EURASIP Journal on
Information Security

RESEARCH Open Access

Transfer learning for detecting unknown network attacks

Juan Zhao[1], Sachin Shetty[2*], Jan Wei Pan[3], Charles Kamhoua[4] and Kevin Kwiat[5]

Abstract

Network attacks are serious concerns in today's increasingly interconnected society. Recent studies have applied conventional machine learning to network attack detection by learning the patterns of the network behaviors and training a classification model. These models usually require large labeled datasets; however, the rapid pace and unpredictability of cyber attacks make this labeling impossible in real time. To address these problems, we proposed utilizing transfer learning for detecting new and unseen attacks by transferring the knowledge of the known attacks. In our previous work, we have proposed a transfer learning-enabled framework and approach, called HeTL, which can find the common latent subspace of two different attacks and learn an optimized representation, which was invariant to attack behaviors' changes. However, HeTL relied on manual pre-settings of hyper-parameters such as relativeness between the source and target attacks. In this paper, we extended this study by proposing a clustering-enhanced transfer learning approach, called CeHTL, which can automatically find the relation between the new attack and known attack. We evaluated these approaches by stimulating scenarios where the testing dataset contains different attack types or subtypes from the training set. We chose several conventional classification models such as decision trees, random forests, KNN, and other novel transfer learning approaches as strong baselines. Results showed that proposed HeTL and CeHTL improved the performance remarkably. CeHTL performed best, demonstrating the effectiveness of transfer learning in detecting new network attacks.

Keywords: Network attacks detection, Machine learning, Transfer learning

1 Introduction

In recent years, cyber attack is a growing serious concern due to its increased sophistication and variations, such as denial-of-service (DoS) tactics and the zero-day attack, posing a great threat to government, military, and industrial networks. Conventional signature-based detection approaches may fail to address the increased variability of today's cyber attacks. Developing novel anomaly detection techniques to better learn, adapt, and detect threats in diverse network environments becomes essential.

Machine learning/data mining approaches have been applied to the attack detection in networked environments to improve the detection rate [1–4]. Data-driven supervised models achieved better accuracy than unsupervised approaches but relied on a large number of labeled malicious samples [5]. As attacks evolved by varying their behaviors, the distributions of feature may change, making the trained models work poorly [6] and unable to detect the new attacks. This is a domain-shift problem, which usually needs recollecting new training data and retraining the model to adapt to the changes in the target domain. However, collecting sufficient labeled data for such continuously rising attack variants is infeasible. Further, detecting evolving attacks usually needs incorporating new features from various network layers [7]. This also needs to retrain the model because of the different feature dimensions.

To address the above problems, we proposed using transductive transfer learning to enhance the detection of new threats [6]. Transductive transfer learning, a novel machine learning technique, can adapt features in a target domain with deficient labeled data by transferring learned knowledge from a related source domain [8]. The intuition behind is the human's transitive inference ability to extending what has been learned in one domain to a new similar domain [9]. Our study is motivated by the fact that

*Correspondence: sshetty@odu.edu
[2]Virginia Modeling Analysis and Simulation Center, Old Dominion University, 23529 Norfolk, USA
Full list of author information is available at the end of the article

most network attacks belong to variants of known network attack families and share common traits in features [6, 10], which suggested a good fit for applying transfer learning.

In this study, source and target domain data refers to the same network environment at a different time. We assumed that attacks in a source domain are already known and labeled, and attacks in a target domain are new and different than the source. We formularized the problem by using source domain data to differentiate new attacks in the target domain. Previously, we developed a transfer learning-enabled detection framework and proposed a feature-based heterogeneous transfer learning, called HeTL [6], to detect unseen variants of attacks. HeTL can find new feature representations for source and target domain by transforming them on a common latent space. Nevertheless, we observed that the performance of HeTL depended on manual pre-settings of a hyper-parameter: relevance between the source and target domain [6]. In this paper, we proposed another approach—a hierarchical transfer learning algorithm with clustering enhancement, called CeHTL, which can cluster source and target domain and compute the relevance between them.

We utilized a benchmark network intrusion dataset NSL-KDD [11]. To stimulate the domain shift, we generated training and testing datasets by sampling attacks from different types of attacks, from big category of attacks (e.g., DoS, R2L), and also the subcategory of attacks (i.e., 22 subtypes). We compared the proposed CeHTL with HeTL [6], as well as any other baselines, including traditional classification without transfer learning and several novel transfer learning approaches. We also evaluated the approaches on imbalanced datasets, which is common in real-world cyber attack practice. We performed sensitivity analysis by tuning parameters and using different sizes of training set. The results showed that CeHTL demonstrated the most stable results, which means that it does not rely on the pre-setting of parameters and thus is more effective in detecting unknown attacks.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 outlines the transfer learning framework. Section 4 describes the proposed approaches. Section 6 presents the experiments, evaluations, and discussions. Finally, we conclude the work in Section 7.

2 Related work

2.1 Network attack detection

One of the well-known techniques for network attack detection is signature-based detection, which is based on an extensive knowledge of the particular characteristics of each attack, referred to as its "signature." One study [12] proposed a methodology to craft traffic with different characteristics. Other studies [13, 14] focused on how to find effective signatures. However, one major limitation of the signature-based technique is its failure to detect new attacks, as their signatures are unknown to the system. In addition, building new signatures needs manual inspection by human experts, which is very expensive and time-consuming, and also introduces an important latency between the discovery of a new attack and the construction of its signature.

Another type of technique for network attack detection is the supervised learning-based technique, which uses instances of known attacks to build a classification model that distinguishes attacks from good programs [1, 3]. Nari and Ghorbani [15] present a network behavioral modeling approach for malware detection and malware family classification. Rafique et al. [16] evaluated the evolutionary algorithms for classification of malware families through different network behaviors. Iglesias and Zseby [17] focused on the feature selection approach to improve the performance of network-based anomaly detection. However, these learning-based techniques share the same limitation as the signature-based detection in that they both perform poorly on new attacks. Since different attacks usually have different distributions of network behaviors, the learned patterns are unable to work accurately. A significant advantage of our approach is its ability to identify an unknown attack that has not been previously investigated.

2.2 Transfer learning

Transfer learning was designed to use knowledge from the source domain, which has sufficient labeled data, to help build more precise models in a related, but different, domain with only a few or no labeled data. Transfer learning approaches can be mainly categorized into three classes [18]. The first class is instance-based [19, 20], which assumes that certain parts in the source data can be reused for the target domain by re-weighting related samples. Dai et al. [20] introduced a boosting algorithm, TrAdaBoost, which iteratively re-weighted the source domain data and the target domain data to reduce the effect of "bad" source data while encouraging the "good" source data to contribute more to the target domains. However, these approaches require a lot of labeled samples from the target domain. The second class can be viewed as model-based approaches [21, 22], which assume both source and target tasks share some parameters or priors of their models. The third class of transfer learning approaches is feature-based [23–25], where a new feature representation is learned from the source and the target domain and is used to transfer knowledge across domains. Shi et al. [26] proposed a heterogeneous transfer learning method, called HeMap, to project the source and target domain onto latent subspace via linear transformations. They assumed the subspace is orthogonal. Pan et al. [24] have performed transfer component analysis (TCA) to reduce the distance between domains by projecting the features onto a shared subspace. Nam et al. [27] then applied TCA to the software defect detection problem. Sun et al. [23] proposed an approach, called Correlation Alignment (CORAL), to project source data onto target data by aligning the second-order statistics of the source and target distributions, which do not need any labeled data from the target domain. The work has been applied to the object detection problem and achieves good results. Shi et al. first proposed a state-of-the-art approach called HeMap [26], which uses spectral embedding to unify the different feature spaces of the target and source datasets, and applies this approach to image classification.

2.3 Transfer learning for network attack detection

Even though transfer learning has many great applications in natural language processing and visual recognition [25, 28], not many studies have applied it to the network attack detection problem. Bekerman et al. [4] mentioned that transfer learning can improve robustness in detecting unknown malware between non-similar environments. However, they did not present much detailed and formal work on this idea. The study in [29] applied an instance-based transfer learning approach in network intrusion detection. However, they require plenty of labeled data from target domain. Gao et al. [30] proposed a model-based transfer learning approach and apply it to the KDD99 cup network dataset. Both of these instance and model-based transfer learning approaches depend heavily on the assumption of homogeneous features. This is often not the case for network attack detection, which typically exhibits heterogeneous features. Another advantage of feature-based approaches is its flexibility to adopt different base classifiers according to different cases, which motivated us to derive a feature-based transfer learning approach for our network attack detection study. To our best knowledge, this paper is the first effort in applying a feature-based transfer learning approach for improving the robustness of network attack detection.

3 Framework of using transfer learning for detecting new network attacks

We have present a transfer learning-enabled network attack detection framework to enhance detecting new network attacks in a target domain in [6]. From a practical standpoint, source and target domains can represent different or the same network environments with different attacks captured at different times and at separate instances. In this study, we primarily consider the latter scenario, wherein the source and target domains comprise different attacks. We assume that the attack in the source domain is known and labeled appropriately, and attacks in the target domain are new and not labeled. Unlike prior studies [29, 30] assuming that the source and target domains should have the same feature sets, our framework supports introducing new features into the target domain. This is relevant to evolving network attacks where the adversary may change their behaviors, resulting in a need to incorporating new features in the network or system layers. Thus, in this scenario, the source and target domains have different attack distributions or feature sets. The goal of the transfer learning framework is to use source domain data to differentiate new attacks from the target domain.

The framework consists of a machine learning pipeline, which includes the following stages: (i) extracting features from raw network traffic data, (ii) learning representations with feature-based transfer learning, and (iii) classification. In the first stage, features are extracted from the raw network trace data with a statistic calculation of the network flow. Second, we used feature-based transfer learning algorithms to learn a good new feature representation from both source and target domains. Then, we fed the new representation to a common base classifier. The choice of a common base classifier can be decision trees, SVM, and KNN.

4 Transfer learning approach via spectral transformation

We model the network attack detection as a binary classification problem, which is to classify each network connection as a malicious or as normal connection. Suppose we are provided with source domain training examples $S = \left\{\vec{x_i}\right\}, \vec{x} \in \mathbb{R}^m$ that have labels $L_S = \{y_i\}$, and target domain data $T = \{\vec{u_i}\}, \vec{u} \in \mathbb{R}^n$. Suppose $\vec{x}$ and $\vec{u}$ are drawn from different distributions, $P_S(X) \neq P_T(X)$, where $P_T(X)$ is unknown, and the dimensions of $\vec{x}$ and $\vec{u}$ are different, $\mathbb{R}^m \neq \mathbb{R}^n$. Our goal is to accurately predict the labels on T.

Since network attacks share similar traits, our approach is to find the common latent subspace and transform the source and target data onto it to get new feature representations, which can then be used in classifcation. We demonstrated the approach in our previous paper [6]. Given source domain data and target domain data with different attacks, the model explores the common latent space, in which the original structure of the data is preserved while the discriminative examples are still far apart.

4.1 Optimization

Given source data $\mathbf{S}$ and target data $\mathbf{T}$, we compute an optimal projection of $\mathbf{S}$ and $\mathbf{T}$ onto an optimal subspace $\mathbf{V_S}$ and $\mathbf{V_T}$ according to the following optimization objective:

$$\min_{\mathbf{V_S},\mathbf{V_T}} \ell(\mathbf{V_S},\mathbf{S}) + \ell(\mathbf{V_T},\mathbf{T}) + \beta D(\mathbf{V_S},\mathbf{V_T}), \tag{1}$$

where $\ell(*,*)$ is a distortion function that evaluates the difference between the original data and the projected data. $D(\mathbf{V_S},\mathbf{V_T})$ denotes the difference between the projected data of the source and target domains. β is a trade-off parameter that controls the similarity between the two datasets.

Thus, the first two elements of (1) ensure that the projected data preserve the structures of the original data as much as possible.

We defined $D(\mathbf{V_S},\mathbf{V_T})$ in terms of $l(*,*)$ as:

$$D(\mathbf{V_S},\mathbf{V_T}) = \|\mathbf{V_T} - \mathbf{V_S}\|^2 \tag{2}$$

which is the difference between the projected target data and the projected source data. Hence, the projected source and target data are constrained to be similar by minimizing the difference function (2).

We applied linear transformation to finding the projected space. We define $\ell(*,*)$ as follows:

$$\ell(\mathbf{V_S},\mathbf{S}) = \|\mathbf{S}-\mathbf{V_S}\mathbf{P_S}\|^2, \ell(\mathbf{V_T},\mathbf{T}) = \|\mathbf{T}-\mathbf{V_T}\mathbf{P_T}\|^2, \tag{3}$$

where $\mathbf{V_S}$ and $\mathbf{V_T}$ are achieved by a linear transformations with linear mapping matrices, denoted as $\mathbf{P_S} \in \mathbb{R}^{k\times m}$ and $\mathbf{P_T} \in \mathbb{R}^{k\times n}$ to the source and target, respectively. $\|X\|^2$ is the Frobenius norm that can also be expressed as a matrix trace norm. In a different view, $\mathbf{P_S}^{\mathbf{T}} \in \mathbb{R}^{m\times k}$ and $\mathbf{P_T}^{\mathbf{T}} \in \mathbb{R}^{n\times k}$ project the original data $\mathbf{S}$ and $\mathbf{T}$ into a k-dimensional latent subspace, where the projected data are comparable $\left(\ell(\mathbf{V_S},\mathbf{S}) = \|\mathbf{S}\mathbf{P_S}^{\mathbf{T}} - \mathbf{V_S}\|^2\right)$. This will lead to a trivial solution $\mathbf{P_S} = 0, \mathbf{V_S} = 0$. We thus apply (3). It can be viewed as a matrix factorization problem, which is widely known as an effective tool to extract latent subspaces while preserving the original data structures.

4.2 Optimization objective 1

Substituting (3) and (2) into (1), we obtain the following optimization objective to minimize with regard to $\mathbf{V_S}, \mathbf{V_T}, \mathbf{P_S}$ and $\mathbf{P_T}$ as follows:

$$\begin{aligned}\min G(\mathbf{V_S},\mathbf{V_T},\mathbf{P_S},\mathbf{P_T}) = \min &\|\mathbf{S}-\mathbf{V_S}\mathbf{P_S}\|^2\\ &+\|\mathbf{T}-\mathbf{V_T}\mathbf{P_T}\|^2\\ &+\beta\cdot\|\mathbf{V_T}-\mathbf{V_S}\|^2)\end{aligned} \tag{4}$$

In our previous work [6], we used a gradient method to get the global minimums by iteratively fixing three of the matrices to solve the remaining one until convergence. The detailed HeTL algorithm was presented in [6].

5 Clustering-enhanced hierarchical transfer learning

In previous study, we have observed that the performance of HeTL depends on the manual presetting of a hyper-parameter—relevance between the source and target domains (β). Inappropriate choice of parameters might lead to suboptimal efficacy results. The row order of the class type for $\mathbf{S}$ and $\mathbf{T}$ could also affect the results of $D(\mathbf{V_S},\mathbf{V_T})$. Practically, we may know little about the new attack in $\mathbf{T}$, so the transformation process in (4) could be misleading.

To address this problem, we proposed a hierarchical transfer learning with clustering enhancement, called CeHTL, through automatically finding the relevance between the source and target domain before we perform the projection. CeHTL first clustered the instances for the target domains, as the source domain already has two natural clusters (classes). By computing the similarity of each cluster and choosing the mapping for two similar clusters in the source and target domains, we can get the correspondence (mapping) of each cluster in the target domain to the source domain. We sorted the instances by order of their cluster labels, so that the rows in matrices $\mathbf{T}$ and $\mathbf{S}$ will have the same class order. Then, we solved objective (4) for the ordered $\mathbf{T}$ and $\mathbf{S}$. We illustrated the comparison between CeHTL with HeTL in Fig. 1. The algorithm for CeHTL is listed in Algorithm 1. We chose K-means++ [31] for clustering and used the Euclidean distance to compute the similarity.

```
Algorithm 1: Clustering Enhanced Hierarchical Transfer Learning (CeHTL)
Input: $\mathbf{T}, \mathbf{S}$
Output: $\mathbf{T_{new}}, \mathbf{S_{new}}$
Initialize: c clusters for each domain, $c = 2$
$C_T$ = kmeans($\mathbf{T}, c$); %$C_T$ is the cluster label for each instance.
$C_S = Y_S$;%$C_S$ is the cluster label for each instance. $Y_S$ is the class label for source domain
If the dimensions of $\mathbf{T}$ and $\mathbf{S}$ is not equal,
$\mathbf{T} = \mathrm{pca}(\mathbf{T})$; $\mathbf{S} = \mathrm{pca}(\mathbf{S})$;
Compute the Euclidean distance between centroid of each cluster in $\mathbf{T}$ and $\mathbf{S}$.
For each cluster in $\mathbf{T}$, choose the similar cluster from $C_S$, which has the smallest Euclidean distance value, to form a similar cluster pair, and assign the same label to each similar pair of clusters.
Sort the matrices [ $\mathbf{T}, C_T$] and[ $\mathbf{S}, C_S$] in the order of $C_T$ and $C_S$, to get the $\mathbf{T_{new}}, \mathbf{S_{new}}$ for the new input for the HeTL algorithm.
```

In case that the source and target domains have heterogeneous feature sets, where $\mathbf{T}$ and $\mathbf{S}$ may have different dimensions, the Euclidean distance cannot be applied. To overcome this problem, we use principal component analysis (PCA) [32] for each source and target domain to perform feature reduction. By choosing the same size of components for source and target domains, they will

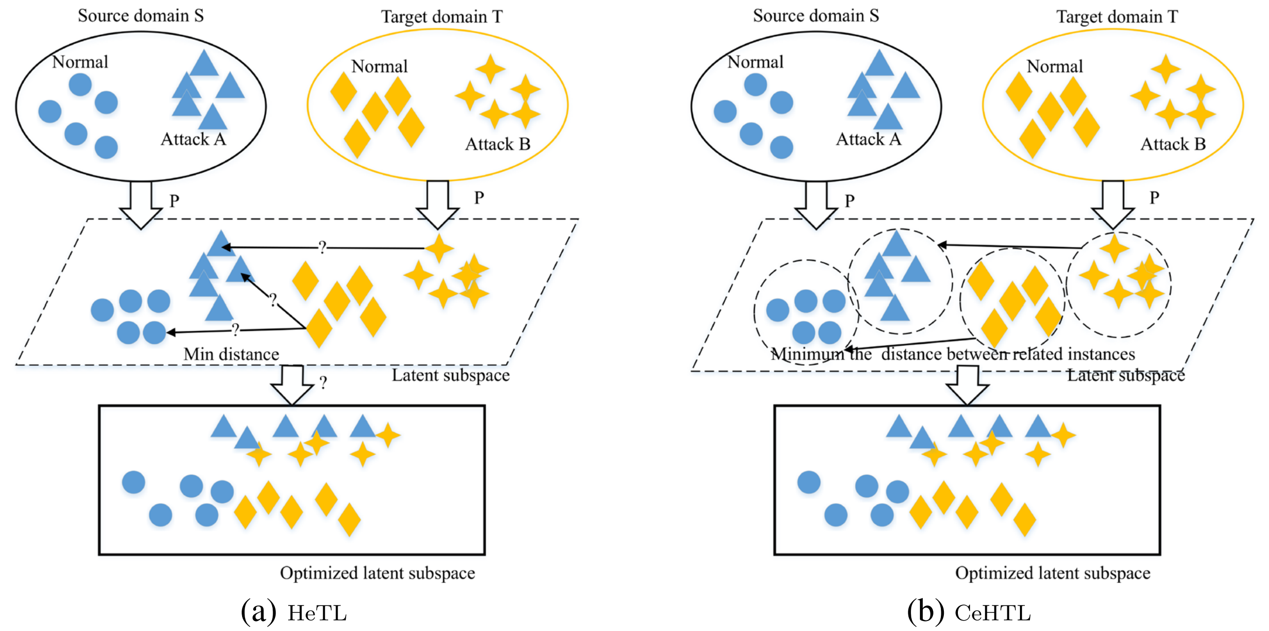


Fig. 1 Comparison between the HeTL and CeHTL

have the same dimensions. The notation description are presented in Table 1.

6 Experimental evaluation

In this section, we evaluated the performance the of proposed transfer learning HeTL and CeHTL for detecting "unknown" network attacks. We addressed the following questions: Does transfer learning approach provide any advantage compared with a single classifier without using transfer learning approach? and Which technique is the most appropriate transfer learning approach? We utilized a benchmark network intrusion dataset—the NSL-KDD benchmark dataset [11] (in Section 6.1). We carried out two experiments to stimulate the "unknown" network attacks and different feature spaces (in Section 6.2). We demonstrated the benefits of HeTL and CeHTL compared to other traditional machine learning algorithms as well as other several novel transfer learning methods (in Section 6.3). We also performed the parameter sensitivity analysis and showed the impact of imbalanced datasets and training data sizes (Section 6.4).

Table 1 Notation descriptions

Notations	Descriptions
$\mathbf{S}$	Source data
$\mathbf{V_S}$	Projected source data
$\mathbf{P_S}$	Projection function to the source space
$\mathbf{T}$	Target data
$\mathbf{V_T}$	Projected target data
$\mathbf{P_T}$	Projection function to the target space
β	Weights of the relevance between the source and target data
k	Dimensions of the projected space
α	Learning rates
Step	Learning step

6.1 Network datasets

NSL-KDD contains network features extracted from a series of TCP connection records captured from a local area network. Each record in the dataset corresponds to a connection labeled as either an normal or attack type. The dataset has 22 different types of attack, which can be grouped into 4 main categories: DoS, R2L, Probe, and User to root (U2R). Tables 2 and 3 provide the details of the attacks and their distribution in the training dataset. Since the portion of U2R is very small, we only focus on DoS, R2L, and Probe.

NSL-KDD contains 41 network features that can be split into 3 groups: (1) basic features deduced from TCP/IP connection packet headers; (2) traffic features, usually extracted by flow analysis tools; and (3) content features,

Table 2 Category of the attack in NSL-KDD

Main categories	Attack
DoS	Neptune, back, land, smurf, teardrop,pod
R2L	buffer_overflow, ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
Probe	ipsweep, nmap, portsweep, satan
U2R	loadmodule, perl, rootkit

Table 3 Number of instances in NSL-KDD

Class	Instances	Percentage
Normal	67343	53.46
DoS	45927	36.46
R2L	995	0.79
Probe	11656	9.25
U2R	52	0.04

requiring the processing of the packet content. Some example of features are listed in Table 4.

6.2 Experimental setting

6.2.1 Detection of unknown network attacks

This experiment is to evaluate the proposed transfer learning approaches for detecting new variants of attacks. Stimulating new attacks is challenging. We can assume attacks in the target data has no labels and differ from attacks in the source domain. We randomly selected malicious examples from one main attack category (e.g., DoS, R2L, Probe) and normal examples as the source domain. Then, we chose a different attack type combined with normal samples for the target domain. We finally generated three groups: DoS→Probe (DoS is the source domain for training and Probe is target domain for testing), DoS→R2L and Probe→R2L). To evaluate the generalization, we also chose attacks from 22 sub-attack types for each source and target set and generated 11 tasks. We repeated the processes ten times and reported the averages and standard deviations. We make the attack data, and the normal data in each domain are balanced unless stated otherwise. We further studied the effects of imbalanced data in Section 6.4.

6.2.2 Network attacks with different feature spaces

To evaluate the performance in detecting attacks using different feature spaces, we used different feature sets for source and target domains, based on the first experiment setting. In network security, there are circumstances that we need to incorporate new features to better detect the attacks. For example, traffic feature is more distinguishable for DoS attack. However, for the R2L attack, the content feature is more distinguishable. This usually need to retrain the model. To stimulate this scenario, we selected the most relative features for the source and target domains using information gain, resulting in unequal feature dimensions. The final selected features were listed in Appendix Tables 5 and 6. Of note, using information gain here is only for generating different feature sets, not for improving the performance. In real practice, features can be changed due to the manual feature engineering as we have less information about the target dataset. The baseline approach manually mapping the target data into the source feature space and applied the traditional classifiers. We compared our transfer learning approach with the baselines.

6.3 Evaluation

We chose the accuracy, F_1 score ($F - $ Measure) and receiver operating characteristic curve (ROC curve) as the performance metrics. F_1 score combines precision and recall to measure the per-class performance of classification or detection algorithms.

We firstly chose C4.5 decision tree (CART), linear SVM, and KNN as the baselines, which were also served for base classifiers for HeTL and CeHTL. We compared HeTL and CeHTL with baselines on three main transfer learning tasks (i.e., DoS→Probe, DoS→R2l, and Probe→R2L). Figures 2 and 3 show the box plots of accuracy and F_1 score on ten iterations on three main tasks. We observed that the baseline models performed poorly, with accuracy of 0.47–0.74 and F_1 score of 0.1–0.65. Our HeTL and CeHTL significantly outperformed the baselines, obtained over 0.70 accuracy and 0.75 F_1 score. CeHTL outperformed HeTL with all three base classifiers in DoS→Probe and in decision tree and KNN in

Table 4 Some selected features in NSL-KDD

Feature name	Description	Feature category
Duration	Duration of the connection	Basic features
Src_bytes	Data bytes from source to destination	Basic feature
Dst_bytes	Data bytes from destination to source	Basic feature
Num_failed_logins	Number of incorrect login in a connection	Content feature
Srv_count	Sum of connections to the same destination port number	Traffic feature
Serror_rate	Percentage of connections that have “SYN” errors among the connections to the same host in the past 2 s	Traffic feature
Srv_serror_rate	Percentage of connections that have “SYN” errors among the connections to the same destination port in the past 2 s	Traffic feature
Dst_host_count	Sum of connections to the same destination IP address	Traffic feature
Dst_host_same_srv_rate	The percentage of connections that were to the same service, among the connections aggregated in dst_host_count	Traffic feature

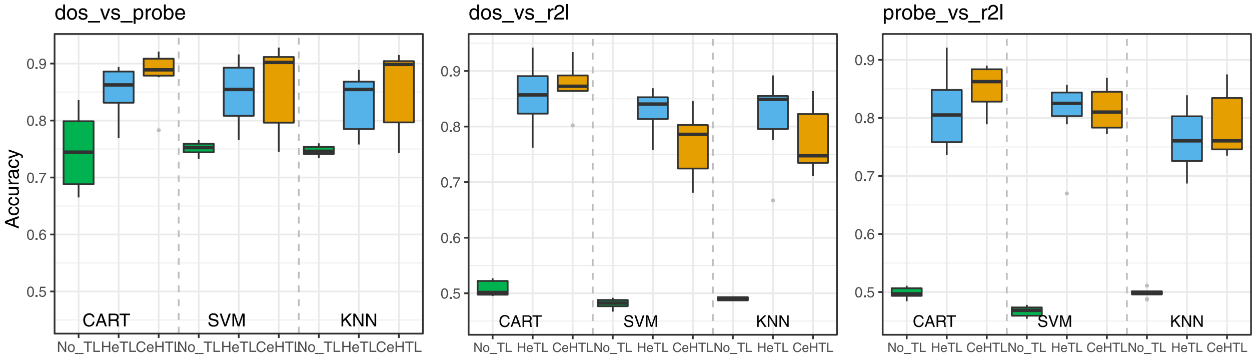


Fig. 2 Box plot of accuracy of transfer learning approaches and baselines on three main tasks

Probe→R2L. CeHTL achieved the best result with an average accuracy and F_1 score of 0.88.

Then, we applied HeTL, CeHTL, and two baseline methods—SVM and HeMap [26], a novel transfer learning approach—to the 11 transfer learning tasks generated by the subtypes of attacks, along with the 3 main tasks. We run the experiment for 10 iterations with different random seeds and reported the average and standard deviations of accuracy and F_1 scores in Figs. 4 and 5. We observed (1) transfer learning approaches outperformed the traditional classifiers without using transfer learning in all 14 tasks, (2) HeTL and CeHTL can improve the accuracy to 0.8–0.9 in 5 tasks, (3) HeTL and CeHTL outperformed HeMap, and (4) CeHTL outperformed all other methods in 10 cases. Figure 6 shows the ROC curves on 3 main transfer learning tasks using KNN as the base classifier. CeHTL achieved the best area under ROC curves (AUC) in 2 DoS→Probe and Probe→R2L (CeHTL 0.93 and 0.91 AUC vs. HeTL 0.82 and 0.65 AUC). Besides HeMap, we compared our approaches with more baselines, TCA [24] and CORAL [23]. Figure 7 showed the results of approaches on 5 classifiers in DoS→R2L. HeTL and CeHTL outperformed all baselines.

Finally, we carried out the second experimental setting, where the source domain and target domain have different feature spaces. We compare the transfer learning approach with the manual mapping approach on DoS→R2L. From the results shown in Fig. 8, we can see that the transfer learning approaches outperformed the baselines.

6.4 Discussion

The study proposed two transfer learning methods, HeTL and CeHTL, on network attack detection methods to address the issues of lacking sufficient labels for new attacks. The results showed that HeTL and CeHTL significantly improved the accuracy compared to the traditional classifiers and other transfer learning methods. Especially, CeHTL performed the best in most of the tasks, especially in DoS→Probe tasks. One of the reason is DoS had more similarities with Probe than R2L, according to the top selected features in Appendix Table 5 and 6. This can improve the accuracy of computing the cluster correspondence, which thus resulted in a better performance.

6.4.1 Parameter sensitivity

Two hyper-parameters, the similarity confidence parameter β and the dimensions of the new feature space k, need to be set for optimization (4). There are several ways to determine the optimum hyper-parameters: (a) the

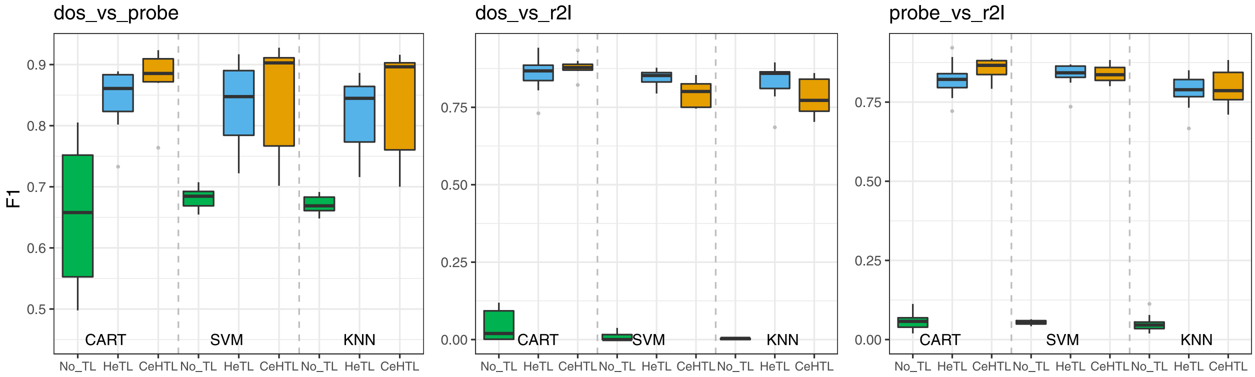


Fig. 3 Box plot of F_1 score of transfer learning approaches and baselines on three main tasks

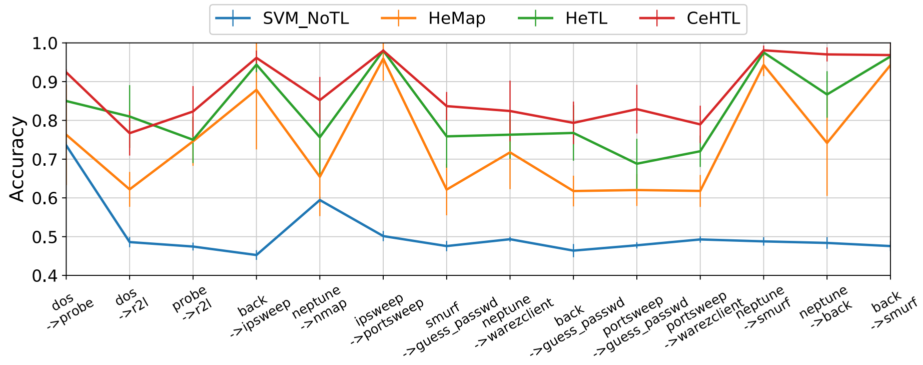


Fig. 4 Performance comparison of accuracy on unknown network attacks detection, sample size = 1000

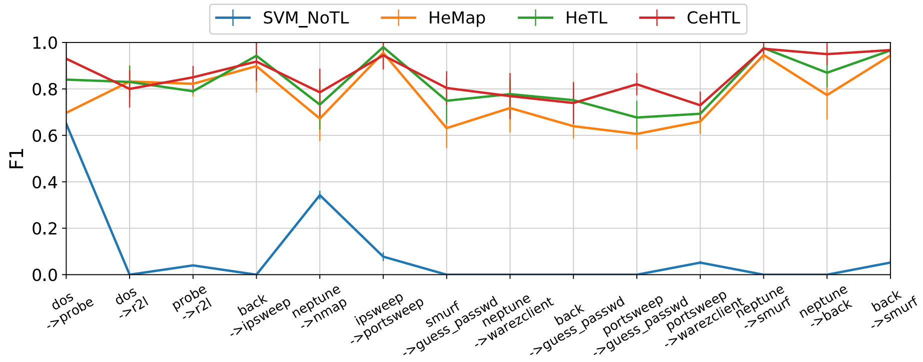
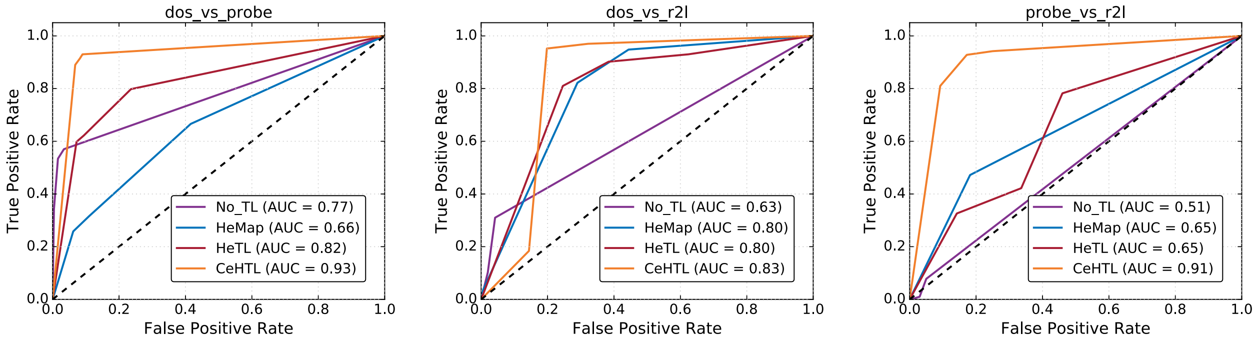


Fig. 5 Performance comparison of F_1 score on unknown network attacks detection, sample size = 1000



(a) (b) (c)

Fig. 6 Performance comparison of ROC curves on the three transfer learning datasets. **a** ROC curve on DoS→Probe. **b** ROC curve on DoS→R2L **c**. ROC curve on Probe→R2L

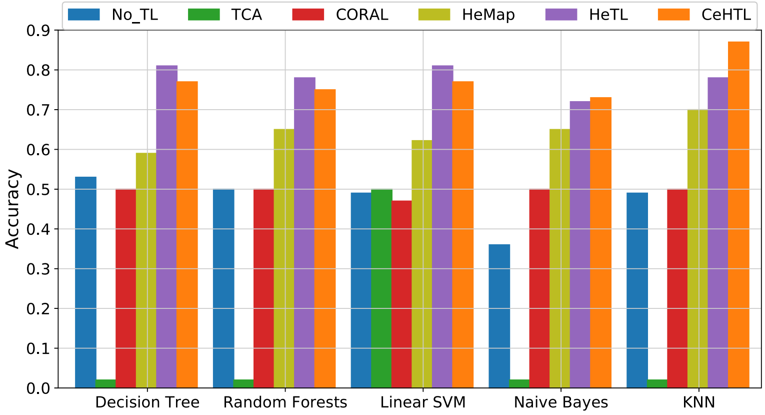


Fig. 7 Performance comparison of feature-based transfer learning approaches on DoS→ R2L

similarity confidence β can be determined by computing the similarity or distance between the source and target data, (b) the optimal number of both parameters can be found by enumerating the number of parameters, or (c) the parameters can be set empirically. However, the first and second approaches need a few labeled data from the target domain, which is not a truly "unknown" situation. We studied the impact of different parameter settings on the performance of detecting attacks. Figure 9 demonstrates the effect on accuracy by using different parameter combinations of β and k (where $\beta \in [0, 1]$ and k ranges from 1 to 6). Figures 10 and 11 demonstrate the average accuracy achieved on parameters β and k.

Compared with HeMap, both HeTL and CeHTL improve the highest accuracy achieved with different parameter settings, shown in Fig. 9. However, HeTL is sensitive to parameter tuning, showing lower accuracy in some specific parameter combinations. CeHTL performs more stably. For example, in DoS→Probe, after several fluctuation, CeHTL can maintain around 0.8 accuracy. For the similarity confidence parameter β, as shown in Fig. 10, CeHTL shows a significant improvement and stays stable from $\beta \geq 0$, because the correspondence has been automatically computed and involved in the transfer learning, so β should be set larger than 0. For the parameter k, in general, CeHTL shows an outstanding and stable performance than other approaches. The results show that CeHTL is more suitable for unknown network detection since we can empirically set the parameters and do not reply heavily on information about the labeled data in the target domain.

6.4.2 The imbalanced data effects

In many real cases, the size of normal and attack data would be not equal. Thus, we investigated the performance of the HeTL and CeHTL on imbalanced data. Figure 12 shows the F_1 score of the transfer learning approaches and baselines in different percentage of the attack data. We observed the baseline method performed poorly on the imbalanced data, especially

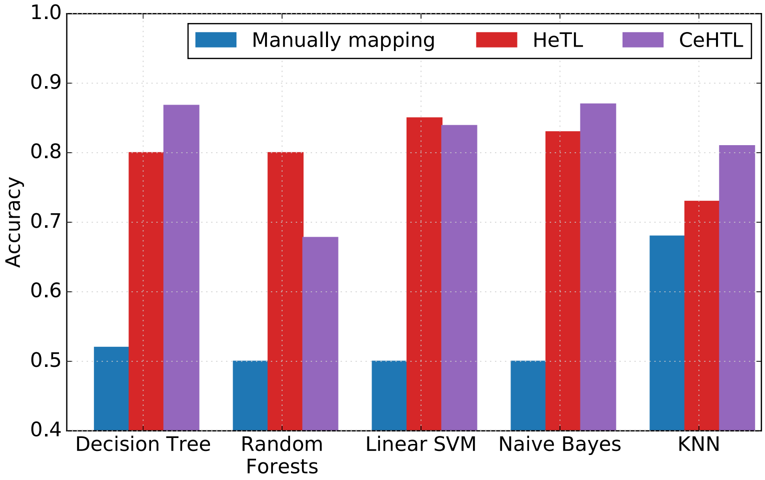


Fig. 8 Performance comparison on heterogeneous spaces on DoS→R2L

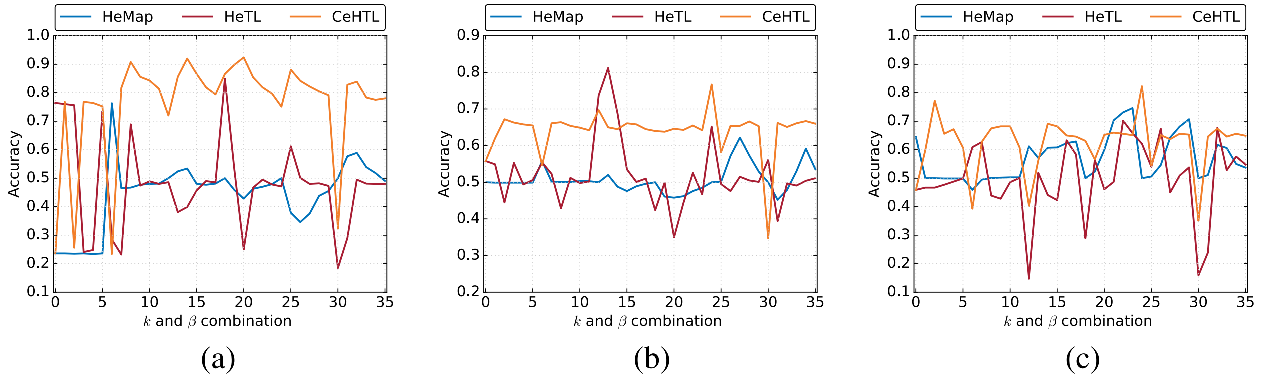


Fig. 9 Accuracy comparison with different combinations of k and β, sample = 1000. **a** DoS→Probe. **b** DoS→R2L. **c** Probe→R2L

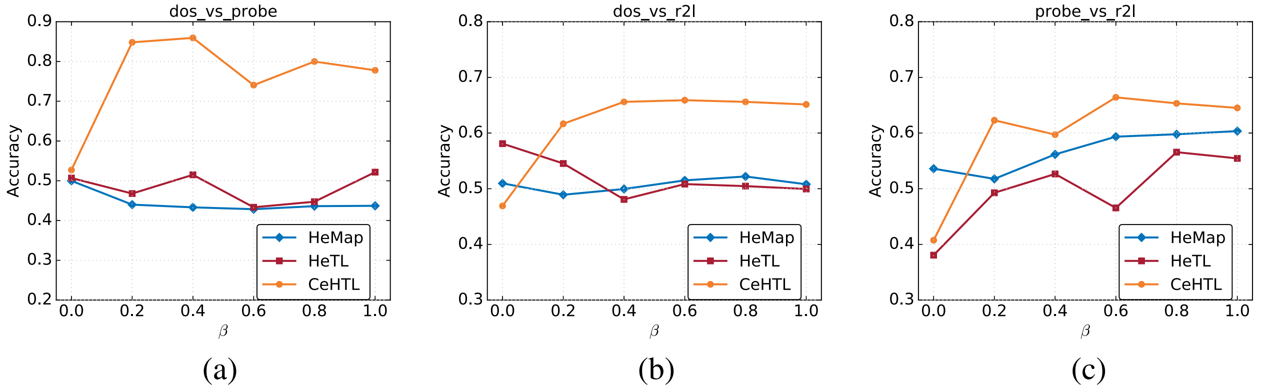


Fig. 10 Study of parameter β sensitivity on three main detection tasks, sample = 1000. **a** DoS→Probe. **b** DoS→R2L. **c** Probe→R2L

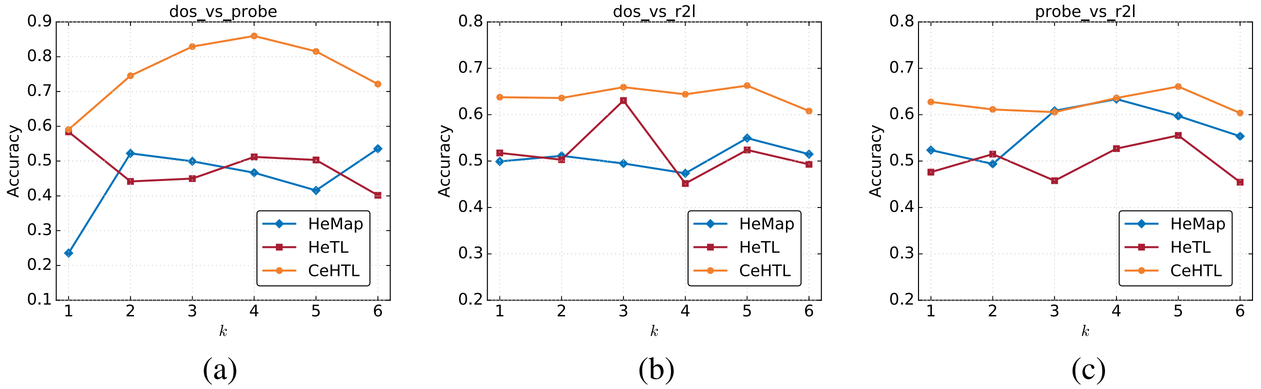


Fig. 11 Study of parameter k sensitivity on the three main detection tasks, sample = 1000. **a** DoS→Probe. **b** DoS→R2L. **c** Probe→R2L

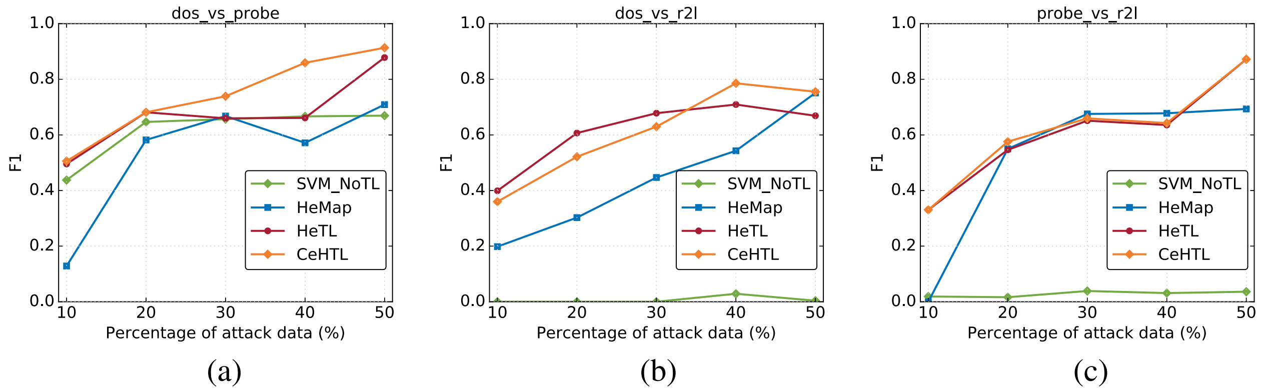


Fig. 12 The performance on imbalanced data by varying the portion of attack data, sample = 1000. **a** DoS→Probe. **b** DoS→R2L. **c** Probe→R2L

in DoS→R2L and Probe→R2L. The transfer learning approaches improved F_1 scores in most cases. Although all the methods had a lower F_1 score in 10% attack data, HeTL and CeHTL boosted the F_1 by 50% when adding another 10% of attack data, and the metric kept rising with increasing the attack data.

6.4.3 *The training size*

We studied how much training data was needed for unknown attack detection. We plot the learning curves in Fig. 13. From the results, we observed that CeHTL gained the best accuracy at a 500 sample size in DoS→Probe and DoS→R2L, and the second best accuracy in Probe→R2L. CeHTL needs the smallest training sample size, which makes it the best option given a limited amount of training data.

7 Conclusion

Machine learning have been employed in detecting the occurrence of malicious attacks. Most machine learning techniques for attack detection are effective only given the assumptions that the training and testing data are from the same distribution. However, in most real cases, continuously evolving attacks and the lack of sufficient labeled datasets hinder the ability of supervised learning techniques to detect new attacks. In this paper, we introduced a feature-based transfer learning framework and transfer learning approaches. We presented a feature-based transfer learning approach using a linear transformation, called HeTL. We also proposed a cluster enhanced transfer learning approach, called CeHTL, to make it more robust in detecting unknown attacks. We evaluated the transfer learning approaches on common classifiers. The results showed the transfer learning approaches improve the performance of detecting unknown network attacks compared to baselines. Spectacularly, CeHTL exhibited higher performance and the ability to be more robust in detecting unknown attacks with no labeled data. The results also demonstrated that the proposed transfer learning techniques can support different feature spaces. In the future, we aim to apply the model to various attack domains, such as malware detection. We also plan to combine transfer learning with deep learning to pre-train the models for practical use.

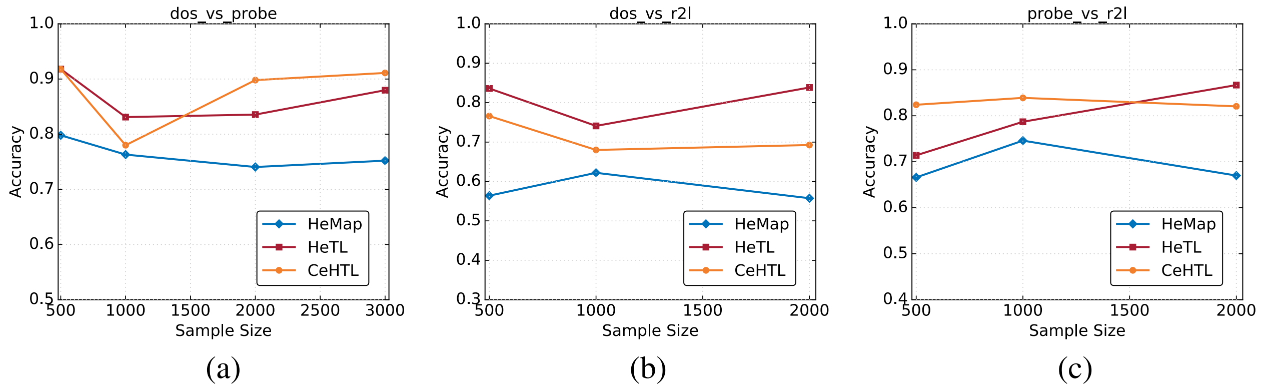


Fig. 13 Learning curves on different training size. **a** DoS→Probe. **b** DoS→R2L. **c** Probe→R2L

Appendix

Table 5 Top features for detecting DoS, used in the second experiment

Rank index	Features	Score
1	srv_serror_rate	0.504
2	serror_rate	0.500
3	flag	0.475
4	dst_host_srv_serror_rate	0.441
5	src_bytes	0.426
6	logged_in	0.417
7	dst_host_serror_rate	0.392
8	diff_srv_rate	0.383
9	dst_bytes	0.334
10	same_srv_rate	0.279
11	service	0.181
12	dst_host_diff_srv_rate	0.173
13	dst_host_same_srv_rate	0.162
14	wrong_fragment	0.161
15	dst_host_srv_diff_host_rate	0.150
16	dst_host_srv_count	0.150
17	count	0.138
18	dst_host_count	0.136
19	srv_diff_host_rate	0.135
20	duration	0.115

Table 6 Top features for detecting R2L, used in the second experiment

Rank index	Features	Score
1	srv_count	0.399
2	count	0.326
3	dst_host_srv_count	0.307
4	service	0.283
5	dst_bytes	0.243
6	src_bytes	0.231
7	hot	0.225
8	is_guest_login	0.215
9	protocol_type	0.208
10	srv_diff_host_rate	0.176
11	dst_host_srv_diff_host_rate	0.175
12	dst_host_same_src_port_rate	0.162
13	num_failed_logins	0.154
14	dst_host_count	0.127
15	flag	0.104

Acknowledgements
Not applicable.

Funding
The research presented in this paper was supported by Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R&E)) agreement FA8750-15-2-0120 and Boeing Data Analytics agreement BRT-L1015-0006.

Availability of data and materials
Not applicable.

Authors' contributions
JZ carried out the data processing, design and implementation of the proposed algorithms, experiment setup, and results evaluation and drafted the manuscript. SS contributed to the conception, experiment design, and evaluation of the proposed approach and results and helped draft the manuscript. JWP provided oversight for data and experimentation and participated in the manuscript editing. CK and KK helped revised the manuscript. All authors read and approved the final manuscript.

Authors' information
Juan Zhao is working as a postdoc fellowship in Department of Biomedical Informatics at Vanderbilt University. Before that, she was a research scientist and Adjunct Graduate Faculty in Tennessee State University. She received her PhD from University of Chinese Academy of Sciences in 2012 and B.E. from Shandong University in 2006, both degrees in Computer Science. She worked as an Associate Research Professor from 2012 to 2015 in Chinese Network Information Center, Chinese Academy of Sciences. Her research interests include machine learning in cyber security, bioinformatics, transfer learning, anomaly detection, feature engineering, natural language processing, and social network analysis.

Sachin Shetty is an associate professor in the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He holds a joint appointment with the Department of Modeling, Simulation and Visualization Engineering and the Center for Cybersecurity Education and Research. He received his PhD in Modeling and Simulation from the Old Dominion University in 2007 under the supervision of Prof. Min Song. Prior to joining Old Dominion University, he was an associate professor in the Electrical and Computer Engineering Department at Tennessee State University. His research interests lie at the intersection of computer networking, network security, and machine learning.

Jan Wei Pan has received his PhD from Virginia Polytechnic Institute and State University and B.E. from University of New South Wales, both degrees in Mechanical Engineering. He was an advanced technologist at The Boeing Company and the company's principal investigator on projects that specialize in machine learning, computer vision, and robotics. He was also a board-certified Professional Engineer in Virginia and an Adjunct Graduate Faculty in Tennessee State University. Currently, he is R&D Director in AutoX Inc. His research activities are related to various aspects of machine learning, such as deep learning, change detection, predictive analytics, data mining, and pattern recognition.

Charles Kamhoua received the BS in electronic from the University of Douala (ENSET), Cameroon, in 1999, and the MS in Telecommunication and Networking and the PhD in Electrical Engineering from Florida International University, in 2008 and 2011, respectively. In 2017, he joined the Network Security Branch of the US Army Research Laboratory, Adelphi, MD. From 2011 to 2017, he worked at the Cyber Assurance Branch of the US Air Force Research Laboratory (AFRL), Rome, NY, as a National Academies Post-doctoral Fellow and became a Research Electronics Engineer in 2012. His current research interests include the application of game theory to cyber security, survivability, cloud computing, hardware Trojan, online social network, wireless communication, and cyber threat information sharing.

Kevin Kwiat has been with the US Air Force Research Laboratory (AFRL) in Rome, NY, for over 32 years. Currently, he is assigned to the Cyber Assurance Branch. He received the BS in Computer Science and the BA in Mathematics from Utica College of Syracuse University and the MS in Computer Engineering and PhD in Computer Engineering from Syracuse University. He is also an adjunct professor of Computer Science at the State University of New York at Utica/Rome, an adjunct instructor of Computer Engineering at Syracuse University, and a research associate professor at the University at Buffalo. His main research interest is dependable computer design.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Vanderbilt University Medical Center, 37203 Nashville, USA. [2]Virginia Modeling Analysis and Simulation Center, Old Dominion University, 23529 Norfolk, USA. [3]AutoX Inc, San Jose, California, USA. [4]US Army Research Laboratory's Network Security Branch, 20783 Adelphi, USA. [5]Haloed Sun TEK, LLC, in affiliation with the CAESAR Group, Sarasota, Florida, USA.

Received: 3 July 2018 Accepted: 18 January 2019
Published online: 21 February 2019

References

1. R. Perdisci, W. Lee, N. Feamster, in *NSDI, vol. 10*. Behavioral clustering of http-based malware and signature generation using malicious network traces (USENIX Association, Berkeley, 2010), p. 14
2. C. Rossow, C. Dietrich, H. Bos, L. Cavallaro, M. V. Steen, F. C. Freiling, N. Pohlmann, in *BADGERS '11 Prof. of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. Sandnet: Network traffic analysis of malicious software, (2011), pp. 78–88
3. N. Stakhanova, M. Couture, A. A. Ghorbani, in *Prof. of the 2011 6th International Conf. on Malicious and Unwanted Software, Malware 2011*. Exploring network-based malware classification (IEEE Computer Society, Washington, DC, 2011), pp. 14–19
4. D. Bekerman, B. Shapira, L. Rokach, A. Bar, in *Communications and Network Security (CNS), 2015 IEEE Conference On*. Unknown malware detection using network traffic classification (IEEE, Los Alamitos, 2015), pp. 134–142
5. K. Bartos, M. Sofka, V. Franc, in *USENIX Security 2016*. Optimized invariant representation of network traffic for detecting unseen malware variants (USENIX Association, Austin, 2016), pp. 807–822
6. J. Zhao, S. Shetty, J. W. Pan, in *Military Communications Conference, (MILCOM)*. Feature-based transfer learning for network security (IEEE, Los Alamitos, 2017)
7. A. Javaid, Q. Niyaz, W. Sun, M. Alam, in *Proceedings of the 9th EAI International Conf. on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), BICT'15*. A deep learning approach for network intrusion detection system (ICST, ICST, 2016), pp. 21–26
8. F. Zhuang, X. Cheng, P. Luo, S. J. Pan, Q. He, in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15. Supervised representation learning: transfer learning with deep autoencoders (AAAI Press, 2015), pp. 4119–4125
9. K. D. Feuz, D. J. Cook, Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR). ACM Trans. Intell. Syst. Technol. **6**(1), 3–1327 (2015)
10. D. Lin, *Network intrusion detection and mitigation against denial of service attack. Technical Report MS-CIS-13-04*. (Department of Computer and Information Science Technical, University of Pennsylvania, 2013)
11. NSL-KDD, UNB IUNB ISCX NSL-KDD DataSet (2016). http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html. Accessed 01 May 2016
12. A. Valdes, K. Skinner, *Adaptive, Model-Based Monitoring for Cyber Attack Detection*. (H. Debar, L. Mé, S. F. Wu, eds.) (Springer, Berlin, Heidelberg, 2000), pp. 80–93
13. M. Hilker, C. Schommer, in *Conf.s in Research and Practice in Information Technology Series, vol. 54*. Description of bad-signatures for network intrusion detection (ACSW, 2006), pp. 175–182
14. H. Han, X.-L. Lu, L.-Y. Ren, in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference On, vol. 1*. Using data mining to discover signatures in network-based intrusion detection (IEEE, Los Alamitos, 2002), pp. 13–17
15. S. Nari, A. A. Ghorbani, in *Prof. of the 2013 International Conf. on Computing, Networking and Communications (ICNC)*. ICNC '13. Automated malware classification based on network behavior (IEEE Computer Society, Washington, 2013), pp. 642–647
16. M. Z. Rafique, P. Chen, C. Huygens, W. Joosen, in *Prof. of the 2014 conference on Genetic and evolutionary computation - GECCO '14*. Evolutionary algorithms for classification of malware families through different network behaviors (ACM, New York, 2014), pp. 1167–1174
17. F. Iglesias, T. Zseby, Analysis of network traffic features for anomaly detection. Mach. Learn. **101**(1-3), 59–84 (2014)
18. S. J. Pan, Q. Yang, A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
19. S. Bickel, M. Brückner, T. Scheffer, in *Prof. of the 24th International Conf. on Machine Learning*. ICML '07. Discriminative learning for differing training and test distributions (ACM, New York, 2007), pp. 81–88
20. W. Dai, Q. Yang, G.-R. Xue, Y. Yu, in *Prof. of the 24th International Conf. on Machine Learning. ICML '07*. Boosting for transfer learning (ACM, New York, 2007), pp. 193–200
21. T. Evgeniou, M. Pontil, in *Prof. of the Tenth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*. KDD '04. Regularized multi–task learning (ACM, New York, 2004), pp. 109–117
22. E. Bonilla, K. M. Chai, C. Williams, Multi-task Gaussian process prediction. Adv. Neural Inf. Process. Syst. **20**(October), 153–160 (2008)
23. B. Sun, J. Feng, K. Saenko, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI 16. Return of frustratingly easy domain adaptation (AAAI Press, 2016), pp. 2058–2065. http://dl.acm.org/citation.cfm?id=3016100.3016186
24. S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2011)
25. B. Kulis, K. Saenko, T. Darrell, in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf. On*. What you saw is not what you get: domain adaptation using asymmetric kernel transforms (IEEE Computer Society, Los Alamitos, 2011), pp. 1785–1792
26. X. Shi, Q. Liu, W. Fan, P. S. Yu, R. Zhu, in *Prof. - IEEE International Conf. on Data Mining, ICDM*. Transfer learning on heterogenous feature spaces via spectral transformation (IEEE, Los Alamitos, 2010), pp. 1049–1054
27. J. Nam, S. J. Pan, S. Kim, in *Prof. of the 2013 International Conf. on Software Engineering*. ICSE '13. Transfer defect learning (IEEE Press, Piscataway, 2013), pp. 382–391
28. B. Long, Y. Chang, A. Dong, J. He, in *WSDM*. Pairwise cross-domain factor model for heterogeneous transfer ranking (ACM, New York, 2012), p. 113
29. S. Gou, Y. Wang, L. Jiao, et al., in *2009 IEEE International Symposium on Parallel and Distributed Processing with Applications*. Distributed transfer network learning based intrusion detection (IEEE, Los Alamitos, 2009), pp. 511–515
30. J. Gao, W. Fan, J. Jiang, J. Han, in *Prof. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Knowledge transfer via multiple model local structure mapping (ACM, New York, 2008), pp. 283–291
31. D. Arthur, S. Vassilvitskii, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. K-means++: the advantages of careful seeding (Society for Industrial and Applied Mathematics, Philadelphia, 2007), pp. 1027–1035. http://dl.acm.org/citation.cfm?id=1283383.1283494
32. H. Abdi, L. J. Williams, Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. **2**(4), 433–459 (2010)