Sustainable Intrusion Detection with New Attack Classification in Private Clouds

Yu Jing[1], Zhiwei Zhang[1,2], Tianzhu Hu[1,2], Zhaoyang Li[1], and Senpeng Liu[1]
[1]School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071, China
[2]Institute of Network Information, Academy of Systems Engineering, Academy of Military Sciences, China

Traditional data-driven intrusion detection systems (IDSs) are typically based on the recognition of some specific features, regulations or patterns belonging to the well-defined known attacks, so they cannot separate new or unknown attacks from abnormities and may even confuse new attacks and legitimate behaviors. With the development of artificial intelligence (AI) technology, it becomes the mainstream technology to improve the detection performance of intrusion detection system. However, the available AI-driven IDSs can hardly classify different types of new attacks separated from abnormities, and they are usually not dedicated to the private cloud, edge or fog computing environments, where the update of the new attack recognition can be very different from that of the public environments. In this article, we present a novel sustainable and AI-driven intrusion detection scheme to support the classification of new attacks in the private clouds. We first adopt the convolutional neural network algorithm to recognize the known attacks, and then propose a new model of recognition and classification for unknown attacks based on network behaviors. We further propose a new approach to update the attacks recognition model for the private clouds. Finally, we provide extensive experiment results to demonstrate that our proposed scheme outperforms the previous IDSs in terms of attack detection accuracy, attack classification accuracy and updating efficiency.

Index Terms—**Intrusion Detection, Deep Learning, New Attack Classification, Private Cloud Computing.**

I. Introduction

NOWADAYS, various intrusion detection systems (IDSs) are widely used in many different environments. They can cooperate with firewalls and antiviruses to protect informational assets from attacks aroused by adversaries or users' abnormal operations. As a type of active defense technique, IDSs monitor network traffic, observe user behaviors, and analyze system records to detect potential treats. Generally, there are two kinds of implemented modes of IDS: network-based IDS (NIDS) and host-based IDS (HIDS).

As a new type of resource organization and service provision model, cloud computing can provide flexible services of computation, storage, and networking on demand. The cloud can also promote IDSs at the same time. Before the conception of cloud computing, there were traditional IDS, which are mostly based on data-driven intrusion detection methods. They typically recognize the intrusion depending on some specific features, regulations, or patterns of well-known and well-defined attacks. As a result, these IDSs cannot separate new or unknown attacks from abnormities, and sometimes they can even confuse new attacks and legitimate behaviors. Then, with the popularization of cloud computing and the rise of artificial intelligence (AI), especially deep learning (DL), over the last decade, more AI algorithms have been introduced to improve traditional IDSs, and these AI-driven or DL-driven IDSs can recognize known and unknown attacks. Unfortunately, few IDSs can further classify new recognized attacks in detail. Moreover, existing IDSs usually are not dedicated to the private cloud, edge, or fog computing environments, in which the updating of the new attack recognition can be very different from that of public environments.

Manuscript received November 8, 2021; revised January 17, 2022. Corresponding author: Zhiwei Zhang (email:zwzhang@xidian.edu.cn).

In this paper, we present a novel sustainable intrusion detection scheme supporting new attack classification for private deployed clouds. The main contributions are as follows: (1) A modified convolutional neural network algorithm that improves the recognition of the known attacks. (2) A new model of recognition and classification for unknown attacks. (3) An update of attack recognition model in private clouds.

In the rest of this paper, we summarize related work in Section II, introduce the architecture and concrete construction of our scheme in Section III and IV respectively, present the experimental analysis in Section V, and offer conclusions in Section VI.

II. Relevant work

In this section, we summarize the literature related to our work, including the intrusion detection model, attack detection for the cloud, and unknown attack detection.

A. Intrusion Detection Model

Early intrusion detection methods usually identify whether system is invaded based on data mining of certain specific features or rules belonging to clearly defined known attacks. In [1], Lee et al. proposed using data mining techniques to discover consistent and useful patterns of system features that describe program and user behavior, and used the set of relevant system features to compute classifiers that can recognize anomalies and known intrusions. However, these IDSs cannot separate new or unknown attacks from anomalies, and even confuse new attacks and legitimate behaviors.

With the development of AI algorithms, intrusion detection based on machine learning has received extensive attention from many researchers. In [2], Van et al. used deep learning techniques to implement an anomaly-based NIDS. It deduces

part of its knowledge from incomplete data and the adaptability. Yin et al. explored how to model an intrusion detection system based on deep learning, and they proposed a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS) in [3]. Lianou et al. proposed a design that adopts a signature-based intrusion detection approach involving both centralized and distributed IDS modules in [4]. In [5], Tama et al. proposed an improved IDS based on hybrid feature selection and two-level classifier ensembles. A hybrid feature selection technique comprising three methods, that is, particle swarm optimization, ant colony algorithm, and genetic algorithm, to reduce the feature size of the training datasets. Mazini et al. proposed a new reliable hybrid method for an anomaly network-based IDS (A-NIDS) using an artificial bee colony (ABC) and AdaBoost algorithms to gain a high detection rate (DR) with a low false positive rate (FPR) in [6].[7] proposes an adaptive ensemble learning model. By adjusting the proportion of training data and setting up multiple decision trees, Sarker et al. construct a MultiTree algorithm. They choose several base classifiers, including decision tree, random forest, kNN, DNN, and design an ensemble adaptive voting algorithm.In [8], Otoum et al. presents a comprehensive analysis of the use of machine and deep learning (DL) solutions for IDS systems in wireless sensor networks (WSNs). They introduce restricted Boltzmann machine-based clustered IDS (RBC-IDS), a potential DL-based IDS methodology for monitoring critical infrastructures by WSNs. In [9], Sarker et al. present an Intrusion Detection Tree ("IntruDTree") machine-learning-based security model that first takes into account the ranking of security features according to their importance and then build a tree-based generalized intrusion detection model based on the selected important features.In [10], Otoum et al. propose a highly scalable and hybrid DNNs framework called scale-hybrid-IDS-AlertNet which can be used in real-time to effectively monitor the network traffic and host-level events to proactively alert possible cyberattacks.However, these solutions are not dedicated to private cloud, edge, or fog computing environments, and cannot accurately identify unknown attacks. Thus, they have high false alarm rates in real environments.

B. Attack Detection for Cloud

With the recent rise of cloud, edge, or fog computing, researchers have begun to study attack detection methods for these special environments. Hatef et al. presented a comprehensive and accurate solution to detect and prevent intrusions in cloud computing systems by using a hybrid method in [11], called HIDCC. Moustafa et al. proposed a collaborative anomaly detection framework (CADF) for detecting cyber attacks from cloud computing environments in [12]. In [13], Idhammad et al. presented a distributed machine learning intrusion detection system for cloud environments. The proposed system was designed to be inserted in the cloud side by side with the edge network components of the cloud provider. Nguyen et al. proposed a novel framework that leverages a deep learning approach to detect cyber attacks in the mobile cloud environment in [14]. Mugunthan et al. proposed method utilizes the hidden Markov Model for observing the flow in the network and the Random forest in classifying the detected attacks from the normal flow in [15]. In [16], system is designed to detect web attacks and is deployed on edge devices. The cloud handles the above challenges in the paradigm of the Edge of Things. Multiple concurrent deep models are used to enhance the stability of the system and the convenience in updating. [17] proposes a new system for detecting DDoS attacks in cloud computing environment. The proposed system is built using voting extreme learning machine (V-ELM), a type of artificial neural network. Dhanapal et al. proposed a novel method to detect slow HTTP DDoS attacks in the cloud in [18]. In [19], a new method is proposed to detect and defend against the DDoS attacks using autonomous multi agent system and the agents use the particle swarm optimization among themselves to have strong communication and accurate decision making. [20] proposes a big data framework to overcome traditional data processing limitations and to exploit distributed resources effectively for the most compute-intensive tasks. The identification of new attacks in private clouds may be very different from that in the public environment. These solutions cannot separate new or unknown attacks from anomalies, and perform poorly when encountering unknown intrusions.

C. Unknown Attack Detection

As for unknown intrusion detection, there has been little research on discovery in this area to our knowledge. In [21], Denatious et al. conducted a survey on data mining techniques applied to intrusion detection systems for the effective identification of both known and unknown patterns of attacks, helping the users to develop secure information systems. Casas et al. proposed an unsupervised network intrusion detection system (UNIDS) capable of detecting unknown network attacks without using any kind of signatures, labeled traffic, or training in [22]. Jongsuebsuk et al. proposed detecting unknown or new network attack types with a fuzzy genetic algorithm approach in [23]. Xuan et al. presented a method of detecting APT attacks based on monitoring access to unknown domains, and this detection method resulted in high effectiveness in the initial stage of APT attacks in [24]. [25] propose a clustering-enhanced transfer learning approach, called CeHTL, which can automatically find the relation between the new attack and known attack. Zhang et al. investigates how the Extreme Value Theory (EVT) is applied to unknown network attack detection system and proposes a network intrusion detection method based on open set recognition in [26]. Based on open-set recognition, Zhang et al. propose the Open-CNN model to implement intrusion detection and detect unknown attacks in [27]. Wang et al. proposes an identification strategy for unknown attack behaviours through the joint learning of spatiotemporal features in [28]. Although these solutions can identify new or unknown attacks from anomalies, they cannot reclassify newly identified attacks and update the identified new attacks to the intrusion detection model.

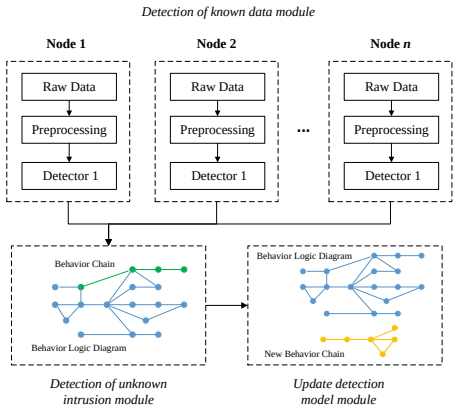

Fig. 1. System architecture.

III. Sustainable intrusion detection with new attacks classification in private clouds

To solve the problem that existing IDSs cannot identify and classify unknown attacks, we present a novel sustainable intrusion detection scheme supporting new attack classification for private deployed clouds.

A. The Concept of Unknown Attack Detection

With the development of artificial algorithms, various classification algorithms have been applied to intrusion detection. However, there are always new attacks in the real world. When these detection models encounter new attacks that are not encountered during training, because the dataset trained by the model has difficulty covering all attack types, it will extract features of new attacks to match features of their known attack types, which can produce a high number of false positives.

After the detection model is deployed, when it is detected that a network behavior is different from the known daily behavior, and this behavior damages the normal operation of the system, we argue that this is an unknown abnormal behavior, mainly an attack.

B. System Architecture

We propose a new intrusion detection method using behavior logic analysis to recognize and classify unknown attacks in the private cloud. The architecture of our system is shown in Fig. 1, and has three main modules: detection of known attack module, detection of unknown attack module, and updated detection model module.

Detection of known attack module: We preprocess the captured raw network data as input to the model. The detection model based on a machine learning algorithm is used to detect the network behavior in the system and determine whether it is normal data or a known malicious intrusion. If neither, it is temporarily marked as unknown.

Detection of unknown attack module: For the unknown, we extract its behavior features and construct a behavior logic chain according to a time sequence, which is matched with the network behavior logic diagram generated by the system based on daily data, and further determine whether it is a normal behavior or a new type of attack. The behavior of the system is complex, so there are many behavioral logic diagrams generated. When the normal behavior chain matches one of the behavior logic diagrams, it is determined as a normal behavior, as shown in Fig.1.

Updated detection model module: Generate the newly discovered logical relationship of intrusion behavior into a network behavior chain to complete the sustainable update of the detection model, which ensures the sustainability of the model update process. Similar behaviors in the future can be found to determine whether it is an intrusion. The yellow nodes represent the generated new behavior logic chain in Fig.1.

IV. The concrete scheme construction of sustainable intrusion detection with new attacks classification

In this section, we introduce the three modules of our proposed method in detail.

A. Detection of Known Attack

Deep learning is developed based on shallow learning research, increasing the number of hidden layers and using massive data training so that it has better feature representation and learning ability. The convolutional neural network is a special deep learning algorithm. Since it was proposed, it has been used to make remarkable achievements in the fields of speech and image recognition. However, the application of convolutional neural networks in the field of network security is still in the developmental stage.

A convolutional neural network can simulate the multi-layer neural network of the human brain, promote the understanding of the complex relationship between network connection data features in the intrusion detection system, and extract more effective information for attack recognition through autonomous learning.

A convolutional neural network is essentially a mapping from input to output. If the input data are trained by the mode of the network itself, the mapping relationship between input and output can be obtained without any precise mathematical expression between the input and output.

The basic structure of a convolutional neural network consists of an input layer, convolution layer, pooling layer, full connection layer, and output layer. The input layer is used to preprocess the original input data, such as de-mean, normalization, and whitening. The convolution layer uses filters to extract features at different levels from the input data by setting a series of parameters. The pooling layer can compress network parameters, reduce overfitting, and reduce network complexity. The neurons in the fully connected layer are connected to all neurons in the upper layer, which is located at the last layer of convolutional neural network to calculate the probability of the category.

Convolutional neural network model training includes forward propagation and error back propagation. First, the prediction results are obtained through forward transmission, and then the error is calculated. Finally, the model parameters are

updated by error back propagation, and the final convolutional neural network intrusion detection model is obtained by iteration.

1) Convolutional calculation

Firstly, in the forward transfer stage, the feature vector is preprocessed and then convolution is performed with the convolution kernel to obtain the feature matrix with the same number and format as the convolution kernel. Defining $\otimes$ as convolution operation, then the convolution process can be expressed as:

$$X^{(i+1)} = f\left(\sum_{k\in M} W_{jk}^{(i)} \otimes X^{(i)} + b_j^{(i)}\right), \tag{1}$$

in which, $W_{jk}^{(i)}$ represents the $i-th$ layer, the $k-th$ input feature vector corresponds to the $j-th$ convolution kernel parameter, X^i represents the input feature vector of the $i-th$ convolution layer, X^{i+1} represents the output of the $j-th$ convolution kernel in the $i-th$ convolution layer, $b_j^{(i)}$ is the bias vector of the $j-th$ convolution kernel in the $i-th$ convolution layer, $f(x)$ is the incentive function, M is the set of all input feature vectors.

2) Pooling operation

The pooling is to extract the features between data, thereby reducing the amount of data transferred to the next layer of network. The down-sampling of the sampling layer based on the principle of local correlation , which can reduce the amount of data and retain sufficient useful information. Suppose the output of the sampling layer after the $i-th$ convolution layer is X^{i+1}, and the input is X^i, which means the output of the $i-th$ convolution layer. The forward transmission process of the sampling process can be expressed as:

$$X^{(i+1)} = subsampling\left(X^{(i)}\right), \tag{2}$$

in which, $subsampling(x)$ is a sampling function, and the common sampling methods are maximum sampling and mean sampling. In the sampling process, the multiplicative bias is 1 and the additive bias is 0. In the training process, a one-dimensional feature vector is finally generated through multiple convolution and sampling operations.

3) Fully connected layer

After convolution and pooling operations, the input data are transformed into feature vectors, which cannot be directly used for classification. It is necessary to further classify the feature vectors by connecting the full connection layer. Full connection layer forward transmission process can be described as:

$$c^{(l+1)} = w^{(l+1)}a^{(l)} + b^{(l+1)}, \tag{3}$$

$$a^{(l+1)} = f\left(c^{(l+1)}\right), \tag{4}$$

in which, $l = 1, 2, \ldots, L$, where L is the total number of layers in the output layer, a^{l+1} represents the output of $(l+1)-th$ layer, c^{l+1} represents the input weighted sum vector of $(l+1)-th$ layer, $f(x)$ is the excitation function, w^{l+1} is the weight of $(l+1)-th$ layer, and b^{l+1} is the bias of $(l+1)-th$ layer.

4) Error calculation

Normally, mean square error (MSE) is used as the error function. For the training set, assuming that the output of the neural network is the number of output neurons, the mean square error can be described as :

$$J(w,b) = E_k = \frac{1}{2}\sum_{i-1}^{n}(\hat{y}_k - y_k)^2, \tag{5}$$

where E_k is the mean square error of the $k-th$ output, which is related to weight w and threshold b .

5) Error back propagation

The principle of the parameter updating process of the neural network model is to adopt the gradient descent strategy, and adjust the parameters with the negative gradient direction of the target as the standard, that is, to update the weights and thresholds according to the reverse results of the error function. Suppose the error function is $J(w,b)$. In fully connected neural networks, the vector expression for parameter update is:

$$w^{(l)} = w^{(l)} - \alpha\frac{\partial J}{\partial w^{(l)}}, \tag{6}$$

$$b^{(l)} = b^{(l)} - \alpha\frac{\partial J}{\partial b^{(l)}}, \tag{7}$$

in which, α is the learning rate. From the chain derivation rule, we can get:

$$\frac{\partial J}{\partial w^{(l)}} = \delta^{(i+1)}\left(a^{(l)}\right)^T, \tag{8}$$

$$\frac{\partial J}{\partial b^{(l)}} = \delta^{(l+1)}, \tag{9}$$

$$\delta^{(l+1)} = \frac{\partial J}{\partial a^{(l+1)}} f\left(c^{(l+1)}\right), (l = 1, 2, 3, \ldots, L-1), \tag{10}$$

in which, δ^0 is called the energy function. According to Formula (5), all parameters of the output layer are updated after several iterations.

The error back propagation of the pooling layer is different from the back error propagation of the fully connected part, the pooling layer reduces the dimension of the feature matrix, so δ^{i+1} needs to be up-sampled $up()$ as the matrix dimension of the convolution layer:

$$\delta^{(i)} = f'\left(c^{(i+1)}\right) \divideontimes up\left(\delta^{(i+1)}\right), \tag{11}$$

in which, $\divideontimes$ represents the multiplication of each element. The calculation process of the error back propagation parameters of the convolution layer is similar to that of the fully connected layer. Similarly, the vector expression of the parameter update is :

$$W_{jk}^{(i)} = W_{jk}^{(i)} - \alpha\left(X^{(i-1)} \divideontimes \left(\delta^{(i)}\right)^T\right), \tag{12}$$

$$b_j^{(i)} = b_j^{(i)} - \alpha\sum_{s,t}\left(\delta^{(l)}\right)_{st}, \tag{13}$$

$$\frac{\partial J}{\partial b_j^{(i)}} = \sum_{s,t} \left(\delta^{(l)}\right)_{st}, \tag{14}$$

$$\frac{\partial J}{\partial W_{jk}^{(i)}} = X^{(i-1)} \divideontimes \left(\delta^{(i)}\right)^T, \tag{15}$$

in which, $(x)_{st}$ means to traverse all elements of x .

B. Detection of an Unknown Attack

We extract the behavior features of the unknown network behavior to construct its behavior logic chain, and match the network behavior logic diagram of the system to further determine whether it is a new type of attack.

1) Recording network behavior

We first define a variety of network behaviors in the private cloud, and then propose a multi-dimensional network behavior recording method combined with the features of network communication data. The quantified expression of the network behavior provides support for the construction of the network behavior logic chain.

The definitions of the main network behaviors is as follows.

Address resolution behavior: The network behavior in which the access node requests the management node to convert the network address to a physical address. It can be further divided into address resolution request behavior and address resolution response behavior.

Dynamic address allocation behavior: When a new node accesses the network, the network behavior that requests the network management node to dynamically assign the unique address identifier. It can also be divided into dynamic address request behavior, dynamic address provision behavior, dynamic address selection behavior, dynamic address confirmation behavior, dynamic address re-request behavior, and dynamic address update behavior.

Data transmission behavior: The network behavior of a network node transmitting data to other nodes.

Connection establishment behavior: The network behavior in which a network node and other nodes establish a communication connection for data transmission. It can be divided into connection establishment request behavior, connection establishment response behavior, and connection establishment confirmation behavior.

Connection release behavior: The network behavior in which a network node and other nodes release the connection after completing data transmission. It can be divided into connection release request behavior, connection release response behavior, and connection release confirmation behavior.

Service request behavior: The network behavior in which a network node provides a service description to a network management node to request service. It can be divided into request service description behavior, request service matching behavior, and request service response behavior.

In private clouds, to identify the above network behaviors, the network communication data packets can be captured, and the network behavior identification can be completed by analyzing the protocol identifier or other protocol type fields in the header of the data packet. Finally, the sub-behavior is determined according to the type of control field in the packet.

2) Construct network behavior transition diagram

Taking the single network data transmission between nodes as the basic unit, and the topology information of each node on each data transmission path, the network behavior information and the temporal relationship between behaviors are recorded. The basic unit that records the network behavior and its temporal relationship on a single data transmission path is called the "information chain".

Taking multiple network behavior information chains within a certain period between the specified source node and the destination node as the input, multiple network behaviors arranged in chronological order within the information chain are extracted, and similar network behaviors among the information chains are merged.

The records of each network behavior are regarded as a multi-dimensional representation vector. The data of each dimension of a single representation vector is normalized and standardized, so that the labeled data of different specifications are transformed into the same specification. Then, the multi-dimensional representation vector corresponding to the same kind of network behavior is calculated by dot product, and the results are compared with the preset threshold (this threshold can be set by the total number of network behavior). If it is greater than this threshold, it is regarded as the similarity of network behaviors. These two network behaviors can be regarded as the same kind, and the same kind of network behaviors can be integrated. The integration of the whole network similar behaviors can be completed by circulating this process.

After merging similar network behaviors in multiple information chains, the state transition probability matrix based on the Markov property is used to determine the network behavior transition diagram. First, we determine the initial state vector of network behavior inference and calculate the first-order state transition probability matrix. Then, we use the first-order state transition probability matrix, the probability value of the current state transition to each state in the next step is calculated, and the logic inference is made.

The process of constructing the first-order state transition probability matrix is as follows: The time step of the initial behavior in the information chain is set to 1, and the time step of the next behavior is set to 2, and so on, until the time step of the last behavior is T. Taking the network behavior, x, occurring at the time step, t, as an example, the number of times, n, of a network behavior, y, appearing in multiple information chains at $t+1$ time is counted, and the ratio of n to the total number of network behavior N is the probability of network behavior x transferring to network behavior y at t time. By applying this process to any two network behaviors, the state transition probability matrix of network behavior at t time can be obtained. For each time step cycle in a time period T, the transition probability matrix for each step of state transition prediction can be obtained. Each node in the network behavior is connected, and the network behavior transition diagram is constructed with the weight of the state transition probability as the edge.

3) Generate network behavior logic diagram

Among the various behavior transition graphs constructed, the number of behavior chains connecting the source node and the destination node is large, and the transfer logic relationship between behaviors is complex. To extract a strong logical network behavior logic chain from each network behavior transition graph, the Path Ranking Algorithm (PRA) algorithm based on graph structure is adopted, and the state transition probability between network behavior nodes is used as a feature for chain prediction reasoning.

Random walk feature selection method is used to prune secondary paths. There are two feature selection metrics : set precision and coverage on each relationship path π . The accuracy calculation formula is as follows :

$$precision(\pi) = \frac{1}{n}\sum_{i} P(s_i \to G_i; \pi), \tag{16}$$

$$\text{cov}erage(\pi) = \sum_{i} I(P(s_i \to G_i; \pi) > 0), \tag{17}$$

in which, $P(s_i \to G_i; \pi)$ is based on the source agent s_i as the starting point, and the probability of reaching the target agent by performing a random walk along the relational path π. The PRA algorithm sets thresholds for precision and coverage respectively. Only when the two measurement values are not less than the relationship path of the set parameters, can they be retained as features . After selecting useful relational paths as features, we need to calculate their eigenvalues $s_{h,p(e)}$for each path network behavior (h, t). $s_{h,p(e)}$ can be understood as the probability that t can be reached from h along the path P. In the initial stage of random walk, $s_{h,p(e)}$ is initialized to 1. If $e = s$, otherwise, it is initialized to 0. The updating principle of $s_{h,p(e)}$ during random walk is as follows :

$$s_{h,p(e)} = \sum_{e' \in range(p')} s_{h,p'(e')}(e') \cdot P(e|e'; r_l). \tag{18}$$

In the above formula, $P(e|e'; r_l) = r_l(e', e)/|r_l(e')|$ indicates the probability that a node e can be reached by walking along the relationship r_l one step from node e'. By calculating the reachability of head-tail pair network behavior on each feature relation path, the values of all features of the head-tail pair network behavior (h, t) can be obtained .

For relation r, after a series of path features $P_r = \{P_1, \ldots, P_n\}$ are obtained by random walk, PRA uses these path features to calculate. The score calculation method of a combination of head network behavior and tail network behavior is as follows :

$$score = \sum_{p_i \in P_r} \theta_i s_{h,p_i(e)} \tag{19}$$

Based on the score of each path, the importance of the relationship path is sorted. We take the highest-ranking value, that is, the behavior chain with the strongest logic in the relational path as the behavior logic chain from the source node to the target node. These behavior logic chains are reduced and merged to form a network behavior logic diagram, shown as Fig. 2.

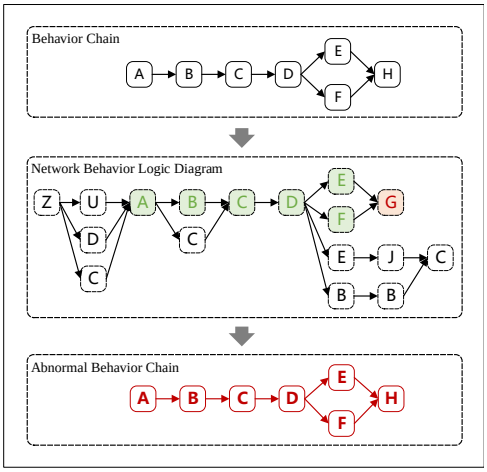

Fig. 2. Unknown abnormal behavior detection process.

4) Extract the network behavior chain for detection

After the current behavior is judged as unknown by the attack detection model, the network behavior temporal relationship information of all tasks from the current time to a fixed time interval is collected for similar behavior merging to form a behavior chain. The network behavior chain after pretreatment is detected and analyzed. When the behavior chain cannot match the network behavior logic diagram, the behavior chain is marked as abnormal, and the network behavior between nodes corresponding to this behavior chain is prevented in time.

In Fig. 2, the nodes A, B, C, D, E, F, and H represent the network behavior between nodes in the private cloud environment. Multiple real-time temporal relationship information chains of network behaviors among nodes in the network are used as input, and similar behaviors are merged to form the behavior chain. For a network behavior chain, the network behavior logic diagram predicts what will happen next based on the network behavior observed in the training stage, to determine whether the unknown behavior is abnormal.

C. Updating detection model

When a behavior chain is marked as abnormal, a new intrusion logical chain is generated according to the network behavior sequence. The construction of new intrusion logical chain mainly includes the following four categories: sequential execution, concurrent execution, new task detection, and loop recognition.

If a sequence always follows a node, that is, the probability is 1, then it is considered that the current behavior and historical behavior sequence are from the same chain, and they are merged into the workflow of this chain. For example, Fig. 3 (a) shows that a sequence $\{A, B, C\}$ whose output is predicted to be $\{D : 1.0\}$ determines that $\{A, B, C, D\}$ comes from a chain.

Figure 3 (b) shows that if the output of the previous module is a set of different network behavior sets with probability sum of 1, there is a divergence point. After we merge the predicted maximum probability behavior into the historical

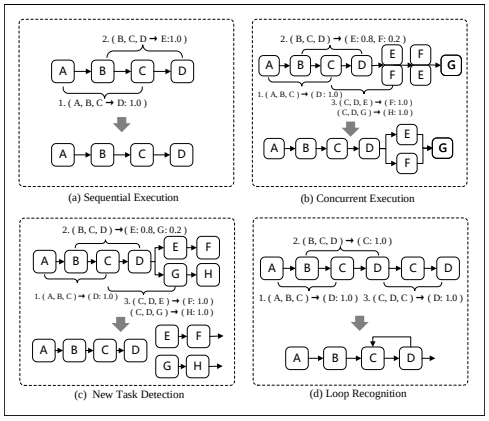

Fig. 3. Types of construction of the new intrusion logic chain.

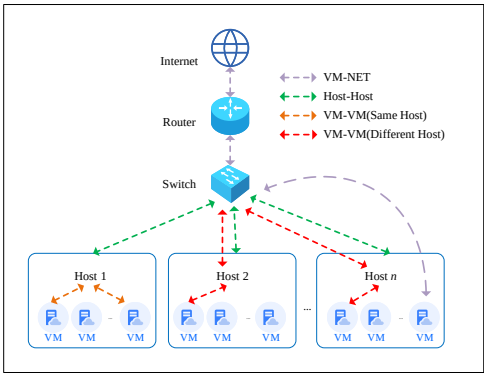

Fig. 4. Private cloud network architecture.

behavior sequence, the next maximum probability behavior will appear again in the next prediction, and its probability will be higher compared with the previous prediction. When all the network behaviors at the divergence point are included in the historical behavior sequence, the prediction will finally become certain. For example, sequence $\{B, C, D\}$, output prediction $\{E : 0.8, F : 0.2\}$; after merging E into historical behavior sequence $\{C, D, E\}$, with the output prediction $\{F : 1.0\}$, it is determined that E and F are from concurrent threads.

Figure 3 (c) shows that if the network behavior of the candidate set predicted by the divergence point does not appear one after another, the next prediction will be a deterministic prediction of a new network behavior by merging a network behavior of the candidate set into a historical sequence. For example, in the case of $\{C, D, E \rightarrow F : 1.0\}\{C, D, G \rightarrow H : 1.0\}$, we stop extending the current network behavior logic chain (stop at D in this case) and begin to build behavior logic chain for new network behavior.

As shown in Fig. 3 (d), if there are repeated fragments in a historical behavior sequence, such as $\{A, B, C, D, C, D\}$, we can identify the repeated fragments $\{C, D\}$ as loops.

V. Experiment Results and Discussion

In our section, we introduce the environment and dataset used in our experiment, and analyze our method from the aspects of effectiveness.

A. Experimental setup

The main difference between network traffic and Internet traffic data in the cloud environment is that the cloud environment contains the internal communication between virtual machines on the same physical machine. Only data collected by traditional intrusion detection methods cannot adequately reflect the network situation in the private cloud environment.

Therefore, in our experiments, we used the CIC-IDS2017 dataset mixed with datasets collected from private cloud environments for evaluating the performance of our model.

Due to the network virtualization, private clouds typically employ a multi-layered network structure, and there are four types of network traffic. (1)VM-NET is used to represent the network traffic between the virtual machine and the outside of the cloud environment. These traffics could pass through the physical machine network port and various internal network equipment, and ultimately achieve the destination address. (2)Host-Host is used to represent the mutual communication flow between physical machines in the cloud environment, which only passed through the internal switches in the cloud environment. (3)VM-VM is used to represent the communication flow between virtual machines in cloud environment, which was divided into two categories: One was the communication between virtual machines on the same physical machine, which did not involve external network equipment. (4)The other was the communication between virtual machines on different physical machines. The path would reach the destination virtual machine through the physical machine network port.

To make the model more consistent with the actual production environment, we collected the communication traffic flow data between one hundred nodes in a private cloud. We collected virtual machine traffic data through the same physical network port and stored it as pcap files. The collected network data was formatted and saved by CICFlowMeter component, and then the extracted feature data was transmitted to the intrusion detection module after data preprocessing. The dataset included common communication protocols such as TCP, UDP, HTTP, HTTPS, FTP, and DHCP. The network virtualization architecture is shown as Fig. 4.

The CIC-IDS2017 intrusion detection dataset was compiled and published by the Canadian Institute of Cyber Security, including benign and latest common attacks: Brute force, DoS, DDoS, XSS, SQL Inject, PortScans and Botnets. The dataset included common network protocols such as HTTP, HTTPS, FTP, SSH, and Email. Furthermore, it provided massive real-world pcap file data. The CICFlowMeter tool could extract network flow features from the network communication data. We combined the data we collected in the actual environment with the data in the CIC-IDS2017 dataset to form a mixed data set with ten types of attack data and one type of normal data. To avoid the problem of data imbalance in multi-classification training, we adopted the mixed sampling method to resample the dataset and thereby improve the detection accuracy of small data samples. Our mixed dataset contains Benign, Bot, DDoS, DoS GoldenEye, DoS Hulk, Dos Slowhttptest, DoS

TABLE I
COMPONENTS OF THE DATASET.

	NUMBER
Benign	100000
Bot	100000
DDoS	100000
DoS GoldenEye	100000
DoS Hulk	100000
DoS Slowhttptest	100000
DoS Slowloris	100000
FTP-Patator	100000
PortScan	100000
SSH-Patator	100000
Web Attack	100000

TABLE II
CLASSIFICATION ACCURACY COMPARISON AMONG DIFFERENT MODELS

	LSTM	DNN	Ours
BENIGN	0.88	0.91	0.99
Bot	0.94	0.23	0.96
DDoS	0.97	0.98	0.99
DoS GoldenEye	0.57	0.58	0.96
DoS Hulk	0.94	0.98	0.98
DoS Slowhttptest	0.95	0.39	0.97
DoS slowloris	0.87	0.65	0.95
FTP-Patator	0.94	0.68	0.97
PortScan	0.97	0.90	0.99
SSH-Patator	0.89	0.20	0.71
Web Attack	0.67	0.08	0.67

Slowloris, FTP-Patator, PortScan, SSH-Patator, Web Attack, each of which has 100,000 samples, as shown in Table I.

B. Analysis of Effectiveness

The experimental environment was built based on a PC with Ubuntu18.04 64-bit system, Intel(R) Xeon Silver 4216 CPU @ 2.1GHz, GeForce RTX 2080 Ti GPU, 128GB RAM. We used the Python 3.8 and scikit-learn machine learning library as programming languages and tools.

The detection of the known data module used the convolutional neural network to discover and classify known intrusions. The convolutional neural network model used a 2D convolution layer, and the number of convolution filters was 32, 64, and 128. We selected 80% from the data set as the training data set and the remaining 20% as the test data set.

It can be seen from Table II that our proposed scheme had the highest accuracy in known intrusion detection, especially the detection accuracy of PortScan and DDoS; two high-frequency network attacks reached 0.99.

In addition, we selected a number of different categories as known categories, including BENIGN, DDoS, DoS, PortScan, and FTP Patator, and other categories as unknown categories. It is worth noting that in the test phase, if an unknown attack as input was identified as known data, it was also considered a decision error.

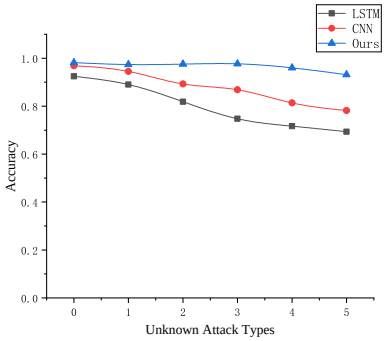

Fig. 5. The accuracy of different model under different unknown attack types.

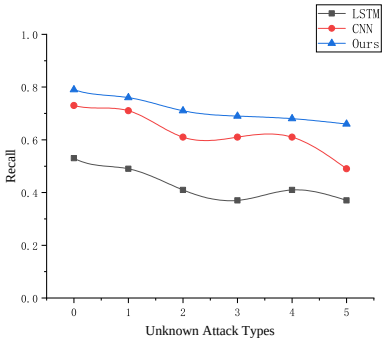

Fig. 6. The recall of different model under different unknown attack types.

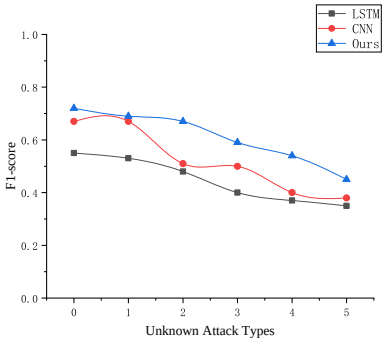

Fig. 7. The F1-score of different model under different unknown attack types.

It can be seen from Fig.5 that our method had similar accuracy in detecting unknown types of intrusions and known types of intrusions, and both had high accuracy. When the model is trained with all data types, it can be seen that the accuracy of the three models is similar. However, after mixing unknown intrusion data in the testing process, other high-precision models have greatly reduced the accuracy of detecting known types of intrusion as the types of unknown intrusion data increase.

VI. Conclusion

This paper proposed a novel intrusion detection scheme for CPSs with private clouds, and it was designed to recognize and classify unknown attacks without any external acknowledgment of these new attacks. Our future work will focus on developing new learning methods to improve the generalization detection ability of intrusion detection models. With the rapid development and widespread application of intelligent equipment as well as intelligent systems, this self-evolution quality will empower the AI-driven protection techniques for military, medical, industrial, and other security- and privacy-sensitive scenarios.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61941114), the Project of Hainan Province Key Research and Development Program (ZDYF2019202), the Key R&D Program of Shaanxi Province (2021ZDLGY03-10), and the Fundamental Research Funds for the Central Universities (JB210301).

References

[1] W Lee, S Stolfo, "Data mining approaches for intrusion detection," *7th USENIX Security Symposium*, pp. 79–94, 1998.

[2] N. T. Van, T. N. Thinh, "An anomaly-based network intrusion detection system using deep learning," in *2017 international conference on system science and engineering (ICSSE) IEEE*, 2017, 210-214.

[3] C. Yin, Y. Zhu, J. Fei, et al., "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, 2017, 5, 21954-21961.

[4] I. lianou, P. kypros, et al., "A signature-based intrusion detection system for the Internet of Things," *Information and Communication Technology Form (2018)*.

[5] B. A. Tama, M. Comuzzi, K. H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system". *IEEE Access*, 2019, 7, 94497-94507.

[6] M. Mazini, B. Shirazi, I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms". *Journal of King Saud University-Computer and Information Sciences*, 2019, 31(4), 541-553.

[7] Gao X, Shan C, Hu C, et al. "An adaptive ensemble machine learning model for intrusion detection". *IEEE Access*, 2019, 7: 82512-82521.

[8] Otoum S, Kantarci B, Mouftah H T. "On the feasibility of deep learning in sensor network intrusion detection". *IEEE Networking Letters*, 2019, 1(2): 68-71.

[9] Sarker I H, Abushark Y B, Alsolami F, et al. "Intrudtree: a machine learning based cyber security intrusion detection model". *Symmetry*, 2020, 12(5): 754.

[10] Ferrag M A, Maglaras L, Moschoyiannis S, et al. "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study". *Journal of Information Security and Applications*, 2020, 50: 102419.

[11] M. A. Hatef, V. Shaker, M. R. Jabbarpour, et al., "HIDCC: A hybrid intrusion detection approach in cloud computing". in *Concurrency and Computation: Practice and Experience*, 2018, 30(3).

[12] N. Moustafa, G. Creech, E. Sitnikova, et al., "Collaborative anomaly detection framework for handling big data of cloud computing". in *2017 military communications and information systems conference (MilCIS)*. IEEE, 2017, 1-6.

[13] M. Idhammad, K. Afdel, M. Belouch, "Distributed intrusion detection system for cloud environments based on data mining techniques", *Procedia Computer Science*, 2018, 127, 35-41.

[14] K. K. Nguyen, D. T. Hoang, D. Niyato, et al., "Cyberattack detection in mobile cloud computing: A deep learning approach," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, 1-6.

[15] Mugunthan S R. "Soft computing based autonomous low rate DDOS attack detection and security for cloud computing". *journal of soft computing paradigm (JSCP)*, 2019, 1(02): 80-90.

[16] Tian Z, Luo C, Qiu J, et al. "A distributed deep learning system for web attack detection on edge devices". *IEEE Transactions on Industrial Informatics*, 2019, 16(3): 1963-1971.

[17] Kushwah G S, Ranga V. "Voting extreme learning machine based distributed denial of service attack detection in cloud computing". *Journal of Information Security and Applications*, 2020, 53: 102532.

[18] Dhanapal A, Nithyanandam P. "The slow HTTP distributed denial of service attack detection in cloud". *Scalable Computing: Practice and Experience*, 2019, 20(2): 285-298.

[19] Kesavamoorthy R, Soundar K R. "Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system". *Cluster Computing*, 2019, 22(4): 9469-9476.

[20] Dinh P T, Park M. "BDF-SDN: A big data framework for ddos attack detection in large-scale sdn-based cloud,"*2021 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2021: 1-8.

[21] D. K. Denatious, A. John, "Survey on data mining techniques to enhance intrusion detection," in *2012 International Conference on Computer Communication and Informatics*. IEEE, 2012, 1-5.

[22] P. Casas, J. Mazel, P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Computer Communications*, 2012, 35(7), 772-783.

[23] P. Jongsuebsuk, N. Wattanapongsakorn, C. Charnsripinyo, "Network intrusion detection with fuzzy genetic algorithm for unknown attacks," in *The International Conference on Information Networking 2013 (ICOIN)*, IEEE, 2013, 1-5.

[24] C. D. Xuan, H. H. Nam. "A method of monitoring and detecting APT attacks based on unknown domainsProcedia," *Computer Science 150 (2019)*, 316-323.

[25] Zhao J, Shetty S, Pan J W, et al. "Transfer learning for detecting unknown network attacks". *EURASIP Journal on Information Security*, 2019, 2019(1): 1-13.

[26] Zhang Y, Niu J, Guo D, et al. "Unknown Network Attack Detection Based on Open Set Recognition". *Procedia Computer Science*, 2020, 174: 387-392.

[27] Zhang Z, Zhang Y, Niu J, et al. "Unknown network attack detection based on open-set recognition and active learning in drone network". *Transactions on Emerging Telecommunications Technologies*, 2021.

[28] Wang H, Mumtaz S, Li H, et al. "An identification strategy for unknown attack through the joint learning of space–time features." *Future Generation Computer Systems*, 2021, 117: 145-154.

Yu Jing Yu Jing is studying for MS degree in computer science and technology in Xidian University,China. Her research interest includes network security and cloud computing security.

Zhiwei Zhang Zhiwei Zhang received his BS degree in network engineering, MS degree in computer systems architecture and PhD degree in Cryptography from Xidian University, China. His research interest includes authentication, access control, data storage security in cloud computing.

Tianzhu Hu Tianzhu Hu received his BS degree in computer science and technology from Northwestern Polytechnical University and MS degree in computer technology from Xidian University at which he is currently studying for a doctorate degree.His research interest includes endogenous safety and intrusion detection.

Zhaoyang Li Li Zhaoyang is studying for MS degree in computer science and technology in Xidian University,China. His research interests include network security, situational awareness, intrusion detection, and vulnerability scanning in cloud computing.

Senpeng Liu Senpeng Liu is studying for MS degree in computer science and technology in Xidian University,China.His research interest includes system virtualization, data storage security in cloud computing.