the cross-entropy loss defined as follows:

\[
L_{SSC} = -\frac{1}{K} \sum_{k=1}^{K} q^{(k)} \top \log(p^{(k)})
\]  

With the help of the scenario semantic, the network data can mimic the characteristics possessed by the IoT domain to some extent and increase its similarity with the IoT domain, while the source II domain is also equipped with rich intrusion knowledge from the source NI domain. Both source domains will be drawn closer toward each other so that the significant heterogeneities will be partially masked, which will therefore ease the source–target KT process.

B. Weighted Implicit Semantic Transfer

1) Implicit Semantic Transfer: Transferring the categorical distribution knowledge is useful not only between source domains but also between source and target domains. The average probabilistic output of source instances belonging to category \(k\) is treated as the teacher, or the “soft label” of category \(k\), which is defined as follows:

\[
q^{(k)}_{S*} = \frac{1}{|X_{SN}^{(k)}|} \sum_{x_i \in X_{SN}^{(k)}} \text{softmax} \left( \frac{C(f(x_i)))}{T_2} \right)
\]  

where \(S* \in \{SN, SI\}\), \(T_2\) is the smoothing temperature hyper-parameter. With the help of the soft label which contains the implicit semantic, we can let the probabilistic output of all labeled target instances \(p_i\) to preserve the implicit semantic by minimizing the soft loss defined as follows:

\[
p_i = \text{softmax}(C(f(x_i))), x_i \in X_{TL}
\]

\[
L_{sf}^{S*}(X_{TL}, Y_{TL}) = \frac{1}{n_{TL}} \sum_{x_i \in X_{TL}, y_i \in Y_{TL}} q^{(y_i)} \top \log(p_i)
\]  

where \(p_i\) is the categorical probabilistic output of the \(i\)th labeled target instance, and the soft loss \(L_{sf}^{S*}\) shortens the divergence of probabilistic outputs between domains. Besides the soft label that is rich of implicit semantic, each labeled target II domain instance also has its corresponding label, i.e., the “hard label.” The hard label will provide a supervised loss that is defined as follows:

\[
L_{hd}(X_{TL}, Y_{TL}) = \frac{1}{n_{TL}} \sum_{x_i \in X_{TL}, y_i \in Y_{TL}} L_{ce}(C(f(x_i)), y_i)
\]  

where \(L_{ce}\) stands for cross-entropy loss.

2) Divergence-Based Weighting Scheme: Considering that the source NI and II domains may have different divergences toward the target II domain, which implicitly indicate their importance during implicit KT, i.e., the degree of knowledge learning achieved by the target II domain. The divergence between source domains and the target II domain \(d_{<S*,T_{L}>}\) are defined as follows:

\[
\mu^{(k)}_{S*} = \frac{1}{|X_{S*}|} \sum_{x_i \in X_{S*}} f(x_i), \mu^{(k)}_{T_L} = \frac{1}{|X_{T_L}|} \sum_{x_i \in X_{T_L}} f(x_i)
\]

\[
d_{<S_{SN},T_{L}>} = \frac{\sum_{k=1}^{K} \|\mu^{(k)}_{S_{SN}} - \mu^{(k)}_{T_{L}}\|_2}{K}
\]

\[
d_{<S_{SI},T_{L}>} = \frac{\sum_{k=1}^{K} \|\mu^{(k)}_{S_{SI}} - \mu^{(k)}_{T_{L}}\|_2}{K}
\]

where \(\mu^{(k)}_{S_{S*}}\) stands for the class \(k\) centroid of domain \(S_{S*}\). Then, the weights \(\omega_{<S*,T_{L}>}\) for source domains during implicit KT are defined as follows:

\[
\omega_{<S_{SN},T_{L}>} = \frac{e^{d_{<S_{SN},T_{L}>}}}{e^{d_{<S_{SN},T_{L}>}} + 1} + 0.25
\]

\[
\omega_{<S_{SI},T_{L}>} = \frac{e^{d_{<S_{SI},T_{L}>}}}{e^{d_{<S_{SI},T_{L}>}} + 1} + 0.25.
\]

The weight is controlled within the range between 0.75 and 1.25, so that source domains will neither completely lose its influence nor have extremely heavy influence. The smaller the divergence is, the smaller the weight is and vice versa. Therefore, if the source domain presents very little divergence with the target domain, then it indicates that the target domain already possesses the knowledge of that source domain, i.e., a relatively high degree of knowledge learning, and hence
that the source domain is suppressed using a smaller weight. Conversely, a large divergence between a source domain and the target domain indicates that the target domain is not yet fully equipped with the knowledge from that source domain, i.e., a relatively low degree of knowledge learning, and hence that source domain will be emphasized by a large weight. By utilizing this weighting mechanism, it can dynamically adapt the relative importance of source domains during implicit KT to maximize knowledge learning.

Hence, the overall weighted implicit semantic loss $L_{WIS}$ is defined as follows:

$$L_{WIS} = (1 - \alpha)L_{bd}(X_{TL}, Y_{TL}) + \alpha(W_{IS} + L_{hd} + L_{sf})$$

(14)

where hyperparameter $\alpha$ balances the influence between soft and hard losses. By optimizing the weighted implicit semantic loss $L_{WIS}$, the correlation within categorical distribution can be preserved by the target in the common feature subspace, and hence can prevent the negative transfer caused by confounded categories without enough discriminability.

C. Hierarchical Explicit Semantic Alignment

1) Pseudo-Label Refiner: The hierarchical explicit semantic alignment mechanism benefits the KT by minimizing the domain divergence from a distance perspective in a hierarchical manner. Considering that utilizing the unlabeled target domain data during divergence minimization would be helpful [1], [17], we perform the pseudo-label assignment process before transferring the hierarchical explicit semantic. Several previous DA methods utilized pseudo-label for unlabeled target data, however, their pseudo-label assignment tended to be inaccurate, which subsequently misled the model training and caused negative transfer. To circumvent the negative effect caused by wrongly assigned pseudo-label, we utilize a PLR based on the ensemble paradigm to improve the pseudo-label assignment accuracy. For each unlabeled target domain instance $x_i \in X_{TU}$, the shared classifier will yield a prediction, which is treated as the neural network label, denoted as $y_i^{<NN>}$.

Various previous DA efforts directly utilized the predicted label as pseudo-label assignment, however, these pseudo-labels are error-prone, especially during the initial training stage. Therefore, we also take the intrinsic geometric knowledge into account. For both the source data and the labeled target data, we calculate the centroid of instances for each class $\mu^{(k)}$, which is defined as follows:

$$\mu^{(k)} = \frac{1}{|X_{SN} \cup X_{SI} \cup X_{TL}|} \left( \sum_{x_i \in X_{SN}^{(k)}} f(x_i) + \sum_{x_m \in X_{SI}^{(k)}} f(x_m) + \sum_{x_n \in X_{TL}^{(k)}} f(x_n) \right)$$

(15)

where $X_{SN}^{(k)}$ means class $k$ $X_{SN}$ instances. After obtaining the centroid of labeled instances for each category, we can assign each unlabeled target instance to the category whose centroid has the highest Cosine similarity with that unlabeled target instance, namely the GS-based label $y_i^{GS}$. The GS label is decided as follows:

$$y_i^{GS} = \arg\max_k CS\left(f(x_i), \mu^{(k)}\right), x_i \in X_{TU}$$

(16)

where $CS()$ is the Cosine similarity. Considering that when the neural network label and the GS label reach a consensus, it gives the pseudo-label a stronger confidence to be correct since it is more unlikely for both the trained classifier and the intrinsic geometric property to reach the same wrong assignment simultaneously. Hence, the PLR forms a refinement mechanism. It will only assign a pseudo-label to the unlabeled target instance if an agreement is reached, or otherwise that unlabeled target instance will not have a pseudo-label assignment and will not be utilized during hierarchical explicit semantic transfer to circumvent error accumulation. We denote the assigned pseudo-label as $\hat{y}_i$ and denote its corresponding feature vector as $x_i$. Hence, the target instance set will be updated as follows:

$$X_T = X_{TL} \cup \{\hat{x}_i\}, Y_T = Y_{TL} \cup \{\hat{y}_i\}$$

(17)

Initially, the model is not stable enough, hence only a few pseudo-label will be assigned and a majority of unconfident pseudo-label assignment will be filtered out to prevent error accumulation. As the training progresses, assignment agreements will be reached for more unlabeled target instances, which will let them to participate in the hierarchical explicit semantic transfer. Eventually, at the later training stage, a majority of unlabeled instances will be assigned with a consistent pseudo-label, which can make the unlabeled target instances be explored as much as possible. Hence, the PLR can filter out pseudo-label assignments that are possibly wrong to prevent the negative transfer, it forms an automatic pseudo-label assignment process without requiring human experience or manually assigned thresholds.

2) Hierarchical Explicit Semantic Transfer—Global Centroid Level: With the help of the PLR, we can perform the hierarchical explicit semantic transfer from two levels. First, a global-level triplet centroid alignment is performed to align category-wise centroids. Specifically, we can calculate the category-wise centroid for the source domain $\mu_S^{(k)}$, target domain $\mu_T^{(k)}$, and the combination of source and target domains $\mu_{ST}^{(k)}$, which are defined as follows:

$$\mu_S^{(k)} = \frac{1}{|X_{SN} \cup X_{SI}^{(k)}|} \sum_{x_i \in X_{SN}^{(k)} \cup X_{SI}^{(k)}} f(x_i)$$

$$\mu_T^{(k)} = \frac{1}{|X_T^{(k)}|} \sum_{x_j \in X_T^{(k)}} f(x_j)$$

$$\mu_{ST}^{(k)} = \frac{1}{|X_{SN} \cup X_{SI}^{(k)} \cup X_T^{(k)}|} \left( \sum_{x_i \in X_{SN}^{(k)} \cup X_{SI}^{(k)}} f(x_i) + \sum_{x_i \in X_T^{(k)}} f(x_j) \right).$$

(18)
Then, we explicitly learn a more robust and discriminative feature representation by minimizing the intracategory divergence, i.e., minimizing the $L_2$-distances between each centroid, which is defined as follows:

$$L_{ESC} = \sum_{k=1}^{K} \left( ||\mu_S^{(k)} - \mu_T^{(k)}||_2^2 + ||\mu_S^{(k)} - \mu_{ST}^{(k)}||_2^2 + ||\mu_T^{(k)} - \mu_{ST}^{(k)}||_2^2 \right).$$

(19)

3) Hierarchical Explicit Semantic Transfer—Local Representative Level: However, only performing the global-level centroid alignment is not enough to achieve fine-grained semantic consistency. As shown in the upper part of Fig. 4, even though the centroids are alignment, the features can still lack concentration as indicated by the gray shaded area, which hurts the semantic transfer. This is due to that centroids only represent the category at the global level, which lack thorough coverage of the whole category in a fine-grained manner. Therefore, the explicit semantic alignment is also performed from a local perspective to achieve a more fine-grained category coverage. For each category $k$ in both the source domain and target domain ($D_T = (X_T, Y_T)$), $R$ representatives are selected via Kmeans+++ clustering, denoted as $r_S^k(i)$ and $r_T^k(i), i \in [1, R], k \in [1, K]$, respectively. Then, we calculate the pairwise distances between source and target representatives for each category as follows:

$$L_{ESR} = \sum_{k=1}^{K} \sum_{j=1}^{R} \sum_{i=1}^{R} \frac{||r_S^k(i) - r_T^k(j)||_2^2}{K \times |r_S^k| \times |r_T^k|}.$$  

(20)

Unlike performing the pairwise divergence minimization for all source and target instances as in [38], the hierarchical explicit semantic works on category-wise representatives, which avoids the severe computational burden without hurting the alignment effectiveness. By explicitly minimizing the intracategory divergence from the global-level centroid perspective and the local-level representative perspective, each domain in the common feature subspace will be more semantically consistent in a concentrated manner as indicated in the lower part of Fig. 4.

D. Overall Optimization Objective

Finally, the ground-truth labels of source domains and the predicted output yielded by the shared classifier will produce a supervision loss $L_{sup}$ as follows:

$$L_{sup}(X_S, Y_S) = \frac{1}{n_S} \sum_{x_i \in X_S, y_i \in Y_S} L_{ce}(C(f(x_i)), y_i)$$  

(21)

while the supervision loss of the labeled target domain has been treated as hard label loss in the implicit semantic transfer as previously mentioned. Overall, the optimization objective of the JSTN model is as follows:

$$\min_{C, E_{SN}, E_{ST}, E_T} \max_D \{L_{sup} + L_{WIS} + \beta L_{ESC} + \lambda L_{ESR} + \gamma L_{SSD} + \eta L_{SSC}\}$$  

(22)

where $\beta, \lambda, \gamma, \text{ and } \eta$ are hyperparameters that control the influence of loss components during optimization. Inspired by [39], we apply the gradient reversal layer (GRL) on the discriminator to train the entire JSTN network in an end-to-end manner using Adam gradient descent. By optimizing the overall objective, the scenario semantic fuses domains by confusing domain discriminator and meanwhile transfer knowledge between source domains. Hence, it forms a domain-invariant feature subspace so that the heterogeneities between source–source domains and source–target domains will be minimized. The weighted implicit semantic increases generalizability through preserving the implicit categorical distribution knowledge, the knowledge from different source domains is weighted based on their relative divergence with the target domain to indicate source importance which reflects the degree of knowledge learning. Meanwhile, the hierarchical explicit semantic learns a robust and semantically consistent common feature subspace with compactness and concentration from the global centroid perspective and local representative perspective so that the intracategory divergence will be shortened. By jointly leveraging these semantics, the KT effectiveness of the model will be enhanced. Upon the above minimax game reaches an equilibrium, the training of feature encoders for each domain, the shared classifier $C$, and the domain discriminator $D$ concludes.

V. EXPERIMENT

A. Data Set and Setup

NI Data Set (NSL-KDD): The NSL-KDD (K) [40] NI data set was released in 2009. It improves the outdated KDD99 data set [41] to reflect modern network attack characteristics. It contains benign traffic with four malicious attack categories, such as Denial of Service (DoS), probing attack, etc. It does not have redundant or duplicate records, the quality of data is significantly improved. Follow [12], we use 20% of the data set, which is a reasonable and affordable amount. Each record is represented using 41 features. We follow Harb et al. [42] to choose 31 most informative features as the feature space.
**NI Data Set (UNSW-NB15):** The UNSW-NB15 (N) NI data set [43] was created in 2015 using the IXIA PerfectStorm tool. The data set also aims to tackle the limitations such as redundant records or missing values of previous IDS data sets, especially under a modern low footprint environment. The data set contains benign network behaviors plus nine attack categories, such as DoS attack, reconnaissance attack, etc. The data set contains 257,646 records, follow previous work [44], we utilize 6000 entries during the model training and evaluation. Each record is represented using 49 features. The preprocessing steps include removing four features that have value 0 for nearly all records.

**NI Data Set (CICIDS2017):** The CICIDS2017 (C) [45] NI data set was released in 2017. It is one of the most up-to-date network traffic data sets. The data is collected using CICFlowMeter. The data set has benign and seven common intrusion attack types to reflect the current trend. The attack types, including DoS, DDoS, Brute Force attack, etc. The portion of the data set (20%) [46] has been provided in the CSV format for ML training, represented using 77 features. We perform preprocessing including data deduplication, and converting categorical attributes to numerical entries. We follow [12] to utilize 20,000 entries of network traffic, a reasonable amount to train an effective IDS. Guided by the information gain-based feature selection work of Stiawan et al. [46] we use the features with top 40 information gain, which can effectively filter out information-scarce features and improve training efficiency.

**II Data Set (UNSW-BOTIOT):** The UNSW-BOTIOT (B) data set [47] was also created in 2017 with a specific focus on realistic II scenarios. It applies five II scenarios in the testbed, including a weather station, a smart fridge, a smart thermostat associated with in-house air-conditioning, etc. The testbed also utilizes the MQTT protocol, a lightweight communication protocol commonly used between II devices. Hence, the data set fills the void of lacking specific consideration for II scenarios. It contains three common II attack categories, such as DoS attack, information theft, etc. Following [44], we utilize 7,500 records during the model training and evaluation. Note that when used as the source II domain, the amount of data is 1/6–1/2 of the amount of source NI data to reflect the diversity of II scenarios. It contains three common II attack categories, including scanning attack, DoS attack, DDoS attack, reconnaissance attack, and password attack. These shared common categories are representatives of the majority of modern intrusions faced by networks and II devices, they account for 99.85%, 99.96%, 54.2%, 100%, and 77.03% amounts of records in the CICIDS2017, NSL-KDD, UNSW-NB15, UNSW-BOTIOT, and UNSW-TONIOT data sets, respectively. Therefore, after transferring the knowledge, most modern intrusion attacks faced by the II domain can be detected.

**Implementation Details:** We implement the JSTN model using the PyTorch [50] DL framework and deploy the experiment on a server equipped with Intel Core i9-9900K CPU and Nvidia Tesla V100 GPU. All feature encoders are two-layer fully connected neural networks with LeakyReLU [51] as the activation function following [1] and [34]. Both the shared classifier C and the domain discriminator D are single-layer neural networks. For hyperparameter settings, we empirically set $\alpha = 0.1$, $\beta = 0.004$, $\lambda = 0.001$, $\gamma = 0.1$, $\eta = 0.001$, $T_1 = 10$, and $T_2 = 5$, and the dimension of the domain-invariant common feature subspace $d_{C,S}$ is set to 3. We also verify the parameter sensitivity in Section V-D to indicate the JSTN can perform stably and robustly under varied parameter settings. Following [1], we optimize the JSTN model using the Adam gradient descent optimizer and set the number of epochs to 1000. Following [44] and [52], we mainly use accuracy on the unlabeled target II data as the evaluation metrics, as well as category-weighted precision (P), recall (R) and F1-score (F) [53] to evaluate the performance. We define true positive $TP^{(k)}$ to be the number of unlabeled target II instances which belong to intrusion category $k$ and are correctly predicted as intrusion category $k$, similar for true negative $TN^{(k)}$, false positive $FP^{(k)}$, and false negative $FN^{(k)}$. Hence, the category-weighted precision, recall, and F1-score are mathematically defined as follows:

\[
\text{Precision} = \frac{1}{K} \sum_{k=1}^{K} \frac{TP^{(k)}}{TP^{(k)} + FP^{(k)}} \cdot \text{Precision}^{(k)}
\]

\[
= \frac{1}{K} \sum_{k=1}^{K} \frac{\lambda^{(k)}}{n_{TU}} \cdot \text{Precision}^{(k)}
\]
TABLE I
ID ACCURACY OF TEN METHODS ON TEN TASKS

| Method   | C + M → T_I | C + W → T_I | C + B → G | N + G → T_I | N + F → T_I | N + B → G | K + M → T_I | K + W → T_I | K + M → G | Avg   |
|----------|-------------|-------------|-----------|-------------|-------------|-----------|-------------|-------------|-----------|-------|
| SB-RF    | 48.84       | 46.76       | 73.40     | 47.36       | 64.87       | 82.30     | 50.50       | 48.81       | 46.65     | 82.30 |
| SB-SVM   | 52.56       | 50.96       | 73.60     | 51.78       | 67.84       | 82.40     | 55.00       | 52.55       | 50.28     | 82.33 |
| SB-NN    | 53.01       | 51.58       | 73.68     | 47.51       | 65.38       | 82.36     | 56.62       | 52.53       | 51.70     | 82.35 |
| SB-TNT   | 52.10       | 50.84       | 73.68     | 52.21       | 70.21       | 85.54     | 61.50       | 58.98       | 50.89     | 82.34 |
| SB-DDAC  | 60.71       | 53.62       | 72.84     | 52.88       | 72.32       | 85.60     | 63.35       | 60.71       | 53.05     | 82.35 |
| SB-DDAS  | 58.98       | 50.84       | 73.68     | 52.21       | 70.21       | 85.54     | 61.50       | 58.98       | 50.89     | 82.34 |
| STN      | 62.29       | 54.92       | 76.83     | 54.99       | 71.83       | 87.00     | 63.95       | 62.79       | 54.44     | 85.57 |
| C yan    | 61.77       | 54.45       | 76.18     | 55.20       | 73.69       | 86.91     | 63.79       | 60.08       | 54.69     | 85.15 |
| JSTN (Ours) | 67.27       | 69.69       | 77.08     | 70.20       | 86.94       | 87.78     | 69.31       | 66.95       | 69.73     | 78.10 |

| Method   | N + M → G | K + B → W | N + G → W |
|----------|-----------|-----------|-----------|
| SB-RF    | 93.33     | 93.32     | 93.28     |
| SB-SVM   | 93.32     | 93.29     | 93.26     |
| SB-NN    | 93.33     | 93.32     | 93.30     |
| SB-TNT   | 93.30     | 93.30     | 93.22     |
| SB-DDAC  | 93.33     | 93.33     | 93.31     |
| SB-DDAS  | 93.33     | 93.32     | 93.30     |
| SB-STN   | 93.48     | 93.33     | 93.19     |
| STN      | 92.99     | 92.94     | 92.39     |
| C yan    | 93.50     | 93.33     | 93.00     |
| JSTN (Ours) | 94.46     | 94.01     | 93.96     |

Recall = \sum_{k=1}^{K} \frac{X(k)_{TU}}{n_{TU}} \cdot \frac{TP(k)}{TP(k) + FN(k)}

F1 = \sum_{k=1}^{K} \frac{X(k)_{TU}}{n_{TU}} \cdot \frac{2 \cdot Precision(k) \cdot Recall(k)}{Precision(k) + Recall(k)}

B. Performance Evaluation

We first analyze the ID performance of the JSTN compared with other state-of-the-art counterparts on several randomly selected representative tasks. The evaluation results are presented in Tables I–IV. In Table I, the default ratio between the amount of labeled and unlabeled target II domain instances is set to 1:2 when T_H = B, and is set to 1:5 otherwise. As indicated in Table I, JSTN outperforms all other comparing methods under all tasks. Specifically, comparing with the double-source counterpart C yan and STN, the best-performed single source method SB-STN, and the best traditional supervised ML method SVM, the JSTN yields a 9.42%, 8.02%, 7.75%, and 13.28% performance boost, which is a significant improvement of detection accuracy. It is natural to observe this since although C yan utilizes both the source NI and II domains to facilitate ID for the target II domain, it does not specifically pay attention to semantic transfers such as the implicit or explicit semantics, which therefore verifies the usefulness of the robust semantic KT utilized by the JSTN. Besides, the STN method does not consider the scenario semantics, which therefore results in hindered ID performance when huge heterogeneities are present between NI and II domains.

To verify the effectiveness of methods under varied n_{TL}:n_{TU} ratios, especially under the extreme case where the amount of unlabeled target II domain data is significantly higher than the amount of labeled target II domain instances, six tasks are...
TABLE III
(Continued From Table II) ID Accuracy Under Varied $n_{TL} : n_{TU}$ Ratios

| $S_{N} + S_{I} \rightarrow T_{I}$ | $C + W \rightarrow B$ | $N + M \rightarrow B$ | $N + W \rightarrow B$ | Overall Avg | 1 : 50 Case Avg |
|----------------------------------|---------------------|---------------------|---------------------|------------|------------|
|                                  | 1 : 2               | 1 : 10              | 1 : 50              | 1 : 2      | 1 : 10     | 1 : 50     |
| SB-RF                            | 46.76               | 46.70               | 46.53               | 50.50      | 50.22      | 50.08      | 47.38      | 47.37      | 47.35      | 65.60      | 65.53      |
| SB-SVM                           | 50.96               | 49.49               | 48.51               | 55.03      | 54.10      | 53.49      | 51.47      | 50.04      | 48.98      | 67.74      | 67.60      |
| SB-NN                            | 51.58               | 48.67               | 48.50               | 56.62      | 53.59      | 53.20      | 51.50      | 50.92      | 50.67      | 67.67      | 67.16      |
| SB-TNT                           | 25.57               | 25.45               | 24.99               | 49.50      | 49.33      | 49.00      | 25.93      | 25.78      | 25.60      | 58.24      | 58.09      |
| SB-DDAC                          | 53.62               | 52.55               | 51.83               | 63.35      | 62.05      | 60.80      | 52.95      | 52.79      | 52.75      | 70.88      | 70.47      |
| SB-DDAS                          | 51.84               | 51.60               | 49.80               | 61.50      | 59.45      | 58.70      | 51.97      | 51.90      | 51.30      | 69.54      | 69.02      |
| SB-STN                          | 54.92               | 54.47               | 52.74               | 63.95      | 62.19      | 62.07      | 54.60      | 54.24      | 54.61      | 71.49      | 70.56      |
| STN                              | 54.29               | 53.06               | 51.78               | 61.98      | 60.91      | 55.94      | 54.08      | 53.24      | 52.61      | 70.66      | 69.50      |
| CWAN                             | 52.40               | 51.87               | 51.77               | 59.69      | 58.32      | 56.28      | 53.03      | 52.12      | 52.10      | 69.90      | 69.30      |
| JSTN (Ours)                      | 69.69               | 66.49               | 61.96               | 69.31      | 67.71      | 67.65      | 69.93      | 67.24      | 67.15      | 77.26      | 76.13      |

TABLE IV
ID Precision, Recall and F1-Score of DA-Based Methods

| $S_{N} + S_{I} \rightarrow T_{I}$ | $N + M \rightarrow G$ | $K + B \rightarrow W$ |
|----------------------------------|---------------------|---------------------|
|                                  | P                   | R                   | F                   |
| SB-TNT                           | 0.871               | 0.932               | 0.900               | 0.543      | 0.735      | 0.624      |
| SB-DDAC                          | 0.871               | 0.932               | 0.901               | 0.720      | 0.784      | 0.734      |
| SB-DDAS                          | 0.870               | 0.932               | 0.900               | 0.639      | 0.758      | 0.671      |
| SB-STN                          | 0.931               | 0.930               | 0.931               | 0.723      | 0.764      | 0.717      |
| STN                              | 0.951               | 0.930               | 0.938               | 0.747      | 0.793      | 0.745      |
| CWAN                             | 0.871               | 0.928               | 0.898               | 0.650      | 0.772      | 0.684      |
| JSTN (Ours)                      | 0.954               | 0.950               | 0.952               | 0.787      | 0.808      | 0.763      |

Randomly selected with the $n_{TL} : n_{TU}$ ratio varied between 1:2 and 1:50. Following [1], [34], and [37], the 1:50 case is sufficient to represent an extremely label-scarce scenario. As shown in Tables II and III, the JSTN outperforms all baseline methods by a large margin. Overall, the JSTN achieves a 7.36%, 6.6%, 5.77%, and 9.52% amount of performance boost compared with the double-source method CWAN, STN, the best-performed single-source baseline SB-STN, and the best-performed traditional supervised ML method SVM, respectively. Specifically, under the extreme 1:50 case, the JSTN also shows robust performance, its performance only drops 1.13% compared with the 1:10 case while still outperforming the best DA method SB-STN and the best ML method NN by 5.57% and 8.97%, which not only verifies the effectiveness of the JSTN when performing semantic KT to facilitate ID in the IoT target domain but also testifies the robustness of the JSTN when working on extremely scarcely labeled target II domain.

To further verify the efficacy of methods under evaluation metrics other than ID accuracy, we utilize precision, recall, and F1-score as additional metrics and present the results on two randomly selected tasks as representatives in Table IV. As we can see, the JSTN achieves the highest precision, recall and F1-score among these tasks over other DA-based baseline methods. By achieving the highest precision, it indicates that the JSTN model achieves the highest correctness among all network traffic that it flags as malicious attacks. Besides, achieving the best recall reveals the JSTN can detect the most amount of malicious traffic out of all malicious behaviors, which indicates its effectiveness in terms of ID. Overall, the highest F1-score indicates the JSTN successfully balances between flagging as many intrusions from all malicious behaviors as possible, and meanwhile avoiding triggering too many false alarms. Hence, together with the overall accuracy as indicated in Tables I–III, the best performance on all evaluation metrics achieved by the JSTN model verifies its superiority.

C. Ablation Study

Component Ablation Study: After performing the overall performance evaluation, we now verify the usefulness of each constituting semantic transfer component of the JSTN. The JSTN variants include the following: 1) $\alpha = 0$, which ablates the weighted implicit semantic transfer; 2) No WI, which turns off the weighting mechanism during implicit semantic transfer; 3) $\beta = 0$, which removes the centroid-level explicit semantic alignment; 4) $\lambda = 0$, which ablates the representative-level explicit semantic alignment; 5) $\beta = \lambda = 0$, which completely turns off the hierarchical explicit alignment; 6) No PLR, which assigns pseudo-label directly from the shared classifier $C$, without using the PLR that is geometric-aware; 7) $\eta = 0$, which removes the categorical distribution preservation used during scenario semantic transfer; 8) $\gamma = 0$, which turns off the domain discriminator $D$, part of the scenario semantic; 9) $S_{NI}$ Only, which only uses a source NI domain, without considering the scenario semantic; and 10) $S_{II}$ Only, which only uses a source II domain, without utilizing the knowledge-rich source NI domain.

The ablation performance on three randomly selected representative tasks is indicated in Table V. The JSTN outperforms all its ablated variants, which verifies that all semantic transfer components are indispensable to facilitate a robust KT. Without any constituting semantic, negative effects will be caused, which therefore leads to impaired ID performance. Among all these components, the weighted implicit semantic contributes around 1.7% of performance improvement, while
the weighting mechanism utilized during implicit semantic raises the performance by 1.38%. The hierarchical explicit semantic alignment contributes 1.76% performance improvement on average. Specifically, only using the centroid-level or the representative-level explicit semantic alignment will cause the performance to drop by 1.51% and 1.65%, respectively, which verifies the importance of the hierarchical explicit semantic alignment. The PLR yields 1.83% performance boost, which verifies the necessity to refine pseudo-labels. In terms of scenario semantic, using the domain discriminator will bring 1.75% performance improvement. As part of the scenario semantic, the categorical distribution knowledge preservation yields 2.47% performance increase. Additionally, the $S_{NI}$ only variant presents a performance reduction of around 1.63% without the help of scenario semantic brought by the $S_{II}$ domain, and the $S_{II}$ only variant reduces the performance by 3.56% due to lack of data and intrusion knowledge contained in it. The degraded performance of single-domain variants further verifies the usefulness of the scenario semantic.

### Necessity of Scenario Semantic

To have a closer look of the importance and necessity of the scenario semantic we proposed, detailed analyses are performed as indicated in Table VI. When transferring knowledge to facilitate the ID of the target II domain, three variants are considered as follows: I, only transfer the knowledge via a single $S_{NI}$ domain; II, transfer the knowledge via two source NI domains $S_{NI1}$ and $S_{NI2}$, to make it a comparable replacement of $S_{II}$, $S_{NI2}$ has the same scale as $S_{NI1}$; and III, transfer the knowledge via the scenario semantic-enabled setting, i.e., an $S_{NI}$ domain facilitated with a small-scale $S_{II}$ domain.

As we can observe from Table VI, the scenario semantic-enabled variant achieves superior performance than the other two variants by 1.63% and 1.45% on average, respectively. Hence, it indicates that a single-source NI domain may present an overly large heterogeneous semantic gap, which will significantly hinder the ID performance without the help from the scenario semantic. Besides, by facilitating the source NI domain $S_{NI}$ with another NI domain, the semantic gap caused by domain heterogeneity is not effectively shortened, which is revealed by nearly the same performance between variants I and II. Furthermore, by leveraging a source II domain which is even 2–6 times smaller than the NI domain counterpart in scale, it can yield positive scenario semantic transfer to bridge the domain gap between heterogeneous NI and II domains, as verified by the superior performance.

### Significant Test Verification

To verify the performance gains achieved by the full JSTN over its ablated variants are statistically significant, i.e., not observed randomly by chance, significant $T$-tests with 0.05 as the significant threshold are performed on three randomly picked tasks, each is repeated ten times. The test results are illustrated in Fig. 5. The significant threshold $−\log(0.05)$ is indicated by the gray shaded area in the center of each subfigure. Each dimension represents an ablated JSTN variant, the higher the value is, the more significant the performance gain is on this ablated component.

As we can see, the full JSTN has a wider coverage on all dimensions under all tasks. A wider coverage than the gray shaded area indicates the test results among all ablated variants are significant. Hence, the usefulness and necessity of all constituting components of JSTN are verified with statistical significance.

### D. Parameter Sensitivity Analysis

To verify the parameter sensitivity of the JSTN model, we vary five major hyperparameters, i.e., $\alpha$, $\beta$, $\lambda$, $\eta$, and $\gamma$ within their corresponding reasonable value ranges. We randomly select four tasks as representatives and plot the results in Fig. 6. The best-performed baseline method for each task is also plotted with the corresponding color in dashed lines.

As we can notice, when parameters vary, the performance of the JSTN model remains relatively stable without incurring severe fluctuation, as indicated by the relatively stable trend of each solid line. Besides, the solid lines stay above their corresponding colored dashed line in nearly all parameter ranges, which means the JSTN outperforms the corresponding best-performed counterpart under nearly all parameter ranges. Therefore, it verifies the robustness and effectiveness of the JSTN model.

### E. Computational Efficiency

We further measure the computational efficiency of the JSTN model. The results on six randomly selected representative tasks are presented in Table VII for average training time per epoch, and in Table VIII for average inference time per unlabeled target instances. We only compare the JSTN with top-performed counterparts STN and CWAN. As indicated in Table VII, the JSTN demonstrates the most efficient per-epoch training speed. The per-epoch performance boost achieved by JSTN is 21.6% more efficient compared with the second-best performed counterpart STN. The most efficient per-epoch training speed reflects that the JSTN enjoys a relatively low computational complexity. The STN, CWAN, and JSTN require 300, 500, and 1000 epochs to train, therefore, the overall training time of these three methods stays comparable with each other. Given that the model training will be performed on relatively resource-rich devices, the computational cost of the JSTN model is satisfying. On the other hand, when
TABLE VI
ID PERFORMANCE WHEN TRANSFERRING SEMANTIC KNOWLEDGE TO FACILITATE TARGET II DOMAIN ID VIA I: ONLY A SINGLE SOURCE NI DOMAIN; II: TWO SOURCE NI DOMAINS; AND III: A SOURCE NI DOMAIN WITH A SMALL-SCALE SOURCE II DOMAIN

| Task | $C + W \rightarrow B$ | $N + F \rightarrow B$ | $N + B \rightarrow W$ | $K + W \rightarrow B$ | $N + W \rightarrow G$ | Avg |
|------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----|
| I: $S_{N1} \rightarrow T_{II}$ | 67.18 | 84.93 | 86.96 | 68.64 | 84.88 | 78.52 |
| II: $S_{N1} + S_{N12} \rightarrow T_{II}$ | 66.30 | 84.84 | 87.04 | 68.28 | 86.03 | 78.70 |
| III: $S_{N1} \rightarrow T_{II}$ | 68.91 | 84.87 | 87.08 | 67.86 | 85.78 | \n
TABLE VII
AVERAGE TRAINING TIME PER EPOCH (MEASURED IN SECOND, THE LOWER THE BETTER)

| $S_{N1} + S_{I}$ | $C + W \rightarrow B$ | $C + B \rightarrow G$ | $N + G \rightarrow B$ | $N + B \rightarrow W$ | $K + W \rightarrow G$ | Avg |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----|
| STN              | 0.72                  | 0.75                  | 0.37                  | 0.38                  | 0.48                  | 0.34 | 0.51 |
| CWAN             | 1.10                  | 1.14                  | 0.59                  | 0.61                  | 0.76                  | 0.55 | 0.79 |
| JSTN (Ours)      | **0.50**              | **0.51**              | **0.31**              | **0.36**              | **0.40**              | **0.30** | **0.40** |

TABLE VIII
AVERAGE INFERENCE TIME PER UNLABELED TARGET INSTANCE (MEASURED IN MICROSECOND ($10^{-6}$ s), THE LOWER THE BETTER)

| $S_{N1} + S_{I}$ | $C + W \rightarrow B$ | $C + B \rightarrow G$ | $N + G \rightarrow B$ | $N + B \rightarrow W$ | $K + W \rightarrow G$ | Avg |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----|
| STN              | 81.2                  | 56.2                  | 57.6                  | 55.1                  | 52.8                  | 58.8 | 59.68 |
| CWAN             | 56.9                  | 49.4                  | 78.1                  | 58.8                  | 65.0                  | 66.2 | 62.40 |
| JSTN (Ours)      | **0.21**              | **0.22**              | **0.20**              | **0.21**              | **0.23**              | **0.18** | **0.21** |

Fig. 5. To verify the performance gains of the full JSTN over its ablated variants are statistically significant, significant $T$-tests with 0.05 as the significant threshold are conducted on three randomly picked tasks. The $-\log(0.05)$ significant threshold is indicated by the gray shaded area. In each dimension, the higher the value is, the more significant the performance gain is. (a) $C + W \rightarrow B$. (b) $K + M \rightarrow G$. (c) $N + F \rightarrow B$.

Fig. 6. Parameter sensitivity of the JSTN for hyperparameters $\alpha$, $\beta$, $\lambda$, $\eta$, and $\gamma$ under their corresponding reasonable ranges on four randomly selected tasks. Each color represents a task. Each color has two lines, the solid line indicates the JSTN performance, and the dashed horizontal line indicates the accuracy of the best-performed baseline method of the corresponding task.

performing the inference for unlabeled target intrusion data, the JSTN achieves the lowest per-instance inference time, as indicated in Table VIII. The JSTN even achieves $10^2$ times performance boost, thanks to the JSTN’s excellent efficiency. Overall, the results verify the computational efficiency of the JSTN model.
Table IX contains the notations used in this article, and their corresponding interpretation. Note that the symbol $** \in \{SN, SI, TL, TU\}$ stands for source network domain, source IoT domain, labeled target domain, and unlabeled target domain, respectively. The $*$ carries the same meaning in the following notations.

### References

[1] S. Li, B. Xie, J. Wu, Y. Zhao, C. H. Liu, and Z. Ding, “Simultaneous semantic alignment network for heterogeneous domain adaptation,” in Proc. 28th ACM Int. Conf. Multimedia, New York, NY, USA, 2020, pp. 3866–3874. [Online]. Available: https://doi.org/10.1145/3394171.3413995

[2] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, “A vision of IoT: Applications, challenges, and opportunities with China perspective,” IEEE Internet Things J., vol. 1, no. 4, pp. 349–359, Aug. 2014.

[3] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, “An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges,” IEEE Internet Things J., vol. 5, no. 3, pp. 1505–1515, Jun. 2018.

[4] Y. Li, X. Cheng, Y. Cao, and L. Yang, “Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT),” IEEE Internet Things J., vol. 1, no. 4, pp. 311–318, Aug. 2014.
Jiashu Wu received the B.Sc. degree in computer science and financial mathematics and statistics from The University of Sydney, Camperdown, NSW, Australia, in 2018, and the M.I.T. degree in artificial intelligence from The University of Melbourne, Parkville, VIC, Australia, in 2020. He is currently pursuing the Ph.D. degree with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and the University of Chinese Academy of Sciences, Beijing, China. His research interests include machine learning and cloud computing.

Yang Wang (Member, IEEE) received the B.Sc. degree in applied mathematics from the Ocean University of China, Qingdao, China, in 1989, the M.Sc. degree in computer science from Carlton University, Ottawa, ON, Canada, in 2001, and the Ph.D. degree in computer science from the University of Alberta, Edmonton, AB, Canada, in 2008. He is currently with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor and with Xiamen University, Xiamen, China, as an Adjunct Professor. He was an Alberta Industry Research and Development Associate from 2009 to 2011, and a Canadian Fulbright Scholar from 2014 to 2015. His research interests include service and cloud computing, programming language implementation, and software engineering.

Binhu Xie is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institution of Technology, Beijing, China. His research interests focus on computer vision and transfer learning.

Shuang Li received the Ph.D. degree in control science and engineering from the Department of Automation, Tsinghua University, Beijing, China, in 2018. He was a Visiting Research Scholar with the Department of Computer Science, Cornell University, Ithaca, NY, USA, from November 2015 to June 2016. He is currently an Assistant Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His main research interests include machine learning and deep learning, especially in transfer learning and domain adaptation.

Hao Dai received the B.Sc. and M.Sc. degrees in communication and electronic technology from Wuhan University of Technology, Wuhan, China, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and the University of Chinese Academy of Sciences, Beijing, China. His research interests include mobile-edge computing, federated learning, and deep reinforcement learning.

Kejiang Ye (Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2008 and 2013, respectively. He was also a joint Ph.D. student with the University of Sydney, Camperdown, NSW, Australia, from 2012 to 2013. After graduation, he worked as a Postdoctoral Researcher with Carnegie Mellon University, Pittsburgh, PA, USA, from 2014 to 2015 and Wayne State University, Detroit, MI, USA, from 2015 to 2016. He is currently an Associate Professor with Shenzhen Institutes of Advanced Technology, Chinese Academy of Science, Shenzhen, China. His research interests focus on the performance, energy, and reliability of cloud computing and network systems.

Chengzhong Xu (Fellow, IEEE) received the Ph.D. degree from the University of Hong Kong, Hong Kong, in 1993. He is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China, and the Director of the Institute of Advanced Computing and Data Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He has published more than 200 papers in journals and conferences. His research interest includes parallel and distributed systems, service and cloud computing, and software engineering. Dr. Xu serves on a number of journal editorial boards, including IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, Journal of Parallel and Distributed Computing, and Science China Information Sciences.