Deep Transfer Learning for IoT Attack Detection

Prof. Abdul Khadar A, Sanjay Yadav S, Mohammed Huzaifa B, Vayalpad Mohammad Thouhid
Department of Information Science and Engineering,
SJC Institute of Technology, Chikkaballapur, Karnataka, India

Abstract*: The digital revolution has substantially changed our lives in which Internet-of-Things (IoT) plays a prominent role. The rapid development of IoT to most corners of life, however, leads to various emerging cybersecurity threats. Therefore, detecting and preventing potential attacks in IoT networks have recently attracted paramount interest from both academia and industry. Among various attack detection approaches, machine learning-based methods, especially deep learning, have demonstrated great potential thanks to their early detecting capability. However, these machine learning techniques only work well when a huge volume of data from IoT devices with label information can be collected. Nevertheless, the labeling process is usually time consuming and expensive, thus, it may not be able to adapt with quick evolving IoT attacks in reality. In this paper, we propose a novel deep transfer learning (DTL) method that allows to learn from data collected from multiple IoT devices in which not all of them are labeled. Specifically, we develop a DTL model based on two AutoEncoders (AEs). The first AE (AE1) is trained on the source datasets (source domains) in the supervised mode using the label information and the second AE (AE2) is trained on the target datasets (target domains) in an unsupervised manner without label information. The transfer learning process attempts to force the latent representation (the bottleneck layer) of AE2 similarly to the latent representation of AE1. After that, the latent representation of AE2 is used to detect attacks in the incoming samples in the target domain. We carry out intensive experiments on nine recent IoT datasets to evaluate the performance of the proposed model. The experimental results demonstrate that the proposed DTL model significantly improves the accuracy in detecting IoT attacks compared to the baseline deep learning technique and two recent DTL approaches.*

Keywords: Deep transfer learning, IoT, cyberattack detection, AutoEncoder

I. INTRODUCTION

The Internet-of-Things (IoT) refers to connected devices, sensors, an actuators used in vehicles, electronic appliances, buildings, and structures. As the sensors, data storage, and the Internet become cheaper, faster, and more integrated together, IoT devices will find more and more applications (e.g., in smart buildings, smart city, intelligent transportation sys- tems, and healthcare). The rapid development of IoT to most corners of life, however, leads to various emerging cyberse- curity threats. This is because IoT devices are often limited in computing capability and energy, making them particu- larly vulnerable to adversaries. IoT devices are more exposed to and unfortunately more difficult to be protected from cyber attacks than computers. Consequently, detect- ing attacks to protect IoT devices from malicious behaviors is critical to broadening the applications of IoT .

IoT attack detection methods can be categorized into signature-based and machine learning-based methods. The signature-based methods seek to find the signatures of IoT attacks in the incoming traffic. These methods require a high prior knowledge of known IoT attacks to define the signatures. The machine learning-based meth- ods, on the other hand, attempt to learn the features of normal and malicious data in the training/offline phase. In the predicting/online phase, these models are used to detect attacks in the incoming traffic. Thanks to the capability to automatically and progressively learn useful information/features from collected data, machine-learning based methods can early detect various IoT attacks

However, the machine learning-based methods only per- form well under an important assumption, i.e., the distributions of the training data and the predicting data are similar [18]. Nevertheless, in many practical applications, this assumption may not be always the case. Especially, in network security, new types of attacks (e.g., zero-day attacks) can be found on a daily basis . As such, the practical IoT data for machine learning models (in the predicting/online

DOI: 10.48175/IJARSCT-10384

ISSN
2581-9429
IJARSCT

phase) is usually very much different from the data used during the training/offline phase. To alleviate the above problem, a large volume of training data with label from multiple IoT devices is often required. However, manually labeling a huge volume of data is very time con- suming and expensive. It, thus, limits the practical deployment of machine learning-based methods in detecting IoT attacks for various scenarios.

Given the above, this work proposes a novel deep trans- fer learning (DTL) approach based on AutoEncoder (AE) to enable further applications of machine learning in IoT attack detection. The proposed model is referred to as Multi-Maximum Mean Discrepancy AE (MMD-AE). MMD-AE can be trained on a dataset including both labeled samples (in the source domain) and unlabeled samples (in the target domain). After training, MMD-AE is used to predict IoT attacks in the incoming traffic in the target domain. Specif- ically, MMD-AE consists of two AEs: AE1 and AE2. AE1 in trained with labeled data while AE2 is trained on the unlabeled data. The whole model, i.e., MMD-AE, is trained to drive the latent representation of AE2 closely to the latent representation of AE1. As a result, the latent representation of AE2 can be used to classify the unlabeled IoT data in the target domain. The major contributions of this paper are as follows:

We propose a novel DTL model based on AEs, i.e., MMD-AE, that allows to transfer knowledge,

i.e., labeled information, from the source domain to the target domain. This model helps to lessen the problem of ‘‘lack label information’’ in collected traffic datasets from IoT devices.

We introduce the Maximum Mean Discrepancy (MMD) metric to minimize the distance between multiple hidden layers of AE1 and multiple hidden layers of AE2. This metric helps to improve the effectiveness of knowledge transferred from the source to the target domain in IoT attack detection systems.

We experiment our proposed method using nine IoT attack datasets and compare its performance with the canonical deep learning model and the state-of-the-art TL models .The experimental results demon- strate the advantage of our proposed model against the other tested methods.

The rest of paper is organized as follows. Section II high- lights recent works on IoT attack detection. In Section III, we define a DTL model and briefly describe the AE archi- tecture. The proposed model is then presented in Section IV. Section V discusses the experiment settings and Section VI

provides detailed analysis and discussion related to exper- imental results. Finally, Section VII concludes with future work.

II. RELATED WORK

There are two main directions for cyberattack detection, i.e., signature-based and machine learning-based approaches, e.g., .The signature-based methods maintain a database of predefined signatures (i.e., patterns) that corre- spond to IoT known attacks and perform the detection task by comparing these to the incoming data stream. Zhang and Green II proposed a lightweight and low-complexity algorithm to prevent Distributed Denial of Service (DDoS) attacks in which each IoT working node has a deep packet inspection to find attack signatures. If a sender repeatedly sends requests with the same content, it will be flagged as malicious requests. Dietz *et al.* proposed a solution to proactively block the spreading of IoT attacks and isolate vulnerable IoT devices. Each IoT device is ver- ified in two steps, i.e., scanning to open ports and services and using predefined list of commonly known credentials to check authentication. After that, a list of predefined rules is used to isolate the vulnerable IoT devices. Nobakht *et al.* proposed a solution for IoT attack detection using Software Defined Network with the OpenFlow protocol to address malicious behaviours and block intruders from accessing the IoT devices. This method incorporates a database of all known in-home IoT devices along with the correspond- ing patterns of potential security risks. Then, the detection method simply maps the IoT traffic with the signatures of security risks stored in the database. The advantage of the signature-based methods is providing a low false positive rate attack detection system. However, they require a prior human knowledge about the behaviours of known IoT attacks to design the database of attack signatures. Thus, the accuracy of these methods depends on the quality of the signature databases. Moreover, if the size of databases is increased, the processing time (i.e., search time) can be excessive [24].

The machine learning-based methods first train the detection models from collected data samples in IoT networks. Then, the trained models are used to classify the new incoming IoT data samples into normal or attack data. The popular traditional machine learning algorithms for IoT attack detection are Decision tree Support Vector Machine (SVM), K-Nearest Neighbour, Bayes Classifier, Neural Net- works. Recently, the deep learning approach is widely used and achieved high performance in detect- ing cyberattacks. Among, deep learning approaches, AE-based models project the original data to a new latent representation space to improve the accuracy in detection tasks. Nevertheless, to train a good machine learning model for detecting IoT attacks, it is usually required to label a huge volume of training data as normal or attack. Moreover, general machine learning models often need to assume that the data distribution of training datasets is similar to the data distribution of predicting datasets. This assumption, however, is usually not practical .Recently, DTL techniques have been used to handle the above issues of machine learning methods where training datafrom a source domain and test data from a target domain aredrawn from different distributions. A DTL model attempts toreduce the distribution divergence between the source domainand the target domain. As a result, the trained knowl-edge of a learning task (e.g., classification) on the sourcedomain can be used to support the learning task on the similar target domain. Gou *et al.*applied aninstance-based DTL approach in network intrusion detec-tion that requires label information from the target domain.Zhao *et al.* [] proposed the feature-based DTL technique

Our proposed DTL model in this paper, i.e., MMD-AE, leverages a non-linear mapping, i.e., AE, to improve the performance of IoT attack detection on the target domain. The key idea of our proposed DTL (compared with previous AE-based DTL methods) is that the knowledge of features in every encoding layers (instead of the only bottleneck layer in previous works) is transferred to the target domain. This helps to force the latent representation of the tar- get domain similarly to the latent representation of the source domain. The experimental results illustrate the effectiveness of our proposed DTL model on the IoT attack detection task in the target domain.

III. FUNDAMENTAL BACKGROUND

This section presents the fundamental background of our proposed model.

TRANSFER LEARNING

Transfer learning (TL) refers to the situation where what has been learned in one learning task is exploited to improve generalization in another learning task. Fig. 1 compares traditional machine learning methods including deep learning and TL models. In traditional machine learning, the datasets

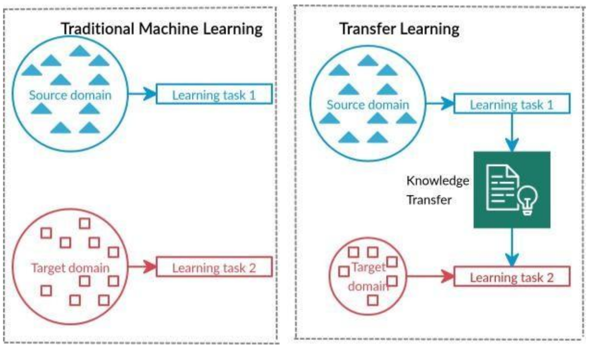


FIGURE 1. Traditional machine learning vs. transfer learning.

and training processes are separated for different learning tasks. Thus, no knowledge is retained/accumulated nor transferred from one model to another. In TL, the knowledge (i.e., features, weights, etc.) from previously trained models in a source domain is used for training newer models in a target domain. Moreover, TL can even handle the problems of having less data or no label information in the target domain. TL is often used to transfer knowledge learnt from a source domain to a target domain where the target domain is different from the source domain but they are related data distributions. We consider a TL method with an input space X .

the number of samples in the source domain and the target domain, respectively. In this paper, the TL model based on a deep neural network, i.e., deep transfer learning (DTL), is trained on the labeled data in the source domain and the unlabeled data in the target domain. After that, the trained model is used for IoT attack detection in the target domain.

DOI: 10.48175/IJARSCT-10384

AUTO ENCODERS

This subsection describes the structure and the training pro- cess of an AutoEncoder (AE) that is fundamental for our DTL model. The reason we develop the TL models based on AE is that these models are proved as the most effective deep neural network for IoT attack detection. Addi- tionally, to prove the effectiveness of the proposed model, we will compare our proposed model with the previous DTL techniques that are also based on AE.

An AE is a neural network trained to reconstruct the network's input at its output . This network has two parts, i.e., encoder and decoder as shown in Fig. 2. Let $\boldsymbol{W}$, $\boldsymbol{W}^{\boldsymbol{S}}$, $\boldsymbol{b}$, and $\boldsymbol{b}^{\boldsymbol{S}}$ denote the weight matrices and the bias vectors of the encoder and the decoder, respectively, and

$X = x^1, x^2, \ldots, x^n$ is a training dataset. $\varphi = (\boldsymbol{W}, \boldsymbol{b})$ and $\theta = (\boldsymbol{W}^{\boldsymbol{S}}, \boldsymbol{b}^{\boldsymbol{S}})$ are parameter sets for training the encoder and the decoder, respectively. Let $q\varphi$ denote the encoder and z^i

denote the representation of the input data x^i. The encoder maps the input x^i to the latent representation z^i (as in (1)). The decoder $p\theta$ attempts to map the latent representation z^i back

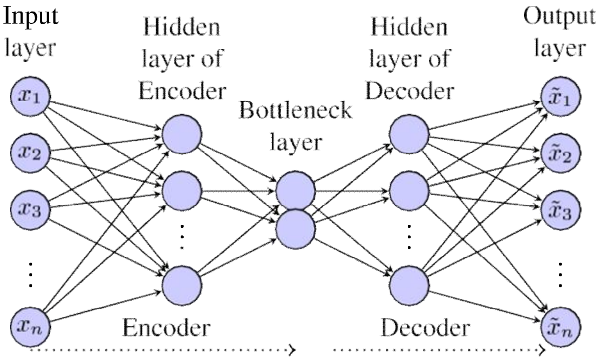


FIGURE 2. Architecture of an AutoEncoder(AE)

into the input space. Therefore, the output of the decoder is formed as the input space, i.e., $x\hat{}^{i}$ (as in (2)).

$z^i = q\varphi(x^i) = af(\boldsymbol{W}x^i + \boldsymbol{b}),$ (1)

$x\hat{}^{i} = p\theta(z^i) = ag(\boldsymbol{W}^{\boldsymbol{S}}z^i + \boldsymbol{b}^{\boldsymbol{S}}),$ (2)

where af and ag are the activation functions of the encoder and the decoder, respectively. Fig. 2 shows an example of AE with input dimension as n, number of layers as 5, bottleneck layer size as 2.

The AE model is trained by minimizing a loss function so called Reconstruction Error (RE). RE is the difference between the input x^i and the output $x\hat{}^{i}$ as in (3). This term encourages the decoder to learn to reconstruct the original

data. If the decoder's output does not reconstruct the data well, it will incur a large cost in this loss term

where $l\ x^i, x\hat{}^{i}$ measures the difference between the input x^i and the output $x\hat{}^{i}$. In the AE model, the mean squared error (MSE) is commonly used

MAXIMUM MEAN DISCREPANCY (MMD)

Maximum mean discrepancy (MMD) is a metric used to estimate the discrepancy of two distributions. MMD is more flexible than Kullback-Libler divergence (KL) [31] thanks to its ability to estimate the nonparametric distance [35]. More- over, MMD does not require to compute the intermediate density of the distributions, thus avoiding the requirement of using a sophisticated optimization [36]. The definition of MMD of two datasets can be formulated as (4) [37].

IV. PROPOSED TRANSFER LEARNING APPROACH FOR IoT CYBERATTACK DETECTION

This section presents our proposed DTL models for IoT attack detection. We first describe the overview of the system structure. After that, the DTL model is discussed in details.

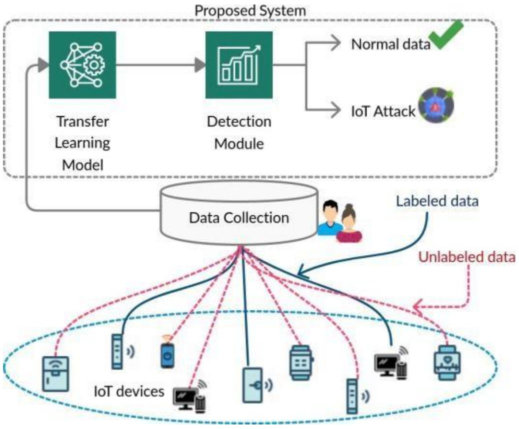
A. SYSTEM STRUCTURE

Fig. 3 presents the system structure that uses DTL for IoT attack detection. First, the data collection module gathers data from all IoT devices. The training data consists of both labeled and unlabeled data. The labeled data is collected from some IoTs devices which are dedicated for labeling data. The labeling process is usually executed in two steps [22]:

each data sample is extracted from captured packets using Tcptrace tool [38], then the data sample is labeled as a normal sample or an attack sample by manually analyzing the flow using Wireshark software. Usually, the number of labeling IoT devices is much smaller than the number of unlabeling IoT devices. Second, the collected data is passed to the DTL model for training. The training process attempts to transfer the knowledge information learnt from the data with label information to data without label information. This is achieved by minimizing the difference between latent representations of the source data and the target data. After training, the trained DTL model is used in the detection module that can classify incoming traffic from all IoT devices as normal or attack data. The detailed description of the DTL model is presented in the next subsection

B. TRANSFER LEARNING MODEL

The proposed DTL (i.e., MMD-AE) model includes two AEs (i.e., AE1 an AE2) that have the same architecture as



Fig;3 Proposed system structure.

We assume that φS , θS , φT , θT are the parameter sets of encoder and decoder of AE1 and AE2, respectively. The first term, 4RE including RES and RET in Fig. 4, attempts to reconstruct the input layers at the output layers of both AEs. In other words, the RES and RET try to reconstruct the input

data xS and xT at their output from the latent representations zS and zT , respectively. Thus, this term encourages two AEs to retain the useful information of the original data at the latent representation. Consequently, we can use latent representations for classification tasks after training. Formally, the 4RE

term is calculated as follows:

$$(x^i , \varphi S , \theta S, x^i , \varphi T , \theta T) = l(x^i , x^{\hat{}i}) + l(x^i , x^{\hat{}i}), \quad (5)$$

DOI: 10.48175/IJARSCT-10384

ISSN 2581-9429 IJARSCT

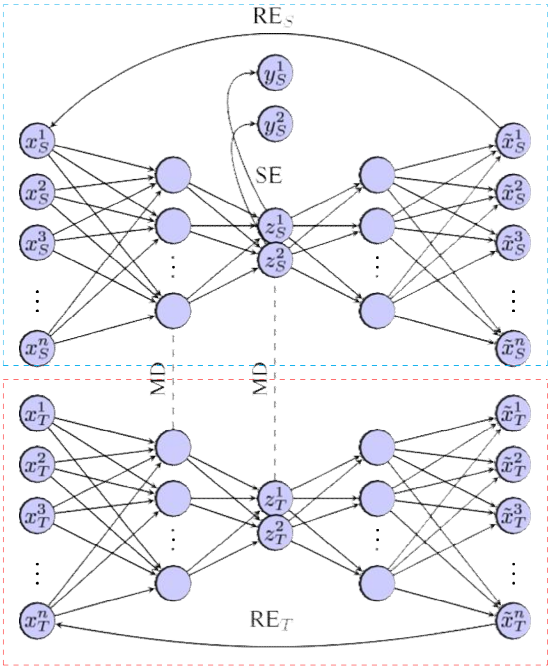


FIGURE 4. Architecture of MMD-AE.

where l function is the MSE function [1$_S$6],$_S x^i$,$_T \hat{x}^i$ $_T$ x^i , $\hat{x}^i$ are the data samples of input layers and the output layers of the source domain and the target domain, respectively.

The second term 4_{SE} aims to train a classifier at the latent representation of AE1 using labeled information in the source domain. In other words, this term attempts to map the value at two neurons at the bottleneck layer of AE1, i.e., z_S , to their label information y_S . This is achieved by using the softmax function [33] to minimize the difference between z_S and y_S . It should be noted that, the number of neurons in the bottleneck layer must be the same as the number of classes in the source domain. This loss encourages to distinguish the latent representation space from separated class labels. Formally, this loss is defined as follows:

where z^i and y^i are the latent representation and labels of the
source data sample x_S . y_S and z_S represent the $j-th$ element

The third term 4_{MMD} is to transfer the knowledge of the source domain to the target domain. The transferring process is executed by minimizing the MMD distances between every encoding layers of AE1 and the corresponding encoding lay- ers of AE2. This term aims to make the representations of the source data and target data close together. The 4_{MMD} loss term is described as follows:

where K is the number of encoding layers in the AE-based model. ξ^k $(x^i$) and ξ^k $(x^i$) are the encoding layers k-th of AE1 and AE2, respectively, MMD(,) is the MMD distance presenting in$_S$ (4). $_T$ $_T$

The final loss function of MMD-AE combines the loss terms in (5), (6), and (8) as in (7).

$4 = 4_{\mathrm{SE}} + 4_{\mathrm{RE}} + 4_{\mathrm{MMD}}.$ (8)

Algorithm 1 presents the pseudo-code for training our proposed DTL model. The training samples with labels in the source domain are input to AE1 while the training samples without labels in the target domain are input to AE2. The training process attempts to minimize the loss function in (8)). After training, AE2 is used to classify the testing samples in the target domain as in Algorithm 2.

```
Algorithm 1 Training the Proposed DTL Model
INPUT:
$xS, yS$ : Training data samples and corresponding labels in the source domain
$xT$ : Training data samples in the target domain OUTPUT: Trained models: AE2.
BEGIN:
Put $xS$ to the input of AE1
Put $xT$ to the input of AE2
$\xi k$ ($xS$ ) is the representation of $xS$ at the layer $k$ of AE1
$zS$ is the representation of $xS$ at the bottleneck layer of AE1
$\xi k$ ($xT$ ) is the representation of $xT$ at the layer $k$ of AE2
Training the TL model by minimizing the loss function in (8)
return Trained models: AE1, AE2. END.
```

```
Algorithm 2 Classifying on the Target Domain INPUT:
$xT$ : Testing data samples in the target domain
Trained AE2 model
OUTPUT: $yT$ : Label of $xT$
BEGIN:
Put $xT$ to the input of AE2
$zT$ is the representation of $xT$ at the bottleneck layer of
$yT$ = softmax ($zT$ )
return $yT$
END.
```

Our key idea in the proposed model, i.e., MMD-AE, com- pared with the previous DTL model is to transfer the knowledge not only in the bottleneck layer but also in every encoding layer from the source domain, i.e., AE1, to the target domain, i.e., AE2. In other words, MMD-AE allows to transfer more knowledge from the source domain to the target domain. One possible limitation of MMD-AE is that it may incur the overhead time in the training process

TABLE 1. Description of IoT datasets.

Dataset	Device Name	Training Attacks	Training size	Testing size
IoT-1	Danmini Doorbell	combo, ack	239488	778810
IoT-2	Ecobee Thermostat	combo, ack	59568	245406
IoT-3	Ennio Doorbell	combo, tcp	174100	181400
IoT-4	Philips_B120N10 Baby_Monitor	tcp, syn	298329	800348
IoT-5	Provision_PT 737E_Security Camera	combo, ack	153011	675249
IoT-6	Provision_PT 838_Security Camera	ack, udp	265862	261989
IoT-7	Samsung_SNH 1011_N Webcam	combo, tcp	182527	192695

ISSN 2581-9429 IJARSCT

IoT-8	SimpleHome XCS7_1002 WHT_Security Camera	combo, ack	189055	674001
IoT-9	SimpleHome XCS7_1003 WHT_Security Camera	combo, ack	176349	674477

since the distance between multiple layers of the encoders in AE1 and AE2 is evaluated. However, in the predicting phase, only AE2 is used to classify incoming samples in the target domain. Therefore, this model does not lead to increasing the predicting time compared to other AE-based models.

V. EXPERIMENTAL SETTING

This section presents the datasets, the performance metrics, the hyper-parameter settings and the sets of the experiments in our paper.

DATASETS

To evaluate the performance of MMD-AE we used nine IoT attack detection datasets from Meidan *et al.* [3]. These datasets were collected from nine commercial IoT devices in their lab. Each IoT dataset includes five or ten DDoS attacks based on types of IoT devices, such as Scanning the network for vulnerable devices (scan), Sending spam data (Junk), UDP flooding (udp), TCP flooding (tcp), and Sending spam data and opening a connection to a specified IP address and port (combo). Each dataset is divided into a training set (70% benign data samples and two random types of attacks) and the testing set (30% benign data samples and the rest of attacks). Thus, many attack types are not included in the training data. Each data sample has 115 attributes extracted from the packet stream. The number of training and testing datasets is presented in Table 1.

EVALUATION METRIC

To evaluate the effectiveness of the proposed model, we use a popular performance metric, i.e., Area Under the Curve (AUC) score. The advantage of AUC includes two aspects. First, it is scale-invariant. In other words, the AUC score measures how well predictions are ranked, rather than their absolute values. Second, AUC is classification-threshold-invariant. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

The AUC score is created by plotting the True Positive Rate (TPR) or Sensitivity[1] against the False Positive Rate (FPR)[2] at various threshold settings. The space under the ROC curve is represented as the AUC score [40]. This measures the average quality of the classification model at differ- ent thresholds.

HYPER-PARAMETERS SETTING

The same configuration is used for all AE-based models in our experiments. This configuration is based on the AE-based models for detecting network attacks in the literature . As we integrate the 4_{SE} loss term to MMD-AE, the number of neurons in the bottleneck layer is equal to the number of classes in the IoT dataset, i.e., 2 neurons in this paper. The number of layers including both the encoding lay- ers and the decoding layers is 5. The ADAM algorithm is used for optimizing the models in the training process. The ReLu function is used as an activation function of AE layers except for the last layers of the encoder and decoder where the Sigmoid function is used. For all datasets, we select 10% of training data as the validation sets for early stopping. This technique helps to stop training process automatically. The performance of each model is evaluated on the validation set at the end of each 10 epochs. If the the AUC score is reduced, the training procedure will be stopped.

EXPERIMENTAL SETS

We carried out three sets of experiments in this paper. The first set is to investigate how effective our proposed model is at transferring knowledge from the source domain to the target domain. We compare the MMD distances between the bottleneck layer of the source domain and the target domain after training when the transferring process is executed

ISSN
2581-9429
IJARSCT
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY

in one, two, and three encoding layers. The smaller MMD distance, the more effective transferring process from the source to the target domain.

The second set is the main result of the paper in which we compare the AUC scores of MMD-AE with AE and two recent DTL models. All methods are trained using the training set including the source dataset with label infor- mation and the target dataset without label information. After training, the trained models are evaluated using the target dataset. The methods compared in this experiment include the original AE (i.e., AE), and the DTL model using the

[1] TPR measures the proportion of actual positive samples that are correctly identified.

[2] FPR measures the ratio between the number of negative samples wrongly categorized as positive samples (false positives) and the total number of actual negative samples

KL metric at the bottleneck layer (i.e., SKL-AE), the DTL method of using the MMD metric at the bottleneck[2] layer (i.e., SMD-AE), and our model (MMD-AE).

The third set is to measure the processing time of the training and the predicting process of the above evaluated methods. The detailed results of three experimental sets are presented in the next section.

VI. RESULTS

This section presents the result of three sets of the experiments in our paper

EFFECTIVENESS OF TRANSFERRING INFORMATION IN MMD-AE

MMD-AE implements multiple transfer between encoding layers of AE1 and AE2 to force the latent representation of AE2 closer to the latent representation of AE1. In order to evaluate if MMD-AE achieve its objective we conducted an experiment in which, IoT-1 is selected as the source domain and IoT-2 is the target domain. We measured the MMD distance between the latent representation, i.e., the bottleneck layer, of AE1 and AE2 when the transfer information is implemented in one, two and three layers of the encoders. The smaller distance is, the more information is transferred from the source domain (AE1) to the target domain (AE2). The result is presented in Fig. 5.

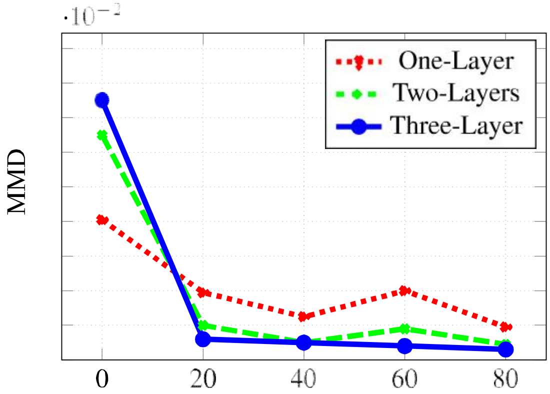


FIGURE 5. MMD of latent representations of the source (IoT-1) and the target (IoT-2) when transferring task on one, two, and three encoding layers.

The figure shows that transferring task implemented on more layers results in the smaller MMD distance value. In other words, more information can be transferred from the source to the target domain when the transferring task is implemented on more encoding layers. This result evidences that our proposed solution, MMD-AE, is more effective than the previous DTL models performing the transferring task only at the bottleneck layer of AE.

PERFORMANCE COMPARISON

Table 2 represents the AUC scores of AE, SKL-AE, SMD-AE and MMD-AE when they are trained on the dataset with

DOI: 10.48175/IJARSCT-10384

ISSN
2581-9429
IJARSCT

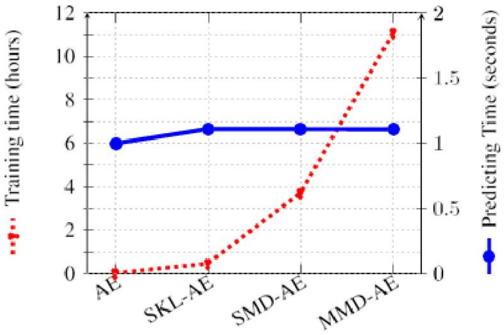


FIGURE 6. Training and testing of AE, SKL-AE, SMD-AE, and MMD-AE when the source domain is IoT-2 the target domain is IoT-1.

label information in the columns and the dataset without information in the rows and tested on the dataset in the rows. In this table, the result of MMD-AE is printed in bold face. We can observe that AE is the worst method among the tested methods. Apparently, when an AE is trained on an IoT dataset (the source) and evaluating on other IoT datasets (the target), its performance is not effective. The reason for this ineffective result is that the predicting data in the target domain is far different from the training data in the source domain.

Conversely, the results of three DTL models are much better than that of AE. For example, if the source dataset is IoT-1 and the target dataset is IoT-3, the AUC score is improved from 0.600 to 0.745 and 0.764 with SKL-AE and SMD-AE, respectively. These results prove that using DTL helps to improve the accuracy of AEs on detecting IoT attacks on the target domain.

More importantly, our proposed method, i.e., MMD-AE, usually achieves the highest AUC score in almost all IoT datasets.[3] For example, the AUC score is 0.937 compared to 0.600, 0.745, 0.764 of AE, SKL-AE and SMD-AE, respec-

tively, when the source dataset is IoT-1 and the target dataset is IoT-3. The results on the other datasets are also similar to the results on IoT-3. These results demonstrate that imple- menting the transferring task in multiple layers of MMD-AE helps the model to transfer the label information from the source to the target domain more effectively. Subsequently, MMD-AE often achieves better results compared to AE, SKL-AE and SMD-AE in detecting IoT attacks in the target domain.

PROCESSING TIME ANALYSIS

Fig. 6 shows the training and the predicting time of the tested model when the source domain is IoT-2 and the target domain is IoT-1.[4] In this figure, the training time is measured in **hours** and the predicting time is measured in **seconds**. It can be seen that, the training process of the DTL methods

[3] The AUC scores of the proposed model in each scenario is presented by the bold text style.

[4] The results on the other datasets are similar to this result

TABLE 2. AUC scores of AE, SKL-AE, SMD-AE, and MMD-AE on nine IoT datasets

Target	Model	Source								
		IoT-	IoT-2	IoT-3	IoT-4	IoT-5	IoT-6	IoT-7	IoT-	IoT-9
IoT-1	AE		0.705	0.542	0.768	0.838	0.643	0.791	0.63	0.600
			0.700	0.759	0.855	0.943	0.729	0.733	0.68	0.705
			0.722	0.777	0.875	0.943	0.766	0.791	0.70	0.705
			0.888	**0.796**	**0.885**	**0.943**	**0.833**	**0.892**	**0.77**	**0.743**
IoT-2	AE	0.54		0.500	0.647	0.509	0.743	0.981	0.77	0.578
		0.54		0.990	0.708	0.685	0.794	0.827	0.64	0.606
		0.56		0.990	0.815	0.689	0.874	0.871	0.77	0.607
		0.93		**0.990**	**0.898**	**0.692**	**0.878**	**0.900**	**0.78**	**0.609**

DOI: 10.48175/IJARSCT-10384

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY
ISSN
2581-9429
IJARSCT

IoT-3	AE	0.60	0.659		0.530	0.500	0.501	0.644	0.80	0.899
		0.74	0.922		0.566	0.939	0.534	0.640	0.93	0.916
		0.76	0.849		0.625	0.879	0.561	0.600	0.91	0.938
		0.93	**0.956**		**0.978**	**0.928**	**0.610**	**0.654**	**0.93**	**0.946**
IoT-4	AE	0.70	0.740	0.817		0.809	0.502	0.944	0.80	0.800
		0.76	0.852	0.837		0.806	0.824	0.949	0.83	0.809
		0.77	0.811	0.840		0.803	0.952	0.947	0.80	0.826
		0.93	**0.857**	**0.935**		**0.844**	**0.957**	**0.959**	**0.87**	**0.850**
IoT-5	AE	0.61	0.598	0.824	0.670		0.920	0.803	0.79	0.698
		0.64	0.639	0.948	0.633		0.923	0.695	0.80	0.635
		0.66	0.576	0.954	0.672		0.945	0.822	0.78	0.833
		0.66	**0.508**	**0.954**	**0.679**		**0.928**	**0.847**	**0.81**	**0.928**
IoT-6	AE	0.82	0.823	0.699	0.834	0.936		0.765	0.83	0.737
		0.86	0.897	0.711	0.739	0.980		0.893	0.78	0.881
		0.87	0.898	0.713	0.849	0.982		0.778	0.86	0.898
		0.92	**0.899**	**0.787**	**0.846**	**0.992**		**0.974**	**0.87**	**0.898**
IoT-7	AE	0.50	0.501	0.626	0.791	0.616	0.809		0.59	0.459
		0.50	0.625	0.865	0.831	0.550	0.906		0.35	0.524
		0.51	0.619	0.865	0.817	0.643	0.884		0.61	0.604
		0.54	**0.621**	**0.888**	**0.897**	**0.858**	**0.905**		**0.61**	**0.618**
IoT-8	AE	0.81	0.599	0.831	0.650	0.628	0.890	0.901		0.588
		0.61	0.636	0.892	0.600	0.629	0.923	0.907		0.712
		0.62	0.639	0.902	0.717	0.632	0.919	0.872		0.629
		0.73	**0.636**	**0.964**	**0.723**	**0.692**	**0.977**	**0.943**		**0.616**
IoT-9	AE	0.82	0.601	0.840	0.851	0.691	0.808	0.885	0.57	
		0.81	0.602	0.800	0.731	0.662	**0.940**	0.855	0.56	
		0.83	0.609	0.892	0.600	0.901	0.806	0.886	0.62	
		0.84	**0.911**	**0.910**	**0.874**	**0.904**	**0.829**	**0.889**	**0.64**	

(i.e., SKL-AE, SMD-AE, and MMD-AE) is more time con- suming than that of AE. One of the reason is that DTL models need to evaluate the MMD distance between the AE1 and AE2 at every iteration while this calculation is not required in AE. Moreover, the training time of MMD-AE is even much higher than those of SKL-AE and SMD-AE since MMD-AE needs to calculate the MMD distance between every encoding layers whereas SKL-AE and SMD-AE only calculate the distance metric in the bottleneck layer.

However, it is important to note that the predicting time of all DTL methods is mostly equal to that of AE. The reason is that the testing samples are only fitted to one AE in all tested models. For example, the total of the predicting time of AE, SKL-AE, SMD-AE, and MMD-AE are 1.001, 1.112, 1.110,

and 1.108 seconds, respectively, on 778, 810 testing samples of the IoT-1 dataset.

VII. CONCLUSION

In this paper, we have introduced a novel DTL-based approach for IoT network attack detection, namely MMD- AE. This proposed approach aims to address the problem of ''lack of labeled information'' for the training detection model in ubiquitous IoT devices. Specifically, the labeled data and unlabeled data are fitted into two AE models with the same network structure. Moreover, the MMD metric is used to transfer knowledge from the first AE to the second AE. Comparing to the previous DTL models, MMD-AE can operate at all the encoding layers instead of only the bottle-neck layer.

We have carried out the extensive experiments to evaluate the strength of our proposed model in many scenarios. The experimental results demonstrate that DTL approaches can enhance the AUC score for IoT attack detection. Further-more, our proposed DTL model, i.e., MMD-AE, operating transformation at all the level of encoding layers of the AEs

DOI: 10.48175/IJARSCT-10384

ISSN
2581-9429
IJARSCT

helps to improve the effectiveness of the transferring process. Thus, the proposed model is meaningful when having label information in the source domain but no label information in the target domain.

One limitation of the proposed model is that it requires more time to train the model. However, the predicting time of MMD-AE is mostly similar to that of the other AE-based models. In the future, one can extend our current work in several directions. First, we will distribute the training process to the multiple IoT nodes by using the federated learning technique to speed up this process. Second, the cur- rent DTL model is developed based on AutoEncoder. In the future, we will attempt to extend this model based on other neural networks such as Deep Adaptation Network (DAN), Adversarial Discriminative Domain Adaptation (ADDA), Maximum Classifier Discrepancy (MCD), and Conditional Domain Adversarial Network (CDAN).

REFERENCES

[1] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, ‘‘Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey,’’ IEEE Commun. Surveys Tuts., vol. 18, no. 4, pp. 2546–2590, Jun. 2016.

[2] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. Davis Guarnizo, and Y. Elovici, ‘‘Detection of unauthorized IoT devices using machine learning techniques,’’ 2017, arXiv:1709.04647. [Online]. Available: http://arxiv.org/abs/1709.04647

[3] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, ‘‘N-BaIoT—Network-based detection of iot botnet attacks using deep autoencoders,’’ IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.

[4] I. Ahmed, A. P. Saleel, B. Beheshti, Z. A. Khan, and I. Ahmad, ‘‘Security in the Internet of Things (IoT),’’ in Proc. 4th HCT Inf. Technol. Trends (ITT), Oct. 2017, pp. 84–90.

[5] N. Vlajic and D. Zhou, ‘‘IoT as a land of opportunity for DDoS hackers,’’
Computer, vol. 51, no. 7, pp. 26–34, Jul. 2018.

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, ‘‘DDoS in the IoT: Mirai and other botnets,’’ Computer, vol. 50, no. 7, pp. 80–84, 2017.

[7] R. Gow, F. A. Rabhi, and S. Venugopal, ‘‘Anomaly detection in complex real world application systems,’’ IEEE Trans. Netw. Service Manage., vol. 15, no. 1, pp. 83–96, Mar. 2018.

[8] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed, and S. A. Khayam, ‘‘A taxonomy of botnet behavior, detection, and defense,’’ IEEE Commun. Surveys Tuts., vol. 16, no. 2, pp. 898–924, 2nd Quart., 2014.

[9] J. Dromard, G. Roudiere, and P. Owezarski, ‘‘Online and scalable unsu- pervised network anomaly detection method,’’ IEEE Trans. Netw. Service Manage., vol. 14, no. 1, pp. 34–47, Mar. 2017.

[10] H. Bahsi, S. Nomm, and F. B. La Torre, ‘‘Dimensionality reduction for machine learning based IoT botnet detection,’’ in Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV), Nov. 2018, pp. 1857–1862.

[11] C. Zhang and R. C. Green II, ‘‘Communication security in Internet of Thing: Preventive measure and avoid DDoS attack over IoT network,’’ in Proc. 18th Symp. Commun. Netw., Alexandria, VA, USA, Apr. 2015, pp. 8–15.

[12] C. Dietz, R. L. Castro, J. Steinberger, C. Wilczak, M. Antzek, A. Sperotto, and A. Pras, ‘‘IoT-botnet detection and isolation by access routers,’’ in Proc. 9th Int. Conf. Netw. Future (NOF), Nov. 2018, pp. 88–95.

[13] M. Nobakht, V. Sivaraman, and R. Boreli, ‘‘A host-based intrusion detec- tion and mitigation framework for smart home IoT using OpenFlow,’’ in Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES), Salzburg, Austria, Aug. 2016, pp. 147–156.

[14] J. Ceron, K. Steding-Jessen, C. Hoepers, L. Granville, and C. Margi, ‘‘Improving IoT botnet investigation using an adaptive network layer,’’ Sensors, vol. 19, no. 3, p. 727, Feb. 2019.

[15] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, ‘‘Learning latent distribution for distinguishing network traffic in intrusion detection system,’’ in Proc. IEEE Int. Conf. Commun. (ICC), May 2019, pp. 1–6.

ISSN 2581-9429 IJARSCT