Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Extraction and mapping of multidimensional data from computer terminals based on transfer probability matrix Markov chain attack graphs

Jun Han[1,✉], Ke Liu[1], Yutong Liu[1], Wenqian Zhang[1], Shaofei Wang[1]

[1] *State Grid Qinghai Electric Power Company Electric Power Science Research Institute, Xining, Qinghai, 810008, China*

ABSTRACT

The existence of a large number of multi-source heterogeneous hosts and application service types in various zones of the power monitoring system leads to difficulties in extracting comprehensive host attack trace data and the problem of fine-grained deep threat detection. This study combines network attack traces extracted from multi-source logs and stores them in attack trace styles. An attack event description model based on key attributes and behavior sequences is constructed. Based on the vulnerability scoring system, an algorithm is designed to map a general attack graph into an absorbing Markov chain attack graph, which provides a computational basis for the analysis of network attacks by calculating the state transfer probability matrix of the attack graph. Finally, the performance of this paper's method for multi-dimensional data feature extraction is explored in a python experimental simulation environment. The simulation results show that the average mapping time of LSTM model for 7 vulnerabilities is 117ms, while the average mapping time of this paper's algorithm is improved by 37ms compared to the LSTM model.Meanwhile, the accuracy, stability, average false detection rate and positive and negative recall rate also achieve good results, which verifies the validity of this method in the practice of power monitoring system management.

Keywords: power monitoring system, transfer probability matrix, attack graph, Markov chain, multidimensional data

1. Introduction

With the continuous development of network information technology such as big data, cloud computing, artificial intelligence, etc., the Internet has become the infrastructure of China's informationization construction [1-2]. At the same time, the network security of the state, enterprises and individuals is facing serious challenges. Attackers carry out network attacks through software

✉ Corresponding author.
E-mail address: gjdwgjdw@163.com (J. Han).
Received 05 February 2024; Revised 12 April 2024; Accepted 05 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-502

application loopholes and insecure network protocols, invade the target hosts and obtain high-level privileges, or implant viruses for profit, which bring immeasurable losses to the state and enterprises. Therefore, it is of great significance to discover potential security risks in the network and take effective protection measures in time to improve the overall security of the network [3-6].

Starting from the long-term background of the development of the network market, for the current computer security vulnerability monitoring technology, security management measures, different categories of network security vulnerabilities and their causes and characteristics, the development of the protection and management program of computer system security vulnerabilities. The computer security assessment is realized with the structure of attack graph as the core, which can better solve the network security problems and realize the optimization of security management and control [7-10].

When analyzing the security of the network, a more comprehensive assessment can often be obtained from the perspective of the attacker, who penetrates the network from a boundary point in the network, uses the vulnerabilities in the network and hosts to gradually transfer, and ultimately obtains the authority of the target state node [11-13]. By simulating the attacker's attack process, analyzing potential attack paths, identifying weaknesses and potential threats in the network and repairing them in a timely manner, it can effectively protect the cyberspace security of the state, enterprises and individuals [14-16].

Attack graph technology, as a means of network security analysis and assessment, can effectively meet the above needs. Attack diagram graphically demonstrates the attack path composed of successive attack behaviors of the attacker, and constructing the attack diagram of the target network topology can not only analyze the attack path of the attacker and protect the key links on the path, but also analyze the attacker's attack intention and movement in real time, and take timely defense and countermeasures [17-20]. As the attack graph technology has a good application prospect in the field of network security assessment, it has become the key research content in academia and industry [21-22].

The research firstly constructs the attack event description model, which is extracted in the multi-source logs of the power monitoring system using python to write the Markov chain model program. A new data structure is established to store the temporary data during the analysis process, and the extracted network attack traces are combined and stored in attack trace style. Based on Markov chain and attack graph mapping, a mapping algorithm for absorbing Markov chain attack graphs is designed, which provides a computational basis for the identification and analysis of network attacks by calculating the state transfer probability matrix of the attack graph. It is also embedded into the vulnerability scoring system, and the performance of the proposed method is tested under python simulation environment.

2. Research on multi-dimensional data extraction and mapping techniques

2.1. Computer terminal multidimensional data extraction

In order to solve the problem that there are a large number of multi-source heterogeneous hosts and application service types in each area of the power monitoring system, it is difficult to extract comprehensive host attack trace data as well as fine-grained in-depth threat detection, and it is necessary to construct a built attack event description model based on the key attributes and behavioral sequences. The attack event description model constructed in the study is used to combine the network attack traces extracted from multi-source logs and store them in an attack trace style.

2.1.1. Definition of key attributes. The key attributes in the model are composed of four attributes: time, source address, destination address, and attack type, and a new attribute “trace weight” is added, which is jointly determined by the URI and POSTData whose trace weight is also higher. In the model,

each attack trace is composed of five key attributes: timeTime, source address sAddress, destination address dAddress, attack type name, and trace weight. With reference to the general characteristics of multi-source logs, almost in every log message itself contains information related to these key attributes, and the trace weight is jointly determined by the correlation information between different kinds of logs.

As the key basis for the preliminary correlation of the attack traces, time information is also essential in the traceability analysis technique, so it is chosen as one of the key attributes of each attack trace.The default logging style of Apache/Nginx and other commonly used web server software will record the time generated by each web request, and take the time generated by the first web request of each web log file as a reference. The time of the first Web request in each Web log file is used as a reference, and the time of the Web request corresponding to the extracted attack trace is subtracted from the time of the first Web request in this log file to get a relative time as the Time attribute of the attack trace in seconds. For those attack types that require multiple logs to be analyzed together to determine, the time corresponding to the first log message is used as the basis for generating the Time attribute.The address composed of IP address and port information can be used to determine the starting point and end point of the Web request, which is also an indispensable key attribute, and it is more convenient to define and analyze the subsequent correlation of the attack event by dividing the address information into the starting point and the end point. It is more convenient to define the attack event and analyze the subsequent correlation. The same Web log file comes from one server, so routinely all Web requests in the same file are sent to the same destination address, i.e., the attribute dAddress is the same. For the sAddress attribute, it can also be obtained directly in each Web log. Also each entry in the security appliance log contains the source and destination addresses, which can be extracted and used directly. The two key attributes, attack type and trace weight, belong to the content obtained after preliminary data processing and are also the core content of correlation analysis. For the attribute of attack type, on the one hand, it can be obtained directly from the security device log records, on the other hand, it can be extracted from the Web server logs and database execution logs, and the attack type and trace weight will be analyzed and obtained from the URIs, POST data, HTTP4 descriptions, methods, HTTP status codes and other information together. The specific extraction process will be described in detail below.

2.1.2. Attack event behavior sequence. After the extraction of all the attack traces in the Web logs is completed, each attack trace is stored using json format data for subsequent analysis, the following definition is a sample of an extracted attack trace:

{“sAddress”:“ip”,“dAddress”:“ip”,“time”:“0”,“name”:“TypeofAttack”,“weight”:“0”)

All the attack traces are categorized according to two attributes, sAddress and dAddress, and traces with the same attributes are considered as the same attack event, which is called a behavioral sequence due to its association.

Attack traces within the same behavior sequence are sorted by time, and the time attribute is denoted by “time” in the sample. In the same attack event, the correlation between different attack traces is the extracted content, which is temporarily represented as “correlation” in the model, and the correlation analysis is not only to analyze the neighboring attack traces in the same behavior sequence, but also to continue to analyze the relationship between one or two attack traces in the interval. The purpose of setting the behavior sequence is to facilitate the serialization of the attack traces and the extraction and analysis between the attack traces. With the setting of behavioral sequences, multiple attack traces belonging to the same attack event can be grouped into a single path, and the serialized data can be used directly when the correlation analysis algorithm scans the dataset, saving most of the algorithm execution time.

2.1.3. Attack trace extraction. Once the features of various cyber attacks have been collected, the process of extraction of attack traces will be carried out. Due to the extraction of attack traces from

logs and other data contains numerous types of attacks, the attack features will be transformed into regular, using python to write Markov chain model program for extraction. Taking the extraction of SQL injection attack traces as an example, while the process of extracting the attack traces, the efficiency of the program operation is also taken into account, and the entire dataset will be scanned as few times as possible when writing the Markov chain model program. When extracting SQL injection attacks, for its keyword features and character features will be scanned in a trip to use the flag bit to directly determine, is to save the program running time with the help of storage space, the establishment of a new data structure for the analysis process of temporary data storage, and at the same time the need to use the file to cache it, to prevent the program from crashing after the loss of temporary data. The last step is to extract the information from the security device logs, which have been labeled with the attack types, and each log also contains the original packet request and respond information, which can be used to assign values to the trace weight attributes in the model. The security device logs contain a small number of cyber attacks with small amount of data and complex types, which are categorized in the same way as other attack types. The data after the completion of the extraction of the assigned attack traces is temporarily stored in a file and used as input data for the next step of correlation analysis. After completing the identification of targets for multi-source log analysis, performing attack event modeling, and extracting Web attack traces from the logs, a certain size of data is obtained, which will be stored according to the attack trace style proposed above.

2.2. Markov chain based attack graph modeling

2.2.1. Attack map definition. Attack graph is a graph-theoretic approach to determine the security of a network by studying the sub-nodes in the network and the interrelationships among the nodes [23-24]. By building actual networks into theoretical graph-theoretic models, attack graphs can give us much to think about in depth, sometimes in unexpected ways. Currently there is not much research in attack graphs, but attack graphs provide us with the idea to construct a model from known aspects, simplify or idealize the actual elements, which allows us to focus on the most important or more important aspects of network security, ignoring the secondary factors, so that we can quickly determine the security of the network under study, avoiding wasting valuable time by some irrelevant aspects and weakening the Security Analysis. In the past, attack graphs were mainly used for the analysis of system security and the design of intrusion detection systems, but attack graphs, as a graph theoretic method in theoretical research, can be applied to many aspects. Any network or system as long as it contains multiple nodes can apply attack graphs to model and conduct research on the overall aspects to achieve their respective purposes. In fact, an attack graph is literally a series of graphs cobbled together from a series of attacks that continually increase in attack capability. It is actually a more mature attack model. An attack graph is generally a directed graph, where each vertex represents an attack state and arcs represent transition relationships between states. An attack graph can be used to focus on whether there is a weak node from which a key point can be reached, i.e., the final target of the attack. It can also be used to focus on whether there is a possibility that a hacker can gradually penetrate the network from the basic functionality provided by the system.

2.2.2. Markov chains. In a stochastic process (or system), in the t_0 moment the state of the process is known, the process in the moment $t > t_0$ of the state of the conditional distribution of the state of the process is not related to the process in the moment t_0 before the state of the nature of the process is known as Markov or no posteriority. Generally speaking, it means that under the condition that the "present" of the process is already known, its "future" does not depend on its "past".

The Markov property is expressed using the distribution function as follows: let the state space of the stochastic process $\{X(t), t \in T\}$ be I, if for any n values $t_1 < t_2 < \cdots < t_n$, $n \geq 3$, and $t_i \in T$ of time t, the distribution function of $X(t_n)$ under conditions $X(t_i) = x_i$, $x_i \in I$, and $i = 1, 2, \cdots, n-1$

is exactly equal to the conditional distribution function of $X(t_n)$ under condition $X(t_n-1)=x_n-1$, that is:

$$\begin{aligned} &P\{X(t_n)\le x_n \mid X(t_1)=x_1, X(t_2)=x_2,\cdots, X(t_n-1)=x_n-1\} \\ &=P\{X(t_n)\le x_n \mid X(t_n-1)=x_n-1\} \end{aligned} \tag{1}$$

Then the process $\{X(t), t\in T\}$ is said to be Markovian, or the process is called a Markov process [25-26]. A Markov process that is discrete in time and state is called a Markov chain, or simply a Markov chain, denoted $\{X_n=X(n), n=0,1,2,\cdots\}$, and it can be viewed as the result of successive observations of the process in discrete states on the time set $T1=\{0,1,2,\cdots\}$. The state space $I=\{a1,a2,\cdots\}$ In the case of chains, Markovianity is usually expressed in terms of conditional distribution rates, i.e., for any positive integers n, r and $0\le t_1<t_2<\cdots\cdots<t_r<m$, t_i, m, $m+n\in T1$, there are $P\{X_m+n=a_j \mid Xt_1=ai_1, Xt_2=ai_2,\ldots, Xt_r=ai_r, Xm=ai\}=P\{Xm+n=a_j \mid Xm=ai\}$, $a_i\in$ called conditional probabilities $P_{ij}(m,m+n)=P\{Xm+n=a_j \mid Xm=a_i\}$ as the transfer probabilities of the Markov chain being in a state a_i at a moment m, conditional on a transfer to a state a_j at a moment $m+n$.

Since the chain starts from any state a_i at moment m and at another moment $m+n$, it must necessarily transfer to one of the $a_1,a_2,\ldots$ states, so $\sum_{j=1}^{\infty}P_{ij}(m,m+n)=1$ (where $i=1,2,\ldots$).

When the transfer probability $P_{ij}(m,m+n)$ is only related to i, j and the time spacing, i.e., $P_{ij}(m,m+n)=P_{ij}(n)$, the transfer probability is said to be smooth and the chain is also said to be chi-square. In a chi-squared martensian chain, the transfer probability $P_{ij}(n)=P\{Xm+n=a_j \mid Xm=a_i\}$ is called a n-step transfer probability, and when $n=1$, the transfer probability becomes a one-step transfer probability.

The matrix consisting of all n-step transfer probabilities is called the n-step transfer probability matrix, and the (i,j) elements of the n-step transfer probability matrix are the n-step transfer probabilities $P_{ij}(n)$ of the system transferring from the in-state i to the state j, and the sum of the elements of each row of this matrix is 1. When $n=1$, the corresponding transfer state probability matrix is called the one-step transfer probability matrix (or simply the transfer probability matrix).

A chi-squared Markov chain (MC) can be defined as a binary $MC=(I,P)$ where I is the system state space and P is the transfer probability matrix.

2.2.3. Mathematical modeling. The standard Markov chain does not specify what the transfer of states is due to, and in order to analyze the network attack graph first the standard Markov chain is extended. An extended Markov chain can be defined as $EMC=(I,P,A)$, where I is the state space, A is the set of all available methods, and P is the transfer probability matrix labeled with methods.

An extended Markov chain $EMC=(I,P,A)$ can be represented using a directed graph, where the nodes of the graph correspond to states in EMC the state space I, the directed edges labeled with weights and methods correspond to the corresponding state transfers, and the weights on the edges are the corresponding transfer probabilities in P. For example, an extended Markov chain EMC_Example has a state space $I=\{1,2,3\}$, a collection of methods $A=\{a_1,a_2,a_3,a_4,a_5,a_6,a_7\}$, and a

transfer probability matrix $p = \begin{bmatrix} \{0.2_{a1}\} & \{0.8_{a2}\} & \{0\} \\ \{0.4_{a3}\} & \{0\} & \{0.6_{a4}\} \\ \{0.5_{a5}, 0.35_{a6}\} & \{0.15_{a7}\} & \{0\} \end{bmatrix}$, and Figure 1 shows the directed graph corresponding to EMC_Example. Each edge is labeled with the transfer probability and the corresponding method, and the sum of the weights on all edges from each node is 1. If this property of directed graphs is referred to as the extended Markov chain property, then any directed graph with the extended Markov chain property can be corresponded to an extended Markov chain.

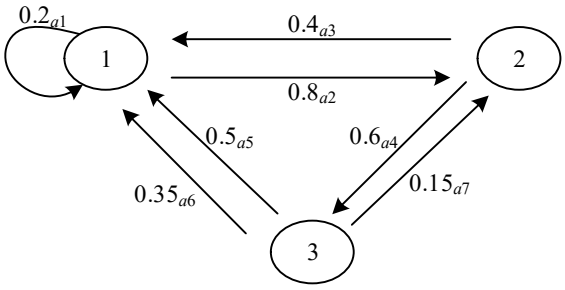

Fig. 1. Direction graph of EMC_Example

The network attack graph is composed of the attack state in which the network attacker is in and the attacker's attack actions, which enable the attacker to leap from one state to another, and the possible attack actions that the attacker may take in a certain state are only related to the attacker's current state. When the attacker is in a certain attack state in the network attack graph and there exist multiple attack actions that can cause the attack state to change, the attacker can choose one of these candidate attack actions to carry out the next attack. Due to the different degree of detail of the attack methods, attack steps, and other information contained in different published weaknesses, the existing attack tools developed by utilizing different weaknesses are available in different states, resulting in different attack complexity for different attack actions. The grading of attack complexity is shown in Table 1. From Table 1, it can be seen that the higher the value of attack complexity, the higher the possibility of success of the attack.

Table 1. Breakdown of attack complexity

Designation	Empower	Level
E6	0.9	Off-the-shelf attack tools with detailed attack steps
E5	0.8	Customize available attack tools with detailed attack steps
E4	0.6	There are no ready-made attack tools but there are more detailed attack steps
E3	0.5	Publicly report and roughly describe attack methods
E2	0.2	Publicly report and mention possible attack methods
E1	0.1	Public report but no attack method
E0	0.05	Publicly report but attack weaknesses that are only theoretically possible or not publicly reported

It is assumed that the attacker uses the attack complexity corresponding to the attack action as the basis of selection when choosing the next attack action, and selects different available attack actions with different probabilities. For example, if an attacker in state S_0 can choose to leap to state S_i using attack action A_i, or choose to leap to state S_2 using attack rule A_2, and A_1 has a complexity level of E_6, A_2 with an offensive complexity level of E_3, the probability of choosing A_1 to make a state leap in state S_0 is $P\left(S_0 \xrightarrow{A1} S_t\right) = \frac{0.7}{0.7+0.3} = 0.7$, and similarly, the probability of choosing A_2 to make a state leap is $P\left(S_0 \xrightarrow{A2} S_2\right) = 0.22$. After attaching the corresponding probability values to the edges from which each state in the network attack graph is originated using the above method, a network attack graph with the extended Markov chain property is obtained, which can correspond to an extended Markov chain.

2.3. Mapping based on Markov chain attack graphs

In this paper, based on the vulnerability scoring system (CVSS), an algorithm is designed to map a general attack graph to an absorbing Markov chain attack graph, which provides a computational basis for subsequent network attack analysis by calculating the state transfer probability matrix of the attack graph. In this section, the absorbing Markov chain, the CVSS vulnerability scoring criteria, and the mapping algorithm for absorbing Markov chain attack graphs are described in detail in turn.

2.3.1. Absorbing Markov chains. A Markov chain is a collection of random sequences $X = \{X_1, X_2, \cdots, X_N\}$ containing a finite number of states that are discrete in time and state. The absence of posteriority of the Markov chain makes the probability P_{ij} of transferring from node S_i to the next node S_j related only to the current state and not to previous states, P_{ij} the state transfer probability of the Markov chain, and P the state transfer probability matrix consisting of P_{ij}.

Absorbing Markov Chain (AMC). A Markov chain that satisfies the following 2 conditions is called an absorbing Markov chain: it has at least one absorbing state, and it is capable of transferring to an absorbing state from every state node. The standard form of the state transfer probability matrix for an absorbing Markov chain is

$$P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} \tag{2}$$

where Q is a transition state transfer probability matrix of size $a \times a$: R is a matrix of size $a \times b$ representing the probability of transferring from a transition state to an absorbing state. 0 is a zero matrix of size $b \times a$. I is a unit matrix of size $b \times b$. P is of size $(a+b) \times (a+b)$ and $a+b$

represents the total number of states.

2.3.2. CVSS vulnerability scoring criteria. CVSS defines the basic exploitability score $Score(v)$ measure of vulnerability of a vulnerability v. The CVSS standard provides a framework for the calculation of the vulnerability exploitability score, defined as follows.

Vulnerability Exploitability Score (VES).CVSS gives a generalized standard for the Vulnerability Exploitability Score, which is calculated as:

$$Score(v) = 20 \times AV \times AC \times Au \tag{3}$$

where AV, AC and Au represent the attack vector (AV), attack complexity (AC) and authentication (Au) of the vulnerability, respectively, and the specific values of different vulnerabilities can be obtained by NVD query. The range of $Score(v)$ is 0 to 10, and the difficulty of exploiting a vulnerability is inversely proportional to $Score(v)$, i.e., the higher the score, the easier the vulnerability can be exploited. The lower the score, the more difficult the vulnerability is to be exploited.

2.3.3. Mapping algorithm for absorbing Markov chain attack graphs. In an attack graph, a successful atomic attack results in one transfer of state, however, in real network environments, the target network environment is usually in a non-ideal state due to the different reserves of attackers' knowledge and rapidly changing attack and defense scenarios. In the non-ideal state, a failed atomic attack can be regarded as a single transfer of a state node to itself. For the calculation of state transfer probability P_{ij}, the overall dynamic security of the network is assessed by calculating the life cycle factor of the vulnerability, analyzing its causal relationship with the attack graph, and synthesizing the impact of the vulnerability life cycle. Vulnerability life cycle $F(t_v)$. The life cycle function of vulnerability V is:

$$F(t_v) = 1 - \left(\frac{k}{t_v}\right)^{\alpha} \tag{4}$$

where t_v is the effective survival time of the vulnerability, calculated as the difference between the date the vulnerability was exposed and the date the vulnerability was scored by CVSS. Constants α and k are defined as shape factors, taking values of 0.26 and 0.00161, respectively.

The mapping algorithm takes into account the lifecycle of vulnerabilities and state transfer failures, and transforms the attack graph into a state transfer probability matrix, which is more reasonable and helps security researchers to analyze the security in a real network environment. Assuming that there are a total of n state nodes, the state changes in the attack graph are represented by a $n \times n$ matrix P, and the value of element P_{ij} in P represents the reachability probability of nodes S_i to S_j. In Algorithm 1, firstly, the time complexity of traversing the n state nodes is $O(n)$. Second, each node traverses the other nodes once to determine if a path exists between the two nodes, and if so, assigns a value to P_{ij}. Otherwise $P_{i,j} = 0$, the process requires $n \times n$ traversals, and the mapping time is $O(n^2)$ the time taken to compute the state transfer probability matrix P corresponding to the attack graph, which has no computational formula.

In the mapping algorithm, the vulnerability lifecycle over time is added as a way to calculate the state transfer probability matrix P corresponding to the attack graph to make it more relevant to the actual network environment. In an absorbing Markov chain, the probability of a state node eventually reaching an absorbing state sums to 1, and thus each row in matrix P satisfies:

$$\sum P_{i,j} = 1 \tag{5}$$

where P_{ij} denotes the success probability of transferring from state node S_i to state node S_j in the generated state transfer probability matrix P.

3. Example Analysis of Multi-Dimensional Data Extraction and Mapping

3.1. Analysis of the effect of multi-dimensional data extraction

3.1.1. Experimental environment. This system is written in python language, because python language has a simple to get started can quickly learn to develop, at the same time, python has cross-platform characteristics, you can run on windows, linux and other platforms, but the experimental environment is selected for the 64-bit Ubuntu 14.04.5LTS, taking into account the convenience of its production of docker images to facilitate the configuration of a single can be free of installation troubles. The version of python in the experimental environment is 3.0, and you need to install some python dependency libraries, which can be installed using the following commands:

pip install python-magic
pip install python-docx
pip install docx2txt
pip install PyPDF2
pip install pdfminer

Once the environment is configured, we can start exploring the transfer probability matrix Markov chain based feature extraction for multidimensional data.

3.1.2. Evaluation indicators. In order to test the accuracy extraction effect of the method, this paper takes the computer multidimensional attack trace data extracted in subsection 2.1.3 as an experimental analysis object, and an attack trace is composed of five key attributes: time T, source address sAddress, destination address dAddress, attack type name, and trace weight. 35,352 raw computer multidimensional attack trace data are obtained, which are stored in the form of datasets, and the training set (28,282) and validation set (7070) are respectively according to the 8:2 ratio, and the accuracy rate, false positive rate, positive recall rate, and negative recall rate are taken as the important evaluation indexes of the extraction effect, and the calculation formulas are shown as follows:

Definition TP: Positive category - number of positive categories, FN : Positive category - number of negative categories, FP : Negative category - number of positive categories, TN : Negative category - number of negative categories.

The AccuracyAccuracy formula is shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + EP} \tag{6}$$

The formula for positive recall $\mathrm{Re}call^{+}$ is shown below:

$$\mathrm{Re}call^{+} = \frac{TP}{TP + FN} \tag{7}$$

The formula for negative recall $\mathrm{Re}call^{-}$ is shown below:

$$\mathrm{Re}call^{-} = \frac{TN}{TP + FN} \tag{8}$$

The formula for calculating the false detection rate W is shown below:

$$W = \frac{FN + TN}{TP + TN + FN + FP} \tag{9}$$

3.1.3. Analysis of experimental results. In order to make the final results more scientific, the experiment takes several tests and produces the results: the five key attributes of time T (887), source address sAddress (1891), destination address dAddress (2432), attack type name (1296), and trace weight (7070-887-1891-2432-1296=564) are used as the test When the set is used, the results of each

index of extraction effect are shown in Table 2. The highest can reach 97.53% accuracy and 94.16% stability, and the average false detection rate with positive and negative recall are also stable, which are 1.76%, 99.06% and 75.61%, respectively.

Table 2. Extract the results of each indicator of the effect

Index	887	1891	2432	1296	564
Accuracy rate	0.9753	0.9713	0.9434	0.9513	0.9416
False detection rate	0.0176	0.0365	0.0576	0.0574	0.0588
Positive recall rate	0.9906	0.9756	0.9528	0.9535	0.9466
Negative recall rate	0.7561	0.7356	0.7253	0.7216	0.7201

In order to further confirm the effect of computerized multidimensional data extraction based on Markov chain (MC), the model in this paper is compared and analyzed with other control models, which include random forest model (RF), support vector machine (SVM), and long and short-term memory neural network (LSTM), the comparison of extraction accuracy rate of each type of model is shown in Table 3, and the comparison of negative recall of each type of model extraction is shown in Table 4. The results of the comparison of the negative recall of various models are shown in Table 4. It can be seen that Markov chain (MC) has a higher extraction accuracy and a more stable false detection rate compared with random forest model (RF), support vector machine (SVM), and long short-term memory neural network (LSTM), which is mainly reflected in the time T (887), source address sAddress (1891), destination address dAddress (2432), attack type name (1296), trace weight (564) in the test, compared with other relatively cold algorithms, Markov Chain (MC) is more malleable and has a wider scope of expansion, thus proving that the algorithm has good results in computerized multidimensional data extraction.

Table 3. Comparison of extraction accuracy of various models

Model	887	1891	2432	1296	564
RF	0.8935	0.8871	0.8988	0.9125	0.8997
SVM	0.9172	0.9145	0.9253	0.9279	0.9286
LSTM	0.9437	0.9458	0.9462	0.9476	0.9491
MC	0.9834	0.9699	0.9756	0.9547	0.9682

Table 4. Comparison results of negative recall rate extracted from various models

Model	887	1891	2432	1296	564
RF	0.4106	0.3766	0.3894	0.3514	0.3611
SVM	0.4956	0.4912	0.4889	0.4936	0.4997
LSTM	0.5821	0.5766	0.5737	0.5915	0.5838
MC	0.7616	0.7406	0.7309	0.7316	0.7418

3.2. Simulation Analysis of Markov Chain Based Attack Graph Modeling

3.2.1. Structure and parameters of the attack map. Vulnerability scanning of each host in the cloud platform using Nessus scanner, the information obtained for each vulnerability is shown in Table 5 with CASS attributes (AV is attack vector, AC is complexity, Au is authentication). Among them, the vulnerability CVE-2017-2620 of the host (WS) terminal is a newly discovered vulnerability at the t_ith moment, and the network connectivity, network configuration, and vulnerability information are inputted into the Markov chain program edited by the Python software, which generates the network structure of the transfer probability attack Fig. PAG_0, and the network structure of the attack Fig.

PAG_0 is shown in Fig. 2. From Fig. 2, it can be seen that there are six possible attack paths, in which the attacker uses the overflow vulnerability CVE-2016-3733 on the WS to obtain the administrator privileges of the WS, and then writes a Trojan horse program on the FS through the NFS interface, which is accidentally executed, thus allowing the attacker to take control of the FS. Of course, the attacker can also directly utilize the overflow vulnerability CVE-2014-0923 on FS to gain access to FS. Then VM2 downloaded the file with Trojan horse from FS and thus was controlled, and the attacker manipulated VM2 to launch an attack on Spark server by exploiting the vulnerability CVE-2018-8012. Similarly, the attacker manipulates FS to launch an attack on virtual machine VM1 and further exploits vulnerability CVE-2016-3043 to launch an attack on the Hadoop platform to steal its important information or files, etc.

Table 5. Vulnerability information

Host	Edition	Loophole	Description	AV/AC/Au
WS	Apache 2.0	CVE-2016-3733	execute code, dos	N/H/N
WS	IE 9	CVE-2019-1339	overflow, execute code	N/H/N
FS	NFS 1.0	CVE-2014-0923	execute code, overflow	N/L/N
VM1	Win7 SP1	CVE-2017-0123	execute code	N/M/N
PM3	Hadoop 2.6	CVE-2016-3043	obtain information	N/L/N
PM4	Spark 2.1	CVE-2018-8012	obtain information	N/M/S
VMM	Xen 4.6	CVE-2017-2615	dos, gain privileges	L/M/N

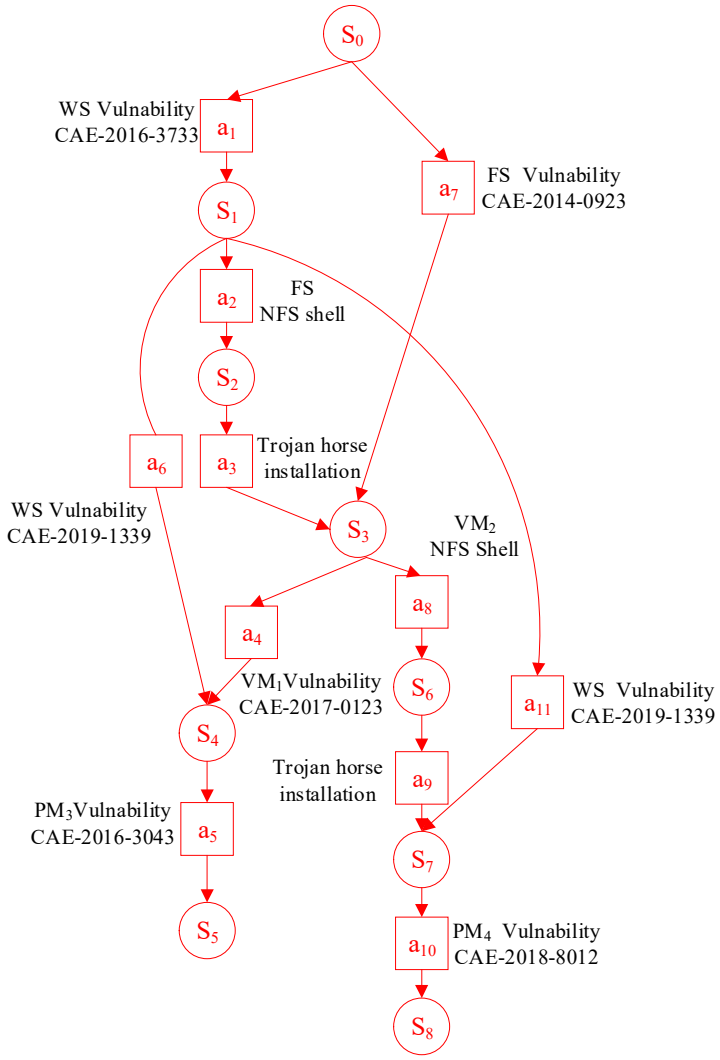

Fig. 2. Attack diagram PAG_0 Network structure.

According to the structure of the transfer probabilistic attack graph in Fig. 2 and the CASS attributes in Table 5, we can determine the weights of the directed edges in the probabilistic attack graph PGA_0, and the weights of the directed edges are shown in Table 6, which shows that the directed edges (S_8, a_{12}) and (a_{12}, S_9) are the new directed edges added at the time when a new vulnerability is found in moment t_i. Based on the network structure and network parameters, the complete transfer probability attack graph PGA_0 is constructed.

Table 6. The weight of the directed edge

Directed edge	Weight	Directed edge	Weight	Directed edge	Weight	Directed edge	Weight
(S_0, a_1)	0.44	(S_3, a_4)	0.78	(S_0, a_7)	1.0	(S_7, a_{10})	0.59
(a_1, S_1)	1.0	(a_4, S_4)	1.0	(a_7, S_3)	1.0	(a_{10}, S_8)	1.0
(S_1, a_2)	0.55	(S_4, a_4)	1.0	(S_3, a_8)	0.55	(S_1, a_{11})	0.44
(a_2, S_2)	0.84	(a_5, S_5)	1.0	(a_8, S_6)	0.84	(a_{11}, S_7)	1.0
(S_2, a_3)	0.55	(S_1, a_6)	0.44	(S_6, a_9)	0.55	(S_8, a_{12})	0.27
(a_3, S_3)	0.84	(a_6, S_4)	1.0	(a_9, S_7)	0.84	(a_{12}, S_9)	1.0

3.2.2. Analysis of the attack graph construction process. The dynamic update of the transfer probabilistic attack graph is shown in Fig. 3, where (a)~(d) denote $t_0 \sim t_3$. For the sake of generality, it is assumed that no alarm event is observed at the moment t_0, which is equivalent to the transfer probabilistic attack graph PAG_0. Fig. 3(a) visualizes the attacker's most probable attack intent and the attack path, where the probability of the state node S_5 is 0.84, and that of the state node S_8 is 0.27, so the attacker's attack intent is S_5, the attack target is the host running Hadoop PM_3, and the maximum probability attack path is $\{S_0, a_7, S_3, a_4, S_4, a_5, S_5\}$. Fig. 3(b) shows the update of the probabilistic attack graph at moment t_1. Assuming $GC_1 \neq \varnothing$, i.e., the attack graph does not have any update operation, the alarm event $O_1 = \{o_1, o_2, o_3\}$ is observed at moment t_1, which corresponds to a confidence level of $PO = \{0.76, 0.55, 0.44\}$, and the candidate attack sequences consist of $\{S_0, a_1, S_1, a_2, S_2, a_3, S_3, a_8, S_6, a_9, S_7, a_{10}, S_8\}$ and $\{S_0, a_1, S_1, a_2, S_2, a_3, S_3, a_4, S_4, a_5, S_5\}$, and the probability of the state node S_5 is calculated to be 0.86 and the probability of the state node S_8 to be 0.27, so that the attacker's intention of attacking remains to be S_5, and the maximum probabilistic attack path is $\{S_0, a_1, S_1, a_2, S_2, a_3, S_3, a_4, S_4, a_5, S_5\}$. Fig. 3(c) shows that if the probabilistic attack graph is updated at the moment t_2 and the vulnerability in PM_3 is repaired at the moment t_2, the corresponding failed action node a_5 and its posterior state node S_5 as well as the related edges should be deleted in the probabilistic attack graph, i.e., the update graph operation $GC_2 = \{< D, a_5, S_5 >\}$ is performed. Assuming that alarm event $O_2 = \{o_{11}, o_{10}\}$ is observed at moment t_2, its corresponding confidence level is $PO = \{0.44, 0.55\}$, and all alarm events O_2 are valid alarms, the target node can be judged to be S_5, the attack intent is S_8, and the maximum probability attack path is $\{S_0, a_1, S_1, a_{11}, S_7, a_{10}, S_8\}$. Fig. 3(d) shows the updated probabilistic attack graph at the moment of t_3, and similarly the attack intent at the moment of t_3 is S_9, the maximum probabilistic attack path is $\{S_0, a_1, S_1, a_{11}, S_7, a_{10}, S_8, a_{11}, S_9\}$, and its corresponding confidence level is 0.47. By constructing a complete and dynamically evolving attack graph, it then helps cloud security administrators to grasp the security posture of the computer terminals from a holistic point of view.

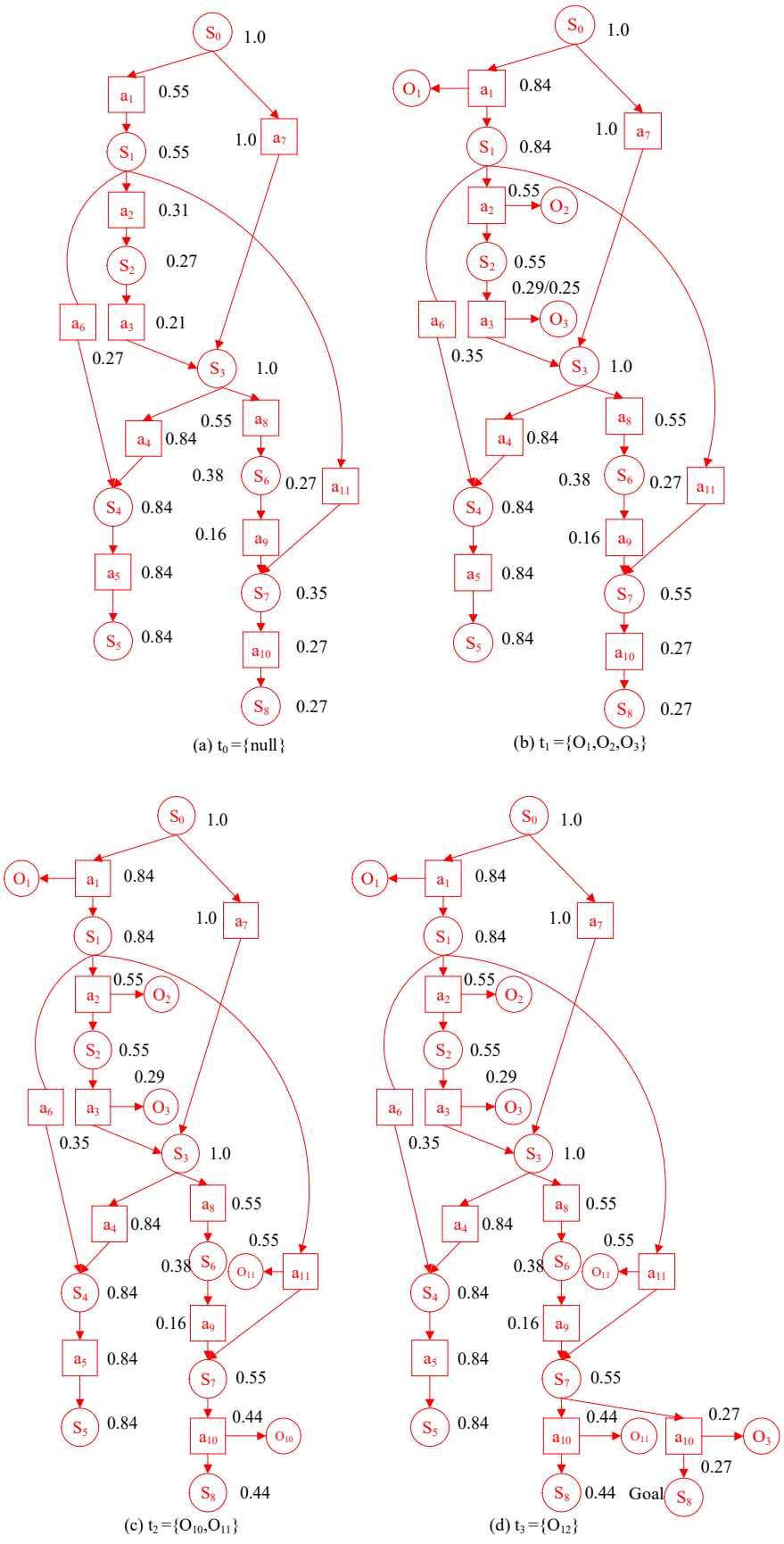

Fig. 3. Dynamic update of transition probability attack graph

3.3. Mapping analysis based on Markov chain attack graphs

3.3.1. CVSS Vulnerability Assessment. The CVSS vulnerabilities are evaluated with the help of the scoring criteria mentioned in subsection 2.3.2.The CVSS vulnerabilities contain seven CVE-2016-3733, CVE-2019-1339, CVE-2014-0923, CVE-2017-0123, CVE-2016-3043, CVE-2018-8012, CVE- 2017-2615, and the CVSS vulnerability assessment results are shown in Table 7. According to the data in the table, it can be seen that the overall scores of AV (Attack Vector), AC (Complexity), and Au (Authentication) for the seven CVSS vulnerabilities are 0.702, 0.701, and 0.695, respectively, and the final exploitable score is 6.84 (6.84=20×0.702×0.701×0.695), which confirms that the CVSS vulnerability assessment criteria proposed in this paper can accurately reflect the CVSS vulnerability situation of computer terminals.

Table 7. CVSS vulnerability assessment results

CVSS vulnerability	AV	AC	Au
CVE-2016-3733	0.695	0.699	0.796
CVE-2019-1339	0.657	0.745	0.651
CVE-2014-0923	0.761	0.654	0.703
CVE-2017-0123	0.794	0.769	0.767
CVE-2016-3043	0.637	0.768	0.703
CVE-2018-8012	0.632	0.657	0.612
CVE-2017-2615	0.739	0.675	0.635
Total	0.702	0.710	0.695

3.3.2. Mapping Algorithm Validation Analysis. In the mapping algorithm validation analysis, the experimental validation is mainly centered on the comparative analysis of the mapping time of the seven CVE-2016-3733, CVE-2019-1339, CVE-2014-0923, CVE-2017-0123, CVE-2016-3043, CVE-2018-8012, and CVE-2017-2615 analysis, in order to study the results are persuasive, set up a control group, and the control group of this experiment is LSTM, the results of the mapping algorithm validation analysis are shown in Figure 4. According to the labels in the figure, it can be seen that the average mapping time of the LSTM model for the seven vulnerabilities is 117ms, while the average mapping time of this paper's algorithm is 80ms, compared to the LSTM model, the average mapping time of the seven vulnerabilities is improved by 37ms, which verifies the priority of this paper's mapping algorithm, and helps security researchers to more quickly and accurately repair the weak links in the network.

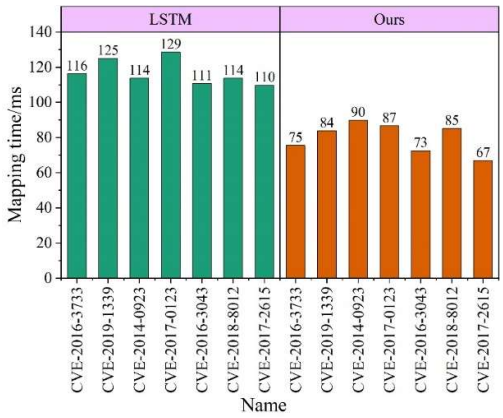

Fig. 4. Mapping algorithm validation analysis results

4. Conclusion

In this paper, the multi-dimensional data extraction and mapping of power monitoring system is investigated for the detection of power monitoring system under comprehensive host attack traces and fine-grained deep threats. First, for the extraction of host attack traces, an attack graph model based on Markov chain is proposed. Second, for the situation that attacks are highly hidden and difficult to be recognized by existing detection means, this paper dynamically eliminates suspicious nodes from the set of directed connectivity domination to ensure the generality of the vulnerability scoring system (CVSS). Numerical simulation results found that the mapping algorithm embedded in the power monitoring system can provide the feature values of multi-dimensional data from the software level, thus greatly reducing the amount of computation to achieve the effect of real-time detection, and the accuracy rate is high, with a certain degree of effectiveness and feasibility, to help security researchers to more quickly and accurately repair the weaknesses of the network, and to maintain the computer security detection management has a contributing role.

Funding

Qinghai Electric Power Company support project (project No: 522807240004) of “Research on Deep Threat Detection and Attack Forensic Technology Based on Multi source Heterogeneous Hosts in Power Monitoring System”.

References

[1] Alladi, T., Chamola, V., Sikdar, B., & Choo, K. K. R. (2020). Consumer IoT: Security vulnerability case studies and solutions. IEEE Consumer Electronics Magazine, 9(2), 17-25.

[2] Aras, E., Ramachandran, G. S., Lawrence, P., & Hughes, D. (2017, June). Exploring the security vulnerabilities of LoRa. In 2017 3rd IEEE international conference on cybernetics (CYBCONF) (pp. 1-6). IEEE.

[3] Sinha, P., Jha, V. K., Rai, A. K., & Bhushan, B. (2017, July). Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: A survey. In 2017 International Conference on Signal Processing and Communication (ICSPC) (pp. 288-293). IEEE.

[4] Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1), 616-644.

[5] Deb, R., & Roy, S. (2022). A comprehensive survey of vulnerability and information security in SDN. Computer Networks, 206, 108802.

[6] Zhang, J. (2019). Detection of Network Protection Security Vulnerability Intrusion Based on Data Mining. Int. J. Netw. Secur., 21(6), 979-984.

[7] Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), 1333.

[8] Wu, S., Zhang, Y., & Cao, W. (2017). Network security assessment using a semantic reasoning and graph based approach. Computers & Electrical Engineering, 64, 96-109.

[9] Dong, X., Wu, F., Faree, A., Guo, D., Shen, Y., & Ma, J. (2019). Selfholding: A combined attack model using selfish mining with block withholding attack. Computers & Security, 87, 101584.

[10] Angelini, M., Blasilli, G., Catarci, T., Lenti, S., & Santucci, G. (2018). Vulnus: Visual vulnerability analysis for network security. IEEE transactions on visualization and computer graphics, 25(1), 183-192.

[11] Ibrahim, M., Al-Hindawi, Q., Elhafiz, R., Alsheikh, A., & Alquq, O. (2019). Attack graph implementation and

visualization for cyber physical systems. Processes, 8(1), 12.

[12] Cetinkaya, A., Ishii, H., & Hayakawa, T. (2019). An overview on denial-of-service attacks in control systems: Attack models and security analyses. Entropy, 21(2), 210.

[13] Al Ghazo, A. T., Ibrahim, M., Ren, H., & Kumar, R. (2019). A2G2V: Automatic attack graph generation and visualization and its applications to computer and SCADA networks. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(10), 3488-3498.

[14] Rizvi, S., Orr, R. J., Cox, A., Ashokkumar, P., & Rizvi, M. R. (2020). Identifying the attack surface for IoT network. Internet of Things, 9, 100162.

[15] Hu, H., Zhang, H., Liu, Y., & Wang, Y. (2017). Quantitative method for network security situation based on attack prediction. Security and Communication Networks, 2017(1), 3407642.

[16] Fu, R., Huang, X., Xue, Y., Wu, Y., Tang, Y., & Yue, D. (2018). Security assessment for cyber physical distribution power system under intrusion attacks. IEEE Access, 7, 75615-75628.

[17] Gangjun, G. O. N. G., Peng, Z. H. A. N. G., Bo, Z. H. O. U., Ren, Q. I. A. N. G., Yue, S. U. N., Leran, C. H. E. N., & Wei, C. H. E. N. (2021). Network security risk assessment of cps system in distribution network based on attack graph. In Journal of Physics: Conference Series (Vol. 1750, No. 1, p. 012078). IOP Publishing.

[18] Ye, J., Cheng, X., Zhu, J., Feng, L., & Song, L. (2018). A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks, 2018(1), 9804061.

[19] Lallie, H. S., Debattista, K., & Bal, J. (2017). An empirical evaluation of the effectiveness of attack graphs and fault trees in cyber-attack perception. IEEE Transactions on Information Forensics and Security, 13(5), 1110-1122.

[20] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 international carnahan conference on security technology (ICCST) (pp. 1-8). IEEE.

[21] Li, F. (2023, April). Network security evaluation and optimal active defense based on attack and defense game model. In 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-7). IEEE.

[22] Kohlios, C. P., & Hayajneh, T. (2018). A comprehensive attack flow model and security analysis for Wi-Fi and WPA3. Electronics, 7(11), 284.

[23] David Herranz Oliveros,Marino Tejedor Romero,Jose Manuel Gimenez Guzman & Luis Cruz Piris. (2024). Unsupervised Learning for Lateral-Movement-Based Threat Mitigation in Active Directory Attack Graphs. Electronics(19),3944-3944.

[24] Ying Zhou,Zhiyong Zhang,Kejing Zhao & Zhongya Zhang. (2024). A novel dynamic vulnerability assessment method for Industrial Control System based on vulnerability correlation attack graph. Computers and Electrical Engineering(PA),109482-109482.

[25] Fengming Liu & Yingda Song. (2025). Analysis of credit ABS based on Markov chain approaches. Finance Research Letters106432-106432.

[26] J.D.K. Bishop & C.J. Axon. (2024). Using natural driving experiments and Markov chains to develop realistic driving cycles. Transportation Research Part D104507-104507.