Dynamic Mining of Wireless Network Information Transmission Security Vulnerabilities Based on Spatiotemporal Dimension

Qiang Chen(✉), Fang Qian, and Yukang Liu

China Southern Power Grid Ultra High Voltage Transmission Company, Guangzhou 510000, China
cgycgy8680@126.com

Abstract. In order to improve the efficiency of dynamic mining for wireless network information transmission security vulnerabilities and improve the accuracy of mining results, this paper proposes a dynamic mining method for wireless network information transmission security vulnerabilities based on the spatiotemporal dimension. Firstly, collect data on security vulnerabilities in wireless network data transmission; Secondly, wavelet transform is introduced to filter and process wireless network information transmission security vulnerability data; Then, in the deep neural network architecture, the instruction level word embedding method based on Word2vec obtains the feature attributes of wireless network information transmission security vulnerabilities; Finally, dynamically mine wireless network information transmission security vulnerabilities based on the spatiotemporal dimension. The experimental results show that the vulnerability dynamic mining method proposed in this paper takes 25.8 s, with an accuracy of 99.0% and a recall rate of 98.1%, which can improve the effectiveness of vulnerability dynamic mining.

Keywords: Spatiotemporal Dimension · Wavelet Transform · Deep Neural Network · Instruction Level Word Embedding · Dynamic Vulnerability Mining

1 Introduction

With the development of wireless internet and the penetration of information technology, there are currently more and more industrial control systems connected to computer networks [1]. Compared with classic information systems, industrial systems mostly use specialized software, devices, and protocols, so the security vulnerabilities of industrial control systems are different from those of information systems. Industrial vulnerabilities can be classified into device vulnerabilities, wireless protocol vulnerabilities, and industrial software vulnerabilities based on their location [2, 3]. Hackers or attackers exploit vulnerabilities to attack the system. If vulnerabilities are discovered and exploited by the attacker in a timely manner, corresponding remedial measures can be

L. Yun et al. (Eds.): ADHIP 2023, LNICST 548, pp. 403–417, 2024.
https://doi.org/10.1007/978-3-031-50546-1_27

taken to effectively reduce the likelihood of the system being attacked. Vulnerability detection and mining technology is used to discover system vulnerabilities in a timely manner. In addition, the vulnerabilities of industrial systems have their own characteristics, so the vulnerability detection and mining methods of traditional information systems may not be fully applicable to vulnerability discovery under the Industrial Internet [4]. Therefore, the study of vulnerability detection and mining technology combined with traditional information system and industrial control system will add a layer of barrier to protect modern industrial control system, escort the vigorous development of Industrial Internet, and even have important significance to protect people's livelihood and national stability. Reference [5] puts forward the method of network Computer security hidden trouble and vulnerability mining, which requires a comprehensive security hidden trouble analysis of the network computer system. Identify potential security risks by reviewing known vulnerability databases, analyzing malware samples, and network attack behaviors. Based on the existing security hazard analysis results, it is necessary to research and develop corresponding vulnerability mining methods. After discovering potential vulnerabilities, verification and repair are required. Trigger vulnerabilities by constructing input with malicious intent and detect whether the system is under attack. Repairs can include modifying source or binary code, adding security checks and error handling mechanisms, or updating system and application patches. Establish a security monitoring system, regularly scan and monitor networks and systems, and promptly identify new security risks and vulnerabilities. At the same time, it is also necessary to pay attention to the latest security technologies and research results, and continuously improve vulnerability mining methods and security protection measures. Reference [6] proposes a network vulnerability mining method based on Apriori risk data analysis to collect and prepare data related to network vulnerabilities. This includes vulnerability databases, network attack logs, system logs, etc. Ensure the integrity and accuracy of data, and carry out appropriate preprocessing, such as removing duplicate data, handling missing values, etc. Perform risk analysis on network vulnerability data using the Apriori algorithm. Based on the results of risk data analysis, specific vulnerabilities can be selected for mining and evaluation. This can be achieved through in-depth analysis of vulnerability combinations and frequent itemsets in association rules. Based on the mining results, evaluate the severity, scope of impact, and potential risks of vulnerabilities. After discovering specific vulnerabilities, corresponding repair and preventive measures need to be taken. This may include application patch updates, system configuration adjustments, network security policy improvements, etc. At the same time, based on the mining results, potential vulnerabilities can be predicted in the future, and proactive preventive measures can be taken to reduce the probability and impact of vulnerabilities occurring. However, the above methods have poor efficiency in dynamically mining vulnerabilities and improving the accuracy of mining results.

Therefore, this article proposes a dynamic mining of wireless network information transmission security vulnerabilities based on spatiotemporal dimensions.

2 Wireless Network Information Transmission Security Vulnerability Data Processing

2.1 Wireless Network Information Transmission Security Vulnerability Collection

Due to the sensitivity and aggressiveness of vulnerabilities, the vulnerability site data does not have a publicly available dataset, making it difficult to obtain. In the early stage, by searching and reading a large amount of literature online, only a portion of SQL injection attack statement data was collected and no vulnerability data was found. Therefore, the experiment requires collecting SQL injection vulnerability data on one's own. There are two ways to collect vulnerabilities, one is to scan specific websites, and the other is to use non-public data authorized by the internship company for research purposes. During the experiment, a total of eight weeks were spent using a scanner to scan over 1500 websites and identify over 1300 vulnerabilities.

The sources of vulnerability websites include a large number of testing stations built through dockers and online vulnerability testing stations publicly available for testing.

Each vulnerability data should contain the following basic information: URL, which is the website URL containing SQL injection vulnerabilities; The HTTP request method is related to the location of the injection point; Injection type, which is a detailed classification of SQL injection types; Vulnerability attack statement, which detects the presence of SQL injection vulnerabilities. By analyzing and extracting the scanning results, the following vulnerability related information was obtained, as shown in Table 1.

Table 1. Scan Results Table

Vulnerability Information	content
URL	Request Target Address
Injection Point	Parameters that can be injected into SQL
HTTP request type	GET, POST Wait for request type
SQL Injecting attack statements	Message for implementing injection attacks
Attack Details	The final SQL injection attack statement and the SQL injection attack statement used during the attack process
HTTP request	Including request type, request header, and request data
HTTP Response information	HTTP requests return information, usually HTML pages or JSON data
Vulnerability classification	Vulnerability type, including CVE and CVSS information
Vulnerability Details	Include vulnerability details and solutions

The scanning results of the scanner, whether displayed through a web page or exported through a report, cannot be input into the model. Therefore, it is necessary to extract the required vulnerability information from the scanner and process it into

a format that is convenient for input into the model. By analyzing the principle of the scanner, all information about the vulnerability was obtained through the API interface. Finally, the vulnerability data was obtained through constructing HTTP requests and stored in MySQL [8].

The internship company obtained approximately 2000 pieces of vulnerability information through authorized penetration vulnerability scanning, covering various databases and injection types of data. The vulnerability information includes URLs, injection points, database types, HTTP request types, and SQL injection attack statements, meeting the requirements of model training. The vulnerability information obtained through the above two methods is processed and the final stored data format is shown in Table 2:

Table 2. Vulnerability Information Data

field	data
domain	http://www.*****
url	http://www.*****
inject_type	SQL injection
request_type	POST
inject_key	q_year
payload	Page = 0&PageSelect = &q_day = &q_month = &q_year = %BF'%BF"

Complete the collection of wireless network information transmission security vulnerability information.

2.2 Wireless Network Information Transmission Security Vulnerability Data Filtering Processing

At present, mainstream wireless network protocols include Modbus, EtherCAT, Powerlink, Porfinet, Ethernet/IP, TSN (Time Sensitive Networks), etc. There are various methods to capture data packets from different wireless network protocols, among which the most direct method is to apply appropriate message packet capture tools to capture data packets generated by industrial control systems from real industrial control network environments as training data. After capturing a sufficient number of data packets from the wireless network environment, it is necessary to perform data preprocessing operations on these raw data [9].

Leopard Mobile data set selected in this chapter is a binary code file, which is disassembled into assembly code by IDA pro tool; The CWE119 dataset is a C language program code file that needs to be compiled into assembly code format. Batch format the compiled assembly code through Python scripts, remove textual description information from the code, and retain function and instruction information. In the above dataset, each sample has a corresponding 0 or 1 label, with 0 indicating vulnerability and 1 indicating

no vulnerability. The ratio of sample size for training, validation, and testing sets is 3:1:1. Before each training and testing, the order of the data is disrupted.

In order to better complete the data mining and processing of wireless network information transmission security vulnerabilities, priority is given to introducing wavelet transform to filter and process wireless network information transmission security vulnerability data [10]; Wavelet transform is a high-performance denoising algorithm that has applications in both data and image fields. It achieves multi-scale refinement of wireless network information transmission security vulnerability data through operations such as scaling and translation, thereby obtaining high and low frequency parts. Further refinement of high and low frequency parts can obtain detailed information of corresponding data. Wavelet analysis theory has been widely applied in fields such as signal and speech analysis.

cIf a represents $\tau(x)$ square integrable function, then the Fourier transform $\beta(x, y)$ needs to satisfy the constraint condition ψ:

$$\psi = \int_{-\infty}^{\infty} \frac{\beta(x, y)}{\tau(x)} - \vartheta_{(x,y)}(t) \tag{1}$$

In the above equation, $\vartheta_{(x,y)}(t)$ represents the wavelet mother function, which is expanded, scaled, and translated to obtain the corresponding wavelet basis function. Then, the continuous wavelet transform $\left|\vartheta_{(x,y)}(t) \cdot \tau(x)\right|$ of wireless network information transmission security vulnerability data is represented in the form of formula (2):

$$\left|\vartheta_{(x,y)}(t) \cdot \tau(x)\right| = \begin{cases} \frac{1}{\sqrt{x}} \int \vartheta_{(x,y)}(t) \times \left(\frac{\alpha - x}{\tau(x)}\right) \\ \left\langle \vartheta_{(x,y)}(t), \tau(x), \alpha \right\rangle \end{cases} \tag{2}$$

In the above equation, x represents the scale factor; α represents the translation factor.

In order to simplify the calculation flow of the computer, it is necessary to discretization all continuous wavelets. Discretization mainly deals with the above two different factors. Applying binary dynamic networks to the wavelet transform process [10, 11] yields the binary wavelet transform c, which corresponds to the following expression:

$$R_{(i,j)} = \vartheta_{(x,y)}(t) \cdot \tau_{ij}(u, v) \cdot \tau(x) \tag{3}$$

In the above equation, $\tau_{ij}(u, v)$ represents the sliding factor coefficient.

At present, some denoising methods for data have certain limitations, mainly targeting partial noise. However, using wavelet transform for denoising can not only achieve satisfactory denoising results, but also run faster than other methods. Priority should be given to conducting wavelet transform operations on wireless network information transmission security vulnerability data containing noise, obtaining wavelet coefficients. Further processing of the wavelet coefficients can obtain the latest wavelet coefficients, and reconstructing the wavelet coefficients can obtain the denoised data. Among them, the wireless network information transmission security vulnerability data noise detection model $\rho(i)$ can be expressed in the form of formula (4):

$$\rho(i) = t(i) + \vartheta(i) \cdot \tau_{ij}(u, v) \cdot \tau(x) \tag{4}$$

In the above equation, $t(i)$ represents wireless network data transmission security vulnerability data containing noise; $\vartheta(i)$ represents real wireless network data transmission security vulnerability data.

Select a suitable wavelet basis and decompose the abnormal transmission data of the cloud platform using the wavelet basis. The decomposition process is shown in Fig. 1.

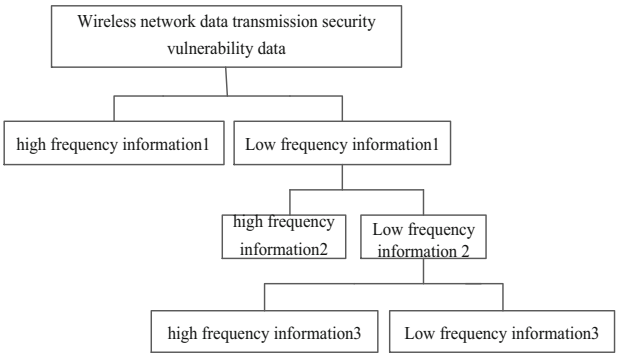

Fig. 1. Schematic diagram of data decomposition for wireless network information transmission security vulnerabilities

After completing the decomposition processing of wireless network information transmission security vulnerability data, select the appropriate wavelet basis to determine the decomposition level, and perform multi-level decomposition processing on the data. $H_{(n)}$ The corresponding calculation formula is:

$$H_{(n)} = \frac{\vartheta(i)}{t(i)-1} \cdot \vartheta_{(x,y)}(t) \cdot \tau_{ij}(u,v) \tag{5}$$

After determining the threshold, perform soft threshold quantization processing $\varpi_{x,y}$ on all high-frequency coefficients using the selected threshold, and the corresponding calculation formula is as follows:

$$\varpi_{x,y} = \sqrt{\frac{\vartheta_{(x,y)}(t) \cdot \tau_{ij}(u,v)}{x-1}} - \frac{\left(x - R_{(i,j)}\right)}{H_{(n)}} \tag{6}$$

Calculate the sensitivity of all wavelet basis analysis wireless network information transmission security vulnerability data, sort and process them, select the least sensitive wavelet basis wireless network information transmission security vulnerability data for wavelet decomposition processing, and then reconstruct the high and low frequency parts, ultimately achieving wireless network information transmission security vulnerability data filtering processing.

2.3 Instruction Level Word Embedding Method Based on Word2vec

In the security leak detection of wireless network information transmission based on deep neural network, the original code data cannot be directly used as the input of neural

network, so it is necessary to convert structured binary code into unstructured vector form. At present, the most common word embedding model is Word2vec, which is a language model that learns low dimensional word vectors rich in Semantic information from massive text corpora in an unsupervised manner. Word2vec word vector model maps words from the original space to the new low dimensional vector space. The similarity between each pair of words can be measured by vector similarity, so Word2vec word vector has good semantic mapping characteristics.

Word2vec includes two training models, namely, the Continuous bag-of-words model (CBOW) and the Skip gram model. CBOW is a model that calculates a word vector based on N words before or N words before and after a word, while Skip gram calculates a vector of several words that appear before and after a word.

Because the CBOW model and Skip gram model are mirror images of each other, a brief introduction to the CBOW model is sufficient here. The basic principle of the CBOW model is as follows:

The CBOW model utilizes contextual words to predict target words. The target word is, its contextual environment is $Context(w)$, and the word set of the entire corpus is w, so the model is transformed into calculating probability:

$$P(w|Context(w)) \tag{7}$$

When training the entire corpus to obtain word vectors, combined with the maximum logarithmic likelihood principle, the objective function form of the CBOW model can be obtained as follows:

$$L = \sum_{w \in W} \log P(w|Context(w)) \tag{8}$$

The specific steps are as follows:

Represent the input sentence as $v(Context(w)_1)$, $v(Context(w)_2)$, ..., $v(Context(w)_n) \in \Re^m$ context word vector based on the number of words, where m represents the dimension of the word vector and n represents the number of word vectors. Accumulate and sum the word vectors in the input layer, i.e. $X_w = \sum_{i=0}^{n} v(Context(w)_i) \in \Re^m$.

Build $N = |D|$ Huffman tree with the frequency of each word as the weight. The Huffman tree has a total of a leaf node and $N - 1$ non leaf nodes, where D is the number of all words in the corpus.

Using the Hierarchical Softmax method to solve the CBOW model, the objective function is transformed into:

$$L = \sum_{w \in W} \sum_{j=2}^{l^w} L(w, j) \tag{9}$$

$$L(w, j) = \{(1 - d_j^w) \cdot \log[X_w^T \theta_{j-1}^W] + d_j^w \cdot \log[1 - \sigma(X_w^T \theta_{j-1}^W)]\} \tag{10}$$

Among them, l^w represents the number of nodes covered in the route p^w of Huffman tree species from the root node to the corresponding leaf node of word w; The Huffman encoding of the f-th node in the path where the word '$d_j^w \in \{0, 1\}$ 'corresponds to' w',

but the root node does not correspond. Code; The word vector of non leaf nodes in the path corresponding to the word $\theta_{j-1}^{w} \in w$; σ is the sigmoid function. At the same time, the CBOW model uses the random gradient ascent method to update parameters.

3 Fuzzy Testing Vulnerability Mining Algorithm Based on Spatiotemporal Dimension

In the algorithm architecture, there are two sub networks, namely generator network G and discriminator network D. One of our design concepts is to design a lightweight vulnerability mining model based on implementing a lightweight model. The model should have the characteristic of reducing computational resource consumption, making it easy to deploy to embedded devices and laying the foundation for the future online learning ability of the model. Therefore, on the premise of meeting the constraints of the spatiotemporal dimension architecture and the requirements mentioned earlier, we design a reasonable simplified architecture diagram based on the spatiotemporal dimension, as shown in Fig. 1. This framework corresponds to the minimax game between the generator and the decider, and the formula can be expressed as follows:

$$\min_{G} \max_{D} V(DG) = E_{x-Px}[\log(D(x))] + E_{z-Px}[-\log(D(G(z)))] \tag{11}$$

Among them, $D(x)$ represents the probability that the discriminator will determine the true data correctly, and $G(z)$ represents the weight matrix output by the generator after inputting noise data.

a. Generator Fig. 3.3 (a) depicts the network structure of the generator of a fuzzy testing vulnerability mining model based on the spatiotemporal dimension. The generator adopts a deconvolution neural network structure and is composed of multiple deconvolution layers. Specifically, unlike traditional convolutional networks, we replace the pooling layer in the generator with the deconvolution layer. Deconvolution, also known as transposed convolution or fractional step convolution, works by exchanging the forward and backward propagation of convolutions. Based on zero padding and non unit step size, the following formula formulates the output size of the generator's deconvolution in this study:

$$a = (i + 2p - k)\%\mathrm{s} \tag{12}$$

$$o' = s(i' - 1) + a + k - 2p \tag{13}$$

Where $o'(o_1 = o_2 = o)$ represents the output size of the matrix, $i'(i'_1 = i'_2 = i')$ represents the input of the matrix, $k(k_1 = k_2 = k)$ represents the size of the convolutional kernel, s represents the step size along both axes, p represents the same filling along both axes, $i(i_1 = i_2 = i)$ represents the input size of the next convolutional layer, and d represents the amount of 0 added to the bottom and right of the input. We idealize parameter settings here, but please note that the formula here also extends to the case where the input matrix in the n dimension is not a square matrix.

Tanh is used in the output layer of the generator, and ReLU is used in the activation function of other layers, such as the input layer and the middle layer. The generator extracts noise data from a uniform noise distribution as input and outputs a two-dimensional matrix as input to the discriminator model. A two-dimensional matrix can be viewed as a sequence, transformed through a character embedding layer, with each row representing one character of the protocol message. It decodes the output of the generator into the generated test case. In order to decode the matrix generated by the generator, we also constructed a BLSTM network. It is considered here that the matrix generated by the generator is equivalence relation to the generated BLSTM network intermediate semantic vector. Characters are output through nonlinear transformation and softmax layer, and the matrix generated by the generator is input to the output part of the BLSTM network as input. The training and optimization strategies of the BLSTM network are introduced in the following text. The loss function of the generator is:

$$E_{z-P_z}[-\log(D(G(z)))] \tag{14}$$

b. Determinator In adversarial training, the discriminator network is designed to guide the training of the generator. Convert the byte representation of each preprocessed actual protocol message into one hot vector representation, and the matrix of the input message packet includes time step dimension and feature vector dimension. Most existing models only consider the time step dimension of the text to obtain a fixed length vector, while ignoring spatial structural features. However, the time step dimension and feature vector dimension are not mutually independent.

In order to fuse the features of two dimensions together, a combined model BLSTM-DCGAN based on BLSTM and CNN is proposed, which enables the discriminator to retain not only the time step dimension but also the feature vector dimension information. A one hot vector is a sequence of fixed length and dimension, which can be regarded as a matrix.

Convert to another vector through the character embedding layer as input to BLSTM. BLSTM not only has the ability to learn sequences forward from LSTM, but also benefits from having a reverse LSTM in its design structure, allowing it to learn sequence features from back to front. The second layer of the BLSTM network, BLSTM, serves as the encoder and generates an intermediate semantic vector as the input for the DCGAN decision maker the calculation of the a character in the BLSTM network, which combines forward and reverse transmission outputs, is as follows i_{th}:

$$h_i = [\overrightarrow{h_i} \oplus \overrightarrow{h_i}] \tag{15}$$

The intermediate semantic vectors $H = \{h_1 h_2 h_{l\_\max}\}$ and $H \in R^{l\_\max \times d}$ can also be seen as a matrix. $l_$max is the maximum frame length of the wireless network protocol, and d is the size of the embedded character in BLSTM. The $l_$max of different wireless network protocols is different, so the calculation steps of the discriminator are different. To simplify the operation here, make $d = l_$max obtain a square input size containing order information. For BLSTM network, we use the cross entropy function as the loss function, which is defined as:

$$H(pq) = -\sum_{x} p(x) \log q(x) \tag{16}$$

Among them, $p(x)$ is the true distribution of the sample, and $q(x)$ is the probability output by the model.

Dropout and L2 regularization terms are used in the above model to reduce the complexity of the model and eliminate the risk of overfitting as far as possible. Since the output of a BLSTM cell is binary, a BLSTM cell's loss function is obtained:

$$L_{BLSTM}^{<t>}(y^{<t>}\hat{y}^{<t>}\omega_1) = -\sum_{j=1}^{c} y_j^{<t>} \log \hat{y_j}^{<t>} + \lambda_1 \|\omega_1\|_2^2 \tag{17}$$

Where C is the size of the embedded character; y is a one hot vector of real characters; $\hat{y}$ is the probability of each class in the softmax layer; λ_1 is the weight of L2 regularization; ω_1 is the weight of the BLSTM layer and output layer. The loss function of BLSTM network to sequence S is:

$$L_{BLSTM}(y\hat{y}\omega_1) = \sum_{t=1}^{T_x} L_{BLSTM}^{<t>}(y^{<t>}\hat{y}^{<t>}\omega_1) \tag{18}$$

Among them, T_x represents the length of the input sequence.

Due to the BLSTM layer's ability to access forward and backward contexts, CNN is used to explore more meaningful information, such as the hierarchical structure of representations. The intermediate semantic vector adds positional features as input through the BLSTM network, and each filter can be regarded as a detector on CNN to detect whether the functional code positional features of the data frame are correct. This is beneficial for the model to quickly learn the format features of the wireless network protocol sequence data. Unlike generators, the discriminator uses batch normalization in all layers except for the input layer; Compared with ReLU, Leaky ReLU is used as the activation function in the model to avoid sample oscillation and model instability. In addition, unlike the generator using deconvolution layers instead of pooling layers, the discriminator uses step convolution layers to replace all pooling layers. In addition, we also use Z-Core to regularization H. Through a series of convolutional operations, we can obtain the output of the discriminator:

$$o' = \left[\frac{i + 2p - k}{s}\right] + 1 \tag{19}$$

The output value of the discriminator obtained at this point is the dynamic mining result of wireless network information transmission security vulnerabilities.

4 Experiment

4.1 Experimental Design

(1) Experimental environment

The specific environment for this experiment is as follows:
CPU: AMD® A8-7200P radeon r5@2.40 GHz

Operating system: CentOS Linux release 7
Running memory: 8 GB

(2) Experimental testing process

Leopard Mobile data set selected in this chapter is a binary code file, which is disassembled into assembly code by IDA pro tool; The CWE119 dataset is a C language program code file that needs to be compiled into assembly code format. Batch format the compiled assembly code through Python scripts, remove textual description information from the code, and retain function and instruction information. In the above dataset, each sample has a corresponding 0 or 1 label, with 0 indicating vulnerability and 1 indicating no vulnerability. The ratio of sample size for training, validation, and testing sets is 3:1:1. Before each training and testing, the order of the data is disrupted.

In the instruction level word embedding model training based on Word2vec, this chapter uses an assembly code corpus, with the first column containing assembly operation instructions and the remaining columns containing operation data. Regular expressions are used to match the numbers in the operation data. The input of the instruction level word embedding model based on Word2vec is assembly code, and the output is a two-dimensional tensor. The input of the convolutional neural network model proposed in this chapter is the two-dimensional tensor output of the word embedding layer mentioned above, and the output is the probability of the code being classified as a bad sample. It is often observed in the training process of convolutional neural network model that, with the increase of the epoch of the training round, the training and verification error of the binary code vulnerability detection model will increase after reaching the local minimum, that is, the binary code vulnerability detection model training consumes more time and the latest parameters are not retained when the training is terminated. Therefore, the experiment in this chapter adopts the early termination strategy, that is, when the error of the binary code vulnerability detection model in the verification set does not further reduce within the epochs specified in advance, the training algorithm will terminate.

4.2 Experimental Result

4.2.1 Wireless Network Information Transmission Security Vulnerability Dynamic Mining Recall Rate

In order to verify the dynamic mining effect of wireless network information transmission security vulnerabilities using the method proposed in this article, the reference [5] method, the reference [6] method, and the method proposed in this article were used to verify the recall rate of wireless network information transmission security vulnerability dynamic mining. The results are shown in Table 3.

According to Table 3, when the resource level is 100 GB, the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using reference [5] method is 76.0%, the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using reference [6] method is 79.2%, and the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using this method is 99.3%; When the resource amount is 500

Table 3. Dynamic Mining Recall Rate of Wireless Network Information Transmission Security Vulnerabilities

Resource quantity/GB	Dynamic Mining of Security Vulnerabilities in Wireless Network Information Transmission and Recall Rate/%		
	Reference [5] Method	Reference [6] Method	proposed method
100	76.0	79.2	99.3
200	79.1	68.3	99.2
300	60.2	79.5	95.6
400	68.8	80.8	96.8
500	79.2	63.6	96.3
600	66.8	68.5	98.1

GB, the recall rate of dynamic mining for wireless network information transmission security vulnerabilities in reference [5] method is 79.2%, the recall rate of dynamic mining for wireless network information transmission security vulnerabilities in reference [6] method is 63.6%, and the recall rate of dynamic mining for wireless network information transmission security vulnerabilities in this method is 96.3%; When the resource amount is 600 GB, the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using reference [5] method is 66.8%, the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using reference [6] method is 66.8%, and the recall rate for dynamic mining of wireless network information transmission security vulnerabilities using this method is 98.1%; The above results indicate that the method proposed in this paper can effectively improve the recall rate of dynamic mining for wireless network information transmission security vulnerabilities.

4.2.2 Dynamic Mining Accuracy of Wireless Network Information Transmission Security Vulnerabilities

In order to verify the efficiency of dynamic mining of wireless network information transmission security vulnerabilities using the method proposed in this paper, the methods of reference [5], reference [6], and the method proposed in this paper were used for English translation error recognition and time-consuming verification. The results are shown in Table 4.

According to Table 4, when the resource size is 100 GB, the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using reference [5] method is 58.3%, the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using reference [6] method is 55.8%, and the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using this method is 98.0%; When the English resource is 500 GB, the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using reference [5] method is 68.8%, the accuracy of dynamic mining for

Table 4. Dynamic Mining Accuracy of Security Vulnerabilities in Online Network Information Transmission

Resource quantity/GB	Dynamic Mining Accuracy of Wireless Network Information Transmission Security Vulnerability/%		
	Reference [5] Method	Reference [6] Method	proposed method
100	58.3	55.8	98.0
200	60.0	78.9	96.3
300	55.1	60.1	99.5
400	63.6	62.3	95.0
500	68.8	65.9	96.6
600	78.0	53.2	99.0

wireless network information transmission security vulnerabilities using reference [6] method is 65.9%, and the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using this method is 96.6%; When the English resource is 600 GB, the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using reference [5] method is 78.0%, the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using reference [6] method is 53.2%, and the accuracy of dynamic mining for wireless network information transmission security vulnerabilities using this method is 99.0%; The above results indicate that the method proposed in this paper can effectively improve the accuracy of dynamic mining of wireless network information transmission security vulnerabilities.

4.2.3 Dynamic Mining of Security Vulnerabilities in Wireless Network Data Transmission Takes Time

In order to verify the efficiency of dynamic mining and identification of wireless network information transmission security vulnerabilities using the method proposed in this paper, reference [4], reference [5], reference [6], and the method proposed in this paper were used to verify the time consumption of dynamic mining of wireless network information transmission security vulnerabilities. The results are shown in Table 5.

According to Table 5, when the resource level is 100 GB, the dynamic mining time for wireless network information transmission security vulnerabilities using reference [4] method is 15.8 s, the dynamic mining time for wireless network information transmission security vulnerabilities using reference [5] method is 11.9 s, the dynamic mining time for wireless network information transmission security vulnerabilities using reference [6] method is 32.9 s, and the dynamic mining time for wireless network information transmission security vulnerabilities using this method is 2.9 s; When the English resource is 600 GB, the dynamic mining time for wireless network information transmission security vulnerabilities using reference [4] method is 282.8 s, the dynamic mining time for wireless network information transmission security vulnerabilities using reference

Table 5. Time consumption for dynamic mining of wireless network information transmission security vulnerabilities

Resource quantity/GB	Dynamic mining of security vulnerabilities in wireless network data transmission takes time/s			
	Reference [4] Method	Reference [5] Method	Reference [6] Method	proposed method
100	15.8	11.9	32.9	2.9
200	48.9	28.9	58.9	5.8
300	82.6	86.2	99.6	12.0
400	198.6	99.1	109.2	18.9
500	252.3	136.9	136.1	22.3
600	282.8	188.7	148.3	25.8

[5] method is 188.7 s, the dynamic mining time for wireless network information transmission security vulnerabilities using reference [6] method is 148.3 s, and the dynamic mining time for wireless network information transmission security vulnerabilities using this method is 25.8 s; The above results indicate that the proposed method can effectively improve the efficiency of dynamic mining of wireless network information transmission security vulnerabilities.

5 Conclusion

The paper proposes a dynamic mining method for wireless network information transmission security vulnerabilities based on spatiotemporal dimensions. Collect wireless network information transmission security vulnerability data, introduce wavelet transform to filter and process wireless network information transmission security vulnerability data, and in the deep neural network architecture, command level Word embedding method based on Word2vec is used to obtain wireless network information transmission security vulnerability feature attributes, and realize dynamic mining of wireless network information transmission security vulnerabilities based on space-time dimensions. The experimental results indicate that the dynamic mining method based on spatiotemporal dimension can effectively discover security vulnerabilities in wireless network information transmission. By real-time monitoring and analysis of wireless network data flow, potential security threats and attack behaviors can be captured. The accuracy of vulnerability dynamic mining can reach 99.0%, and dynamic mining methods can timely detect security vulnerabilities in wireless network information transmission and provide corresponding warnings. By identifying abnormal behaviors and patterns, we can quickly respond and take corresponding security measures to avoid or reduce potential security risks. The dynamic vulnerability mining takes 25.8 s, indicating that our method can improve the effectiveness of vulnerability dynamic mining.

References

1. Liu, Y.: Research on wireless communication network data security situation awareness based on deep learning. Inf. Rec. Mater. **23**(08), 182–185 (2022)
2. Lv, G., Ju, L.: A method for identifying network security vulnerabilities in power systems based on data mining. Electrotech. J. (02), 49–51 (2023)
3. Ding, J.: Research on software security vulnerability automatic mining method based on big data technology. J. Taiyuan Norm. Univ. (Nat. Sci. Ed.) **21**(01), 45–50 (2022)
4. Yin, Y.: Research on network protocol vulnerability mining technology based on fuzzy. Microcomput. Appl. **37**(09), 8–10+16 (2021)
5. Zhang, M.: Analysis of network computer security hazards and vulnerability mining technology. Wirel. Internet Technol. **19**(10), 13–15 (2022)
6. Guan, J., Shi, G., Chen, H.: Research on network vulnerability mining based on Apriori risk data analysis. Comput. Simul. **39**(01), 343–347 (2022)
7. Gu, M., et al.: Software security vulnerability mining based on deep learning. Comput. Res. Dev. **58**(10), 2140–2162 (2021)
8. Li, M., Zhu, M.: A security vulnerability detection method for wireless communication networks based on ant colony algorithm. Comput. Meas. Control **30**(10), 51–56 (2022)
9. Wang, X., Wang, C., Li, Q., Ren, T.: A method for mining network security vulnerabilities in power systems based on black box genetic algorithm. J. Shenyang Univ. Technol. **43**(05), 500–504 (2021)
10. Jiang, Z., Fan, L.: Intelligent monitoring system for wireless communication network security vulnerability based on machine learning. Electron. Des. Eng. **29**(15), 115–119 (2021)