Research Article

Predicting the APT for Cyber Situation Comprehension in 5G-Enabled IoT Scenarios Based on Differentially Private Federated Learning

Xiang Cheng ^{1,2}, Qian Luo,¹ Ye Pan,¹ Zitong Li,² Jiale Zhang,² and Bing Chen ²

¹The Second Research Institute of CAAC, Chengdu 610041, China

²College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 21106, China

Correspondence should be addressed to Bing Chen; cb_china@nuaa.edu.cn

Received 17 September 2020; Revised 16 October 2020; Accepted 9 April 2021; Published 22 April 2021

Academic Editor: Luigi Coppolino

Copyright © 2021 Xiang Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Driven by the advancements in 5G-enabled Internet of Things (IoT) technologies, the IoT devices have shown an explosive growth trend with massive data generated at the edge of the network. However, IoT systems exhibit inherent vulnerability for diverse attacks, and Advanced Persistent Threat (APT) is one of the most powerful attack models that could lead to a significant privacy leakage of systems. Moreover, recent detection technologies can hardly meet the demands of effective security defense against APTs. To address the above problems, we propose an APT Prediction Method based on Differentially Private Federated Learning (APTPMFL) to predict the probability of subsequent APT attacks occurring in IoT systems. It is the first time to apply a federated learning mechanism for aggregating suspicious activities in the IoT systems, where the APT prediction phase does not need any correlation rules. Moreover, to achieve privacy-preserving property, we further adopt a differentially private data perturbation mechanism to add the Laplacian random noises to the IoT device training data features, so as to achieve the maximum protection of privacy data. We also present a 5G-enabled edge computing-based framework to train and deploy the model, which can alleviate the computing and communication overhead of the typical IoT systems. Our evaluation results show that APTPMFL can efficiently predict subsequent APT behaviors in the IoT system accurately and efficiently.

1. Introduction

With the continuous development of 5G-enabled IoT technologies, numerous mobile applications have emerged with various requirements in terms of intelligence, latency, and bandwidth [1]. However, enormous risks and hidden dangers of information security still exist in the applications of the 5G-enabled Internet of Things (IoT). It is mainly caused by the characteristics of the IoT systems, which are lacking update, having longer lifetimes, delayed patched, and facing consequences of compromise [2, 3]. Among the diverse attacks, Advanced Persistent Threat (APT) belongs to a class of advanced multiple-step attacks. Ascribed to its permeability, concealment, and pertinence, the APT could bring severe threats to the IoT systems [4]. For the sake of defending these increasingly complex and potential security risks, researchers and organizations have put forward

various detection technologies, such as intrusion detection technology, malicious code detection technology, and vulnerability detection technology [5–8]. However, the above-mentioned methods have difficulty in meeting the higher requirements of protection for 5G-enabled IoT systems since APT attacks usually adopt the way of step-by-step penetration and long-term latency to achieve the final purpose of confidential data exfiltration [9].

Recently, the technology of Cyber Situation Awareness (CSA) has been put forward by researchers to solve the above problems. The cyber situation comprehension is a phase of the CSA process that focuses on analyzing the detected malicious activities semantics and the possible internal relationships among them [10]. This kind of technology has the ability to infer the attacker's intention and predict the probability of subsequent attacks, which is quite useful for detecting APT in 5G-enabled IoT systems. Hence, in this

paper, we aim to propose an effective and robust cyber situation comprehension method to predict the probability of subsequent APT attacks occurrence after recognizing APT attacks in the 5G-enabled IoT system.

However, predicting the APT attack in IoT systems could face the following challenges: (1) *Unbalanced Datasets*. Since the APT is a multistep attack model, it is difficult for a single organization to capture the data that can cover the complete APT stages and sufficient attack patterns [11]. In addition, since the different organizations will face different APT attacks, it will contribute to the imbalance of APT data. (2) *Isolated Data Island*. As the data generated by a single organization are not sufficient to describe the complex APT process, integrating the data from several organizations to train a sharing model is a promising way to defend against APT [12, 13]. (3) *Limited Resources of IoT Devices*. The IoT devices are usually resource-constraint where their storage capacity and computing power are usually limited. It is not feasible to assign the large-scale data analysis and process to the IoT devices directly [14]. (4) *Arising Privacy Issues*. Conventional APT prediction methods all need to collect the private information of each device, such as the system logs and device IDs, which could arise significant privacy challenges [15].

To meet the problems above, we proposed an APT Prediction Method based on Differentially Private Federated Learning (APTPMFL) to predict the probability of subsequent APT attacks occurring in IoT scenarios. The contributions we have made are shown as follows:

- (i) We proposed a novel APT prediction method, named APTPMFL, which utilizes the federated learning framework to aggregate suspicious activities in the IoT systems. The IoT devices can unite with edge servers to train the prediction model locally using system logs, just uploading the parameter updates to the security service cloud.
- (ii) To protect IoT device data privacy against untrusted edge servers, we adopt a differentially private data perturbation mechanism to perturb the Laplacian random noises to the IoT device training data features, so as to achieve the privacy-preserving property of users' training data.
- (iii) We present an edge computing-based framework to deploy the prediction model in typical IoT systems. The edge servers can not only share the computing overhead for the IoT devices but also alleviate the communication cost between IoT devices and the security service cloud.

The rest of the paper is organized as follows. Section 2 summarizes the related works of attack prediction in cybersecurity and privacy-preserving deep learning. Section 3 provides an overview of the federated learning-based APT prediction architecture for IoT systems, which contains a description of the proposed APTPMFL and the edge computing framework for deploying the APT prediction method. Section 4 presents the design details of the APTPMFL, which consists of the federated learning

approach and the APT attack prediction. Section 5 shows a view of our experiments and analysis. Section 6 presents some conclusions.

2. Background and Related Work

In 1999, the US Air Force Communications and Information Center originally applied the situation awareness technology to the data fusion analysis of multiple NIDS detection results. They claimed that the multisensor data fusion technology provides an important functional framework for the next generation of the intrusion detection system and CSA [4] system, which can fuse the data of multisource heterogeneous IDS, identify the intruders, attack frequency, threat degree, and so on. The CSA applies the theory and method of situation awareness to the field of cybersecurity so that network security managers can grasp the security status of the dynamic network environment and acquire defending decision support. We also give a general functional model of CSA, as shown in Figure 1. The model includes the cyber situation perception phase, cyber situation comprehension phase, and cyber situation projection phase. The functions of each phase are briefly summarized as follows:

- (i) The function of cyber situation perception is to identify the activities in the system, that is, reduce the noise of the raw data generated by security equipment and information management system to get the valid information and analyze the correlation of them to identify the objects in the system. In this way, the abnormal activities will stand out.
- (ii) Cyber situation comprehension usually focuses on recognizing the malicious activities and correlating the semantics of them. In this way, the attacker's intention can be inferred and the subsequent attacks can also be predicted.
- (iii) Cyber situation projection can analyze and evaluate the threat of attack activities to each object in the information system based on the two steps above. This phase focuses on estimating the effects that attacks have produced and may produce on the objects. By projecting the results of CSA to a certain system object, the state of the object can be obtained in the current situation. Although we want to recognize and analyze the activities, the final result of CSA should be expressed as the influence of these activities on the system objects, not just the identification of activities. Cyber situation projection is a process of feedback understanding, fusing the states of various objects observed from the system to form a situation and then evaluate the significance of the situation to each object.

As an important part of cyber situation comprehension, the attack prediction can analyze the logical relationship between attack behaviors and infer the possible changing trends. The purpose of attack prediction is to infer the subsequent malicious actions by understanding the intention of them. At present, the hot topic of attack prediction is

FIGURE 1: The framework of Cyber Situation Awareness.

related to the four topics as follows: attack/intrusion prediction, attack projection, attack intention recognition, and network security situation forecasting [5]. The tasks of attack intention recognition and attack projection are tied to intrusion detection. The core task of them is to predict an adversary's next step moving and his ultimate goal. The attack/intrusion prediction is much more general as it only focuses on predicting malicious activity occurring. The network security situation forecasting is essentially a generic concept related to CSA. The output of network security situation forecasting is a forecast of the number of malicious activities and vulnerability fluctuation in the network.

For solving the challenge of attack activities prediction, Polatidis et al. [6] presented a recommender system that can be applied to defense the cyber threat effectively and practically by making predictions about the ensuing attack behaviors in attack graphs. The Bayesian classifier was developed by Okutan et al. [7], utilized to predict the attack probability in a given day by processing signals extracted from social media and overall events. Huang et al. [8] worked on industrial cyber-physical systems (ICPSs) security and proposed a novel risk assessment approach in virtue of a Bayesian network to model the propagation of malicious activities and predict the probability of IoT devices being attacked. Okutan et al. [9] designed an innovative, automatic attack prediction system called CAPTURE which comprehensively uses generated signals to train a Bayesian classifier used to forecast the cyber threat. Dowling et al. [10] implemented dynamic and adaptive honeypots to capture malicious datasets used to analyze attack types and model temporal attack patterns, and the probability of each attack type proceeding at a certain slot of the day can be calculated. A novel attack prediction method based on information exchange and data mining is presented in [11], which defines rules to describe the general malicious patterns by extracting information from numerous alerts. A system based on

machine learning named MLAPT is suggested in [12]. The proposed system developed eight modules to detect various technics of APT and the machine-learning-based prediction framework takes associated alerts as input to calculate the probability of alerts to evolve a full APT scenario. A data-snapshot-based malware prediction approach is described in [13]. Using recurrent neural networks, this approach can predict malicious executable files in the early stage of software execution. The literature [14] designed a Bayesian game framework based on game theory to analyze multiple APT attack stages and deceptive strategies. Behaviors of APT actors can be predicted by the perfect Bayesian Nash equilibrium (PBNE) to make a defensive strategy. A targeted complex attack network model entitled TCAN is developed in literature [15]. The model predicts the optimum attack path by means of constructing a dynamic attack graph and monitor state change.

Due to the fact that we adopt a differentially private federated learning mechanism to predict APT attacks, we have also surveyed the recent researches on privacy-preserving deep learning. To optimize the efficiency of Deep Neural Networks, the model partition technique [16] has been proposed, which can assign the loosely coupled hidden layers [17] to a third party. Meanwhile, in order to prevent user's privacy data stolen by malicious servers, some researches have been presented from different perspectives [18]. A privacy-preserving deep learning approach proposed by Abadi and Chu [19] provides a Gaussian Perturbation mechanism [20] for the clipped gradient. It is a very effective solution to protect the data privacy of users. To prevent each user's updates from leaking to untrusted third party by learning model sharing, Geyer et al. proposed a device-side differential private federated learning framework [21]. In addition, a secret sharing-based method has been applied to a high-dimensional data sum aggregation protocol by Bonawitz et al. [22]. Unfortunately, all of these privacy

solutions above have to rely on the existence of trusted aggregating servers which work on perturbing the global model parameters and safely assign noise parameters to each user. It means that the aggregator servers can read the model parameters of each individual. Therefore, it is necessary to adopt a practical mechanism to protect the privacy of IoT devices from untrusted third parties in APT attack prediction based on federated learning.

3. APT Prediction for IoT Systems

3.1. System Architecture. In this work, we provide an edge computing framework (shown in Figure 2) that can partition the APTMFL method across the security service cloud, edge servers, and IoT devices. All of the IoT devices take part in the differentially private federated learning protocol provided by a security service cloud, so as to acquire the APT prediction service. Due to malicious edge servers and untrusted security service cloud even have possibility of sneaking into the process of federated learning, to guarantee the involved IoT devices will not suffer the threats of training data breach, we can divide the local learning model into the device-side part and edge-side part which is called ICP-GRU model. The Inception (ICP) models are deployed on IoT devices to extract features from multiple scales of log data. The Gated Recurrent Units (GRU) models are deployed on the edge servers to learn the evolution of APT scenarios. The ultimate goal is not to detect APT attacks but to predict the probability of the evolution of the APT scenario to the next stage which can contribute to the cyber situation comprehension in IoT scenarios. The involved entities and their function are shown in Figure 2.

3.1.1. IoT Devices. The IoT devices represent various kinds of devices (such as industrial personal computers, smart meters, and monitoring equipment). Each IoT device is capable of communication and computing, which can execute the local log data feature extraction procedure (the device-side part ICP model) and transfer its extracted features to the adjoining edge server through 5G base stations.

3.1.2. Edge Server. The edge servers are equipped with much stronger computation and storage resources compared to the IoT devices, usually assigned at the edge of the IoT systems and working as the computation unit between security service cloud and IoT devices. The role of the edge server is a participant in federated learning with the purpose of training the APT prediction model (the edge-side part GRU model). Federated learning is a distributed machine-learning approach with efficient communication and privacy protection. The edge servers receive the features extracted from IoT devices through 5G base stations to train a model locally and update the model parameters to the security service cloud. The participated edge servers can learn various APT attack patterns without exchanging datasets and monitor IoT devices to launch APT attack prediction for cyber situation comprehension.

FIGURE 2: System architecture.

3.1.3. Security Service Cloud. Each edge server updates the model parameters to the security service cloud after training the model locally; after that, the security service cloud aggregates the parameters into a global model and assigns the aggregated model to each edge server. Based on that, all participants only need to interact with the security service cloud for model parameters without exchanging their own data, which protects data privacy and improves transmission efficiency. In such a condition, different edge servers could belong to different organizations, and the security service cloud maintains a repository of APT attack patterns for participants. Meanwhile, the challenge of imbalanced APT data can also be alleviated thanks to the distributed learning model.

3.2. Threat Model and Assumptions. Generally, a full APT attack scenario consists of the following attack phases. Initially, an APT performer gains access to the system illegitimately through the point of entry. Then, the attacker will establish a connection with a C&C. After that, the attacker discovers and collects assets within the organization for privilege escalation and lateral movement. Eventually, the adversary will destroy infrastructures or exfiltrate confidential data of the organization to achieve the ultimate goal. During this process, the APT performer will persist for an extended period of time and use numerous technics. The proposed APTMFL focuses on learning the occurring APT scenarios to acquire its evolution features. Under such circumstances, once a new APT alert comes, we can measure the probability of the subsequent malicious activity occurring by inputting the relevant log instances into the ICP-GRU model.

Among these entities involved in the ICP-GRU model, the IoT devices are assumed as the trusted entities which benefit from the security services by collaboratively executing the training process with other IoT devices. Unfortunately, the third parties, that is, security service cloud and edge servers, have the possibility of being honest but curious [23]. In particular, they can faithfully perform the federated

learning process and correctly calculate and send results. However, they are curious about the privacy contained in the log data and try to acquire the privacy data [24]. In addition, we assume that the model will not be poisoned by attackers during the model training process. At last, we assume the integrity of the data collection framework which completely records the operations of the system, and the data will not be falsified by attackers.

4. APTMFL Design

4.1. Federated Learning Approach. In order to realize APT attack prediction in IoT scenario to achieve security situation comprehension, we introduce an efficiently differentially private federated learning model in edge computing IoT system, named ICP-GRU (the learning procedure is shown in Figure 3). Considering performance as well as privacy issues, this machine-learning model is divided into the device-side part and the edge-side part. Thereby, the computation overhead on the IoT devices is also able to be reduced [25]. The ICP-GRU relies on the edge computing architecture in the IoT scenario and splits the learning protocol across the participants. Specifically, the local federated learning training procedure is divided into two phases: device-side ICP and edge-side GRU. According to the division mechanism, the inception convolution layers are assigned to the device-side while the remaining layers (i.e., Gated Recurrent Units layers) are deployed on the edge side. In this approach, IoT devices only extract and perturb the lightweight and simple features of log instances. To guarantee the system log data in rigorous privacy protection, we provide a differential privacy mechanism in our ICP-GRU scheme, where the extracted features from log instances are perturbed by the deliberate Laplace noise before being transmitted to the edge server through 5G base stations. The security service cloud in the ICP-GRU model is designed to aggregate and average the local updates provided by the edge servers.

To provide proper data for this APT prediction approach based on differentially private federated learning, we also standardize and normalize the original system log. Since the attributes of some features are character types, such as pname, q_domain, and referer, all the symbolic features needed to convert into numerical types before feeding the dataset into the neural network. At the same time, the value of each feature dimension is inconsistent, and the range of values is also obviously different. Some data with high values on high-magnitude features perhaps have a large weight, thus ignoring some hidden information on low-level data. Therefore, we provide a log instance construction module at the beginning of the ICP-GRU model to preprocess the raw data.

4.1.1. Inception Convolution and Data Perturbation. As CNN achieved excellent performance in image processing, it is also constantly being applied to other fields. However, traditional CNN only focuses on extracting local features and neglects the aggregation of multiple local features. To flexibly mitigate this problem, Google proposed a

convolutional neural network architecture in the GoogleLeNet network called Inception. The Inception module aggregates 1×1 , 3×3 , and 5×5 convolution kernels and max-pooling into one layer. Multiple convolution kernels extract information of different scales of the dataset, and the fusion of features can obtain a better representation of the log instance. We adopt the Inception module to perform convolution operation which extracts features from multiple scales of the log instances dataset which can make the neural network much smarter.

After data preprocessing, the dataset arranged according to timestamp is fed into the inception convolution module in the form of data flow for training. Considering the contextual correlation between data, the data flow is split into fixed-size vectors, and each vector contains n pieces of data. After processing each vector, an $n \times m$ feature matrix can be formed where m represents the number of features of each data. The inception convolution module will extract the features of the dataset through convolution kernels of 1×1 , 2×2 , and 3×3 and max-pooling of 2×2 , and the same convolution should be utilized in order to match the width and height of the output matrix. Nevertheless, the Inception module is resource-consuming when performing the convolution operation. Therefore, the 1×1 convolution is inserted before 2×2 and 3×3 convolution to reduce the feature dimension and increase the computation speed.

As the federated learning approach usually trains the sensitive data locally, it can provide a basic privacy guarantee to the involved IoT devices. However, these sensitive device data such as updated parameters (i.e., features and gradients) still have probability stolen by the untrusted security service cloud and edge servers. Therefore, it is urgent to formulate a robust preserving mechanism to keep the confidentiality of each IoT device against the untrusted security service cloud and edge servers. Thereby, we perturb the features extracted by the ICP model, so as to protect the privacy of the IoT device logs. To meet the aforementioned challenge, we formulate a differentially private data perturbation mechanism to defense the untrusted entities acquiring the privacy information that exists in the features extracted by the IoT device-side ICP model. The ICP model can be regarded as a deterministic function: $x_l = F(x_r)$, where x_r is the private device log data and x_l represents the l -th layer output of the ICP model. In order to achieve privacy protection, the differential privacy approach is applied to the ICP model and our private federated learning protocol is constructed in the edge computing-based IoT scenario. As an efficient means to achieve ϵ -differential privacy, the controlled Laplace noise is added to the extracted features. The Laplace noise is sampled from the Laplace distribution with scale $(\Delta F/\epsilon)$ into the output x_l . According to the definition of differential privacy, the global sensitivity for a query $f: D \rightarrow \mathbb{R}$ can be defined as follows:

$$\Delta f = \max_{d \in D, d' \in D'} |f(d) - f(d')|. \quad (1)$$

4.1.2. GRU Networks. The GRU is a type of RNN that is similar to LSTM, which is proposed to address the problems of long-term memory and gradient in backpropagation. The

FIGURE 3: ICP-GRU learning process.

reason for adopting GRU is that it not only achieves the effect equivalent to LSTM but also can save more computing resources, which can greatly improve training performance.

Similar to RNN, the hidden state h^{t-1} transmitted from the previous node and the current input x^t constitute the inputs of GRU. Combining x^t and h^{t-1} , GRU will get the output of the current node as y^t and the hidden state h^t transmitted to the next node. Initially, two gates are obtained by the previous state h^{t-1} and the current node input x^t . As shown in formulas (2) and (3), r indicates the reset gate, z represents the update gate, and σ represents the *Sigmoid* function.

$$r = \sigma(W^r x^t + U^r h^{t-1}), \quad (2)$$

$$z = \sigma(W^z x^t + U^z h^{t-1}). \quad (3)$$

After getting the gate signal, the reset gate is used to get the reset data $h^{t-1} \odot r$, that is, h^{t-1} . Then input h^{t-1} and x^t into the $\tanh(\cdot)$ activation function so as to the output range will fall in $[-1, 1]$. h' here mainly involves the currently entered data x^t , and adding h' to the current hidden state in a targeted manner is equivalent to memorizing the state at the current moment. The formal description of h' is shown as follows:

$$h' = \tanh(W^h x^t + U^h (h^{t-1} \odot r)). \quad (4)$$

Finally, the GRU carries out two operations (forgetting and memorizing) at the same time. With the acquired update gate whose value range falls in $[0, 1]$, $(1 - z) \odot h^t$ defines how much the previous memory is forgotten, and $z \odot h'$ defines how much of the h' containing the current node information is to be keep around. The more close the update gate to "1" is, the more the memory will be reserved, and the more close the update gate to "0" is, the more the memory will be forgotten. The formal description of h_t is shown as follows:

$$h_t = (1 - z) \odot h^t + z \odot h'. \quad (5)$$

Multiscale features hidden in the data flow can be extracted after the ICP model is processing plenty of log instances. The features are subsequently fed into the GRU network in serialized form to learn the temporal features by selectively learning and forgetting. The model parameters will be continuously updated by means of gradient back-propagation, and a powerful APT attack prediction model will be obtained in the wake of multiple rounds of iterations.

4.1.3. Federated Learning Procedure. The APTPMFL method proposed in the paper is applied to the IoT edge computing environment. As shown in Figure 4, the federated learning process for APT prediction consists of the following steps:

- (1) The edge servers act as participants in federated learning and request the ICP-GRU model from the security service cloud.
- (2) The security service cloud assigns an initialized model to each edge server once the participants' requests are received, and the corresponding ICP model is transmitted to the connected IoT devices through 5G communication.
- (3) Each IoT device inputs locally collected data into the model for training, learning the private log data independently, and perturbs the features by controlled Laplace noise.
- (4) The edge servers get the features transmitted from the IoT devices for further GRU learning and update the local model parameters to the security service cloud once achieving model training.
- (5) The security service cloud aggregates the local updated models into a global model.
- (6) The security service cloud will deliver the aggregated global model to each edge server again. After multiple rounds of retraining, the global model is aggregated by the security service cloud, and the APT attack patterns will be acquired.

Eventually, the security service cloud delivers the global model to each participant which is employed to predict the process of APT attackers in the 5G-enabled IoT system. The pseudocode of the overall APTPMFL method is presented in Table 1 and the notations used in the pseudocode are listed in Table 2.

4.2. APT Attack Prediction. APT attackers penetrate the target 5G-enabled IoT system for an extended period of time and launch attacks slowly making defenders have to monitor the system behavior incessantly, which brings great challenges to process data efficiently and detect attacks accurately. However, this low-and-slow attack is a double-edged sword. The long span of time between the APT attack phases leaves enough time for defenders to predict the attacker's next move. When an APT attack at a certain stage is detected

FIGURE 4: Federated learning process.

TABLE 1: The APTPMFL pseudo-code.

APTPMFL algorithm
(1) procedure IoT DEVICE SIDE
(2) Nullification operation: $x_r' = x_r \odot I_n$
(3) Features extraction: $x_l = F(x_r')$
(4) Norm bounding: $x_l' = x_l / \max(1, x_{lco}/B)$
(5) Adding perturbation: $\bar{x}_l = x_l' + \text{Lap}(B/\delta I)$
(6) Upload $\bar{x}_l \leftarrow \bar{x}_l$ to the edge server
(7) end procedure
(8) procedure EDGE SERVER SIDE
(9) EVENT : Receive perturbed features $\bar{x}_i$
(10) Initialize: $w_t^{(i)} \leftarrow w_t^{(\text{global})}$
(11) for each $E_k \in (1, \dots, E)$ do
(12) for batch $b \in B$ do
(13) $g_t^{(i)} = \nabla_{w_t} L(w_t, \bar{x}_i)$
(14) $w_{t+1}^{(i)} = w_t^{(i)} - \sigma \cdot g_t^{(i)}$
(15) end for
(16) $w_{t+1}^{(i)} = w_t^{(i)} - w_t^{(i)}$
(17) end for
(18) Upload $\Delta w_{t+1}^{(i)}$ to the SECURITY SERVICE CLOUD
(19) end procedure
(20) procedure SECURITY SERVICE CLOUD SIDE
(21) EVENT : Receive local model updates $\Delta w_{t+1}^{(i)}$
(22) Initialize global parameters $w_t^{(\text{global})}$
(23) for all $i \in [1, n]$ do
(24) $w_{t+1}^{(\text{global})} = w_t^{(\text{global})} + (1/n) \sum_{i=1}^n \Delta w_{t+1}^{(i)}$
(25) end for
(26) Send $w_{t+1}^{(\text{global})}$ to the edge server
(27) end procedure

and the corresponding alerts are generated, suspicious logs can be continuously analyzed to predict the attacker's behavior so that the proper defensive measures can be chosen before the APT attackers achieve their ultimate goal. In this

TABLE 2: The notations used in the pseudocode.

Notation	Definition
x_r	IoT device sensitive raw data
x_r'	Nullification result for x_r
$\bar{x}_i$	Perturbed features
I_n	Nullification matrix
x_l	l-th layer output of DNN
x_l'	Nullification result for x_l
l	Partitioned layer
$F(x_r)$	Deep neural network
$L(w_t, \bar{x}_i)$	Objective loss function
B	Norm bounding
E	Local epoch
δ	Local learning rate
$w_t^{(i)}$	Parameters for IoT device i at t
$g_t^{(i)}$	Gradient for IoT device i at t
$\Delta w_t^{(i)}$	Local model update at t
$w_t^{(\text{global})}$	Global model parameters at t
σ	Noisy scale

section, we first provide the log instance construction procedure for APT behavior learning and prediction. Then, we present an APT prediction procedure.

4.2.1. Log Instance Construction. The log data collected on the IoT devices infected by APT attacks are used for model training and APT attack prediction. The detection system will generate a series of alerts when a certain APT attack step is detected in the 5G-enabled IoT systems. Although the recognized APT attacks have triggered the alerts, there are still plentiful related unaggressive malicious behaviors that have not been detected which could be a stepping stone for the attacker to launch the next step attack activity.

Consequently, alert attribute values will be analyzed to discover the threatened IoT devices after some alerts have emerged. The log data generated in the targeted IoT devices will be constructed into log instances. The types of log instances depend on the data providers and the operating system installed in IoT devices. On the assumption that all log data derive from Windows Embedded Compact (Windows CE) IoT devices, various types of logs provided by different application programs or record facilities are shown in Table 3. The log data are supposed to transform into a uniform format for the purpose of processing data effectively. As shown in Table 4, 14 features are selected from the log data to construct the log instance. Due to the fact that different log data sources and not all features are contained in each log instance, each log instance can be described as a 14-tuple: $A(I(\log)_m) = (a_1, a_2, \dots, a_{13}, a_{14})$. If a log instance only has features a1, a3, and a5, the other feature values are set to zero.

4.2.2. APT Prediction Procedure. Once the ICP-GRU model has accomplished learning the APT attack activity behaviors in the target 5G-enabled IoT system, it can be used for predicting the probability of later APT attack activities. We abstract 4 APT stages under the IoT environment, that is,

TABLE 3: Log types from different recorders.

Type	Logs	Recorders
1	Object access logs	Audit
2	Process create logs	Audit
3	WFP connect logs	Audit
4	HTTP logs	Internet explorer
5	DNS logs	Tshark
6	Authentication logs	Syslogd

TABLE 4: Log instance features extracted from log records.

Type	Logs	Features	Annotation
1	Log3	h_ip	Host IP address
2	Log3	d_ip	Destination IP address
3	Log3	h_port	Host port number
4	Log3	d_port	Destition port number
5	Log3	Type	Request/response
6	Log5	q_domain	D queried domain name
7	Log5	r_ip	DNS resolved IP address
8	Log2	Ppid	Base-16 parent process ID
9	Log1 Log2 Log3	Pid	Base-16 process ID
10	Log1 Log2 Log3 Log6	Pname	Process
11	Log1	Objname	Object name
12	Log4	res_code	Response code
13	Log4	Referer	Refer of requested URI
14	Log1-Log6	Timestamp	Event timestamp

point of entry, C&C communication, asset/data discovery, and data exfiltration. The log instances can be classified into benign and suspicious through the Log Instance Community Detection algorithm proposed in our previous work [13], and only suspicious instances can be collected to train the model. The overall APT prediction process is presented in Figure 5.

Suppose that the APT stages i and $i + 1$ are detected at the time of T_1 and T_2 , respectively; all of the suspicious log instances within the time window $[T_1 - \tau, T_2]$ should be fed to the ICP-GRU model to learn attack behavior features. It means that when the IoT devices suffer from these suspicious activities, the APT attacker will conduct the $i + 1$ stage attack with a high probability. In case a certain APT stage is detected at time T_1 and corresponding alerts are triggered, log instances of threatened devices are recorded starting at $T_1 - \tau$. The suspicious log instances are fed into the prediction model in real time, and the output of the model is a probability value in the range of $[0, 1]$ which indicates the probability of the system suffering the next stage APT attack. Once the output value surpasses the prediction threshold λ , the 5G-enabled IoT system will be in danger of the next stage of an APT attack scenario with high probability.

5. Experimental Evaluation

For the sake of carrying on a detailed and objective evaluation of our proposed APT prediction method (APTPMFL), we implement a federated learning prototype

on the system log data which are generated within the typical seven APT attack scenarios (Op-Clandestine Fox, Hacking Team, APT on Taiwan, Tibetan and HK, Op-Tropic Trooper, Russian Campaign, and Attack on Aerospace) [26].

5.1. Datasets and Experimental Setup

5.1.1. Datasets. It is unfortunate that the appropriate system log dataset and attack alert dataset associated with typical APT attacks are not acquirable. However, our previous work [26] has accomplished the construction of the APT scenario and log instance correlation. Therefore, we adopt the labeled log instances and recognized APT scenarios generated in this work as simulated data to evaluate the effectiveness of APTPMFL. Table 5 presents the details about these datasets. Each dataset is generated by reconstructing a kind of typical APT scenario. These APT scenarios are designed and launched by different APT attack teams; the attackers exploit various vulnerabilities and adopt different attack strategies. We provide these various datasets that can adequately verify the predicting effect of our method for different APT scenarios.

5.1.2. Experimental Settings. In order to make the laboratory environment reflect the characteristics of the real IoT system as similar as possible, we link four identical hosts (a Red Hat Linux operating system running on a host with an Intel Core i7-8550u 2.53 GHz CPU, 16 GB RAM) to deploy an edge computing network based on federated learning. The laboratory environment is shown in Figure 6. The first host works as the security service cloud, and the other three hosts work as the edge servers. The reason for deploying the identical computing resource on the three hosts is that we, respectively, set eight virtual machines in the victim edge servers as virtual IoT devices. Half of the edge servers' computing resource is shared by the virtual IoT devices. It means that an edge server owns 8 GB RAM, and each virtual IoT device occupies 1 GB RAM. This resource allocation scheme is very consistent with the real IoT system. The APTPMFL is proposed to meet the challenge of the IoT device's resource limitation. Even though we do not implement the real IoT operation flow, each virtual IoT device just has 1 GB RAM resource and without any updated patches can competently simulate and testify the efficiency of our method in the IoT system.

To accomplish the federate learning training process of APTPMFL, we allocate the labeled log instances to each virtual machine (simulates the IoT devices) and allocate the recognized APT scenarios to the edge servers. Then, the virtual machines will transmit part of the logs to the edge servers for training the ICP-GRU models and keep the other logs for testifying the prediction performance. As the baseline of the experiments, we adopt the standard federated learning approach presented in the literature [27]. The local training configurations are that the prediction train for $E = 20$ local epochs with the initial learning rate $\delta = 0.1$. Each module of APTPMFL works on the corresponding locations based on the proposed edge computing-based framework.

FIGURE 5: APT prediction process.

TABLE 5: Datasets used in our experiments.

Dataset	Input size	Training samples	Testing samples
Op-Clandestine	14×1	5600	1200
Fox	14×1	5800	1240
Hacking Team	14×1	5750	1232
APT on Taiwan	14×1	6300	1350
Tibetan and HK	14×1	6100	1307
Op-Tropic Trooper	14×1	7800	1671
Russian Campaign	14×1	4900	1050
Attack on Aerospace	14×1		

Finally, we verify the performance of our method based on some evaluation indicators.

The essence of proposing the APTMFL is to predict the probability of subsequent APT attacks occurring in IoT scenarios. We implemented the algorithm and federated learning framework on the laboratory edge servers and security service cloud. For the sake of verifying whether the proposed method can effectively predict the probability of subsequent APT attacks occurring in the laboratory environment, we select the $F1$ score and false-positive rate (FPR) to indicate the performance of APTMFL. The reason for adopting the $F1$ score instead of the common indicators $Recall = TP / (FN + TP)$ and $Precision = TP / (TP + FP)$ is that the two indicators above are mutually exclusive in some cases, needing a harmonic mean to balance respective defects. The parameters TP , FN , and FP , respectively, count the number of true-positive prediction probabilities, the number of false-negative prediction probabilities, and the number of false-positive prediction probabilities. Thereby, the formal description of the $F1$ -score is shown in formula (5). The FPR focuses on representing the proportion of false alert of the organization that is in danger of the next APT attack stage. The formal description of FPR is shown in formula (6).

$$F_1 = \frac{2Recall \cdot Precision}{Recall + Precision}, \quad (6)$$

$$FPR = \frac{FP}{FP + TN}. \quad (7)$$

5.2. Evaluation of APTMFL

5.2.1. APT Prediction Performance. We will evaluate the prediction performance of the APTMFL by analyzing the results of the FPR and $F1$ -scores. The prediction threshold λ can influence the prediction result as more next step APT attack alerts will generate with its value higher. We have evaluated the prediction performance of APTMFL on the 7 typical APT attack scenarios, such as Op-Clandestine Fox, Hacking Team, APT on Taiwan, Tibetan and HK, Op-Tropic Trooper, Russian Campaign, and Attack on Aerospace. The corresponding system logs are reconstructed by one of our previous works [26].

The performances of APTMFL predicting the 7 typical APT attacks are shown in Figures 7 and 8. It is easy to get the result that both the FPR and $F1$ will reduce with the value of prediction threshold λ increasing. It is due to the fact that the lower λ will make more log instances detected as prestep of APT attack activities and the benign log instances will have a higher possibility to be incorrectly detected. We work hard for getting a proper threshold to make our method achieve preferable prediction performance on the 7 typical APT attack scenarios. Fortunately, when the value of the prediction threshold λ is 0.75, the $F1$ -scores are not too low (at around 80%) and the FPR can drop to an acceptable level (not exceeding 5%).

5.2.2. Efficiency of Federated Learning. We have conducted a set of experiments to evaluate the federated learning performance with varying the quantity of IoT devices (from 3 to 24). The number of epochs that each IoT device collaborates

FIGURE 6: Laboratory environment setup.

FIGURE 7: The F_1 varies with threshold λ fluctuation. FIGURE 8: The FPR varies with threshold λ fluctuation.

with edge server training the ICP-GRU local model is set as 20 and the number of communication rounds between edge server and security service cloud is set as 5. Thereby, each local model has been trained in a total of 100 epochs. It is a sufficient number of training epochs due to the fact that we have accomplished training the ICP-GRU model in a centralized learning scenario and got convergence after 98 epochs. We allocated a randomized subset of training IoT device logs from the constructed dataset to each virtual IoT device and the proposed APTMFL method performance has been evaluated with the number of devices involved in the federated learning model varying. We repeated this experiment seven times for each APT scenario, with random resampling of the training

datasets. The value of prediction threshold λ is set as 0.75 to balance the F_1 -score and FPR. The evaluation results of seven APT scenarios are shown in Figure 9 which demonstrates that the APTMFL method with more participating IoT devices can acquire better FPR and the F_1 -score deteriorates only slightly. Besides, we also find another feature of APTMFL which is that the time complexity will approximately linearly increase with the quantity of log instance increasing.

The proposed APTMFL method can provide better privacy for IoT devices contributing to without training data sharing during the training procedure. However, the ICP-GRU model still has some limitations. Comparing with training this model in a centralized framework, it will

(a)

(b)

(c)

(d)

FIGURE 9: Continued.

FIGURE 9: Evolution of F_1 and FPR as we increase the number of IoT devices. (a) Attack on Aerospace. (b) Hacking Team. (c) Tibetan and HK. (d) Russian Campaign. (e) Op-Tropic Trooper. (f) APT on Taiwan. (g) Op-Clandestine Fox.

TABLE 6: Effect of using federated learning compared to centralizing approach.

Metric	Centralized learning (%)	Federated learning			
		6 participants (%)	12 participants (%)	18 participants (%)	24 participants (%)
F_1 -score	89.8	88.7	86.5	84.5	81.4
FPR	4.85	4.87	4.89	4.94	4.96

inevitably lose some accuracy of APT prediction. To compare this inevitable loss in accuracy, we set another experiment to retrain four ICP-GRU models using the entire training dataset by dividing it among 6, 12, 18, or 24 IoT devices and comparing these to an ICP-GRU trained in a

centralized framework. The evaluation values of F_1 -scores and FPR are the average of their effectiveness on 7 APT scenarios when the threshold λ is set as 0.75. Table 6 shows a small decrease in F_1 -scores as we increase the number of IoT devices, and the FPR does not fluctuate evidently. This small

drop in F_1 -scores will not be concerned, because we can fix the threshold λ to amend it to an appropriate level.

6. Conclusions

We present APTPMFL, a federated learning-based APT prediction method deployed on the 5G-enabled IoT scenario. A model containing multiple APT attack patterns is trained in a distributed learning manner and the well-trained model will be implemented to predict the probability of subsequent APT attacks occurring in 5G-enabled IoT scenarios. As a result, the experiments show that APTPMFL successfully predicts the probability of subsequent APT activities with acceptable accuracy and low false rates.

Data Availability

As this work was supported by the National Key Research and Development Program of China which involves security and secrecy in the military domain, the datasets are not suitable for public exposure.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Key Research and Development Program of China, under Grant no. 2019YFB2102000.

References

- [1] K. Palani, E. Holt, and S. Smith, "Invisible and forgotten: zero-day blooms in the IoT," in *Proceedings of the IEEE International Conference on Pervasive Computing & Communication Workshops*, pp. 1–6, Melbourne, Australia, March 2016.
- [2] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Computer Networks*, vol. 109, pp. 127–141, 2016.
- [3] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra, "Dynamic defense strategy against advanced persistent threat with insiders," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 747–755, Hong Kong, China, March 2015.
- [4] T. Bass, "Intrusion detection systems and multisensor data fusion," *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.
- [5] M. Husák, J. Komárková, E. Bou-Harb, and P. Celeda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2019.
- [6] N. Polatidis, E. Pimenidis, M. Pavlidis, and H. Mouratidis, "Recommender systems meeting security: from product recommendation to cyber-attack prediction," *Engineering Applications of Neural Networks*, vol. 11, no. 3, pp. 508–519, 2017.
- [7] A. Okutan, S. J. Yang, and K. McConky, "Predicting cyber attacks with Bayesian networks using unconventional signals," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, pp. 1–13, Oak Ridge, TN, USA, April 2017.
- [8] K. Huang, C. Zhou, Y.-C. Tian, S. Yang, and Y. Qin, "Assessing the physical impact of cyberattacks on industrial cyber-physical systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8153–8162, 2018.
- [9] A. Okutan, G. Werner, K. McConky, and S. J. Yang, "POSTER: cyber attack prediction of threats from unconventional resources (CAPTURE)," in *Proceedings of the ACM SIGSAC Conference*, pp. 2563–2565, Dallas, TX, USA, October 2017.
- [10] S. Dowling, M. Schukat, and H. Melvin, "Using analysis of temporal variances within a honeypot dataset to better predict attack type probability," in *Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 349–354, London, UK, December 2017.
- [11] M. Husák and J. Kašpar, "Towards predicting cyber attacks using information exchange and data mining," in *Proceedings of the International Wireless Communications Mobile Computing Conference (IWCMC)*, pp. 536–541, Caen, France, July 2018.
- [12] I. Ghafir, M. Hammoudeh, V. Prenosil et al., "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, no. 1, pp. 349–359, 2018.
- [13] M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," *Computers & Security*, vol. 77, no. 2, pp. 578–594, 2018.
- [14] L. Huang and Q. Zhu, "Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks," *Performance Evaluation Review*, vol. 46, no. 2, pp. 52–56, 2018.
- [15] W. Niu, X. Zhang, G. Yang, R. Chen, and D. Wang, "Modeling attack process of advanced persistent threat using network evolution," *IEICE Transactions on Information and Systems*, vol. E100-D, no. 10, pp. 2275–2286, 2017.
- [16] S. A. Osia, A. S. Shamsabadi, and A. Taheri, "A hybrid deep learning architecture for privacy-preserving mobile analytics," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–21, 2017.
- [17] J. Dean, G. Corrado, R. Monga, and K. Chen, "Large scale distributed deep networks," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1223–1231, Lake Tahoe, NV, USA, December 2013.
- [18] J. L. Zhang, J. Wang, Y. C. Zhao, and B. Chen, "An efficient federated learning scheme with differential privacy in mobile edge computing," in *Proceedings of the 4th EAI International Conference on Machine Learning and Intelligent Communications (MLICOM)*, pp. 538–550, Nanjing, China, August 2019.
- [19] M. Abadi and A. Chu, "Deep learning with differential privacy," in *Proceedings of the 23th ACM Conference on Computer and Communications Security (CCS)*, pp. 308–318, Vienna, Austria, October 2016.
- [20] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.
- [21] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: a client level perspective," in *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1–7, Montréal, Canada, December 2018.
- [22] K. Bonawitz, V. Ivanov, B. Kreuter, and A. Marcedone, "Practical secure aggregation for privacy-preserving machine

- learning,” in *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*, pp. 1175–1191, Dallas, TX, USA, October 2017.
- [23] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: deep learning for the internet of things with edge computing,” *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
  - [24] S. Chang and C. Li, “Privacy in neural network learning: threats and countermeasures,” *IEEE Network*, vol. 32, no. 4, pp. 61–67, 2018.
  - [25] J. L. Zhang, Y. C. Zhao, and J. Wang, “FedMEC: improving efficiency of differentially private federated learning via mobile edge computing,” *Mobile Networks and Applications*, vol. 3, pp. 1–13, 2020.
  - [26] X. Cheng, J. Zhang, and B. Chen, “Cyber situation comprehension for IoT systems based on APT alerts and logs correlation,” *Sensors*, vol. 19, no. 18, p. 4045, 2019.
  - [27] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proceedings of the AISTATS*, pp. 1–10, Lauderdale, FL, USA, April 2017.