Open Set Dandelion Network for IoT Intrusion Detection

JIASHU WU and HAO DAI, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China and University of Chinese Academy of Sciences, China
KENNETH B. KENT, University of New Brunswick, Canada
JEROME YEN and CHENGZHONG XU, University of Macau, China
YANG WANG, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

As Internet of Things devices become widely used in the real-world, it is crucial to protect them from malicious intrusions. However, the data scarcity of IoT limits the applicability of traditional intrusion detection methods, which are highly data-dependent. To address this, in this article, we propose the Open-Set Dandelion Network (OSDN) based on unsupervised heterogeneous domain adaptation in an open-set manner. The OSDN model performs intrusion knowledge transfer from the knowledge-rich source network intrusion domain to facilitate more accurate intrusion detection for the data-scarce target IoT intrusion domain. Under the open-set setting, it can also detect newly-emerged target domain intrusions that are not observed in the source domain. To achieve this, the OSDN model forms the source domain into a dandelion-like feature space in which each intrusion category is compactly grouped and different intrusion categories are separated, i.e., simultaneously emphasising inter-category separability and intra-category compactness. The dandelion-based target membership mechanism then forms the target dandelion. Then, the dandelion angular separation mechanism achieves better inter-category separability, and the dandelion embedding alignment mechanism further aligns both dandelions in a finer manner. To promote intra-category compactness, the discriminating sampled dandelion mechanism is used. Assisted by the intrusion classifier trained using both known and generated unknown intrusion knowledge, a semantic dandelion correction mechanism emphasises easily-confused categories and guides better inter-category separability. Holistically, these mechanisms form the OSDN model that effectively performs intrusion knowledge transfer to benefit IoT intrusion detection. Comprehensive experiments on several intrusion datasets verify the effectiveness of the OSDN model, outperforming three state-of-the-art baseline methods by 16.9%. The contribution of each OSDN constituting component, the stability and the efficiency of the OSDN model are also verified.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Transfer learning**; **Neural networks**; • **Security and privacy** → **Intrusion detection systems;**

This work is supported by the National Key R&D Program of China (No. 2021YFB3300200), National Natural Science Foundation of China (No. 92267105), Guangdong Special Support Plan (No. 2021TQ06X990), the Third Xinjiang Scientific Expedition Program (Grant No. 2021XJKK1300), Shenzhen Science and Technology Plan Project (Shenzhen-Hong Kong-Macau Category C, No. SGDX20220530111001003), Key-Area Research and Development Program of Guangdong Province (No. 2021B010140005) and the Chinese Academy of Sciences President's International Fellowship Initiative (Grant No. 2023VTA0001, No. 2023DT0003).
Authors' addresses: J. Wu and H. Dai, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China and University of Chinese Academy of Sciences, Beijing, Beijing 100049, China; e-mails: {wujiashu21, daihao19}@mails.ucas.ac.cn; K. B. Kent, University of New Brunswick, Fredericton, New Brunswick E3B 5A3, Canada; e-mail: ken@unb.ca; J. Yen and C. Xu, University of Macau, Taipa, Macau 999078, China; e-mails: {jeromeyen, czxu}@um.edu.mo; Y. Wang (Corresponding author), Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China; e-mail: yang.wang1@siat.ac.cn.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1533-5399/2024/02-ART4
https://doi.org/10.1145/3639822

Additional Key Words and Phrases: Domain adaptation, internet of things, intrusion detection, open-set domain adaptation, dandelion network

ACM Reference Format:
Jiashu Wu, Hao Dai, Kenneth B. Kent, Jerome Yen, Chengzhong Xu, and Yang Wang. 2024. Open Set Dandelion Network for IoT Intrusion Detection. *ACM Trans. Internet Technol.* 24, 1, Article 4 (February 2024), 26 pages. https://doi.org/10.1145/3639822

1 INTRODUCTION

Internet of Things (IoTs) devices become prevalent in many real-world applications [6, 22, 38, 40]. However, they tend to be computational and energy-constrained, which hinder the deployment of effective intrusion detection mechanisms. Together with the lack of maintenance, these limitations compromise the security of IoT devices, making them vulnerable to attacks [20, 42]. To protect the safety of IoT devices, an effective intrusion detection mechanism becomes indispensable [39].

The intrusion detection for IoT has drawn wide attention from the academic community. For instance, signature-based intrusion detectors were proposed [8, 25, 26], which detected malicious behaviours by pattern matching with sophisticated rule repositories. With the rapid growth of machine learning techniques, some machine learning and deep learning-based intrusion detectors were also proposed [28, 29, 44] and achieved satisfactory performance. However, these traditional intrusion detection methods either require a sophisticated, thorough and up-to-date rule repository, or a fully annotated training dataset. These prerequisites either require comprehensive expertise knowledge to build and update, or require a tremendous amount of efforts to annotate. Besides, due to the limited storage and communication capability of the IoT device and the concerns of user privacy, it further hinders the availability of an IoT intrusion rule repository or training dataset. Under such data-scarcity [39], these traditional intrusion detectors suffer from compromised performance.

To work around the data-scarcity, domain adaptation-based (DA) intrusion detection methods [17] can be leveraged by transferring the intrusion knowledge from a knowledge-rich source network intrusion (NI) domain to assist the intrusion detection for the target IoT intrusion (II) domain. Popular solutions [39, 42] performed intrusion knowledge transfer and meanwhile masked the heterogeneities between different domains and achieved satisfying outcomes.

Despite the effectiveness of these DA-based methods, they operate under the assumption that both source and target domains share exactly the same type of intrusions. However, this assumption is sometimes unrealistic in the real-world as the IoT intrusion domain can constantly confront newly-emerged intrusion strategies [23]. Therefore, this assumption hinders the applicability of traditional DA-based intrusion detectors. As a more general solution, Open-Set Domain Adaptation (OSDA) [10, 30] relaxes this assumption and allows the target domain to contain newly-emerged intrusions unobserved in the source NI domain. Some OSDA methods were proposed [14, 18, 21], which tackled this challenging setting via hyperspherical feature space learning, semantic recovery learning and progressive graph learning, and so on. However, these research efforts all suffered from some drawbacks, such as failing to utilise the graph embedding alignment in the learned hyperspherical feature space, lacking the exploitation of the correction effect of the semantics, and so on, which therefore provided room for improvement of a more effective OSDA-based intrusion detector.

In this article, inspired by the structure of the dandelion, we propose the Open-Set Dandelion Network (OSDN) based on unsupervised heterogeneous DA in an open-set manner. The OSDN model tackles the IoT data scarcity by transferring intrusion knowledge from a knowledge-rich

source NI domain to assist the knowledge-scarce IoT target domain. It relaxes the closed-set assumption and can effectively detect both known and unknown intrusions faced by IoT devices, making it applicable in real-world applications. To achieve this, the OSDN model forms the source domain into a dandelion-like feature space with the goal of grouping each intrusion category compactly and meanwhile separating different intrusion categories, i.e., achieving inter-category separability and intra-category compactness, the foundation for an accurate intrusion detector to work on. The dandelion-based target membership mechanism then constructs the target dandelion. Then, the dandelion angular separation mechanism is leveraged to enhance inter-category separability, together with the dandelion embedding alignment mechanism, which transfers intrusion knowledge via a graph embedding perspective. The discriminating sampled dandelion mechanism is also used to promote intra-category compactness. Besides, trained using both known and generated unknown intrusion knowledge, the intrusion classifier produces probabilistic semantics, which forms a semantic dandelion and in turn emphasises easily-confused categories and provide correction for better inter-category separability. Holistically, these mechanisms form the OSDN model that can effectively transfer intrusion knowledge for more accurate IoT intrusion detection.

In summary, the contributions of this article are three-fold as follows:

- We realise the benefits of the Open-Set DA technique to perform intrusion knowledge transfer and facilitate more accurate intrusion detection for the data-scarce IoT scenarios. The OSDA-based intrusion detector also relaxes the closed-set assumption, making it a more robust intrusion detector in the real-world.
- We formulate the intrusion feature space into a dandelion-like feature space. The proposed OSDN model leverages mechanisms such as the dandelion angular separation mechanism (DASM), the dandelion embedding alignment mechanism (DEAM), the discriminating sampled dandelion mechanism (DSDM) and the semantic dandelion correction mechanism (SDCM) to promote inter-category separability and intra-category compactness in the dandelion feature space, which is the foundation for an accurate intrusion detector to work on.
- We conduct comprehensive experiments on five widely recognised intrusion detection datasets and verify the effectiveness of the OSDN model against three state-of-the-art baselines. A 16.9% performance boost is achieved. Besides, the contribution of each OSDN constituting component, the stability and the efficiency of the OSDN model is also verified.

The rest of the article is organised as follows: Section 2 categorises related works and summarises the research opportunities. Section 3 presents model preliminaries and the OSDN model architecture, followed by Section 4, in which the detailed mechanisms constituting the OSDN model are presented. Section 5 presents the experimental setup and detailed experimental analyses. The last section concludes the article. We provide an acronym table (Table 5) and a notation table (Table 6 and 7) for better readability in Appendix A.

2 RELATED WORK

In this section, we introduce the related works in a categorised manner and outline our research opportunities. In Figure 1, we summarise the traditional IoT intrusion detection methods, their data dependency and their drawbacks, which reflect the merits of the domain adaptation-based intrusion detection methods for the data-scarce IoT scenarios. The OSDN method belongs to the open-set domain adaptation-based intrusion detector.

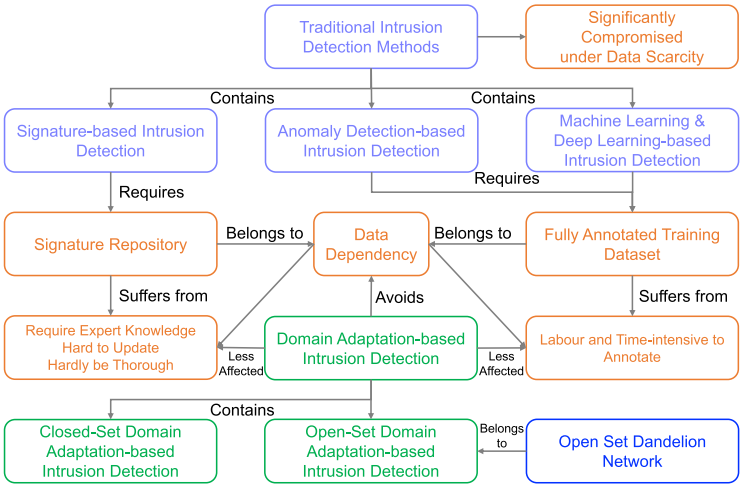

Fig. 1. Summarisation of IoT intrusion detection methods, the data dependency and drawback of traditional intrusion detection methods, and the merits of domain adaptation-based intrusion detection methods. The OSDN method belongs to the open-set domain adaptation-based intrusion detector.

2.1 Traditional Intrusion Detection

Intrusion detection has drawn wide attention from the research community. Traditional intrusion detection methods, including signature-based intrusion detectors [7, 25, 26], which require a sophisticated rule repository for decision-making. It can only detect malicious intrusions if their patterns match certain rules in the repository. Anomaly-based intrusion detectors [4, 5, 33, 37] are also popular. These methods need to go through a comprehensive training process based on a well annotated training dataset to learn the patterns of normal traffic behaviours and then flag any traffic that deviates from the normal patterns. With the rapid advance of machine learning and deep learning techniques, ML and DL-based intrusion detectors are also widely used. Possible methods include multi-kernel SVM [29], isolation forest [9] and deep learning models such as autoencoders [24, 28] and capsule network [44], and so on.

However, all these traditional intrusion detection methods may be hindered by the IoT data-scarcity due to their strong data dependency on a well-built intrusion rule repository or a finely-annotated training dataset. Building an intrusion rule repository requires sophisticated expertise knowledge, and can hardly be thorough and up-to-date. Besides, finely annotating a training dataset is both labour and time-intensive. Without enough annotated datasets, the learning process of anomaly-based, ML-based and DL-based methods is significantly hindered, resulting in compromised efficacy. Therefore, it naturally leads to the domain adaptation-based solutions, which can work under data-scarce IoT scenarios by performing intrusion knowledge transfer, a merit that outperforms traditional intrusion detection methods.

2.2 Domain Adaptation for Intrusion Detection

Domain adaptation can transfer intrusion knowledge from a knowledge-rich source domain to facilitate more accurate intrusion detection for the target domain. Hence, it possesses the merit to comfortably work under the data-scarce IoT scenario. Wu et al. [42] proposed a Joint Semantic Transfer Network, aiming to address the IoT intrusion detection problem under the semi-supervised heterogeneous DA setting. Later, the Geometric Graph Alignment method was also proposed by Wu et al. [39] to tackle the intrusion detection for completely unsupervised target

IoT domains. There are other DA methods such as [19, 41, 43], which performed intrusion knowledge transfer via Wasserstein distance minimisation, adversarial learning, Pareto optimal solution searching, adaptive recommendation matching, and so on.

However, the traditional domain adaptation methods work under the closed-set assumption that the intrusion categories in both source and target domains are exactly the same. Hence, these methods cannot tackle the case in which new IoT intrusions emerge as time goes by, limiting their applicability in the real-world.

2.3 Open-Set Domain Adaptation for Intrusion Detection

Open-Set DA methods relax the closed-set assumption of traditional DA methods and allow the target IoT domain to possess new intrusions unobserved in the source domain. Jing et al., [14] presented an open-set DA method with semantic recovery to better exploit the semantic information of the unknown target intrusions. However, it put no effort to explore the possibility brought by the hyperspherical structure formulation with excellent inter-category distinguishability. Li et al., [18] explored the open-set DA problem via the angular margin separation network. Despite its effectiveness, it lacked finer alignment achievable by graph embedding and ignored the correction effect of the semantics. Besides, Luo et al., [21] investigated the graph embedding-based open-set DA solution. However, the proposed Progressive Graph Learning (PGL) also failed to investigate the usefulness of angular-based hyperspherical space with excellent separability and compactness.

2.4 Research Opportunity

The OSDN model transfers intrusion knowledge via the dandelion-based feature space that emphasises both inter-category separability and intra-category compactness, which is lacked by previous open-set DA methods as in [14, 21]. Besides, the graph embedding alignment can achieve both finer feature space alignment and tighter intra-category structure via adversarial learning. Such mechanisms were not attempted in [14, 18]. Moreover, the OSDN model leverages the semantic dandelion correction mechanism, the utilisation of the semantic dandelion fills the void in [18]. The semantic correction is also lacked in these aforementioned methods. By combining these methods to form a holistic framework, the OSDN model can perform finer intrusion knowledge transfer and benefit IoT intrusion detection.

3 MODEL PRELIMINARY AND ARCHITECTURE

In this section, we introduce the preliminaries and the architecture of the proposed OSDN model.

3.1 Model Preliminary

The OSDN model works under the unsupervised open-set DA setting with heterogeneities exist between domains. Following common notations in [42], we denote the source NI domain $\mathcal{D}_S$ as follows:

$$\begin{aligned} &\mathcal{D}_S = \{\mathcal{X}_S, \mathcal{Y}_S\} = \{(x_{S_i}, y_{S_i})\}, i \in [1, n_S], \\ &x_{S_i} \in \mathbb{R}^{d_S}, y_{S_i} \in [1, K], \end{aligned} \tag{1}$$

where $\mathcal{X}_S$ contains n_S source NI domain traffic features, each feature vector is represented in d_S dimensions. $\mathcal{Y}_S$ is the corresponding intrusion category label within a total number of K categories, one normal category and others are intrusion categories. Similarly, the target II domain $\mathcal{D}_T$ is defined as follows:

$$\begin{aligned} &\mathcal{D}_T = \{\mathcal{X}_T, \mathcal{Y}_T\} = \{(x_{T_i}, y_{T_i})\}, i \in [1, n_T], \\ &x_{T_i} \in \mathbb{R}^{d_T}, y_{T_i} \in [1, K'], K' > K. \end{aligned} \tag{2}$$

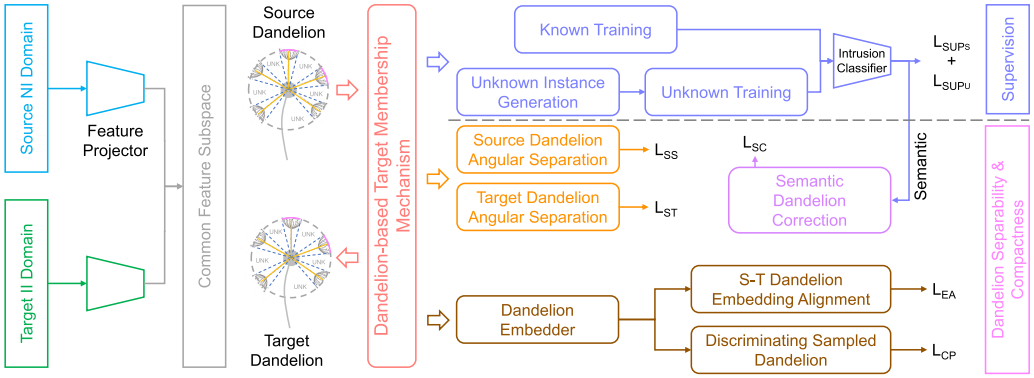

Fig. 2. The architecture of the OSDN model and the interrelationships between the OSDN's constituting components.

Under the open-set DA setting, the intrusion categories of the source NI domain is a subset of the intrusion categories of the target II domains, i.e., $\mathcal{Y}_S \subset \mathcal{Y}_T, K' > K$. Both domains share K common intrusion categories. Furthermore, the target II domain contains $K' - K$ new intrusion categories unobserved in the source domain. Under the unsupervised setting, the ground truth labels of the target II domain remain agnostic during the training process. As a heterogeneous DA problem, heterogeneities present between domains, e.g., $d_S \neq d_T$.

3.2 The OSDN Architecture

The architecture of the OSDN model has been presented in Figure 2. To perform intrusion knowledge transfer, features in each domain will be normalised to form a unit hyperspherical space and then be projected into a d_C-dimensional common feature subspace (the grey box) by its corresponding feature projector (the trapezoids). The feature projector E is defined as follows:

$$f(x_i) = \begin{cases} E_S(x_i) & \text{if } x_i \in \mathcal{X}_S \\ E_T(x_i) & \text{if } x_i \in \mathcal{X}_T \end{cases} \tag{3}$$
$$f(x_i) \in \mathbb{R}^{d_C}.$$

As illustrated in Figure 3, the common feature subspace aims to group each shared intrusion category in a compact manner (each pappus of the dandelion, i.e., intra-category compactness), and meanwhile achieves excellent separability between intrusion categories, i.e., inter-category separability. For these unknown new intrusions in the target II domain, since their number is agnostic, therefore, instead of deliberately grouping them in a brute-force manner, the dandelion-analogous common feature subspace allows them to spread in any gap between pappuses to promote distinguishability between shared and unknown intrusion categories. As visualised in Figure 3, by making the common feature subspace analogous to the structure of the dandelion, i.e., achieving excellent intra-category compactness and inter-category separability, the shared classifier C can then make accurate intrusion detection decisions.

The source dandelion can be formed directly since the source domain is completely supervised. Then, the dandelion-based target membership mechanism (the red box in Figure 2) is used to form the target dandelion based on the spatial relationship between target instances and the source dandelion. Once both source and target dandelions are formed, the dandelion angular separation mechanism (the orange boxes in Figure 2) is utilised to enhance the inter-category separability in each dandelion. Besides, a dandelion embedder is leveraged to generate graph embeddings for

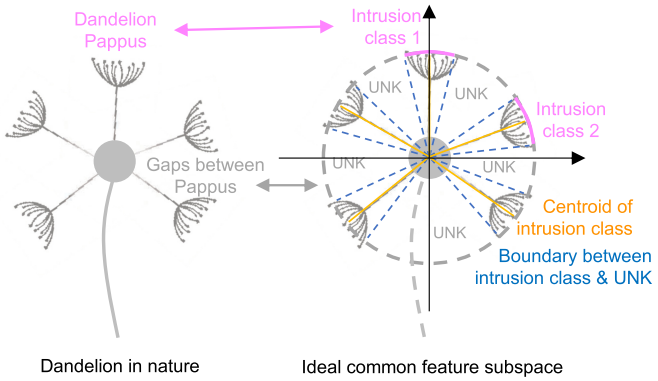

Fig. 3. The analogy between the structure of a dandelion and the dandelion-like common feature subspace. Each pappus corresponds to a shared intrusion category, it needs to be compact and well-separated from other intrusion categories (pappuses), simultaneously achieving both intra-category compactness and inter-category separability. Target unknown intrusion categories reside in the gaps between pappuses to achieve distinguishability. The analogy between dandelion and the ideal common feature subspace leads to the naming of the OSDN.

dandelions and it is used in two ways (the brown boxes in Figure 2): the graph embeddings of source and target dandelions are aligned to promote better alignment between domains; moreover, sampled child dandelions are produced and their graph embeddings need to confuse a discriminator to achieve finer intra-category compactness. To better train the shared intrusion classifier C, the source domain data provides supervision information. To equip the intrusion classifier with knowledge of target unknown intrusions, unknown instances residing in the pappus gaps in the source dandelion are generated for unknown intrusion training. Lastly, the probabilistic semantic yielded by the shared intrusion classifier also works as a correction to deliberately emphasise easily-confused categories and remind the dandelion angular separation mechanism to separate them, forming a correction loop (the purple box in Figure 2).

Finally, by forming these mechanisms into a holistic model, fine-grained intrusion knowledge transfer can be achieved and the shared and unknown intrusion categories will be well-separated so that the shared classifier C can enjoy excellent intrusion detection efficacy for the target II domain.

4 THE OSDN ALGORITHM

In this section, we present the detailed mechanism of each OSDN constituting component and the overall optimisation objective of the model.

4.1 Dandelion-based Target Membership Mechanism (DTMM)

The source dandelion can be easily formed based on its supervision information. Then, the source dandelion will guide the membership decision for unsupervised target instances to form the target dandelion. For each source intrusion category i, the maximum intra-category deviation d^i_{max} will be calculated as follows:

$$\begin{aligned} d^{(i)}_{max} &= \max\left(1 - \mathrm{COS}\left(x^{(i)}_{S_j}, \mu^{(i)}_S\right)\right), j \in \left[1, n^{(i)}_S\right], \\ \mu^{(i)}_S &= \frac{1}{n^{(i)}_S} \sum_{j=1}^{n^{(i)}_S} x^{(i)}_{S_j}, \end{aligned} \tag{4}$$

where $COS()$ stands for Cosine Similarity, $n_S^{(i)}$ denotes the number of instances in the ith intrusion category in the source domain, $\mu_S^{(i)}$ denotes the mean of the source intrusion category i and $x_{S_j}^{(i)}$ means the jth instance of the source ith intrusion category. Then, each target instance will be assigned to its nearest source category i if it resides within the maximum deviation range of source category i, i.e.,

$$y_{T_j}^D = \begin{cases} \underset{i}{\operatorname{argmin}}(1 - COS(x_{T_j}, \mu_S^{(i)})) & \text{if } 1 - COS(x_{T_j}, \mu_S^{(i)}) \le d_{max}^{(i)} \\ K+1 & \text{otherwise}, \end{cases} \tag{5}$$

where $y_{T_j}^D$ represents the dandelion-based membership for the jth target instance x_{T_j}. Otherwise, that target instance will be assigned to the unknown category $K+1$ to avoid deteriorating the compactness of its closest intrusion category. Unlike methods such as [14, 18] that perform K-means clustering of unknown intrusions, the OSDN assigns all unknown intrusions into a single category $K+1$ and hence does not rely on the availability of the prior knowledge on the number of unknown intrusion categories and is more practical in the real-world. Besides, the OSDN model does not deliberately enforce all unknown target instances to reside at a single place, it allows unknown intrusions to reside at any pappus gap in the target dandelion. Deliberately aligning unknown target instances coming from different intrusion categories may cause negative transfer.

4.2 Dandelion Angular Separation Mechanism (DASM)

To increase the separability between known intrusion categories and meanwhile enhance the discriminability between known and unknown intrusion categories, i.e., enlarge the gap between pappuses, the OSDN model will achieve these goals from an angular perspective. First, the centroid of each intrusion category will be calculated. Then, the source category pair-wise Cosine similarity matrix CS_S will be calculated as follows:

$$\begin{aligned} CS_S &= \begin{bmatrix} CS_S^{11} & CS_S^{12} & \cdots & CS_S^{1K} \\ CS_S^{21} & CS_S^{22} & \cdots & CS_S^{2K} \\ \vdots & \vdots & \ddots & \vdots \\ CS_S^{K1} & CS_S^{K2} & \cdots & CS_S^{KK} \end{bmatrix}, \\ CS_S^{ij} &= COS\left(\mu_S^{(i)}, \mu_S^{(j)}\right), \end{aligned} \tag{6}$$

where CS_S^{ij} represent the Cosine similarity between the ith and jth intrusion category of the source NI domain. By minimising the sum of the upper triangle of the matrix CS_S, it enlarges the inter-category angular divergence. The source dandelion separation loss $\mathcal{L}_{SS}$ is defined as follows:

$$\mathcal{L}_{SS} = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} CS_S^{ij}. \tag{7}$$

The target dandelion Cosine similarity matrix CS_T and the corresponding target dandelion separation loss $\mathcal{L}_{ST}$ are defined similarly. By minimising both $\mathcal{L}_{SS}$ and $\mathcal{L}_{ST}$, it promotes better dandelion inter-category separability from an angular perspective.

4.3 Dandelion Embedding Alignment Mechanism (DEAM)

To further promote a finer alignment between the source and target dandelions, a dandelion graph embedder is used to produce the graph embeddings for both dandelions. To achieve this, each

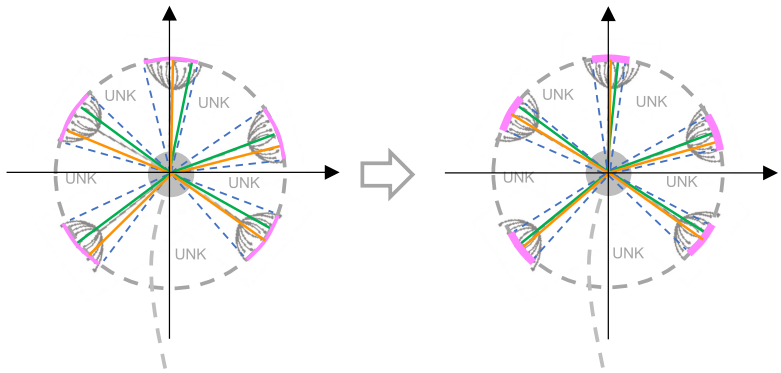

Fig. 4. Illustrating example of the OSDN discriminating sampled dandelion mechanism to enhance intra-category compactness.

dandelion is formulated as a graph, defined as follows:

$$\begin{aligned} G_S &= < V_S, E_S > \\ V_S &= \left\{ V_S^{(i)} \right\}, i \in [1, K], V_S^{(i)} = \mu_S^{(i)} \\ E_S &= \left\{ E_S^{i,j} \right\}, E_S^{i,j} = \left\| \mu_S^{(i)} - \mu_S^{(j)} \right\|_2^2, i \in \{\mathfrak{p}\} \cup [1, K], j \in \{\mathfrak{p}\} \cup [1, K], i \neq j, \end{aligned} \tag{8}$$

where G_S denotes the source dandelion graph, V_S and E_S stand for vertices and edges in G_S, respectively and G_T is defined similarly. Each vertex $V_S^{(i)}$ is the centroid of the corresponding intrusion category. The graph is fully connected and each vertex is also connected with the origin, denoted as $\mathfrak{p}$.

In the OSDN model, we apply the Feather network [32] as the graph embedder. As a graph embedding algorithm, it enjoys several merits: first, the Feather network can work in an unsupervised manner, which works comfortably under the data-scarce IoT scenario; second, the Feather network enjoys a linear time complexity as proved in [32], the low complexity can enhance the efficiency of the intrusion detection model in real-world applications; finally, the Feather network is comprehensively verified [32] to have superior graph embedding performance.

Using the graph embedder, each dandelion graph will be mapped into a d_G-dimensional graph embedding space, in which the more geometrically similar between dandelion graphs, the more similar the graph embeddings will be. Then, the dandelion embedding alignment loss $\mathcal{L}_{EA}$ is defined as follows:

$$\mathcal{L}_{EA} = ||\phi_S - \phi_T||_2^2, \phi_S, \phi_T \in \mathbb{R}^{d_G}, \tag{9}$$

where ϕ_S denotes the graph embedding of the source domain dandelion. By minimising the dandelion embedding alignment loss, both dandelions will be further aligned and hence will promote better intrusion knowledge transfer, as verified by experimental evidences in Section 5.6.

4.4 Discriminating Sampled Dandelion Mechanism (DSDM)

To further boost the intra-category compactness and hence promote better known-intrusion separability and unknown-intrusion discriminability, a discriminating sampled dandelion mechanism is proposed. As illustrated in Figure 4, one instance per intrusion category is randomly sampled to form a new child dandelion, such as the orange and the green dandelion in Figure 4. The more compact each intrusion category is, the more similar the embeddings of child dandelions will be. Hence, the OSDN achieves this goal via a discriminating perspective. First, both source and target

domain intrusion features will be fused in the common feature subspace to form a fused dandelion, then, N child dandelions will be sampled, where the ith pappus in each child dandelion is a randomly selected instance from the ith category from the fused dandelion. Next, a discriminator is confused using the discriminating sampled dandelion loss $\mathcal{L}_{CP}$, defined as follows:

$$\mathcal{L}_{CP} = \frac{1}{2}(log(D(\phi_S)) + log(D(\phi_T))) + \frac{1}{N}\sum_{j=1}^{N}\left(1 - log\left(D\left(\phi^j_{\mathcal{DD}_*}\right)\right)\right), \tag{10}$$

in which $\mathcal{DD}_S$, $\mathcal{DD}_T$, and $\mathcal{DD}^j_*$ denote the source, target and jth sampled dandelion, ϕ denotes the dandelion graph embedding and $D()$ denotes the discriminator. By assigning $\mathcal{DD}_S$ and $\mathcal{DD}_T$ with label 1 and assign sampled child dandelions with label 0, letting the network to minimise the $\mathcal{L}_{CP}$ will confuse the discriminator to be incapable to distinguish whether the given dandelion embedding is generated from a randomly sampled dandelion or not. Meanwhile, the discriminator will try to stay unconfused. Once the minimax game between the network and the discriminator reaches an equilibrium, the graph embeddings of source, target and sampled child dandelions will become indistinguishable, which in turn enhances the intra-category compactness, as illustrated in Figure 4.

4.5 Semantic Dandelion Correction Mechanism (SDCM)

The source NI domain is completely supervised, however, it lacks the knowledge of unknown intrusions in the target II domain. Therefore, directly using the source NI domain supervision to train the shared intrusion classifier C will significantly hinder its ability to detect unknown intrusions. To tackle this issue, the OSDN model generates n_R instances residing in the gaps between source dandelion pappuses, and treat these generated instances as unknown intrusions to equip the intrusion classifier C with the ability to detect both known and unknown intrusions under the open-set DA setting. The overall supervision loss of known and unknown training $\mathcal{L}_{SUP}$ is defined as follows:

$$\begin{aligned}
\mathcal{L}_{SUP} &= \mathcal{L}_{SUP_S} + \mathcal{L}_{SUP_U} \\
&= \frac{1}{n_S}\sum_{j=1}^{n_S}\mathcal{L}_{CE}(C(f(x_j)), y_j) + \frac{1}{n_R}\sum_{j=1}^{n_R}\mathcal{L}_{CE}(C(f(x_j)), y_j) \\
y_j &= \begin{cases} y_{S_j} & \text{if } x_j \in \mathcal{X}_S \\ K+1 & \text{if } x_j \in \mathcal{X}_R \end{cases},
\end{aligned} \tag{11}$$

where $\mathcal{L}_{CE}$ denotes the cross entropy loss and $\mathcal{X}_R$ represents generated unknown instances for unknown training.

Once the intrusion classifier C is well-trained, it can then yield probabilistic semantics for each intrusion data instance j, i.e., the inter-category probabilistic correlations, denoted as p_j. Therefore, the semantic information can also form new semantic dandelions $\mathcal{DD}_{S*}$ in the semantic space, defined as follows:

$$\mathcal{DD}^{(i)}_{\mathcal{SS}} = \frac{1}{n^{(i)}_S}\sum_{j=1}^{n^{(i)}_S} p^{(i)}_{S_j}, \mathcal{DD}^{(i)}_{\mathcal{ST}} = \frac{1}{|y^D_T = i|}\sum_{j=1}^{|y^D_T=i|} p^{(i)}_{T_j}, \tag{12}$$

where $\mathcal{DD}^i_{\mathcal{SS}}$ denotes the ith pappus of the source semantic dandelion $\mathcal{DD}_{\mathcal{SS}}$, $n^{(i)}_S$ represents the number of source ith category instances, y^D_T denotes the membership assigned to target instances by the source dandelion in Section 4.1. Then, the Cosine similarity matrix CS_{SM} between both

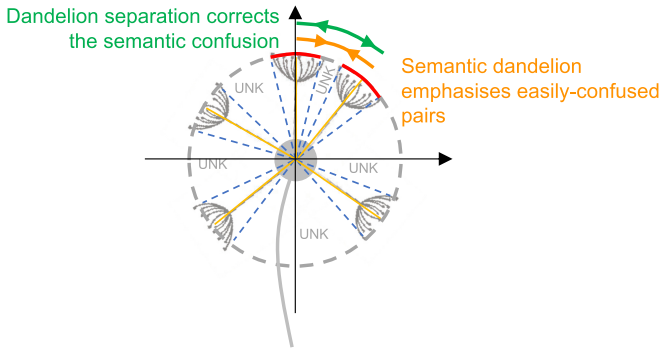

Fig. 5. The OSDN semantic dandelion correction mechanism. It will point out easily confused intrusion category pairs from the probabilistic semantic perspective (the orange part), which will act as a correction to the dandelion angular separation mechanism (the green part).

semantic dandelions are calculated as follows:

$$CS_{SM} = \begin{bmatrix} CS_{SM}^{11} & CS_{SM}^{12} & \cdots & CS_{SM}^{1K} \\ CS_{SM}^{21} & CS_{SM}^{22} & \cdots & CS_{SM}^{2K} \\ \vdots & \vdots & \ddots & \vdots \\ CS_{SM}^{K1} & CS_{SM}^{K2} & \cdots & CS_{SM}^{KK} \end{bmatrix}, \tag{13}$$
$$CS_{SM}^{ij} = COS(\mathcal{DD}_{SS}^{(i)}, \mathcal{DD}_{ST}^{(j)}).$$

Ideally, the ith intrusion category from both source NI and target II domain should share similar inter-category probabilistic semantics, while different intrusion categories from both domains should have their inter-category probabilistic semantics diverge from each other. To achieve this, the OSDN model minimises the semantic dandelion correction loss $\mathcal{L}_{SC}$ as follows:

$$\mathcal{L}_{SC} = \frac{2}{K(K+1)} \sum_{i=1}^{K} \sum_{j=i}^{K} CS_{SM}^{ij}. \tag{14}$$

By minimising the $CS_{SM}^{ij}, i \neq j$, inter-category probabilistic semantics will be diverged from each other, leading to better inter-category discriminability. It is worth noting that the $\mathcal{L}_{SC}$ also minimises the CS_{SM}^{ii}, i.e., maximising the divergence between cross-domain same-category probabilistic semantics. The rationale is as follows: if minimising the CS_{SM}^{ii} can easily compromise the semantic of the ith intrusion category, then it indicates the ith intrusion category can be easily confused with other categories from the probabilistic semantic perspective, as indicated in Figure 5. Therefore, deliberately minimising CS_{SM}^{ii} can exploit and emphasise easily-confused intrusion category pairs, i.e., pointing out a possible point to correct for the dandelion angular separation mechanism. Consequently, by utilising this correction mechanism, it can further boost the dandelion separation efficacy, as supported by experimental evidences in Section 5.6 and in turn enhances the intrusion detection accuracy.

4.6 Overall Optimisation Objective

Overall, the optimisation objective of the OSDN model is defined as follows:

$$\begin{aligned} &\min_{E_S, E_T, C} (\alpha_S \mathcal{L}_{SUP_S} + \alpha_U \mathcal{L}_{SUP_U} + \beta_S \mathcal{L}_{SS} + \beta_T \mathcal{L}_{ST} + \delta \mathcal{L}_{EA} + \theta \mathcal{L}_{SC} + \gamma \mathcal{L}_{CP}) \\ &\max_{D} (\gamma \mathcal{L}_{CP}), \end{aligned} \tag{15}$$

where $\alpha_S, \alpha_U, \beta_S, \beta_T, \delta, \theta$ and γ are hyperparameters controlling the influence of the corresponding loss components. We utilise the gradient reversal layer [11] for the discriminator, which acts as an identity function during forward propagation and reverses the gradient during backpropagation to achieve an end-to-end optimisation process for the OSDN model. Once the above minimax game reaches an equilibrium, the intrusion knowledge is transferred in a fine-grained manner, and the intrusion detection efficacy can therefore benefit.

5 EXPERIMENT

To verify the effectiveness of the OSDN model, we perform experiments on five comprehensive and representative intrusion detection datasets with three state-of-the-art baseline counterparts. We also verify the performance stability of the OSDN model under varied openness settings and manipulated hyperparameter settings and demonstrate the contribution and necessity of each OSDN constituting component. Finally, we verify the computational efficiency of the OSDN model.

5.1 Experimental Datasets

We use five comprehensive intrusion detection datasets. Network intrusion detection datasets include NSL-KDD, UNSW-NB15 and CICIDS2017. IoT intrusion detection datasets include UNSW-BOTIOT and UNSW-TONIOT.

Network Intrusion Dataset: NSL-KDD This dataset [36] contains benign network traffic and four types of real-world intrusions, such as probing attacks, Denial of Service (DoS) attacks, and so on. It enjoys excellent data quality compared with its previous version [13]. We follow [2] to use a reasonable amount of 20% of the dataset during experiments. Following [12], we use the top-31 most informative features out of 41 features as the feature representation and denote the dataset as *K*.

Network Intrusion Dataset: UNSW-NB15 The dataset [27] was released in 2015 and was constructed on a comprehensive security testing platform commonly used by the industry. It includes normal network traffic with nine categories of modern intrusion patterns, such as DoS attack, reconnaissance attack, and so on, and possesses high data quality. We perform data preprocessing to remove four features out of the original 49 features that have a value of 0 for nearly all records. We denote the dataset as *N*.

Network Intrusion Dataset: CICIDS2017 This dataset [34] was released in 2017 and contained up-to-date intrusion trends that include seven intrusion categories, represented in 77 dimensions. We use 20% of the dataset provided by its creator, and perform preprocessing steps such as categorical-numerical data conversion. We follow [35] to use the top-40 most informative features, and denote the dataset as *C*.

IoT Intrusion Dataset: UNSW-BOTIOT This dataset [15] was released in 2017. It is constructed on a realistic testbed involving commonly-used IoT devices such as the weather station, smart fridge, and so on, and utilises the common lightweight IoT communication protocol MQTT. The dataset contains four up-to-date intrusion categories, represented in 46 dimensions. We follow the advice from the dataset creator to use the top-10 most informative features. The dataset is denoted as *B*.

IoT Intrusion Dataset: UNSW-TONIOT The dataset [3] was released in 2021 and involved up-to-date IoT protocols and standards. The testbed used is sophisticated, with seven types of real IoT devices such as the GPS tracker, the weather meter, and so on, and capturing heterogeneous features. The dataset contains nine types of common IoT intrusions [1], such as the DoS attack, scanning attack, and so on. We follow [31] to leverage 10% of the dataset, and select two IoT devices, i.e., the GPS tracker and the weather meter, denoted as *G* and *W*, respectively.

Dataset Comprehensiveness and Intrusion Methods The datasets used during experiments are comprehensive and representative. First, these datasets are widely recognised by the intrusion detection research community with a broad range of usage. Second, these datasets are recently released and contain modern intrusion trends and patterns, some of them are released in 2021. Third, these datasets all involve widely recognised testbeds. The IoT datasets also involve real-world IoT devices deployed in a real-world environment. Finally, the network and IoT datasets have at most eight shared intrusion categories, with a coverage of 100%, 55%, 100%, 100% and 98% on NSL-KDD, UNSW-NB15, CICIDS2017, UNSW-BOTIOT and UNSW-TONIOT, respectively. The transferrable intrusion knowledge reflects modern intrusion trends. Hence, the datasets used are sufficient to verify the effectiveness of the OSDN model.

5.2 Implementation Details

We implement the OSDN model using the deep learning framework PyTorch. The feature projectors are implemented as a single-layer neural network and use LeakyRelu as the activation function. Likewise, both the intrusion classifier C and the discriminator D are also implemented as single-layer neural networks.

We apply cross validation with grid search to tune hyperparameters. Since all experiments share a single set of hyperparameter settings, the tuning effort is not too laborious. The default hyperparameter settings are as follows: $\alpha_S = 0.8$, $\alpha_U = 0.1$, $\beta_S = \beta_T = 0.75$, $\delta = 0.001$, $\theta = 1.0$, $\gamma = 1.0$, number of sampled dandelions $N = 10$ and number of sampled unknown instances $n_R = 100$. Additionally, the stability and robustness of the OSDN model with manipulated hyperparameters in their corresponding reasonable ranges are also verified in Section 5.8.

During evaluation, we follow [42] to use accuracy, category-weighted precision (P), recall (R) and F1-score (F) as evaluation metrics. Their definitions are as follows:

$$Accuracy = \frac{\sum_{k=1}^{K}(TP^{(k)} + TN^{(k)})}{n_T}, \tag{16}$$

$$Precision = \sum_{k=1}^{K} \frac{|\mathcal{X}_T^{(k)}|}{n_T} \cdot Precision^{(k)} = \sum_{k=1}^{K} \frac{|\mathcal{X}_T^{(k)}|}{n_T} \cdot \frac{TP^{(k)}}{TP^{(k)} + FP^{(k)}}, \tag{17}$$

$$Recall = \sum_{k=1}^{K} \frac{|\mathcal{X}_T^{(k)}|}{n_T} \cdot Recall^{(k)} = \sum_{k=1}^{K} \frac{|\mathcal{X}_T^{(k)}|}{n_T} \cdot \frac{TP^{(k)}}{TP^{(k)} + FN^{(k)}}, \tag{18}$$

$$F1 = \sum_{k=1}^{K} \frac{|\mathcal{X}_T^{(k)}|}{n_T} \cdot \frac{2 \cdot Precision^{(k)} \cdot Recall^{(k)}}{Precision^{(k)} + Recall^{(k)}}, \tag{19}$$

where the true positive $TP^{(k)}$ denotes the number of category k intrusions being correctly detected, similar for $TN^{(k)}$, $FP^{(k)}$ and $FN^{(k)}$. During experiments, we evaluate the performance in two modes: the ACC mode which evaluates the prediction with the corresponding ground truth intrusion label, and the IND mode, which treats all known and unknown intrusions as a single intrusion class.

As an open-set DA method, following Kundu et al. [16], we define openness O as follows:

$$O = 1 - \frac{K}{K'}. \tag{20}$$

The openness O lies in the range between 0 and 1, the larger the openness is, the more unknown classes will be in the target II domain.

5.3 State-of-the-art Baselines

We use three state-of-the-art baseline methods to verify the superiority of the OSDN model, which include AMS [18], SR-OSDA [14], and PGL [21]. The AMS method attempts the OSDA problem by formulating a framework with four phases. In phase 1, a discriminative representation of seen classes is learned to benefit the seen and unseen intrusion separation performed in the second phase. After performing the seen and unseen separation and the target domain is pseudo-labelled, the phase 3 further optimises the feature representation. Both phase 2 and 3 also form an iterative loop, which gradually improves the quality of intrusion recognition quality. Finally, phase 4 learns a re-projection, which promotes the generalisability of unseen intrusion recognition without sacrificing the ability to correctly recognise the seen classes. The SR-OSDA method deals with the OSDA problem by firstly separating seen and unseen intrusion instances progressively via a threshold-based pseudo-label assignment mechanism and the K-means clustering. Then, the intrusion knowledge transfer is performed by mapping both domains into a domain-invariant and discriminative feature space. Finally, the semantic information is utilised to better exploit the unknown target intrusions, so that they are not deliberately confounded together, which causes negative transfer. The PGL method integrates a graph neural network with the episodic training strategy and meanwhile applies adversarial learning to bridge the gap between two intrusion domains. In the episodic training strategy, the model progressively enlarges the labelled set via pseudo-labelling and utilise the pseudo-labelled target samples for episodic training. On top of it, the graph neural network is benefitted to perform more accurate intrusion detection. We summarise their differences with the OSDN model as follows:

— From the dandelion-based feature space perspective, the AMS method attempts this direction. However, it lacks other mechanisms such as the graph embedding-based dandelion alignment and the dandelion compactness enhancement, and also fails to form the semantic dandelion and explore its correction effect.
— From the graph embedding perspective, the PGL method utilises the graph embedding during knowledge transfer. However, the PGL completely ignores the benefit brought by utilising the graph embedding in a dandelion-based feature space.
— From the semantic alignment perspective, all these methods lack effort to build a semantic hyperspherical space to guide the inter-category separation in the dandelion-based feature space, leaving a void to be filled.

Therefore, these state-of-the-art methods are comparable and representative to verify the effectiveness of the OSDN model.

5.4 Intrusion Detection Performance

The intrusion detection accuracy of nine randomly selected tasks with varied openness has been presented in Table 1. As we can observe, the OSDN model outperforms other baseline counterparts by a large margin, achieving a 20.9% and 12.9% performance improvement under two modes, respectively. We also measure the intrusion detection performance using three other metrics and present the results in Figure 6. Under both modes, the OSDN model is positioned at the top-right corner in all three tasks, indicating that the OSDN model achieves the best precision and recall performance compared with other methods, and hence it is natural to observe the best F1-score is also yielded by the OSDN model. The best precision performance indicates the highest amount of intrusions flagged by the OSDN model are correct, while the best recall performance demonstrates the OSDN model can successfully flag as many intrusions as possible. As a harmonic mean of precision and recall, the best F1-score performance further verifies the OSDN model can elegantly balance between flagging as many intrusions as possible and simultaneously avoid triggering too

Table 1. The Intrusion Detection Accuracy Results

Tasks	K→G, $\mathcal{O}$ = 0.6		N→W, $\mathcal{O}$ = 0.4		C→W, $\mathcal{O}$ = 0.5		K→B, $\mathcal{O}$ = 0.5		K→G, $\mathcal{O}$ = 0.2	
Methods	ACC	IND	ACC	IND	ACC	IND	ACC	IND	ACC	IND
AMS	42.90	54.26	36.02	58.30	42.70	58.98	42.12	62.48	44.02	57.14
SRODSA	44.02	57.15	34.32	57.28	37.25	57.36	43.13	62.02	43.56	55.78
PGL	40.42	57.17	43.85	58.38	45.63	62.18	42.80	59.52	39.95	57.14
OSDN	**76.18**	**89.94**	**61.79**	**64.51**	**59.20**	**63.10**	**53.78**	**67.42**	**75.83**	**90.11**
Tasks	K→W, $\mathcal{O}$ = 0.71		C→B, $\mathcal{O}$ = 0.5		C→W, $\mathcal{O}$ = 0.66		C→W, $\mathcal{O}$ = 0.33		Average	
Methods	ACC	IND	ACC	IND	ACC	IND	ACC	IND	ACC	IND
AMS	38.36	59.44	49.26	65.92	41.44	57.98	42.98	59.22	42.20	59.30
SRODSA	36.24	57.88	49.56	66.33	38.44	56.36	38.78	58.46	40.59	58.74
PGL	34.52	46.89	51.53	67.68	39.92	60.10	45.05	61.41	42.63	58.94
OSDN	**75.05**	**78.31**	**56.22**	**69.56**	**57.32**	**62.31**	**56.39**	**64.94**	**63.53**	**72.24**

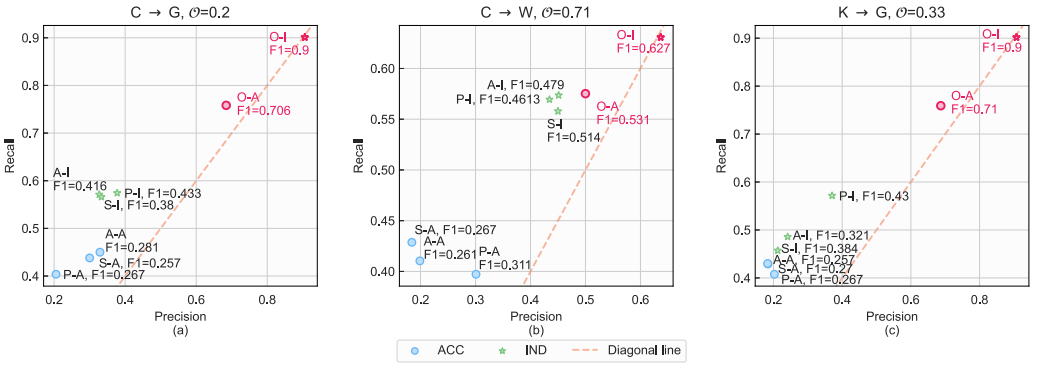

Fig. 6. Precision, Recall and F1-Score performance on three tasks under two modes. A-A denotes the performance of method AMS under ACC mode. PGL, SR-OSDA and OSDN (Ours) are denoted as P, S and O, respectively. The X-axis and Y-axis represent precision and recall, respectively. The F1-score is marked as text in the diagram. The red diagonal line marks $f(x) = x$.

many false alarms. The same result is also verified by the OSDN's nearest proximity from the red diagonal line among all methods as shown in Figure 6. Hence, it demonstrates the real-world applicability of the OSDN model as an intrusion detector.

5.5 Robustness and Stability under Varied Openness

We first present the performance of the OSDN model and its baseline counterparts under varied openness in Figure 7. We can observe that the OSDN model stably outperforms its baseline counterparts under varied openness evaluated using both accuracy and F1-score. Besides, compared with other baseline methods, the OSDN model shows a flatter trend with less severe fluctuation. Hence, it demonstrates the robustness of the OSDN method under varied openness levels.

We further evaluate the OSDN model against two more tasks under both large and small openness ranges. The results are shown in Figure 8. The task in Figure 8(a) and (b) has a relatively higher openness range and the task in Figure 8(c) and (d) presents a relatively lower openness range. From both Figure 7 and Figure 8, the OSDN model maintains a relatively stable trend without heavy fluctuation even when the openness varied significantly. Therefore, the OSDN's capability to detect

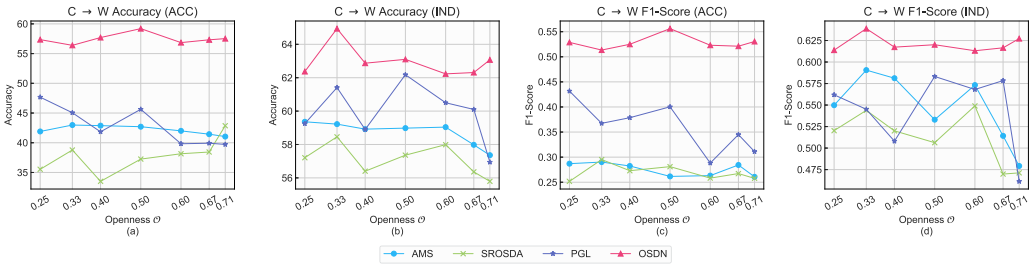

Fig. 7. Intrusion detection accuracy and F1-score performance under varied openness levels. The accuracy results under two modes are shown in (a)–(b). The F1-score results under two modes are shown in (c)–(d).

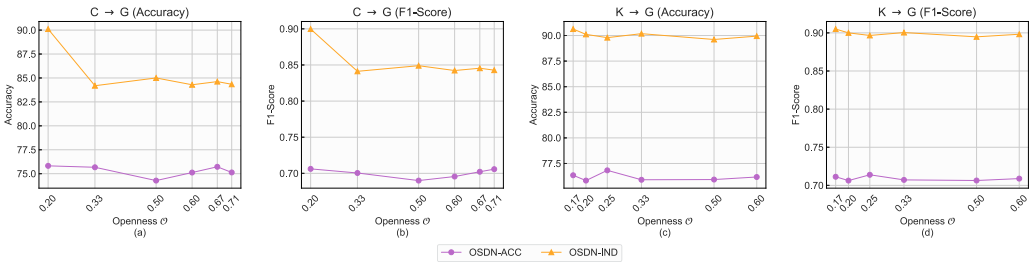

Fig. 8. Intrusion detection accuracy and F1-score performance for two tasks under large and small openness levels in (a)–(b) and (c)–(d), respectively.

unknown intrusions in the target II domain under varied openness levels is verified and can enhance its real-world usefulness.

5.6 Ablation Study

To verify the positive contribution and the necessity of each constituting component of the OSDN model, six groups of ablation studies are performed and the corresponding results are demonstrated in Table. 2. In the ablation group *A*, the unknown training mechanism is ablated, which causes the accuracy to drop around 5.3% and 3.6% under two modes, respectively. In the ablation group *B*, either the source or target dandelion angular separation mechanism (B_1 and B_2), or both of them (B_3) are dropped. As we can observe, lacking any of the DASM will result in a significant performance reduction, hence it verifies the necessity of the DASM for both domains. Besides, using the DASM for only one domain dandelion will further deteriorate the intrusion detection efficacy. The reason is that when both DASMs are turned off, other mechanisms such as the semantic dandelion correction and discriminating sampled dandelion mechanism will still partially achieve the dandelion separation effect. However, only using a single DASM will end up with a severe dandelion misalignment. Hence, worse performance is observed for the ablation groups B_1 and B_2.

The dandelion embedding alignment mechanism is removed in the ablation group *C*. Without it, the performance drops by 10.5% and 6.6% under two modes, respectively. A heavier performance drop is observed in the ablation group *D*, in which the semantic dandelion correction mechanism is eliminated. Without this mechanism, there will be no semantic-assisted correction for under-separated intrusion categories, resulting in compromised intrusion detection efficacy. In the ablation group E_1, the discriminating sampled dandelion mechanism is turned off, and in the ablation group E_2, the traditional instance domain discriminator substitutes the proposed DSDM. As we

Table 2. Ablation Study Results for Five Ablation Study Groups

Group	Experiment Setting		N→W, $\mathcal{O}$ = 0.40		C→G, $\mathcal{O}$ = 0.71		K→W, $\mathcal{O}$ = 0.71		Average	
			ACC	IND	ACC	IND	ACC	IND	ACC	IND
A	$\alpha_U = 0$		54.87	60.45	72.36	82.97	69.00	73.06	65.41	72.16
B	$\beta_{\mathcal{S}} = 0$	$\beta_{\mathcal{T}} = 0$								
B1	×	✓	52.62	59.82	68.51	80.45	60.24	67.08	60.46	69.12
B2	✓	×	55.29	60.49	66.96	80.25	63.31	68.37	61.85	69.70
B3	×	×	54.47	61.44	72.20	83.72	67.10	72.23	64.59	72.46
C	$\delta = 0$		55.89	62.22	67.34	80.05	57.27	65.24	60.17	69.17
D	$\theta = 0$		45.15	58.35	58.10	77.02	48.22	58.86	50.49	64.74
E	Discriminating Strategy									
E1	$\gamma = 0$		56.55	61.61	67.21	78.43	59.89	66.71	61.22	68.92
E2	Domain Adv		54.13	60.03	72.55	83.69	60.03	66.70	62.24	70.14
F	No DA		44.22	57.26	42.87	60.60	43.16	57.13	43.42	58.33
Full	$\alpha_U = 0.1, \delta = 0.001$ $\beta_{\mathcal{S}} = \beta_{\mathcal{T}} = 0.75$ $\gamma = 1.0, \theta = 1.0$		**61.79**	**64.51**	**75.13**	**84.34**	**75.05**	**78.31**	**70.66**	**75.72**

can see, completely lacking the adversarial learning significantly hinders the intrusion detection performance, which yields a 9.4% and 6.8% performance reduction under two modes, respectively. Although the substituting domain adversarial learner slightly increases the performance compared with the ablation group E_1, it still presents a performance that is much lower than the full OSDN model.

Finally, in the ablation group *F*, the domain adaptation mechanism is completely turned off to verify that the HDA mechanism plays an indispensable role. As we can see, removing the intrusion knowledge transfer performed by the HDA mechanism significantly degrades the intrusion detection performance, which is the worst among all ablated groups. Therefore, it justifies the necessity of having the HDA mechanism, and shows that the HDA mechanism makes a non-negligible contribution towards more accurate IoT intrusion detection.

Overall, the full OSDN model outperforms all its ablated counterparts by a significant margin, which indicates that all constituting components of the OSDN model contribute positively towards finer intrusion knowledge transfer and hence are indispensable for achieving excellent intrusion detection performance.

We further verify the statistical significance of each component's contribution via the significance T-test with the significance threshold of 0.05. The results are presented in Figure 9. The grey area in the middle stands for the significance threshold $-log(0.05)$. Among each dimension of the radar chart, the higher the value is, the more statistically significant the contribution is for that corresponding component. As we can observe, under all three tasks and all two modes, the coloured areas present a wider coverage than the grey shaded area. The results verify the statistical soundness of all components' contributions.

5.7 Separability and Compactness Analysis

The ideal common feature subspace should have the inter-category divergence as large as possible to achieve good separability and meanwhile have the intra-category variation as small as possible to achieve compactness. To verify the constituting components of the OSDN model contribute positively towards these goals, we follow Equation (7) to calculate the separability from an angular

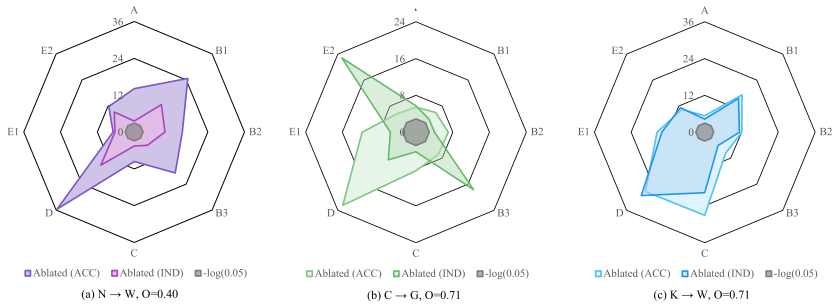

Fig. 9. Hypothesis testing results under the significance threshold of 0.05 to verify the statistical significance of the contribution made by each OSDN constituting component. For better visualisation, we omit ablation group F due to its significant differences compared with the full OSDN method.

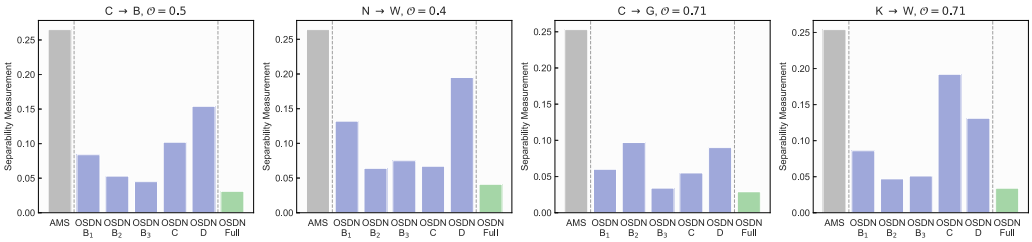

Fig. 10. Separability measurement results on four tasks between the hyperspherical-based baseline AMS, the full OSDN model and ablated groups of the OSDN model that affect the separability.

perspective on the source-target combined dandelion $\mathcal{DD}_{S\cup T}$, defined as follows:

$$
\begin{aligned}
CS_{S\cup T} &= \begin{bmatrix} CS^{11}_{S\cup T} & \boldsymbol{CS^{12}_{S\cup T}} & \cdots & \boldsymbol{CS^{1K}_{S\cup T}} \\ CS^{21}_{S\cup T} & CS^{22}_{S\cup T} & \cdots & \boldsymbol{CS^{2K}_{S\cup T}} \\ \vdots & \vdots & \ddots & \vdots \\ CS^{K1}_{S\cup T} & CS^{K2}_{S\cup T} & \cdots & CS^{KK}_{S\cup T} \end{bmatrix}, \\
CS^{ij}_{S\cup T} &= \mathrm{COS}\left(\mu^{(i)}_{S\cup T}, \mu^{(j)}_{S\cup T}\right), \\
\mu^{(i)}_{S\cup T} &= \frac{1}{n^{(i)}_S + n^{(i)}_T}\left(\sum_{j=1}^{n^{(i)}_S} x^{(i)}_{S_j} + \sum_{j=1}^{n^{(i)}_T} x^{(i)}_{T_j}\right),
\end{aligned}
\tag{21}
$$

where $CS_{S\cup T}$ denotes the inter-pappus Cosine similarity matrix of the source-target combined dandelion and $CS^{ij}_{S\cup T}$ represents the Cosine similarity between the ith and jth pappus of the source-target combined dandelion. Then, the separability measurement SP is defined as follows:

$$
SP = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} CS^{ij}_{S\cup T}, \tag{22}
$$

the smaller the separability measurement SP is, the better the separability is for the source-target combined dandelion. We present the separability measurement results between the hyperspherical-based baseline AMS, the separability-related ablated groups and the full OSDN model in Figure 10.

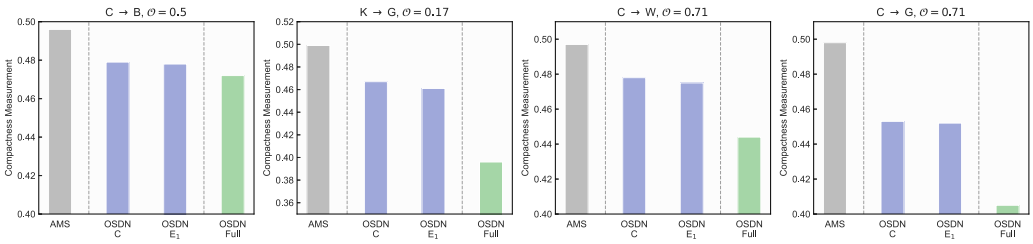

Fig. 11. Compactness measurement results on four tasks between the hyperspherical-based baseline AMS, the full OSDN model and ablated groups of the OSDN model that affect the compactness.

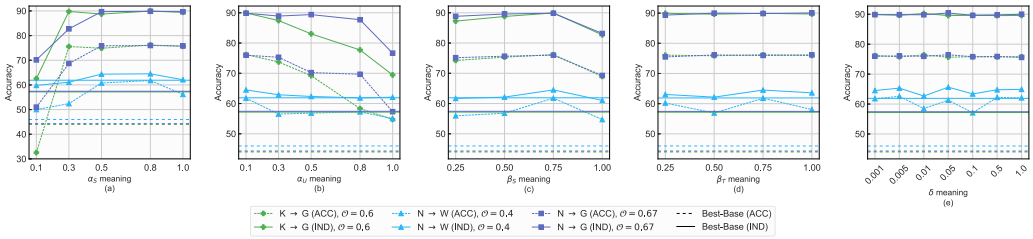

Fig. 12. Hyperparameter sensitivity analysis for hyperparameter α_S, α_U, β_S, β_T, and δ under their corresponding reasonable range. The dashed lines and solid lines indicate two modes, respectively. The horizontal lines indicates the best-performed baseline counterpart.

As we observe, both the full OSDN model and its ablated groups enjoy better separability compared with the AMS baseline. Moreover, the full OSDN model presents the best inter-category separability by achieving the lowest *SP* measurement. Hence it verifies the positive contribution of OSDN's constituting components towards enhancing inter-category separability, and the superior performance of the OSDN model over its hyperspherical-based counterpart.

We then follow Equation (4) to measure the compactness by the average category-wise maximum deviation d_{max}, defined as follows:

$$d_{max} = \frac{1}{K} \sum_{i=1}^{K} d_{max}^{(i)} , \tag{23}$$

the smaller the d_{max} is, the better the compactness performance is. Again, the measurement results presented in Figure 11 indicate the OSDN model outperforms both the baseline method AMS and its compactness-related ablated groups by a large margin. Hence, the excellent intra-category compactness achieved by the OSDN model is verified.

Overall, by achieving the best inter-category separability and intra-category compactness, the OSDN model can lead to more accurate intrusion detection performance.

5.8 Hyperparameter Sensitivity Analysis

We verify the stability and robustness of the OSDN model under varied hyperparameter settings within their corresponding reasonable range. The results are presented in Figure 12 and Figure 13. The dashed lines and solid lines indicate two modes, respectively. The horizontal lines indicates the best-performed baseline counterpart.

We observe the OSDN model performs relatively stable without showing significant fluctuation in nearly all hyperparameter settings. As well, the OSDN model constantly outperforms the

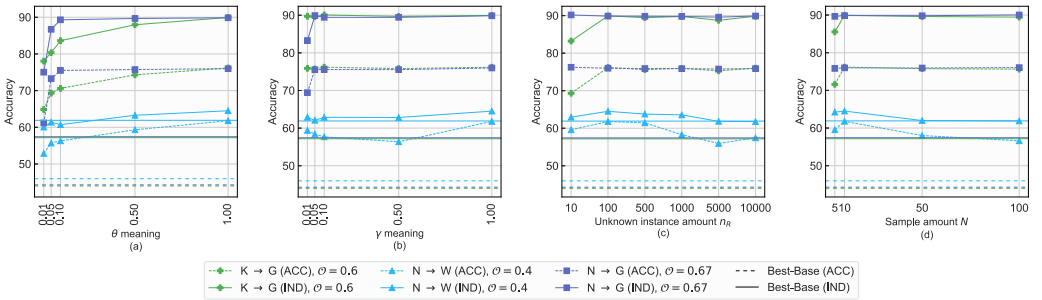

Fig. 13. Hyperparameter sensitivity analysis for hyperparameter θ, γ, unknown instance amount n_R used in the intrusion classifier unknown training and the sampled child dandelion number N used in the DSDM, under their corresponding reasonable range.

Table 3. Total Training Time, Measured in Minutes

Methods	C→G, $\mathcal{O}$ = 0.67			K→B, $\mathcal{O}$ = 0.5			C→W, $\mathcal{O}$ = 0.25			Avg
AMS	14.69			14.65			15.01			14.78
SROSDA	14.57			14.42			14.55			14.51
	N \ n_R	$=10^2$	$=10^3$	N \ n_R	$=10^2$	$=10^3$	N \ n_R	$=10^2$	$=10^3$	
OSDN	$=10$	4.47	5.46	$=10$	4.42	5.41	$=10$	4.47	4.76	**9.65**
	$=10^2$	14.06	15.00	$=10^2$	14.49	14.43	$=10^2$	14.02	14.83	

The performance of the OSDN model under different performance-sensitive hyperparameter settings are tested.

Table 4. Inference Time Per Network Traffic Instance, Measured in Milliseconds (10^{-3} Second)

Methods	C→G, $\mathcal{O}$ = 0.67			K→B, $\mathcal{O}$ = 0.5			C→W, $\mathcal{O}$ = 0.25			Avg
AMS	1.68			1.65			1.68			1.67
SROSDA	1.63			1.63			1.67			1.64
	N \ n_R	$=10^2$	$=10^3$	N \ n_R	$=10^2$	$=10^3$	N \ n_R	$=10^2$	$=10^3$	
OSDN	$=10$	0.100	0.099	$=10$	0.098	0.099	$=10$	0.101	0.103	**0.100**
	$=10^2$	0.098	0.100	$=10^2$	0.099	0.100	$=10^2$	0.101	0.102	

The performance of the OSDN model under different performance-sensitive hyperparameter settings are tested.

best-performed baseline method under nearly all hyperparameter settings. The OSDN model applies a single set of hyperparameter setting when facing different data domains and different tasks and still achieves such a stable level of performance. Hence, it verifies the stability and robustness of the OSDN model under manipulated hyperparameter settings.

5.9 Intrusion Detection Efficiency

We finally verify the training and intrusion detection inference efficiency of the OSDN model. The training time taken has been summarised in Table 3, and the inference time per network traffic instance has been summarised in Table 4. We only compare the OSDN model with the top-two best-performing baseline methods. As shown in Table 3, under varied settings of the OSDN model, the OSDN model performs more efficiently compared with its counterparts in nearly all settings. Since the model training can be performed on computationally-sufficient devices such as network

gateway servers, therefore, the training efficiency of the OSDN model is satisfactory. Besides, as indicated in Table 4, the OSDN model significantly outperforms its baseline counterparts in terms of the inference time taken to examine a network traffic instance. Therefore, the results verify the efficiency of the OSDN model, and demonstrate its real-world applicability as an efficient and accurate intrusion detector.

6 CONCLUSION

In this article, we propose the OSDN based on unsupervised heterogeneous domain adaptation in an open-set manner. The OSDN model tackles the IoT data scarcity by transferring intrusion knowledge from source NI domain to promote more accurate intrusion detection for the target IoT domain. The relaxation of the closed-set assumption lets the OSDN model detect both known and newly-emerged unknown intrusions in the IoT intrusion domain, hence it is more applicable in the real-world. The OSDN model achieves this by first forming the source domain into a dandelion-like feature space that emphasises inter-category separability and intra-category compactness. Then, the dandelion-based target membership mechanism constructs the target dandelion for intrusion knowledge transfer. The dandelion angular separation mechanism is used to promote inter-category separability, while the dandelion embedding alignment mechanism facilitates knowledge transfer from a graph embedding perspective. Also, the discriminating sampled dandelion mechanism is used to promote intra-category compactness. Trained using both known and generated unknown intrusion information, the intrusion classifier yields probabilistic semantics that can emphasise easily-confused categories and hence provide correction for the inter-category separation mechanism. Holistically, these mechanisms form the OSDN model and benefit in a more effective intrusion detection for IoT scenarios. Comprehensive experiments on five intrusion datasets are conducted. The OSDN model outperforms three state-of-the-art baseline methods by 16.9%. The effectiveness of each OSDN constituting component, the stability and the efficiency of the OSDN model are also verified. For future research, it is worthwhile to extend the OSDN model to the multi-source setting, in which the intrusion knowledge from multiple source domains can jointly benefit the open-set intrusion knowledge transfer. Besides, we can also consider using a category-wise attention mechanism during the intrusion knowledge transfer to account for diverse knowledge transfer sufficiency for each intrusion category. We will leave these as our future research directions.

A APPENDIX

A.1 Acronym Table

Table 5. The Acronym Table and the Corresponding Interpretation (Based on the Order of Appearance in the Article)

Acronym	Interpretation
OSDN	Open-Set Dandelion Network
DA	Domain Adaptation
NI	Network Intrusion
II	IoT Intrusion
OSDA	Open-Set Domain Adaptation
DASM	Dandelion Angular Separation Mechanism
DEAM	Dandelion Embedding Alignment Mechanism
DSDM	Discriminating Sampled Dandelion Mechanism
SDCM	Semantic Dandelion Correction Mechanism
ML	Machine Learning
DL	Deep Learning
CS	Cosine Similarity
EA	Embedding Alignment
CP	Compactness
SUP	Supervision
SM	Semantic
SC	Semantic Correction
CE	Cross Entropy

A.2 Notation Table

Table 6. The Notation Table and the Corresponding Interpretation (Based on the Order of Appearance in the Article)

Notation	Interpretation
$\mathcal{D}_S$	Source NI domain
$\mathcal{X}_S$	Source NI domain traffic features
$\mathcal{Y}_S$	Source NI domain traffic intrusion labels
x_{S_i}	The i^{th} traffic instance in $\mathcal{X}_S$
y_{S_i}	The intrusion label of x_{S_i}
n_S	Number of instances in $\mathcal{X}_S$
d_S	Instance dimension of $\mathcal{X}_S$
K	Number of intrusion categories in $\mathcal{D}_S$
K'	Number of intrusion categories in $\mathcal{D}_T$
$f(x_i)$	The feature projector
E_S	The source feature projector
E_T	The target feature projector
d_C	The dimension of the common feature subspace
$d_{max}^{(i)}$	The maximum intra-category deviation of source intrusion category i
$COS()$	Cosine Similarity
$n_S^{(i)}$	Number of instances in the i^{th} source intrusion category
$\mu_S^{(i)}$	Mean of the source intrusion category i
$x_{S_j}^{(i)}$	The j^{th} instance of source i^{th} intrusion category
$y_{T_j}^D$	The dandelion-based membership for the j^{th} target instance x_{T_j}
CS_S	The source category pair-wise Cosine similarity matrix
CS_S^{ij}	The Cosine similarity between the i^{th} and j^{th} source intrusion category
$\mathcal{L}_{SS}$	Source dandelion separation loss
$\mathcal{L}_{ST}$	Target dandelion separation loss
G_S	The source dandelion graph
V_S	Vertices in G_S
E_G	Edges in G_S
$V_S^{(i)}$	The i^{th} vertex in the G_S
E_S^{ij}	The edge connecting $V_S^{(i)}$ and $V_S^{(j)}$
$\mathfrak{p}$	The origin
$\mathcal{L}_{EA}$	Dandelion embedding alignment loss
ϕ_S	The graph embedding of the source domain dandelion
$\mathcal{L}_{CP}$	Discriminating sampled dandelion loss
$D()$	The discriminator
$G_{\mathcal{DD}_S}$	The graph embedding of the source dandelion
$G_{\mathcal{DD}_T}$	The graph embedding of the target dandelion
$G_{\mathcal{DD}_*^j}$	The graph embedding of the j^{th} sampled dandelion
N	The amount of child dandelion being sampled
$\mathcal{L}_{SUP}$	The overall supervision loss
$\mathcal{L}_{SUP_S}$	The source supervision loss
$\mathcal{L}_{SUP_U}$	The unknown supervision loss
$\mathcal{L}_{CE}$	The cross entropy loss
n_R	The amount of unknown instances being generated
$\mathcal{X}_R$	The generated unknown instances for unknown training
C	The intrusion classifier

Table 7. The Notation Table and the Corresponding Interpretation (Continued)

Notation	Interpretation
$p_{S_j}^{(i)}$	The probabilistic semantic of the j^{th} source instance in category i
$\mathcal{DD}_{SS}$	The source semantic dandelion
$\mathcal{DD}_{SS}^{(i)}$	The i^{th} pappus of the source semantic dandelion
CS_{SM}	The Cosine similarity matrix between semantic dandelions
CS_{SM}^{ij}	The Cosine similarity between $\mathcal{DD}_{SS}^{(i)}$ and $\mathcal{DD}_{ST}^{(j)}$
$\mathcal{L}_{SC}$	The semantic dandelion correction loss
α_S, α_U	Hyperparameter controlling $\mathcal{L}_{SUP_S}$ and $\mathcal{L}_{SUP_U}$, respectively
β_S, β_T	Hyperparameter controlling $\mathcal{L}_{SS}$ and $\mathcal{L}_{ST}$, respectively
δ	Hyperparameter controlling $\mathcal{L}_{EA}$
θ	Hyperparameter controlling $\mathcal{L}_{SC}$
γ	Hyperparameter controlling $\mathcal{L}_{CP}$
$TP^{(k)}$	True positive of category k
$\lvert\mathcal{X}_T^{(k)}\rvert$	Number of target instances in intrusion category k
O	The openness level
$\mathcal{DD}_{S\cup T}$	The source-target combined dandelion
$CS_{S\cup T}$	The inter-pappus Cosine similarity matrix of $\mathcal{DD}_{S\cup T}$
$\mu_{S\cup T}^{(i)}$	The i^{th} pappus of $CS_{S\cup T}$
$CS_{S\cup T}^{ij}$	The Cosine similarity between $\mu_{S\cup T}^{(i)}$ and $\mu_{S\cup T}^{(j)}$
d_{max}	The average category-wise maximum deviation

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Prof. André Brinkmann for his invaluable assistance in enhancing the quality of this article.

REFERENCES

[1] Ghada Abdelmoumin, Danda B. Rawat, and Abdul Rahman. 2021. On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. *IEEE Internet of Things Journal* 9, 6 (2021), 4280–4290.

[2] Eirini Anthi, Lowri Williams, Małgorzata Słowińska, George Theodorakopoulos, and Pete Burnap. 2019. A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal* 6, 5 (2019), 9042–9053.

[3] Tim M. Booij, Irina Chiscop, Erik Meeuwissen, Nour Moustafa, and Frank TH den Hartog. 2021. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal* 9, 1 (2021), 485–496.

[4] Giampaolo Bovenzi, Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, Valerio Persico, and Antonio Pescapé. 2023. Network anomaly detection methods in IoT environments via deep learning: A fair comparison of performance and robustness. *Computers and Security* 128 (2023), 103167.

[5] Giampaolo Bovenzi, Giuseppe Aceto, Domenico Ciuonzo, Valerio Persico, and Antonio Pescapé. 2020. A hierarchical hybrid intrusion detection approach in IoT scenarios. In *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–7.

[6] Hao Dai, Jiashu Wu, Yang Wang, Jerome Yen, Yong Zhang, and Chengzhong Xu. 2023. Cost-efficient sharing algorithms for DNN model serving in mobile edge networks. *IEEE Transactions on Services Computing* 16, 4 (2023), 1–14.DOI: https://doi.org/10.1109/TSC.2023.3247049

[7] Christian Dietz, Raphael Labaca Castro, Jessica Steinberger, Cezary Wilczak, Marcel Antzek, Anna Sperotto, and Aiko Pras. 2018. IoT-botnet detection and isolation by access routers. In *Proceedings of the 2018 9th International Conference on the Network of the Future*. IEEE, 88–95.

[8] Felix Erlacher and Falko Dressler. 2022. On high-speed flow-based intrusion detection using snort-compatible signatures. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2022), 495–506. DOI: https://doi.org/10.1109/TDSC.2020.2973992

[9] Mojtaba Eskandari, Zaffar Haider Janjua, Massimo Vecchio, and Fabio Antonelli. 2020. Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet of Things Journal* 7, 8 (2020), 6882–6897.
[10] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. 2020. Open set domain adaptation: Theoretical bound and algorithm. *IEEE Transactions on Neural Networks and Learning Systems* 32, 10 (2020), 4309–4322.
[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.
[12] Hany M. Harb, Afaf A. Zaghrot, Mohamed A. Gomaa, and Abeer S. Desuky. 2011. Selecting optimal subset of features for intrusion detection systems. *Advances in Computational Sciences and Technology* 4, 2 (2011), 179–192.
[13] Setz Hettich. 1999. The uci kdd archive. http://kdd.ics.uci.edu (1999).
[14] Taotao Jing, Hongfu Liu, and Zhengming Ding. 2021. Towards novel target discovery through open-set domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9322–9331.
[15] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* 100 (2019), 779–796.
[16] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. 2020. Towards inheritable models for open-set domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12376–12385.
[17] Shuang Li, Binhui Xie, Jiashu Wu, Ying Zhao, Chi Harold Liu, and Zhengming Ding. 2020. Simultaneous semantic alignment network for heterogeneous domain adaptation. In *Proceedings of the 28th ACM International Conference on Multimedia*. Association for Computing Machinery, New York, NY, 3866–3874. DOI: https://doi.org/10.1145/3394171.3413995
[18] Xinhao Li, Jingjing Li, Zhekai Du, Lei Zhu, and Wen Li. 2022. Interpretable open-set domain adaptation via angular margin separation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*. Springer, 1–18.
[19] Jian Liang, Kaixiong Gong, Shuang Li, Chi Harold Liu, Han Li, Di Liu, Guoren Wang, et al. 2021. Pareto domain adaptation. *Advances in Neural Information Processing Systems* 34 (2021), 12917–12929.
[20] Yang Lu and Li Da Xu. 2018. Internet of things (IoT) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal* 6, 2 (2018), 2103–2115.
[21] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. 2020. Progressive graph learning for open-set domain adaptation. In *Proceedings of the International Conference on Machine Learning*. PMLR, 6468–6478.
[22] Antonio Mangino, Morteza Safaei Pour, and Elias Bou-Harb. 2020. Internet-scale insecurity of consumer internet of things: An empirical measurements perspective. *ACM Transactions on Management Information Systems* 11, 4 (2020), 24 pages. DOI: https://doi.org/10.1145/3394504
[23] Sk Tanzir Mehedi, Adnan Anwar, Ziaur Rahman, Kawsar Ahmed, and Rafiqul Islam. 2022. Dependable intrusion detection system for IoT: A deep transfer learning based approach. *IEEE Transactions on Industrial Informatics* 19, 1 (2022), 1006–1017.
[24] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Proceedings of the 2018 Network and Distributed Systems Security Symposium*. 1–15.
[25] Robert Mitchell and Ray Chen. 2013. Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, 5 (2013), 593–604.
[26] Robert Mitchell and Ray Chen. 2013. Behavior-rule based intrusion detection systems for safety critical smart grid applications. *IEEE Transactions on Smart Grid* 4, 3 (2013), 1254–1263.
[27] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Proceedings of the 2015 Military Communications and Information Systems Conference*. IEEE, 1–6.
[28] Ghulam Muhammad, M. Shamim Hossain, and Sahil Garg. 2020. Stacked autoencoder-based intrusion detection system to combat financial fraudulent. *IEEE Internet of Things Journal* 10, 3 (2020), 2071–2078.
[29] Sarumathi Murali and Abbas Jamalipour. 2019. A lightweight intrusion detection for sybil attack under mobile RPL in the internet of things. *IEEE Internet of Things Journal* 7, 1 (2019), 379–388.
[30] Pau Panareda Busto and Juergen Gall. 2017. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*. 754–763.
[31] Han Qiu, Tian Dong, Tianwei Zhang, Jialiang Lu, Gerard Memmi, and Meikang Qiu. 2020. Adversarial attacks against network intrusion detection in IoT systems. *IEEE Internet of Things Journal* 8, 13 (2020), 10327–10335.
[32] Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge*

Management. Association for Computing Machinery, New York, NY, 1325–1334. DOI: https://doi.org/10.1145/3340531.3411866

[33] Pratik Satam and Salim Hariri. 2020. WIDS: An anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol. *IEEE Transactions on Network and Service Management* 18, 1 (2020), 1077–1091.

[34] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP* 1 (2018), 108–116.

[35] Kurniabudi, Deris Stiawan, Darmawijoyo, Mohd Yazid Bin Idris, Alwi M. Bamhdi, and Rahmat Budiarto. 2020. CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access* 8 (2020), 132911–132921.

[36] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. Ieee, 1–6.

[37] Mahbod Tavallaee, Natalia Stakhanova, and Ali Akbar Ghorbani. 2010. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 5 (2010), 516–524.

[38] Wattana Viriyasitavat, Li Da Xu, Zhuming Bi, and Assadaporn Sapsomboon. 2019. New blockchain-based architecture for service interoperations in internet of things. *IEEE Transactions on Computational Social Systems* 6, 4 (2019), 739–748. DOI: https://doi.org/10.1109/TCSS.2019.2924442

[39] Jiashu Wu, Hao Dai, Yang Wang, Kejiang Ye, and Chengzhong Xu. 2023. Heterogeneous domain adaptation for IoT intrusion detection: A geometric graph alignment approach. *IEEE Internet of Things Journal* 10, 12 (2023), 1–1. DOI: https://doi.org/10.1109/JIOT.2023.3239872

[40] Jiashu Wu, Hao Dai, Yang Wang, Yong Zhang, Dong Huang, and Chengzhong Xu. 2022. PackCache: An online cost-driven data caching algorithm in the cloud. *IEEE Transactions on Computers* 72, 4 (2022), 1–8. DOI: https://doi.org/10.1109/TC.2022.3191969

[41] Jiashu Wu, Yang Wang, Hao Dai, Chengzhong Xu, and Kenneth B. Kent. 2023. Adaptive bi-recommendation and self-improving network for heterogeneous domain adaptation-assisted IoT intrusion detection. *IEEE Internet of Things Journal* 10, 15 (2023), 1–1. DOI: https://doi.org/10.1109/JIOT.2023.3262458

[42] Jiashu Wu, Yang Wang, Binhui Xie, Shuang Li, Hao Dai, Kejiang Ye, and Chengzhong Xu. 2023. Joint semantic transfer network for IoT intrusion detection. *IEEE Internet of Things Journal* 10, 4 (2023), 3368–3383. DOI: https://doi.org/10.1109/JIOT.2022.3218339

[43] Binhui Xie, Shuang Li, Fangrui Lv, Chi Harold Liu, Guoren Wang, and Dapeng Wu. 2022. A collaborative alignment framework of transferable knowledge extraction for unsupervised domain adaptation. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 6518–6533.

[44] Haipeng Yao, Pengcheng Gao, Jingjing Wang, Peiying Zhang, Chunxiao Jiang, and Zhu Han. 2019. Capsule network assisted IoT traffic classification mechanism for smart cities. *IEEE Internet of Things Journal* 6, 5 (2019), 7515–7525.

Received 28 March 2023; revised 21 October 2023; accepted 4 January 2024