2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE) | 978-1-6654-0396-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigDataSE50710.2020.00020

IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT

Yulin Fan[1,2], Yang Li[1,2], Mengqi Zhan[1,2], Huajun Cui[1], Yan Zhang[1,2]
[1]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
{fanyulin, liyang, zhanmengqi, cuihuajun, zhangyan80}@iie.ac.cn

***Abstract*—5G and edge computing promote the development of Internet of Things (IoT). In the near future, 5G will be used as infrastructure to connect all walks of life. At the same time, numerous resource-constrained IoT devices make attacks easier and more frequent, resulting in more and more serious harm. 5G needs to provide security for the IoT it carries. 5G IoT security faces three major challenges. First, due to the heterogeneity, diversity and personalization of IoT networks, it is impossible to use a single unified detection model. Second, data in various industries exists in the form of isolated islands so it is hard to share in the light of privacy protection. Third, data island makes some industries produce too little data to train a powerful intrusion detection model. Therefore, 5G needs a personalized, distributed and effective intrusion detection system that can integrate all IoT information under the premise of protecting the privacy of each IoT data.**

In this paper, we propose IoTDefender, an intrusion detection framework for 5G IoT based on federated transfer learning. 5G edge computing well supports the layered and distributed structure of IoTDefender. IoTDefender carries out data aggregation by federated learning and builds customized detection models by transfer learning. It enables all IoT networks to share information without leaking privacy. Consequently, IoTDefender owns excellent generalization ability, which can highly improve the detection of unknown attacks. The experimental results demonstrate that IoTDefender is more effective (91.93% detection accuracy on average) than traditional method. Furthermore, IoTDefender produces a lower false positive rate than that of a single unified model, which means it has advantages in personalization.

***Keywords*—5G IoT, MEC, intrusion detection, federated transfer learning**

I. Introduction

Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction [1]. IoT connections will reach almost 25 billion globally by 2025, up from 12 billion in 2019 [2]. 5G has the advantages in wide coverage, large connection, low latency and safety, which can promote the development of IoT. 5G defines three application scenarios: Enhanced Mobile Broadband (eMBB), Ultra-reliable and Low Latency Communications (uRLLC) and Massive Machine Type Communications (mMTC) [5]. The latter two scenarios of uRLLC and mMTC are highly related to IoT. Thus, they can be subsumed under the concept of 5G IoT [4]. The uRLLC is oriented to special applications such as Internet of vehicles, industrial control system, telemedicine, etc. In the mMTC scenario, 5G's powerful connection ability can quickly promote the deep integration of various vertical industries such as smart city, smart home, environmental monitoring, and realize the interconnection of everything [3]. 5G IoT is not aimed at a certain scenario or field. It is the infrastructure to integrate all industries and involve all walks of life.

Mobile Edge Computing (MEC) pushes the cloud computing capability close to the radio access networks. MEC is an important technology of 5G. It realizes data processing at the edge rather than sending to the cloud. Therefore, MEC can contribute to 5G in terms of low latency and high reliability. The combination of cloud and MEC can also relieve the traffic pressure and greatly improve the operation efficiency. MEC can better support 5G IoT for several reasons. 1) Fast and real-time massive data processing. MEC nodes are distributed and close to the device, which makes it more efficient to support local business processing and execution with low latency. 2) Data security and privacy protection. Compared with cloud computing, MEC is able to limit the operation of data within the firewall of edge gateway, and reduce the long-distance transmission of data, thus reducing the risk of data theft and privacy exposure. 3) The mobile and positioning support for IoT devices. For some location-based applications, such as navigation, the terminal equipment can provide relevant location information and data to the edge nodes for processing, judgment and decision-making, while the request can be responded quickly. Therefore, it can be predicted that 5G IoT will be deployed in a hierarchical manner as sensor network-MEC-cloud platform, as shown in Fig. 1.

5G, as the infrastructure connecting the IoT in all walks of life, needs to provide security measures. An important measure to secure networks is intrusion detection system (IDS). For IDS of IoT security, a lot of researches based on machine learning(ML) have been made. Researchers have tried a variety of ML methods, such as support vector machine (SVM), k-nearest neighbor (KNN), fuzzy logic, artificial neural network (ANN) ,etc [6]. However, these studies only focus on the IoT in a single industry, while 5G IoT contains a variety of IoTs from all walks of life. The IDS for 5G IoT faces the following three challenges:

First, 5G IoT is a geographically distributed, heterogeneous and hierarchical system, which requires personalized models instead of a single unified model. Different IoT network are deployed in complex and diverse physical environments, such

978-1-6654-0396-2/20/$31.00 ©2020 IEEE
DOI 10.1109/BigDataSE50710.2020.00020

as home, factory, city and so on. Their network function, architectures and terminal devices are quite different, which means they own different network traffic pattern and face different security threat. Therefore, a unified IDS model is not practical and personalized models should be suggested. Furthermore, due to the existence of MEC and its strong support for IoT, the IDS in 5G IoT should be deployed in a distributed and hierarchical manner.

Second, 5G IoT needs to ensure data isolation and privacy. Data privacy has always been an important issue for the security field. Due to security and privacy issues, data often exists in the form of isolated islands. Data of different industries, different enterprises and organizations in the same industry, cannot be shared. Although a large quantity of data exists in different networks, it is impossible to collect, integrate and use them. Therefore, we need to study the IDS model of 5G IoT, which can integrate the information from all IoTs under the condition of protecting the privacy of user data, so as to better reflect the advantages of 5G as infrastructure.

Third, the small amount of data leads to inefficient training model. First, compared with high-end devices, some IoT devices generate little traffic, which is often triggered by infrequent user interaction, such as in intelligent parking, remote meter reading, etc. In addition, since privacy data is hard to share, the attack event samples of each data island are very few. The limited local data is not comprehensive enough to train excellent IDS models. In that case, it is difficult for the IDSs to find unknown attacks when new attacks occur. The IDS model of 5G IoT should be able to integrate the attack data from various IoTs to settle this issue, and improve the overall capability of detection and defense in 5G IoT.

To solve the aforementioned problems, a novel intrusion detection system based on federated transfer learning is proposed in this paper. Federated transfer learning enables different enterprises or institutions to train their own personalized models by learning knowledge from each other without sharing them directly. The distributed characteristic of 5G MEC also supports the design of federated transfer learning framework.

In this paper, we propose IoTDefender, the first federated transfer learning framework for IDS of 5G IoT. On the Security Cloud side, IoTDefender aggregates the information from MEC platforms without leaking privacy by federated learning. On the MEC side, IoTDefender trains personalized models by transfer learning, detects abnormal traffic and sends alerts to specific IoT network within its coverage area. In summary, this paper makes the following contributions:

- To the best of our knowledge, IoTDefender is the first framework to apply federated transfer learning to the IDS of 5G IoT. It aggregates data from different IoT networks securely and achieves a good detection model for each IoT network through knowledge transfer and federation. Our framework is hierarchical, flexible and extensible, which can be easily used in many different IoT network.
  - It utilizes federated learning to aggregate information and ensure data privacy of each IoT network.
  - It utilizes transfer leaning to achieve personalized model for each IoT network.
  - With federated transfer leaning, IoTDefender has strong generalization ability and can detect unknown attacks effectively.
- We conduct extensive experimental analysis using both private and public datasets to simulate the heterogeneous IoT environment in the real world. The private datasets are from two different smart home networks, one smart camera monitoring network and one traditional network respectively. The public dataset is CICIDS2017 [39]. Therefore, IoTDefender is practicable and valuable to the real 5G IoT system.
- With the data privacy well preserved, we achieve a good performance of intrusion detection with an increase of about 3.13% compare to traditional ML methods.

The rest of the paper is organized as follows. Sect.II presents the related work. Sect.III describes IoTDefender in detail. Implementation and evaluation are presented in Sect.IV. Discussion are addressed in Sect.V. Finally, the paper is concluded in Sect.VI.

II. Related work

A. IDS of IoT

The domain of IDS for IoT has been extensively studied [7]–[10]. Raza et al. [8] proposed a SVELTE, the first IDS to detect spoofing and sinkhole attack in IPv6-connected IoT. Bostani and Sheikhan [10] suggested a hybrid anomaly-based IoT IDS, which supports the detection of sinkhole and selective-forwarding attacks in 6LowPAN networks. These solutions focus on routing attacks inside the wireless sensor network, which are constrained by protocol and only applicable to some specific IoT. In contrast, IoTDefender is intrusion detection system deployed outside the radio network, which is general and does not oriented to any specific protocol.

In general, IDS can be divided into rule-based and anomaly-based detection methods. For anomaly-based IoT IDS, many researchers are interested in using ML algorithms to design it [11]–[15]. Kitsune [13] is an online unsupervised intrusion detection system using an integrated self-encoder, and has been tested in the home IoT and IP camera networks. Lopez-Martin et al. [12] proposed an unsupervised anomaly NIDS for IoT based on Conditional Variational AutoEncoder (CVAE). Hodo et al. [14] used Multi-Layer Perceptron (MLP) as a supervised Artificial Neural Network (ANN) in an off-line IoT IDS. Their analysis is built on internet packet traces and tends to detect DoS attacks in the IoT network.

The above approaches rely on clustering large numbers of packets to detected intrusions in a centralized way. They not only require a lot of computing resource, but also are inflexible and hard to extend. As a matter of fact, 5G IoT systems are mostly distributed in different regions and industries, and each IoT has its own traffic and business characteristics. Therefore, the distributed, personalized, flexible and scalable IDS has an important application prospect. Fortunately, fog computing and edge computing can help us achieve this goal.

Several methods proposed the IDS mechanism for IoT using distributed architecture based on fog computing nodes and ML [16]–[19]. For example, Shailendra Rathore [18] proposed a fog-based attack detection framework that relies on the fog computing paradigm and an ELM-based Semi-supervised Fuzzy C-Means method. Prabavathy et al. [19] proposed a method using Online Sequential Extreme Learning Machine (OSELM) based on fog computing. The distributed mechanism respects security, interoperability, flexibility, scalability and heterogeneity aspects of IoT systems. However, due to data privacy, the above distributed methods can hardly help different networks to share knowledge securely. Also,they can hardly reflect the situation of the real world because all of them use only one IDS dataset like NSL-KDD for experiment. As an infrastructure carries IoT of all walks of life, 5G operators are able to provide security detection capabilities. Therefore, we need to design an IDS framework which can integrate the data of various industries on the premise of protecting the privacy of users. IoTdefender uses federated transfer learning to achieve this goal.

B. Federated learning

Federated learning was first proposed by Google in 2016 [36]. Its design goal is to carry out efficient learning among multiple participants or computing nodes on the premise of ensuring the information security during data exchange, protecting personal data privacy, and ensuring legal compliance. Federated learning is able to solve the data island problems effectively. A comprehensive secure federated learning framework and a survey of existing works are provided in [21] by Yang et.al. According to [21], federated learning can be divided into three categories: 1) horizontal federated learning, which is applicable to the scenarios where two datasets are different in samples and the same in feature space; 2) vertical federated learning, which is applicable to the scenarios where two datasets are the same in samples but different in feature space; 3) federated transfer learning, which is applicable to the scenarios where two datasets are different not only in samples but also in feature space. The IoTDefender belongs to federated transfer learning category.

Federated learning has been used in computer vision, healthcare, natural language processing, etc [20]. In the field of IDS system of IoT, Thien Duc Nguyen presents the DÏoT [24], a self-learning distributed system that first applies federated learning to aggregate behavior profiles for detecting device-type specific anomaly autonomously. However, DÏoT still has some limitations: 1) Both the Security Cloud and security gateway use a unified model architecture without considering the needs of personalized IoT models; 2) DÏoT lacks the support of public datasets (the model is random at the beginning), so it hardly detects new or unknown attacks.

C. Transfer learning

Transfer learning aims at transferring knowledge from existing domains to a new domain without rebuilding models [29], which is used to solve the problem that the model cannot be effectively trained because of the small data island. A comprehensive survey to the transfer learning is provided in [29]. Transfer learning has made success in healthcare [34], recommendation [32], object detection [33] and autonomous vehicles [31]. As far as we know, it has not been applied in network security. Deep domain adaptation is a branch of transfer learning [28]. IoTDefender can make use of domain adaptation to solve the issue of different distribution between traditional network and IoT.

The combination of federated learning and transfer learning can break the data barriers between different organizations. Several researches on federated transfer learning have been discussed in [23]. Yiqiang Chen propose FedHealth [22], the first federated transfer learning framework for wearable healthcare to classify human activities. We try to train the personalized IDS model for attack detection in 5G IoT in this paper.

III. The Proposed Framework

A. Problem Definition

In IoTDefender, we combine N different MEC platforms to get traffic data from the edge of the 5G IoT network.The MEC platforms are denoted by $\{P_1, P_2, ..., P_N\}$ and the traffic data that MEC platform collect are denoted by $\{D_1, D_2, ..., D_N\}$. In our problem, we want to protect the privacy of different IoT network data rather than directly sharing them. In addition, we need a model that can better adapt to the local network environment and owns generalizition ability to detect unknown attack. Obviously, the way of combining all the data $D = D_1 \cup D_2 \cup ... \cup D_N$ is more likely to obtain a poweful unified model M_{UNI} that is able to achieve the goals. It is because the larger and richer dataset is, the more knowledge it contains. However, it's not good for privacy. While training a personalized model M_{TRA} locally only based on local dataset D_i is often limited by the scale and privacy of data.

We want to collaborate all the data to train the federated transfer model M_{IoTDef} while any MEC platform P_i will not disclose their data D_i to each other. The goal of IoTDefender is to establish a collaborative framework to improve the performance of M_{IoTDef} beyond individual model M_{TRA} and ensure its accuracy as close as possible to that of M_{UNI}.

B. Overview of the Framework

IoTDefender aims to achieve excellent detection accuracy through federated transfer learning and ensuring data security. We assume that there are three MEC platforms as the clients of federated learning and one Security Cloud platform as the server, which can be extended to more general situations. The overall architecture of IoTDefender is shown in Fig. 1.

IoTDefender framework has three layers. The top layer is Security Cloud platform operated by 5G operator which owns a large amount of data and computing resource to train the server model. The Security Cloud is different from IoT cloud platform in that it integrates all IoT security detection information within its coverage and can be worked as a part of 5G security infrastructure. The bottom layer is IoT devices

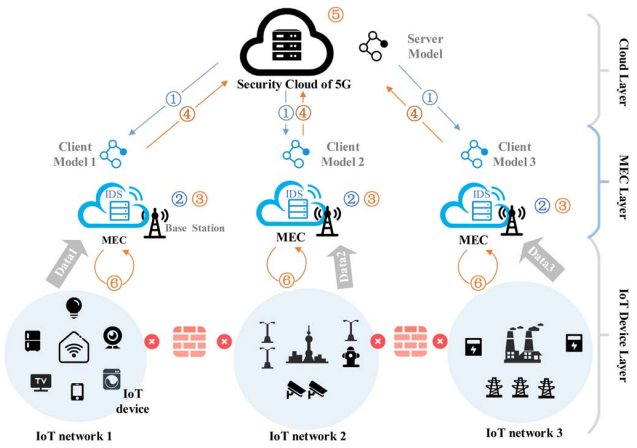


Fig. 1. Architecture of IoTDefender.

layer with a variety of intelligent IoT endpoints. Note that different IoT networks are distributed at different locations and do not share information with each other. The MEC platforms at the middle layer are appropriate to host the IDS component and work as local access gateways to the Security Cloud. Thus, each MEC platform can obtain traffic data from IoT to train the client model and detect attacks. MEC platforms are not limited by resources so that IoTDefender should have capability to store and process data from all the sensor networks and should provide quick response in a short time.

The model training in IoTDefender framework mainly includes six procedures as shown in Fig. 1. ① First, the server model is trained according to the public IDS dataset and distributed to all MEC platforms. ② Then, each of MEC platforms can train their client model based on their local private IDS data. In this step, the data distribution between the Security Cloud and MEC platforms is different. Transfer learning is performed to make the client model customized for each IoT. ③ Next, each MEC platform computes the logits of client model based on the public dataset as the input. Note that the logits indicate the output result of the layer before softmax layer. ④ Later, MEC platforms upload the logits to the Security Cloud. ⑤ The server integrates them and transmits the new logits to MEC clients.⑥ Finally, Each MEC client trains client model on public dataset to make its logits approach to the new logits. After that, each of them trains client model again on private dataset for a few epochs to get a personalized client model. The step 3 to 6 procedures are repeated throughout the training process. Note that all the steps do not disclose any user data or information. After the training process is completed, the personalized IDS model generated in the final transfer learning process is used to detect intrusion.

The federated learning paradigm is the basic computing model of IoTDefender. It deals with model building and knowledge sharing without leaking privacy during the whole process. After the server model is generated, it cannot be directly applied to the clients IoT, since it is obvious that the samples in the Security Cloud have highly different probability distribution and feature spaces with the samples in each MEC platform. Therefore, transfer learning is applied to local IoT network to construct a pesonalized client model.

C. Federated Learning

IoTDefender adopts the federated learning to resolve the problem of data isolation. This step mainly consists of two critical parts: server and clients model learning. In Security Cloud side, the server trains the server model based on public dataset and sends the initialized server model parameters to clients. In MEC side, each client trains its client models based on the sever model and private dataset. Then, the clients upload their updated parameters like weights or gradients to the server for aggregation. In this paper, the clients upload logits to the server [35]. During aggregation, the server aligns all clients' parameters. It can perform an average operation to get new parameters. And the updated parameters are returned to the clients, each client starts the next iteration. The above procedure will be repeated until the whole training process converges.

Considering the calculation burden and efficiency, the clients can upload parameters every night. For each MEC platform, the client model has a good generalization ability, since it integrates the knowledge from both the Security Cloud and all other MEC platforms in an implicit way. The learning objectives for the server and client models are denoted as below respectively:

$$arg\min_{\omega,b} L = \sum_{i=1}^{n} l(y_i, f_S(x_i)) \tag{1}$$

$$arg\min_{\omega^k,b^k} L_k = \sum_{i=1}^{n} l(y_i^k, f_k(x_i^k))) \tag{2}$$

where k is the client number, $l(\cdot,\cdot)$ denotes the loss function, (x_i, y_i) and (x_i^k, y_i^k) are data instances from the Security Cloud and MEC platform, n and n^k denote size of public dataset and private dataset. ω, b denote the weight and bias that will be learned. f_S denotes the sever model, and f_k denotes the client model.

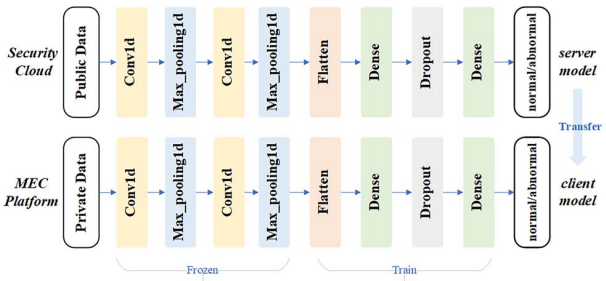


Fig. 2. Transfer learning process.

D. Transfer Learning

Federated learning has solved the issue of data privacy and data islands. Another important issue is data heterogeneity. If we directly apply the sever model to clients, it still performs poorly due to the great distribution difference between the MEC and the cloud data. In addition, the sever model in the Security Cloud only learn the coarse features from the large dataset of traditional network, whereas fail to learn the fine-grained information of a particular IoT network. It has been proved that in deep neural networks, features in the lower levels are highly transferable since they focus on learning common and low-level characteristics. The higher layers will learn more specific features to the task [30]. Therefore, after obtaining the server model, MEC clients can perform deep transfer learning to achieve personalized client model.

The transfer learning process is shown in the Fig. 2. The neural network consists of two convolution layers (conv1d, conv2d), two maximum pool layers (max_pooling1, max_pooling2), two full connection layer (Dense), one flatten layer (Flatten) and one dropout layer (Dropout). The input is network data and outputs are traffic class (normal or abnormal). We regard that the lower layer will extract the basic features of network traffic, while the higher layer will extract the special features of network traffic. Concretely, we keep the lower layer (conv and max_pooling) frozen and adjust the parameters of the upper layer. It is a simple transfer learning approach called Fine-tune, which finetunes the model without considering the distribution divergence between domains [30].

In our task, we have large public dataset from traditional network as source domain and private data from IoT network as target domain. IoTDefender adopts domain adaptation to solve the problem that the probability distribution of source domain and target domain is inconsistent [28].To this end, we borrow the idea from [27] called DDC. In order to measures the distance between two domains, a kernel learning method called maximum mean variance (MMD) is used. The greater the MMD value, the greater the difference between the two datasets. MMD is defined as:

$$D_{MMD}(X_S, X_T) = \| \frac{1}{|X_S|} \sum_{x_s \in X_S} \varphi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \varphi(x_t) \| \tag{3}$$

where $\| \cdot \|$ denotes the squared matrix Frobenius norm, X_S and X_T denotes the source and target domain, $|X_S|$ and $|X_T|$ denotes the number of source and target domain samples. $\varphi(\cdot)$ denotes nonlinear mapping function. We add this distance to the loss of the network for training, and then get the loss function as:

$$L = L_C(X_L, y) + \lambda D^2_{MMD}(X_S, X_T) \tag{4}$$

where L_C is classification loss function. λ balances the proportion of classification task and domains distance. The training target is to minimize the total loss.

E. Learning Process

The learning procedure of IoTDefender is described in Algorithm 1. After all training procedure is completed, IoTDefender can work continuously to process the new emerging labled data. In this case, the longer IoTDefender is used, the more powerful the model can be.

Algorithm 1 The learning procedure of IoTDefender

Input: public dataset D_0, private dataset D_k
Output: trained model f_k , $k = 1, 2, \ldots, n$

1: **Initialization:** Train a CNN model f_S with public dataset D_0 on Security Cloud.
2: **Distribution:** The server model f_S is distributed to all the MEC platforms.
3: **Transfer learning:** Each MEC platform trains client model f_k on public and private dataset D_0, D_k using (4).
4: **Federated Process:**
5: **for** round = 1, 2..., r **do**
6: Each MEC platform calculate the logits l_k on public dataset D_0 and upload it to the Security Cloud platform.
7: The Security Cloud aggregates the logits of all MEC platforms and calculates the average logits l_{avg}.
8: The Security Cloud sends l_{avg} to all MEC platforms.
9: Each MEC platform trains client model f_k on public dataset D_0 to make its logits close to l_{avg}.
10: **Transfer learning:** Each MEC platform trains client model f_k on the private dataset D_k again based on Fine-tune.
11: **end for**

IV. Experiments

In this section, we evaluate the performance of the proposed IoTDefender via two experiments on known and unknown attacks detection.

A. Datasets

We evaluate IoTDefender capabilities of detecting attacks in four different networks. We use CICIDS2017 [39] as public dataset, NSL-KDD [37] and three IoT datasets [13], [38] as private dataset. Table I shows some information of the five datasets, including attack types and number of total packets. NSL-KDD is drawn from traditional network which is denoted as P4. The remaining three IoT datasets are drawn from three different real IoTs, two of which are smart home networks denoted as P1 and P2 respectively, and the other is IP cameras video surveillance network denoted as P3. The details of the three IoTs are described as below:

1) The first smart home network P1 consists of two typical smart home devices – SKT NUGU (NU 100) and EZVIZ Wi-Fi Camera (C2C Mini O Plus 1080P) and some laptops or smart phones. All devices were connected to the same wireless network [38]. 2) The second smart home network P2 is also a Wi-Fi network composed of 9 IoT devices, including one thermostat, one baby monitor, one webcam, two different doorbells, four different cheap security cameras and three PCs

TABLE I
THE DATASETS TO EVALUATE IOTDEFENDER

Dataset	Attack type	# Total Packets
CVE-CIC-IDS	PortScan, DDoS, FTP-Patator, SSH-Patator, Bot, Heartbleed	2300825
IOT-dataset 1(P1)	ARP Spoofing, Dos, Scanning, Mirai	127358
IOT- dataset 2(P2)	Mirai	764137
IOT- dataset 3(P3)	ARP MitM, Dos, Fuzzing, OS Scan	721276
NSL-KDD(P4)	DoS, Probe, R2L, U2R	494020

[13]. 3) The third real IP camera video surveillance network P3 consists of two deployments of four HD surveillance cameras. The cameras are connected to the Digital Video Recorder (DVR) via a site-to-site VPN tunnel [13].

B. Implementation

We conduct two experiments to evaluate the effectiveness of IoTDefender. The first experiment is to evaluate the basic ability of attack detection, and the second is to prove the generalization ability. On both Security Cloud and MEC platforms, we design the IDS models based on convolutional neural network (CNN) because deep neural network can learn nonlinear characteristics of data and is easy to perform knowledge transfer. The CNN network is shown in Fig. 2. The small batch random gradient descent (SGD) is used for optimization. In addition, we use 60% of each dataset for training and 40% for testing. We set the learning rate of batch processing to 0.1, the batch size to 64 and the training epochs to 10.

In order to verify the effectiveness of IoTDefender (IoTDef), we compare its performance with traditional ML method, such as KNN, Adaboost (AB), Random Forest (RF) and CNN in the first experiment. In addition, we record the performance of transfer learning (TF), in which the server model is retrained by local data with only transfer learning, without federated learning. Moreover, we also record the performance of using federated learning only (FED), which means it do not perform transfer process. Its basic idea is similar to DÏoT [24].

TABLE II
DETECTION ACCURACY(%) OF THE TEST CLIENT

Client	AB	RF	KNN	CNN	TF	FED	IoTDef
P1	97.12	98.96	99.57	99.54	99.56	99.56	99.60
P2	99.84	98.14	97.65	99.10	99.31	99.66	99.75
P3	77.82	77.99	74.67	84.23	82.84	85.07	86.37
P4	80.58	80.67	80.62	77.71	75.71	78.06	81.99
AVG	88.84	88.94	88.12	90.14	89.35	88.84	91.93

We evaluate IoTDefender with the following standard measures, which can be calculated based on the confusion matrix.

$$ACC = (TP + TN)/(TP + TN + FP + FN) \quad (5)$$

TABLE III
DETECTION ACCURACY (%) OF UNKNOWN ATTACKS

Client	Unknown attack	AB	RF	CNN	TF	IoTDef
P1	Mirai	92.10	96.63	99.51	99.75	99.89
	Dos	77.03	71.18	77.80	99.90	99.97
	AVG1	84.56	83.90	88.65	99.82	99.93
P2	Mirai	-	-	-	42.31	81.30
P3	OS Scan	79.3	77.46	71.07	81.22	82.98
AVG					84.60	92.81

$$TPR = TP/(TP + FN) \quad (6)$$

$$FPR = FP/(FP + TN) \quad (7)$$

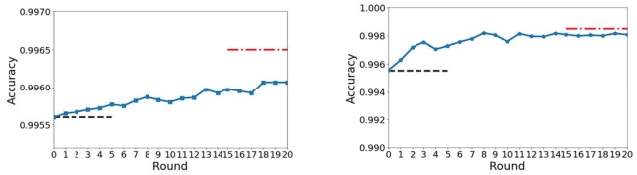
where TP is true positive indicating the quantity of attack samples correctly detected, TN is true negative indicating the quantity of normal samples correctly detected, FN is false negative indicating the quantity of attack samples incorrectly detected, and FP is false positive indicating the quantity of normal samples incorrectly detected.

C. Performance Results of Detection Accuracy

The detection accuracy is shown in Table II. The accuracy of IoTDefender and TF is higher than that of traditional methods (KNN, AB, RF, CNN). IoTDefender combines data from diverse networks, so it works better than traditional models. Compared with TF and FED, IoTDefender improves the accuracy about 2.58% and 3.09% respectively. In a word, IoTDefender is effective to detect anomaly. Fig. 3 shows the test accuracy of IoTDefender of iteration rounds. The black dashed line (on the left) represents the test accuracy of a model after transfer learning with the public dataset and its own private dataset. It is the lower limit of IoTDefender. The red dash-dot line (on the right) represents the would-be performance of a unified model M_{UNI} based on the whole datasets (including public and all private datasets). It is also the upper bound of IoTDefender. After several rounds of learning, the accuracy of IoTDefender is almost close to its upper bound. For example, it reaches 99.8% of the upper limit in P2. In fact, IoTDefender trades a small loss of accuracy for data privacy protection.

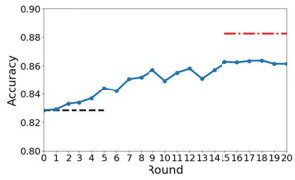
D. Performance Results of Model Generalization

In order to prove the superiority of the generalization ability of the federated transfer learning model, we set up another experiment to verify that IoTDefender can not only detect the attacks contained in local training dataset, but also identify unknown attacks with the help of the public and other local datasets. We first remove the data of some certain attacks in training set and maintain them in test set. Specifically, we remove Mirai and Dos from P1 training set while keeping other data sets unchanged, but keep them in test set. Similarly, we remove Mirai from P2, OS Scan from P3 in training set and maintain them in test set. Then, we repeated the previous experiment as mentioned in last subsection. Finally, we compare the detection accuracy of the unknown attack

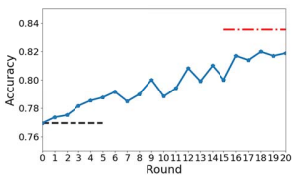


(a) P1

(b) P2

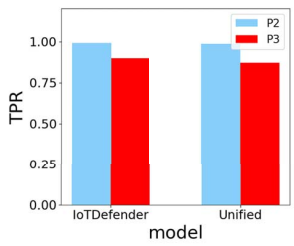


(c) P3

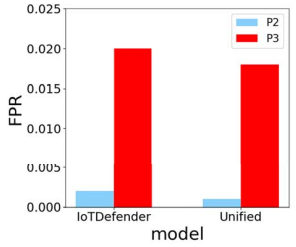


(d) P4

Fig. 3. The test accuracy of the client models.



(a) TPR



(b) FPR

Fig. 4. TPR and FPR of P2 and P3.

of IoTDefender with other models (see the results in the Table III). We run the two experiments 5 times to record the average values.

If there is no Mirai attack at the P1 training set, traditional models cannot learn the behavior of Mirai attack. So they hardly recognize it when Mirai attack appears in the network for the first time. However, IoTDefender utilizes federated transfer learning to help P1 learn the knowledge of Mirai from P2 and public dataset implicitly. Therefore, even P1 owns few data, it can detect new attacks that never occured before, and the result proves that. For P2, since Mirai is the only attack in P2, traditional method cannot be performed when we remove it. In this case, TF uses the sever model directly and IoTDefender uses the collaborated model of P1, P3 for testing. As a result, IoTDefender can highly improve the accuracy of detection when compared with TF, because the public dataset does not contain data of Mirai while P1 does. P2 can learn the knowledge of Mirai from P1 through federated learning.

As shown in the Table III, the unknown attacks detection accuracy of TF is higher than that of traditional models, while the accuracy of IoTDefender is higher than that of all other models. TF can use the large public dataset so it performs better. Compared with TF, IoTDefender produces higher accuracy with 8.21% improvement. This fully proves that IoTDefender has strong generalization ability. The reason behind this is that the IDS module at each MEC platform in IoTDefender can learn the knowledge from other clients and server during federation and transfer process.

E. Model Personalization Analysis

In addition, due to the heterogeneity of datasets, utilizing a unified model M_{UNI} or only federated learning model M_{FED} is more likely to suffer from increasing false positive rates or decreasing sensitivity of the model. IoTDefender can overcome these defects since it builds a personalized IDS model for each IoT network through transfer learning. Each client model focuses on the characteristic behavior of one single IoT, resulting in a more specific, accurate and personalized IDS model. In order to evaluate the benefit of using personalized models, we compared IoTDefender with the unified model M_{UNI} by means of accuracy, TPR and FPR. The results in Fig. 4 indicate that IoTDefender has lower FPR although it does not do well as M_{UNI} in TPR and accuracy (Fig. 3). Therefore, IoTDefender has advantages on personalization which makes it more practical to deploy in real world.

V. Discussion

IoTDefender is a general intrusion detection framework in 5G IoT, which can be further expanded to more complex scenarios. In this section, we discuss several possible directions for expansion:

1) In the IoTDefender implementation in this paper, we just build four client IoTs for the federated learning process. With more and more clients joining the system, how to choose the federated clients is an important problem. There is no need to choose all the clients in federate process. A potential way is to combine the information of 5G slice. The networks under the similar slice could be a group of federated clients.

2) Due to the issue of slow network speed in some 5G IoT areas, it is difficult to ensure reliable communication. The possible methods, like compression scheme and local update, can be used to optimize the federation algorithm [25], [26]. They are able to reduce the communication rounds and make the transmission parameters as small as possible to ensure the transmission quality.

3) With the development of attack technology, the attack mode is also constantly changing. Original dataset is out of date and new knowledge is required. It is possible to train network traffic data online with incremental learning method to make the model more powerful.

VI. Conclusion

In this paper, we propose IoTDefender, a novel intrusion detection framework based on federated transfer learning for 5G IoT security. IoTDefender captures network traffic from different IoT networks, processes it on the MEC platform and obtains personalized detection models without disclosing data and privacy. Experiments demonstrate the effectiveness and validity of IoTDefender by detecting malicious packets with 91.93% accuracy on average. The generalization ability is also proved by its significant accuracy increase for unknown attacks detection when compared with other methods.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China (Grant No. 61202419 and No. 61572497), and the Project of Beijing Science and Technology Commission (Grant Y9C003B114).

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645-1660, 2013.

[2] GSMA. "The mobile economy," 2020. Retrieved from https://www.gsma.com/mobileeconomy/.

[3] Vermesan, and P. Friess, Internet of things-from research and innovation to market deployment. River publishers. Aalborg, pp. 231-237, 2014.

[4] Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," IEEE Communications Standards Magazine, vol. 2, no. 1, pp. 36-43, 2018.

[5] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," IEEE Access, vol. 6, pp. 3619-3647, 2017.

[6] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," IEEE Communications Surveys & Tutorials, pp. 1-1, 2019.

[7] M. Surendar, and A. Umamakeswari, "InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN," in 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, pp. 1903-1908.

[8] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," Ad hoc networks, vol. 11, no. 8, pp. 2661-2674, 2013.

[9] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the RPL-connected 6LoWPAN networks," in Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security. Association for Computing Machinery, pp. 31-38.

[10] H. Bostani, and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," Computer Communications, vol. 98, pp. 52-71, 2017.

[11] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K. K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," IEEE Transactions on Emerging Topics in Computing, vol. 7, pp. 314-323, 2016.

[12] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot," Sensors, vol. 17, no. 9, p. 1967, 2017.

[13] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," arXiv preprint arXiv:1802.09089, 2018.

[14] E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2016 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, pp. 1-6.

[15] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," Pattern Recognition Letters, vol. 119, pp. 3-11, 2019.

[16] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," Journal of Communications and Networks, vol. 20, no. 3, pp. 291-298, 2018.

[17] F. Hosseinpour, P. Vahdani Amoli, J. Plosila, T. Hämäläinen, and H. Tenhunen, "An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach," International Journal of Digital Content Technology and its Applications, vol. 10, no. 5, pp. 34-46, 2016.

[18] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," Journal of Communications and Networks, vol. 20, no. 3, pp. 291-298, 2018.

[19] S. Rathore, and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," Applied soft computing, vol. 72, pp. 79-89, 2018.

[20] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," arXiv preprint arXiv:1908.07873, 2019.

[21] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 2, pp. 1-19, 2019.

[22] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," arXiv preprint arXiv:1907.09173, 2020.

[23] Y. Liu, T. Chen, and Q. Yang, "Secure federated transfer learning," arXiv preprint arXiv:1812.03337, 2018.

[24] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 756-767.

[25] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," arXiv preprint arXiv:1610.02527, 2016.

[26] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," arXiv preprint arXiv:1610.05492, 2016.

[27] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," arXiv preprint arXiv:1412.3474, 2014.

[28] M. Wang, and W. Deng, "Deep visual domain adaptation: A survey," Neurocomputing, vol. 312, pp. 135-153, 2018.

[29] S. J. Pan, and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345-1359, 2009.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in neural information processing systems 27. MIT Press, pp. 3320-3328.

[31] C. Fellicious, "Transfer Learning and Organic Computing for Autonomous Vehicles," arXiv preprint arXiv:1808.05443, 2018.

[32] G. Hadash, O. S. Shalom, and R. Osadchy, "Rank and rate: multi-task learning for recommender systems," in Proceedings of the 12th ACM Conference on Recommender Systems. Association for Computing Machinery, pp. 451-454.

[33] K. Kumar Singh, S. Divvala, A. Farhadi, and Y. Jae Lee, "Dock: Detecting objects by transferring common-sense knowledge," in Proceedings of the European Conference on Computer Vision (ECCV). Springer, pp. 492-508.

[34] R. Venkataramani, H. Ravishankar, and S. Anamandra, "Towards Continuous Domain adaptation for Healthcare," arXiv preprint arXiv:1812.01281, 2018.

[35] D. Li, and J. Wang, "FedMD: Heterogenous Federated Learning via Model Distillation," arXiv preprint arXiv:1910.03581, 2019.

[36] H. B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-efficient learning of deep networks from decentralized data," arXiv preprint arXiv:1602.05629, 2016.

[37] M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, 2009

[38] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, H. K. Kim, "IoT network intrusion dataset," IEEE Dataport, 2019. [Online]. Available: http://dx.doi.org/10.21227/q70p-q449. Accessed: May. 28, 2020.

[39] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018