IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON COMMUNICATIONS IN HARSH ENVIRONMENTS

Received November 3, 2020, accepted November 14, 2020, date of publication November 26, 2020,
date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3040856

Abnormal Detection of Wireless Power Terminals in Untrusted Environment Based on Double Hidden Markov Model

KEHE WU[1], JIAWEI LI[1], AND BO ZHANG[2]
[1]School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
[2]Global Energy Interconnection Research Institute, Nanjing 210008, China

Corresponding author: Bo Zhang (zhangbo@geiri.sgcc.com.cn)

This work was supported by the State Grid Corporation of China—Research on Security Defense Technology for Closed-Source Power Industrial Control System Science and Technology Project under Grant 5455HT20200119.

ABSTRACT The wireless power terminals are deployed in harsh public places and lack strict control, facing security problems. Thus, they are faced with security problems such as illegal and counterfeit terminal access, unlawful control of connected terminals, etc. The intrusion detection system based on machine learning and artificial intelligence significantly improve the terminal side's abnormal detection capacity. In this article, we aim at identifying the abnormal behavior of wireless power terminals based on a double Hidden Markov Model (HMM), which solves the computational complexity problem caused by high dimensions in intrusion detection systems using a single HMM. The lower-layer HMM is used to identify the discrete single network abnormal behavior. Simultaneously, the upper-layer can obtain more extended period attack behavior in multiple independent abnormal events identified by the low-level. The experiment results indicate that the intrusion detection system using proposed double HMM can effectively detect the terminal's abnormal behavior and identify the network attack behavior for an extended period.

INDEX TERMS HMM, abnormal detection, power IoT device.

I. INTRODUCTION

In recent years, the situation of network security is gradually rigorous. Network attacks initiated by terminal devices often occur. The destructive power of attacks increases obviously, and the scope of influence tends to expand. In October 2016, an anonymous attacker launched a large-scale DDoS attack by illegally controlling webcam, DVR, and other terminal devices, causing severe damage to Dyn, a DNS service provider on the east coast of the United States. The attack caused a total outage of several well-known Internet services of its customers (including Twitter, Amazon, PayPal, etc.), resulting in more than half of the Americans unable to access the Internet. Gartner, a research organization, predicts that by 2020, there will be more than 20 billion terminal devices globally, and more than a quarter of network attacks on enterprises will involve terminal devices. By the end of 2019, a total of 2526 control servers have been found to control more than 1254000 terminals, posing a serious potential security threat to the stable operation of the Internet.

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

The above situation shows that the trend of network attacks extending to the terminal side is apparent. The number of network attacks launched against terminals will continue to grow in the future. Various terminal security at the end of the network has become a key component of complete network security.

With the development of the power IoT, the types of IoT terminals connected to the smart grid are also increasing. According to statistics, 25 kinds of 15.8512 million terminals are connected to the State Grid management information area. Among them, the number of wireless power terminals is the largest, accounting for 84.87%. Due to the lack of effective monitoring means, it is difficult to find the abnormal behavior of wireless power terminals in time, which will enable the abnormal behavior to continue to destroy, obtain the company's critical data, and further launch network attacks. It is very urgent to design security measures to prevent network security incidents in such a complex and severe situation. However, it is unrealistic to avoid security attacks altogether. We can only find and block the abnormal behavior of wireless power terminals as far as possible.

The idea of abnormal behavior detection technology is to establish a benchmark behavior model based on standard network data and then compare the detected data with the benchmark behavior to determine whether there is an abnormal. Compared with other technologies, abnormal behavior detection is sufficient to identify unknown attacks in network flow. The development of machine learning technology further improves the technical advantages of abnormal behavior detection, which can detect abnormal flow and identify existing threats, specific attack types, and unknown new attack means. The anomaly detection mechanism's essence is to analyze, understand, and characterize network behavior and identify or classify abnormal flow instances. Therefore, from the perspective of machine learning, abnormal detection is a classification problem. The methods usually include supervised, semi-supervised, and unsupervised abnormal detection.

The supervised model requires a dataset with instances of tagged normal and abnormal categories. In this case, the typical method is to establish a prediction model for standard and abnormal classes and compare new data instances with the model to determine which class it belongs to. Compared with the standard cases in training data, the number of abnormal cases is usually much less. Meanwhile, to obtain accurate and representative tags, especially those for exception categories, is challenging.

Semi-supervised mode assumes that the training data only has tagged instances for standard categories. Since semi-supervised mode does not require tags for exception categories, it is more widely applicable than supervised techniques. A typical method is to build a model for the class, corresponding to normal behavior and use the model to identify anomalies in test data.

The unsupervised mode does not require training data and is more challenging to achieve current goals. An implicit assumption is needed: normal conditions occur much more frequently than exceptions in the test data. If the hypothesis does not exist, the false positive rate will be high.

Hidden Markov Model (HMM) is a classical model for modeling and analyzing sequence behavior. It has been widely used in many fields, such as speech recognition, natural language processing, and so on [1]. Due to the temporal characteristics of intrusion behavior, HMM's application in the field of intrusion detection has been widely concerned. However, the statistical learning algorithm used in the existing HMM schemes will increase exponentially with the increase of the analysis packets' data volume. In large dimension state space, HMM converge difficultly, which leads to training failure. The behavior recognition of HMM is only related to the current state; therefore, it will ignore the multi-state network attacks across large time scales. In order to solve the above problems, this article proposes an abnormal detection architecture based on a double HMM with two layers. The lower-layer realizes fine-grained abnormal behavior detection by detecting the network data flow frame by frame. Also, it identifies the specific abnormal attack behavior and then obtains the time series of the attack behavior. On this basis, the upper-layer realizes the identification of network attacks in a considerable period.

The structure of this article is as follows: Section 2 introduces the related work. Section 3 gives the framework of abnormal behavior identification of wireless power terminals based on double HMM. Section 4 tests the framework and discusses its performance. The last section summarizes this article.

II. RELATED WORK

In 1987, Denning first proposed the abstract model of abnormal detection and regarded intrusion detection as a security defense measure of computer systems [2]. According to the statistics of behavior portraits, Denning's general model is mainly based on host audit records to generate system behavior portraits and discover intrusion behavior. This model is a real-time, intrusion detection system model. In addition to the host audit record, the system call of the operating system kernel also reflects the program's running behavior in the computer system. Reference [3] uses the system call data set generated by different programs to accurately represent the program's normal behavior through the data modeling method and is used to detect intrusion.

Wagner and Dean proposed an abnormal detection model based on program analysis, which can construct a control flow model by static analysis of source code, instead of building a learning model from program tracking [4]. In reference [5], the control flow abnormal detection can be judged according to the control flow-sensitive attributes (i.e., the ability to analyze the execution sequence of statements) and the mutually orthogonal context-sensitive attributes (the ability to distinguish the call context at runtime). It proposes a static analysis algorithm to construct the control flow and context-sensitive models, in which context sensitivity can reduce the impossible control flow paths to be considered in the intrusion detection system. In reference [6], the Dyck model describes how NFA (nondeterministic finite automaton) is related to context-sensitivity. There is a certain balance between context-sensitive and runtime overhead. In reference [7], a context-sensitive automaton PDA (push-down automaton) is constructed, reducing the time complexity. In reference [8], several techniques are proposed to improve context-sensitivity, such as renaming system calls to distinguish different calls of the same function. The Dyck model's code connects the entry and returns the objective function's address with the call point.

The model can distinguish the call point and improve the context-sensitivity. In reference [9], CFI (control flow integrity) usually means that the program execution must follow the predetermined CFG path. The CFI property's execution can be realized by modifying the source code and object code related to control flow transfer and embedding control flow policy in the binary file. The subsequent CFI technology improves the front and back edge processing and kernel rootkit detection. In reference [10], static analysis can

be used to reduce the cost of CFI. In reference [11], Zhang and Sekar proposed a method based on static analysis, which can be used on binary files to reduce CFI's execution cost.

Furthermore, a control flow integrity framework is proposed to demonstrate the replication of functions and function pointers to prevent control-flow hijacking. Reference [12] improves CFI technology, and the monitoring system realized pays more attention to the call part of the control flow. Both data flow and control flow have specific effects on anomaly detection. Data dependency analysis has been used to model and detect malicious behaviors. Research has confirmed the validity of system call modeling parameters, such as anomaly detection according to the string distribution in reference [13].

In reference [14], WIT (Write Integrity Testing) technology can prevent memory error attacks. It can predict writable objects through static analysis. Wit technology also realizes the integrity of control flow and ensures the consistency of indirect control transmission and control flow graph during runtime. In reference [15], DFI (data flow integrity, DFI) attribute, first proposed by Castro, Costa, and Harris, refers to the consistency requirements between the runtime data stream and the static predicted data stream and demonstrates the detection process of DFI to control and uncontrolled data attacks.

The above work mainly analyzes the program statically's possible execution control flow, so it needs to deal with the considerable program execution space. The system call stack information of program execution reflects the program's actual execution process, so it can better reflect the program's behavior. Reference [16] proposed a new method for abnormal detection using call stack information. Experimental results show that this method can detect attacks that other methods cannot detect. In reference [17], the combination method of static analysis and dynamic learning is adopted. In this method, program tracing is used to define the basic static generation model. The hybrid pushes down automata (HPDA) to describe the call stack information to obtain the program's control flow efficiently. However, this model is not a probabilistic method and cannot record, model, and predict branches. In reference [18], probabilistic data mining technology is used to analyze attack behavior. Warrender *et al.* proposed the first probabilistic learning work for program behavior modeling. Probabilistic abstract interpretation in reference [19] is used to calculate and limit the knowledge gain associated with information dissemination. In reference [20], the probability of program path execution was estimated by Monte Carlo simulation. In reference [21], Sampson *et al.* Provided a framework for expressing and verifying the probability of variables in programs based on the Bayesian network model. In reference [21], a probabilistic modeling method is proposed to predict new and invisible programs' properties. An intrusion detection model for recording and evaluating call sequences is based on n-gram. This method collects call sequences (such as system calls) to form a collection of allowed call sequences, and any new or unordered call sequences are classified as exception sequences. However, this method is limited to the need to enumerate and store all possible call sequences, which affects its scalability.

Hidden Markov Model (HMM) is a classical model for modeling and analyzing sequence behavior and has been widely used in many fields such as speech recognition, natural language processing, etc. Due to intrusion behavior's temporal characteristics, the application of HMM in intrusion detection has also received extensive attention [22]. In reference [23], researchers proposed using HMM to compare two parallel abnormal detection methods. The execution graph model in reference [24] is constructed by learning the program runtime's execution mode, that is, the return address on the call stack related to the system call, and using the inductive attributes in the call sequence.

III. ABNORMAL DETECTION STRUCTURE OF WIRELESS POWER TERMINALS BASED ON DOUBLE HMM

A. OVERALL STRUCTURE

According to the smart grid system's security characteristics, this article adopts the model of network isolation and security access based on no connection. The model includes four parts: communication front-end processor, network security isolation (short for isolation), network security access gateway(short for gateway), acquisition front processor. Also, it is equipped with a self-defined private protocol for communication. First, the overall access model is described with single isolation and single gateway architecture, as shown in the following(see Figure 1):

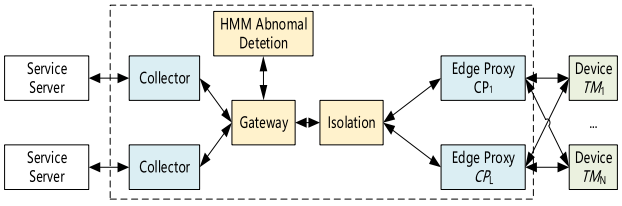


FIGURE 1. Overall structure with single isolation and single gateway.

Communication front-end processor CP: it has the socket link to access and maintain a large number of terminals, initiate a small number of sockets to connect to the isolator, and have the ability to filter private protocols and forward application layer messages (private protocols). Meanwhile, the communication front-end and the acquisition front-end interactive terminal access information to provide addressing service for the messages sent by the master station.

Network security isolation N.I.: with physical isolation and analytical isolation capabilities. In physical isolation, the classic 2 + 1 physical isolation design idea is adopted, composed of two systems: the front and post systems. The high-speed multi isolation card channel based on PCIe is used to communicate with each other, and the TCP / IP protocol of the network layer is shielded physically. The isolation card uses a high-performance FPGA chip, adopts scalable

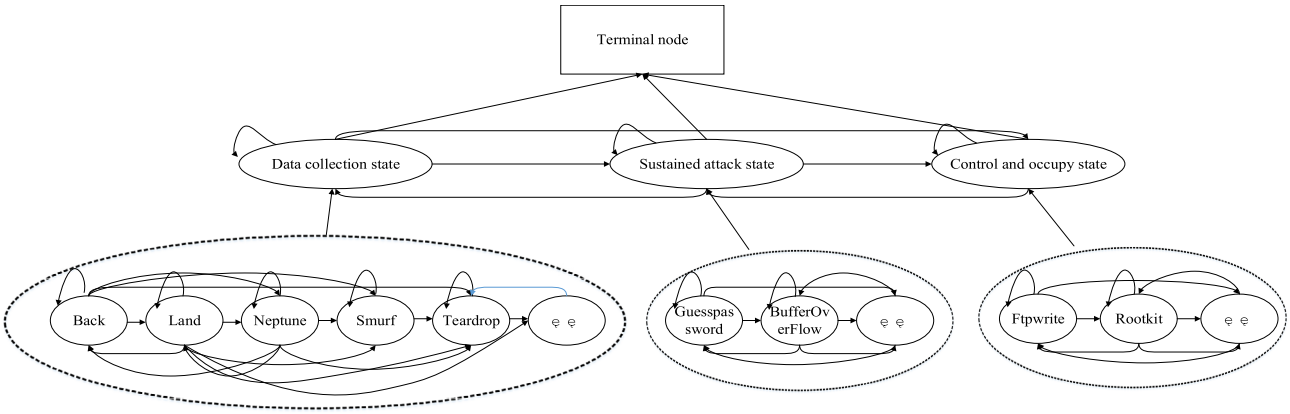


FIGURE 2. Double HMM for abnormal behavior detection.

multi-channel mode to support high-speed isolated switching, uses four channels by default, and single-channel can support 1Gbps traffic.

Network security access gateway N.G.: as a server, it provides socket access for isolation post and acquisition front. The gateway has the ability of message encryption and decryption. The gateway decrypts the data reported by the terminal. After processing, the message is forwarded to the acquisition front end in plaintext. The plaintext message sent from the acquisition front is encrypted by the gateway and sent to the ciphertext terminal.

Acquisition front processor A.P.: it encapsulates the private protocol. It is used to address the C.P. The corresponding to the private protocol's encapsulation and landing is the terminal side. After receiving the private protocol from the gateway, the A.P. unpacks the private protocol, extracts the business data, and transfers it to the master station. It receives the business data from the master station and assembles the private protocol. Next, the communication front node is designated to assist in addressing the corresponding terminal.

HMM, abnormal detection HAD: with terminal abnormal behavior detection ability. After the terminal access authentication, HAD has real-time analysis of terminal transmission message content to ensure that the legitimate terminal will not be used as a springboard to carry out network attacks. The HAD uses the HMM to analyze and predict terminal transmission behavior and excavate terminal behavior deviation. Thus, HAD prevents the legitimate terminal from abnormal behavior attacks.

B. DOUBLE HIDDEN MARKOV MODEL

HMM is a parameterized probability model used to describe the statistical characteristics of a random process. It is a double random process. One is the Markov chain, which describes states' transition, and the other random process describes the relationship between states and observations. HMM defines three kinds of probability: the initial state probability vector α, the state transition probability matrix P, and the observation probability matrix O. HMM can be expressed by these three probabilities, namely $\lambda = \{A, B\pi\}$. A represents the state transition matrix of the implicit state, which describes the transition probability between each state in the HMM model; B represents the observable state chain, which is related to the implicit state in the model; π represents the initial probability matrix, which refers to the probability matrix of the initial implicit state.

According to the characteristics of attack behavior, each attack behavior event can be described by several attack actions, and each attack action is composed of a set of abnormal behavior data time series. Therefore, we can construct a double HMM with two layers to describe aggressive behavior characteristics in a considerable period (see Figure 2).

Each layer is an HMM sequence, and the upper HMM uses the possible visible state sequence of each HMM in the lower layer to construct the second layer's training data. It will be used to train the upper HMM, which will use information from the lower HMM to learn new patterns that the lower HMM may not recognize.

Use $M = \{A_1, B_1, \pi_1, A_2, B_2, \pi_2, H\}$ to represent the double HMM, A_1, B_1, π_1 and A_2, B_2, π_2 represent the lower and upper HMM, respectively. H represents the conditional probability matrix of the upper HMM to the lower HMM.

For a specific network behavior, its parameter set M is:

1) State transition matrix A_i: In the i-th HMM, the current state can only be transferred to the next state but cannot return to the previous state, $\boldsymbol{a}_{1,2}^{(i)} = \boldsymbol{P}(\boldsymbol{S}_t^{(i)} = \boldsymbol{S}_2^i \mid \boldsymbol{S}_{t-1}^{(i)} = \boldsymbol{S}_1^i)$, $i = 1, 2$.

2) The state output probability matrix B_i, which represents the probability that the state will output an observation value at the current moment. defined as:

$$\boldsymbol{B}_J^i = \begin{bmatrix} b_1^{(i)}(c_1^i) & \cdots & b_1^{(i)}(c_k^i) \\ \vdots & & \vdots \\ b_5^{(i)}(c_1^i) & \cdots & b_5^{(i)}(c_k^i) \end{bmatrix}$$

While $b_2^{(i)}(c_m^i) = \boldsymbol{P}(\boldsymbol{c}_m^i \mid \boldsymbol{S}_t^{(i)} = \boldsymbol{S}_2^i)$, $i = 1, 2$.

3) The initial state probability distribution π_i, since the transition of the state always starts from the S_B state, it has the following definition: $\boldsymbol{\pi}_J^{(i)} = \{\boldsymbol{\pi}_1^{(i)} = 1, \boldsymbol{\pi}_2^{(i)} = 0, \ldots, \boldsymbol{\pi}_5^{(i)} = 0\}, i = 1, 2$

4) The conditional probability matrix of the upper HMM to the lower HMM as H:

$$\boldsymbol{H}^{(1,2)} = \begin{bmatrix} \mathrm{h}_{1,1}^{(1,2)} & \mathrm{h}_{1,2}^{(1,2)} & 0 & 0 & 0 \\ 0 & \mathrm{h}_{2,2}^{(1,2)} & \mathrm{h}_{2,3}^{(1,2)} & 0 & 0 \\ 0 & 0 & \mathrm{h}_{3,3}^{(1,2)} & \mathrm{h}_{3,4}^{(1,2)} & 0 \\ 0 & 0 & 0 & \mathrm{h}_{3,4}^{(1,2)} & \mathrm{h}_{4,5}^{(1,2)} \\ 0 & 0 & 0 & 0 & \mathrm{h}_{5,5}^{(1,2)} \end{bmatrix}$$

H represents the probability that there is a lower-layer attack action time sequence under the condition of the upper-layer attack action sequence state, for example, where $h_{2,3}^{(1,2)} = \mathbf{P}(\boldsymbol{S}_t^{(1)} = \boldsymbol{S}_3^2 | S_t^{(2)} = \boldsymbol{S}_2^1)$. The initial value selection in the parameter training is:

$$\boldsymbol{H}_0^{(1,2)} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Suppose the observation sequence is $\mathrm{O} = \{O_1, O_2, \ldots, O_T\}$, where each observation value is composed of observation values based on large-scale attack behavior features, and small-scale attack behavior features, using $\boldsymbol{O}_t = \{\boldsymbol{c}_t^1, \boldsymbol{c}_t^2\}$ represents the observed value at the t-the moment. The observation vector length is T state sequence, $\boldsymbol{S} = \{(\boldsymbol{S}_1^1, \boldsymbol{S}_1^2), (\boldsymbol{S}_2^1, \boldsymbol{S}_2^2), \ldots, (\boldsymbol{S}_T^1, \boldsymbol{S}_T^2)\}$. Therefore, a series of unknown network behaviors $\{O_1, O_2, \ldots, O_n\}$ and a parameter set $\lambda_j^{(1,2)}$ describing behavior J are given. Unknown behavior $\boldsymbol{O}_c = \{(\boldsymbol{c}_1^1, \boldsymbol{c}_1^2), (\boldsymbol{c}_2^1, \boldsymbol{c}_2^2), \ldots, (\boldsymbol{c}_T^1, \boldsymbol{c}_T^2)$ and the similarity of the multi-scale feature HMM of behavior J (see Equation 1) by Bayes Criterion Formula $\mathbf{P}(\boldsymbol{O}_c | \lambda_j^{(1,2)})$ is obtained.

$$\begin{aligned} \boldsymbol{a}_1(i, \mathrm{j}) &= \boldsymbol{\pi}_i^{(1)} \boldsymbol{\pi}_i^{(2)} \boldsymbol{b}_i^{(1)}\left(\boldsymbol{c}_1^1\right) \boldsymbol{b}_j^{(2)}\left(\boldsymbol{c}_1^2\right), \text{one} \leq \mathrm{i}, \mathrm{j} \leq 5 \\ \boldsymbol{a}_t(i, \mathrm{j}) &= \boldsymbol{b}_i^{(1)}\left(\boldsymbol{c}_t^1\right) \boldsymbol{b}_j^{(2)}\left(\boldsymbol{c}_t^2\right) \sum, \\ \sum &= \sum_{m,\mathrm{n}} [\boldsymbol{a}_l(\mathrm{m}, \mathrm{n}) \boldsymbol{a}_{m,\mathrm{i}}^{(1)}, \boldsymbol{a}_{m,\mathrm{j}}^{(2)}, \boldsymbol{h}_{m,\mathrm{j}}^{(1,2)}] \end{aligned} \tag{1}$$

Likelihood probability (see Equation 2):

$$\mathbf{P}\left(\boldsymbol{O}_c \mid l_J^{(1,2)}\right) = \sum_{\forall m,\mathrm{n}} [\boldsymbol{a}_T(\mathrm{m}, \mathrm{n})] \tag{2}$$

C. OBSERVABLE STATE CHARACTERISTICS OF ABNORMAL TERMINAL BEHAVIOR

How to extract data with abnormal behavior characteristics from monitoring sequence and observe these characteristics to reflect the terminal's abnormal behavior plays a vital role in determining abnormal detection accuracy. A total of 4 statistical observation features are selected to determine the terminal's behavior, which not only fully shows the change of terminal behavior, but also effectively avoids the complexity of calculation. The four-movement characteristics are as follows:

(1) Familiar characteristics $\mathrm{F}Ch_{TM_i}$: the higher the number of historical communications between the terminal and the communication front-end, the greater the familiarity between them. Familiarity will affect the terminal's trust under evaluation, and the familiarity between them depends mainly on the number of communications after the terminal is connected. Therefore, the familiar characteristics of the terminal can be expressed by Equation (3):

$$\begin{aligned} \mathrm{F.C.h_{TMi}} &= \frac{\mathrm{sum\{FL_{TMi}\}}}{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{sum\{FL_{TMi}\}}} \\ &= \begin{bmatrix} h_{1,1}^{(1,2)} & h_{1,2}^{(1,2)} & 0 & 0 & 0 \\ 0 & h_{2,2}^{(1,2)} & h_{2,3}^{(1,2)} & 0 & 0 \\ 0 & 0 & h_{3,3}^{(1,2)} & h_{3,4}^{(1,2)} & 0 \\ 0 & 0 & 0 & h_{3,4}^{(1,2)} & h_{4,5}^{(1,2)} \\ 0 & 0 & 0 & 0 & h_{5,5}^{(1,2)} \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \end{aligned} \tag{3}$$

(2) Similar characteristics of business behavior BCh_{TM_i}: In order to calculate the similarity trust degree of business behaviors, the same number of message types transmitted by the terminal TM_i Moreover, the same type of terminal can be used to calculate. Therefore, if the terminal TM_i The terminal of the same type transmits more of the same message type, that is, participating in the same business activity; it will have a higher degree of trust. Equation (4) gives the calculation of the similarity trust of business behaviors. Use $PT_{TM_i}^{same}$ to indicate the number of the same message type transmitted by the terminal TM_i and the same type of terminal, and $PT_{TM_i}^{all}$ to indicate the total number of message types transmitted by the terminal. If most of the message types transmitted by the terminal are the same as those transmitted by other terminals of the same type, the credibility of this terminal is higher. Conversely, although it cannot be directly determined that the terminal is untrustworthy, it can be suspected that the terminal has been attacked and abnormal behavior has occurred.

$$\begin{aligned} &B\mathrm{Ch_{TMi}} \\ &= \frac{\sum (\mathrm{PT_{TMi}^{same}} - \mathrm{PT_{TMi}^{all}})(\mathrm{PT_{Type}^{same}} - \mathrm{PT_{Type}^{all}})}{\sqrt{\sum (\mathrm{PT_{TMi}^{same}} - \mathrm{PT_{TMi}^{all}})^2} \sqrt{\sum (\mathrm{PT_{Type}^{same}} - \mathrm{PT_{Type}^{all}})^2}} \end{aligned} \tag{4}$$

(3) Access behavior characteristics ACh_{TM_i}: The smart grid edge computing terminal's network access behavior has a certain regularity, so the network access behavior of the

terminal can also reflect its abnormal state. The calculation method is given in Equation (5):

$$ACh_{TMi} = \begin{cases} 1 - \dfrac{\varepsilon(DA^{t}_{TM_i}, DA^{old}_{TM_i})}{L}, & \varepsilon(DA^{t}_{TM_i}, DA^{old}_{TM_i}) \geq L \\ 0, & others \end{cases} \tag{5}$$

$\varepsilon(DA^{t}_{TM_i}, DA^{old}_{TM_i})$ represents the difference between the terminal's destination address access behavior in the current time and its historical access address, which can be solved by editing distance. When the difference of its access behavior is more significant than a value L, the trust degree is zero. This formula shows that the more regular the terminal's access behavior, the higher the trust degree. If the access behavior is too different, it means that the terminal node has minimal data contact with the destination address node it visits or that the terminal node has abnormal behavior.

(4) Data load behavior characteristics DCh_{TM_i}: After the terminal is connected, the real-time data load characteristics in the data interaction process also reflect the terminal's trust to a certain extent. Under normal conditions, the data load of the terminal $DATA_{TM}$ should present a regular Gaussian distribution. When the terminal is attacked or used to carry out an attack, the terminal's data load will be mainly attacked, deviating from the average data load. Therefore, the characteristics related to the terminal data load behavior can be calculated by Equation (6):

$$\mathrm{D.C.h_{TMi}} = \begin{cases} 1/\log_{1+q}\left(\dfrac{\mathrm{DATA_{TM_i}} - \delta\left(\mathrm{DATA_{TM_{type}}}\right)}{\frac{\delta\left(\mathrm{DATA_{TM_{type}}}\right)}{10}} + 1 + \mathrm{q}\right), \\ \qquad \mathrm{DATA_{TM_i}} \geq \delta(DAT\,\mathrm{A_{TM_{type}}}) \\ 1, \qquad others \end{cases} \tag{6}$$

Use the above four different behavior characteristics to construct an observable state set of terminal behavior:

$$\mathrm{O_{TM_i}} = \left\{\mathrm{F.C.h_{TM_i}}, \mathrm{B.C.h_{TM_i}}, \mathrm{A.C.h_{TM_i}}, \mathrm{D.C.h_{TM_i}}\right\}$$

On this basis, a terminal behavior observation feature matrix within the observation period can be constructed. Each column of the matrix T represents the terminal's TM_i familiarity characteristics, business behavior, similar characteristics, access behavior, similar characteristics, data load behavior characteristics, and constructed observation characteristics. The matrix is as follows:

$$\mathrm{O^{T}_{TM_i}} = \begin{bmatrix} \mathrm{F.C.h^{1}_{TM_i}} & \mathrm{F.C.h^{2}_{TM_i}} & \cdots & \mathrm{F.C.h^{t}_{TM_i}} \\ \mathrm{B.C.h^{1}_{TM_i}} & \mathrm{B.C.h^{2}_{TM_i}} & \cdots & \mathrm{B.C.h^{t}_{TM_i}} \\ \mathrm{A.C.h^{1}_{TM_i}} & \mathrm{A.C.h^{2}_{TM_i}} & \cdots & \mathrm{A.C.h^{t}_{TM_i}} \\ \mathrm{D.C.h^{1}_{TM_i}} & \mathrm{D.C.h^{2}_{TM_i}} & \cdots & \mathrm{D.C.h^{t}_{TM_i}} \end{bmatrix}$$

By observing and analyzing the above characteristics, we can detect and analyze the abnormal behavior of the smart grid edge computing terminal caused by the type of attack to evaluate the terminal risk. So far, we have determined the input and output parameter set of the HMM model (see Figure 3).

D. CALCULATION OF STATE TRANSITION PROBABILITY OF ABNORMAL TERMINAL BEHAVIOR

The traditional Markov risk assessment model assumes that the state transition probability matrix of the system does not change with time. However, in the smart grid edge network environment, the state transition probability is continually changing, especially in network attacks. Therefore, this article updates the state transition probability matrix in real-time according to the time transition probability of the attack state switching on the network from the perspective of time. First, we determine the difficulty of each stage of the attack by calculating the ratio of the time spent in each attack stage to the time it takes to complete the entire attack to objectively calculate the difficulty of state transition between each stage of the same attack. Second, the attack state's transition probability can be calculated according to the difficulty of transition between different attacks. The smaller the difficulty, the greater the transition probability and vice versa.

We define T as the time cost of the whole process, and t_i represents the attack time cost of state i of the whole attack process. We define the attack process as $A = \{A_i\}$, one $\leq i \leq M$. A_i represents the state i of the attack, and M representing the division of the attack phase. With the above definition, we can define the difficulty of an attack as $D = \{D_i\}$, one $\leq i \leq M$.D_i represents the state i of attack difficulty in Equation 7.

$$D_i = \frac{t_i}{\sum_{i=1}^{M} t_i}, \quad t_i \in T \tag{7}$$

Then give the general formula of the state transition probability matrix P,

$$p_{ij} = \frac{1/d_{ij}}{\sum_{i=1}^{M} 1/D_i} \tag{8}$$

$D_i \in D$, d_{ij} represents the difficulty form from state i to state j. Here:

$$d_{ij} = \begin{cases} D_j, & i < j \\ D_i, & i = j \\ D_i, & i > j \end{cases}$$

Equation 8 expresses the transition probability of a node from state i to state j to be attacked.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. TEST ENVIRONMENT CONSTRUCTION

To verify the effectiveness and performance of the algorithm proposed in this article, related experiments have been done in a laboratory environment. The topology of the experimental environment is shown below (see Figure 4):

The configuration of the relevant experimental environment is shown (see Table 1).

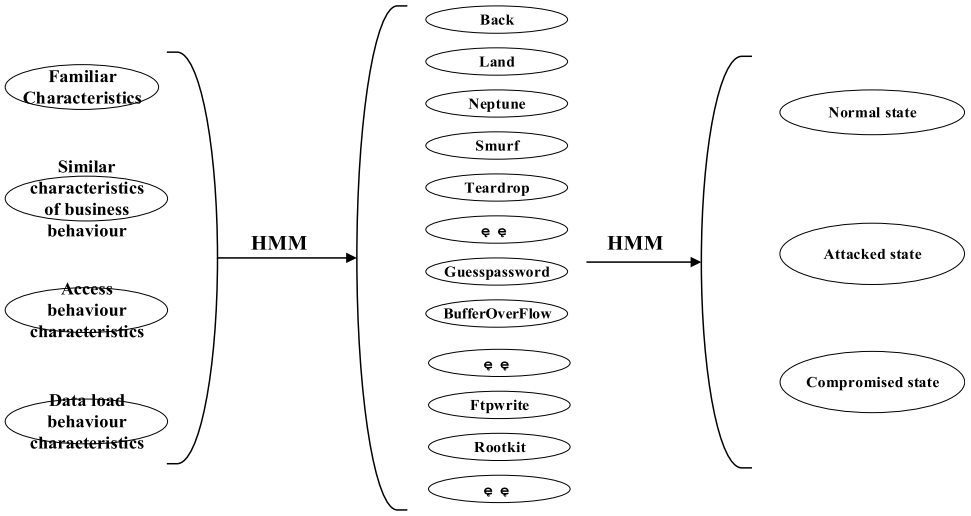


FIGURE 3. Input and output parameter set of double HMM model.

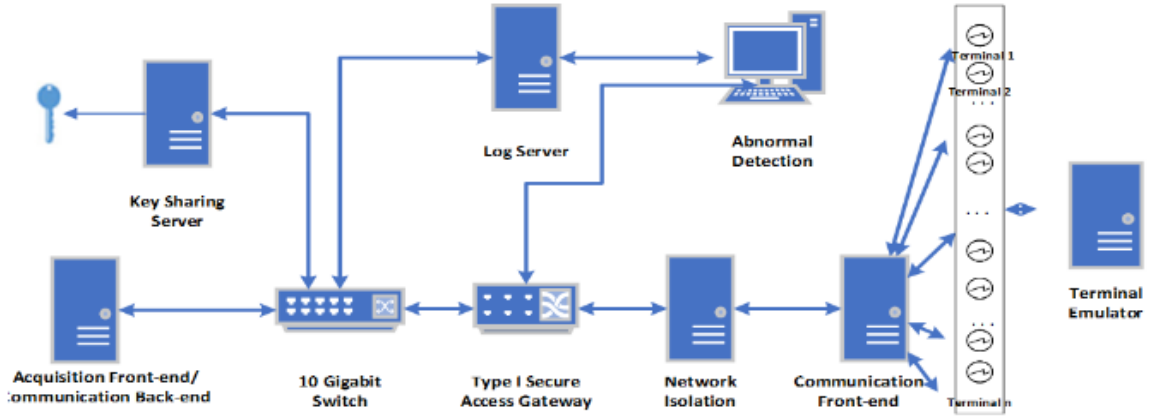


FIGURE 4. The topology of the experiment environment.

B. ANALYSIS OF EXPERIMENTAL RESULTS OF ABNORMAL BEHAVIOR DETECTION AND EVALUATION

First, acquiring the data set from the NS2 simulation database, and dividing it into three different attack stages, according to the details and specific functions of each data set. The attack stages include the data collection stage, continued attack stage, and the node occupation stage. Second, calculate the average attack time based on each attack's statistical analysis (see Table 2–4).

From Table 2–4, calculate the average attack time of each stage, then using Equation 8 to obtain the probability transition matrix:

$$\mathrm{P} = \begin{Bmatrix} P_{GG} & P_{GA} & P_{GC} \\ P_{AG} & P_{AA} & P_{AC} \\ P_{CG} & P_{CA} & P_{CC} \end{Bmatrix} = \begin{Bmatrix} 0.994 & 0.00596 & 0.00004 \\ 0.0898 & 0.910 & 0.0002 \\ 0.007 & 0.003 & 0.990 \end{Bmatrix}$$

Then, we give the initial probability of the node at each stage. Assuming the initial probability of nodes in different stages: $\pi = \{\pi_1, \pi_2, \pi_3\} = \{1, 0, 0\}$, where $\sum \pi_i = 1$.

The experience of the observation matrix can be based on the expert's experience and set as:

$$\mathrm{Q} = \begin{Bmatrix} q_{G(G)} & q_{G(A)} & q_{G(C)} \\ q_{A(G)} & q_{A(A)} & q_{A(C)} \\ q_{C(G)} & q_{C(A)} & q_{C(C)} \end{Bmatrix} = \begin{Bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{Bmatrix}$$

Next, use the γ_t^k Viterbi algorithm to calculate the abnormal value. First, understand the general process of the Viterbi algorithm, as follows:

Step 1: Initialization. $\gamma_t^{(i)} = \pi_k b_t(O_k), 1 \le i \le N, 1 \le k \le$ sum(Number of nodes).

Step 2: Recursion or loop.

$\gamma_t(j) = max_{1 \le j \le N} \left\lfloor \gamma_{t-1}(i) p_{ij} \right\rfloor b_t(O_t), 2 \le t \le T, 1 \le j \le N.$

Step 3: Result. $P^* = max_{1 \le j \le N} \lfloor \gamma_t(i) \rfloor.$

TABLE 1. Specific configuration parameters of the experimental environment.

Type	Equipment Name	Quantity	Hardware Configuration	Equipment Usage
Server	Key sharing server	1	2.6GHz CPU2*8 CPU	Running the key sharing service system
	Master station server	1		Simulating master station system
	Terminal emulator server	1		Simulating edge computing terminal
	Access gateway	1		Running access gateway system
	Network isolation	1		Running network isolation system
	Abnormal detection server	1		Running abnormal detection system
	Log server	1		Log collection and traffic analysis
	Acquisition & communication	1		Acquisition front end and communication front end
Testing Machine	Test machine	1	Intel®I5 16G，1T	Running test tools and software for functional and performance testing
Testing Software	Random number detection	1	/	Random number detection software
	Function test	1	/	Gateway function test software
	Performance test	1	/	Gateway performance test software

Calculate $\gamma_t k = [0.8, 0.1, 0.1]$. and assume $\xi^k = 0, 10, 30$, $\varphi_t^k = \{\varphi_1, \varphi_2, \varphi_3\} = \{0.28, 0.33, 0.39\}$. We can obtain $\gamma_t^k \varphi_t^k = \{0.54, 0.215, 0.245\}$. Finally, give the known abnormal value of the node: (may wish to set the value of the first node) R_t^1

$$R_t^1 = \phi_t^1 * \xi^1 = \begin{bmatrix} 0.54 & 0.215 & 0.245 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \\ 30 \end{bmatrix}$$

In the same way, we can obtain the abnormal value of the remaining nodes. $R_t^2 = 6, R_t^3 = 6.5, R_t^4 = 5.5, R_t^5 = 5 R_t^6 = 7, R_t^7 = 7.5$.

TABLE 2. Attack time statistics during the data collection stage.

Attack Name	Attack Time/s	Maximum Duration of Attack /s	Minimum Duration of Attack /s	Average Duration of Attack /s
Back	2203	14	0	0.129
Land	21	0	0	0
Neptune	107201	0	0	0
Pod	264	0	0	0
Smurf	280790	0	0	0
Teatdrop	979	0	0	0
Lesweep	1247	7	0	0.0345
Nmap	231	0	0	0
Satan	1589	11	0	0.043

TABLE 3. Attack time statistics during the continued attack stage.

Attack Name	Attack Time/s	Maximum Duration of Attack /s	Minimum Duration of Attack /s	Average Duration of Attack /s
Guesspassword	53	60	0	2.717
Pdf	4	12	0	4.5
BufferOverFlow	30	321	0	91.7
Imap	12	41	0	6
Loadmodule	9	103	0	36.2222
Multihop	7	718	0	184
Perl	3	54	25	41.3333
Postsweep	1040	42488	0	1915.2990

TABLE 4. Attack time statistics during the node occupation stage.

Attack Name	Attack Time/s	Maximum Duration of Attack /s	Minimum Duration of Attack /s	Average Duration of Attack /s
Ftpwrite	8	134	0	32.375
Rootkit	10	708	0	100.8
Spy	2	337	299	318
Warezmaster	20	156	0	15.05
WarezClient	1020	15168	0	6152.178

According to the abnormal value of the seven nodes in the simulation, the entire network's abnormal value can be calculated. First, determine the time weight in the Equation. The attacks of nodes are different at different times, and the degree of exposure is also different. Take one day as an example. The time of the day is divided into three time periods: $T_1 : 0 : 00 \sim 8 : 00$, $T_2 : 8 : 00 \sim 16 : 00$, $T_3 : 16 : 00 \sim 24 : 00$. The attacks in the second period are the most active. It also has the greatest impact on the entire network. Followed by the third period, and finally, the first period. The quantitative value of the importance of these periods based on professional knowledge. The relative importance weights of the three-time periods can be obtained after normalization: $w_{T_1} = 0.11, w_{T_2} = 0.67, w_{T_3} = 0.22$

From the above, we can get the abnormal value of the entire network $R_t' = 1.49$.

As the time slice is divided into 3 hours, the attacks in Table 4 were applied to network nodes between 9 o'clock and 12 o'clock, and 21 o'clock to 24 o'clock, and the

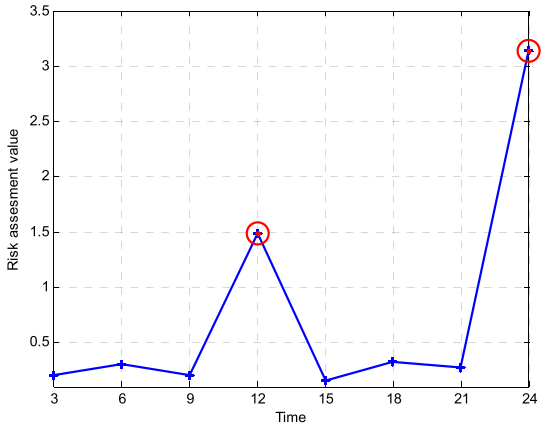


FIGURE 5. Validation of this article method.

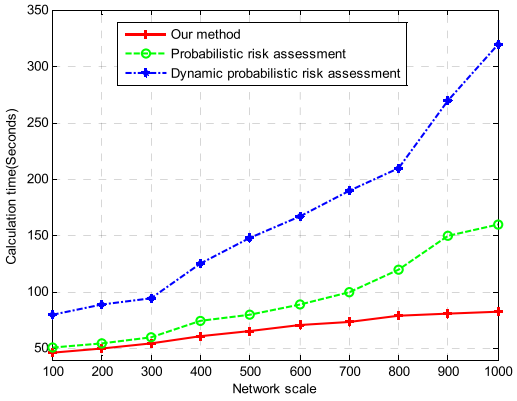


FIGURE 6. Comparison of calculation time required by different methods.

abnormal value increased significantly from the normal state (no attack), as shown by the red circle in the figure (see Figure 5). Among them, a total of 3 attacks in Table 4 were applied between 21:00 and 24:00, and the abnormal value was significantly increased from 12:00 (one attack), which was about twice the value at 12:00. The experimental simulation results show that this article method can effectively identify and evaluate the terminal's abnormal behavior in the network.

As shown in Figure 6, compared with the Probabilistic Risk Assessment and Dynamic probabilistic risk assessment methods, the anomaly recognition and evaluation calculation method proposed in this article requires the shortest calculation time under different network scales. Moreover, with the increase of the network scale, the time required for this article's method shows a slow increase trend, so the method proposed in this article can be better suited for large-scale network environments.

V. CONCLUSION

This article proposes a method for identifying abnormal behaviors of wireless access power terminals based on double HMM, which solves the computational complexity problem caused by high dimensions in intrusion detection systems. The lower-layer is used to identify discrete individual network abnormal behaviors. The upper-layer obtains a longer span of attack behavior from multiple independent abnormal events identified by the lower-layer. The experiment shows that our method can effectively detect the terminal's abnormal behavior and identify the network attack behavior through a long-time span.

REFERENCES

[1] T. Seals. (2015). *75% of Companies are Insider Threat Victims*. [Online]. Available: https://www.infosecurity-magazine.com/ news/75-of-companies-are-insider-threat/

[2] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 1, pp. 9–42, Jun. 2001, doi: 10.1142/ S0218001401000836.

[3] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh, U.K.: Edinburgh Univ. Press, 1990.

[4] Y. Wen, "Text mining using HMM and PMM," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 2001.

[5] H. Bunke and T. Caelli, *Hidden Markov Models: Applications in Computer Vision*. Singapore: World Scientific, 2001.

[6] R. J. Boys, D. A. Henderson, and D. J. Wilkinson, "Detecting homogeneous segments in DNA sequences by using hidden Markov models," *J. Roy. Stat. Soc., C (Appl. Statist.)*, vol. 49, no. 2, pp. 269–285, Jan. 2000, doi: 10.1111/1467-9876.00191.

[7] Z. Anming and J. Chunfu, "Study on the applications of hidden Markov models to computer intrusion detection," in *Proc. 5th World Congr. Intell. Control Autom.*, Hangzhou, China, Jun. 2004, pp. 4352–4356, doi: 10.1109/WCICA.2004.1342335.

[8] S. Shin, S. Lee, H. Kim, and S. Kim, "Advanced probabilistic approach for network intrusion forecasting and detection," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 315–322, 2013, doi: 10.1016/j.eswa.2012.07.057.

[9] M. H. Kabir, M. R. Hoque, K. Thapa, and S.-H. Yang, "Two-layer hidden Markov model for human activity recognition in home environments," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Jan. 2016, Art. no. 4560365, doi: 10.1155/2016/4560365.

[10] E. Parzen, *Stochastic Processes*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999.

[11] S. Karlin and H. E. Taylor, *A First Course in Stochastic Processes*. St. Louis, MO, USA: Elsevier, 2014.

[12] J. Lamperti, "Markov transition function," in *Stochastic Processes: A Survey of the Mathematical Theory*. New York, NY, USA: Springer, 2012, pp. 106–131.

[13] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, 1966, Art. no. 15541563.

[14] CERT Insider Threat Center. (2011). *The CERT Insider Threat Database*. [Online]. Available: https://insights.sei.cmu.edu/insider-threat/2011/08/thecert-insider-threat-database.html

[15] H. Eldardiry, E. Bart, J. Liu, J. Hanley, B. Price, and O. Brdiczka, "Multi-domain information fusion for insider threat detection," in *Proc. IEEE Secur. Privacy Workshops*, San Francisco, CA, USA, May 2013, pp. 45–51, doi: 10.1109/SPW.2013.14.

[16] A. Gamachchi, L. Sun, and S. Boztas, "Graph-based framework for malicious insider threat detection," in *Proc. 50th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Hilton Waikoloa Village, HI, USA, 2017.

[17] G. D. Forney. (2005). *The Viterbi Algorithm: A Personal History*. [Online]. Available: https://arxiv.org/abs/cs/0504020

[18] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Harlow, U.K.: Pearson, 2014.

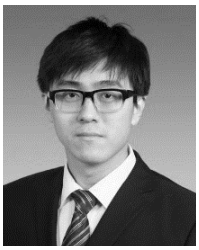
[19] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD)*, Portland, OR, USA, 1996, pp. 226–231.

[20] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *Proc. IEEE Secur. Privacy Workshops*, San Francisco, CA, USA, May 2013, pp. 98–104, doi: 10.1109/SPW.2013.37.

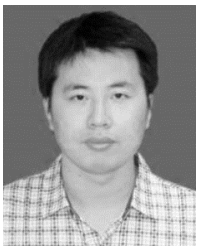
[21] J. A. Bilmes, *Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Berkeley, CA, USA: International Computer Science Institute, 1997.

[22] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 621–636, Mar. 2018.
[23] B. Atli, "Anomaly-based intrusion detection by modeling probability distributions of flow characteristics," Ph.D. dissertation, Aalto Univ., Helsinki, Finland, 2017. Accessed: Jul. 14, 2018. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201710307348
[24] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," 2017, *arXiv:1701.02145*. [Online]. Available: http://arxiv.org/abs/1701.02145

KEHE WU received the Ph.D. degree from North China Electric Power University, Beijing, in 2009. He is currently a Professor with North China Electric Power University, the Director of the Chinese Association for Artificial Intelligence and the Beijing Engineering Research Center of Electric Information Technology, and a Committee Member of the China Electric Power Information Standardization Committee and the Professional Electric Power Information Committee of the Chinese Society for Electrical Engineering. His research interests include computer vision, pattern recognition, and deep learning.



JIAWEI LI received the master's degree from North China Electric Power University, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Control and Computer Engineering. His research interests include wireless security and cybersecurity.



BO ZHANG received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Chia, in 2018, and the M.Sc. degree from Southeast University in 2012. He is currently with the Global Energy Interconnection Research Institute, China. His research interests include cybersecurity in smart grid and network security situation awareness.