Automatic Detection Method of Website Vulnerabilities Based on an Associated Data Drive

Xiaoli Li*, Ling Zhao, Haobin Shen, Hanlin Du and Zhida Guo

Huizhou Power Supply Bureau of Guangdong Power Grid Co., Ltd., Huizhou 516000, China
E-mail: lixiaoli223dd@163.com
**Corresponding Author*

Received 29 July 2024; Accepted 21 January 2025

Abstract

In order to reduce the probability of website users being attacked and maintain the safety of website operation, this study proposes an automatic vulnerability detection method of websites based on associated data. We use plug-ins to scan the website in all directions, establish a scanning database, and classify and store the scanned web data. By applying optimized an a priori association rule algorithm, key features are extracted from web scan data, which are then transformed into input samples for a K-means clustering algorithm. The aim is to efficiently extract feature attributes of website vulnerability data and ultimately construct a text vectorized representation of vulnerability data. Convolutional neural networks can automatically detect website vulnerabilities by using the constructed text vector as input. Experimental verification shows that this method demonstrates comprehensive data coverage, efficient processing speed, and high-precision recognition performance. It not only significantly reduces the clustering analysis time, but also ensures the accuracy and timeliness of vulnerability detection.

Keywords: A priori algorithm, website, website vulnerability, automated detection, clustering algorithm, convolutional neural network.

Journal of Web Engineering, Vol. 24_2, 217–242.
doi: 10.13052/jwe1540-9589.2423

1 Introduction

With the rapid progress of Internet technology, web applications [1, 2] have become the core component of modern social life and work, and their importance is increasingly prominent and irreplaceable. However, web applications face the risk of being exploited by hackers, with common methods including cross site scripting (XSS), SQL injection, and other attack methods, which have become the focus of security threats. Due to the uneven level and experience of developers, some developers have not paid attention to the user's input data or the information carried on the page (such as Cookies [3]), legal judgment has become necessary, which leads to website loopholes. Due to the increasing network security threats and frequent user data leakage incidents, in order to protect users' privacy and sensitive information, related vulnerability detection methods have attracted the extensive attention of scholars.

Li et al. proposed a new vulnerability detection method for complex path environments [4], which combines two major techniques: guided fuzzy testing and selective symbol execution. Among them, guiding fuzzy testing is responsible for filtering the critical execution paths of the program, while selective symbolic execution conducts in-depth analysis of these potential vulnerability paths, effectively improving the efficiency and accuracy of vulnerability detection. Firstly, the technology obtains program vulnerability information through static analysis; then, using the oriented fuzzy testing technology, the test cases that can cover the vulnerability function can be generated quickly. Finally, the path that can trigger the vulnerability in the vulnerability function is executed symbolically, and the test case that triggers the vulnerability is generated, so as to detect the vulnerability. Researchers such as Ma have developed an innovative web access control vulnerability detection technique [5], which adopts a state deviation analysis framework and combines it with a white box testing strategy. By deeply parsing the code, this technology can extract access control constraints and build an expected access model for web applications. Subsequently, dynamic analysis techniques are used to capture actual access behavior during operation, and the two are compared to identify state deviations, thereby accurately locating access control vulnerabilities. Based on this technology, they successfully developed the prototype tool ACVD, which demonstrates excellent capabilities in identifying unauthorized access and various other access control defects, effectively improving the accuracy and efficiency of vulnerability detection. Web vulnerability detection based on stain analysis and symbol

execution [6] studied by Liu et al., proposes a finer-grained stain analysis method for web vulnerabilities and generates more accurate object state records in the process of code analysis, judges the execution path of the code, accurately records the transmission process of the stain and sink, and verifies the accessibility of the vulnerability position by using the symbol execution tool, thus completing the detection of the vulnerability. Researchers such as Wen [7] explored the application of graph convolutional networks in the field of source code vulnerability detection, which is an innovative detection method. Firstly, the detection method converts the program source code into CPG containing syntax and semantic feature information; Wen et al. used RGCN to construct a graph structure representation of the source code and trained a neural network model to predict vulnerabilities in the code. On the other hand, Gong et al.'s [8] research is based on the BiLSTM model, which innovatively extracts method bodies from source code to form a set, constructs ASTs for each method to extract statement sets, anonymizes variable names, and assigns unique node numbers to construct node sets. Subsequently, through data flow and control flow analysis, the dependency relationships between nodes are revealed, forming feature representations and transforming them into feature matrices. The matrix is labeled based on the existence of vulnerabilities and becomes a training sample. To improve the model's understanding of sequence context, BiLSTM was selected and incorporated into the attention layer to enhance model performance and achieve efficient vulnerability detection. Although the above studies have achieved certain results in the field of vulnerability detection, they still face multiple challenges. Specifically, these studies have limitations in the coverage range of vulnerabilities detected, the detection process is time-consuming, and the parameter tuning of neural network models is not yet ideal. Especially crucial is that their detection accuracy still needs to be improved to cope with increasingly complex network security threats.

When there are loopholes in the website [9, 10], attack hackers can attack web applications by exploiting loopholes in websites, and hackers can implant trojans by means of loopholes in web pages, thus gaining some important data and benefits. Based on this, this study proposes an automatic detection method of website vulnerabilities based on associated data, which effectively combines associated data and convolutional neural network to diagnose website vulnerabilities, and verifies the effectiveness of the method through experiments.

2 Website Vulnerability Automatic Detection

Using associated data to drive automatic detection of website vulnerabilities refers to the process of promoting, perfecting, and improving website vulnerability detection by using relevant data information and relevance. By analyzing the relationship between different data, we can get a more comprehensive understanding of website vulnerabilities, thus improving the accuracy, efficiency and comprehensiveness of vulnerability detection. Specifically, the application of an associated data driver in website vulnerability detection includes data integration and association, and vulnerability association analysis. Therefore, in the research process, this paper uses a priori association rules to mine vulnerability data after website data scanning. It uses a K-means clustering algorithm to extract features from vulnerability data and then uses these features as inputs to convolutional neural networks to achieve an automated vulnerability detection process.

2.1 Website Data Scanning

Data acquisition plays a fundamental role as the primary step in automatically detecting website data vulnerabilities. This process extensively covers network ports [11], databases [12], web interfaces, security baselines, weak password detection, and comprehensive website scanning through network scanning technology. Given the diversity and complexity of scanning objects, we innovatively adopt plugin technology to efficiently collect data. These plugins are written in C language, seamlessly integrate with network interfaces, and customize exclusive plugins for various scanning objects, to ensure the comprehensiveness and flexibility of data collection. A scan plugin overview is shown in Table 1.

Table 1 Overview of scanning plugins

Scanning Plug-in ID	Category	Name	Description
001	Port scanning	Port	Service analysis
002	Database dictionary scan	Dict	Oracle, MySQl, DB2, PostgreSQL scan
003	Web scan	HTTP	Web services scanning
004	Safety baseline scanning	SNMP	Third party access security analysis
005	Weak password scanning	General	Weak password scanning exists in the system
006	Website scanning	Overflow	Website security analysis

Table 1 clearly shows the unique plugin architecture customized for each website. Each plugin encapsulates a specialized scanning function to ensure clarity and no confusion during the scanning process. These plugins exist in the form of Windows Dynamic Link Libraries (DLLs), each equipped with a unique identifier (ID), and each scanning function is called through carefully designed unique function names. The function naming follows the prefix rule of "_", combined with the function description, serial number, and plugin ID to form a standardized format such as "_XXX * * 00" to enhance the readability and maintainability of the code.

When the network remains active, it performs network scanning tasks one by one based on the plugin sequence listed in Table 1.

(1) Perform a network port scan as a preliminary exploration of potential attack paths, and use plugin number 001 to comprehensively search for network open ports,
(2) Use the 002 plugin to conduct in-depth scanning of the database dictionary, focusing on identifying and parsing system tables and fields.
(3) During the software construction phase, programming vulnerabilities such as SQL injection [13], cross site scripting, and covert links pose a security risk to the OWASP Top 10. The use of the 003 plugin to detect these threats aims to prevent website data tampering, information leakage, and the risk of servers being illegally controlled.
(4) Security configuration verification is the cornerstone of security management and a key technology to ensure stable operation. This process begins with building a security configuration baseline that complies with the organization's information security policy. By deploying the 004 plugin, specialized configuration audits can be carried out on key system products such as Oracle and WebLogic.
(5) The 005 plugin focuses on weak password detection, covering protocols such as TELNET, FTP, SSH, POP3, SMB, SNMP, RDP, SMTP, REDIS, and Oracle. It enhances detection capabilities with default and custom dictionary libraries. The scan results are integrated into the website database and stored through customized file names and classification, providing an efficient foundation for subsequent feature analysis.

2.2 Improved A Priori Association Rules to Extract Features from Website Data

The website data obtained through scanning includes two categories: vulnerability free and vulnerability containing. The a priori algorithm, with

its concise operation process and intuitiveness, is good at revealing the association rules between data [14], and can operate effectively even under lower data quality requirements. Based on the website data samples obtained from the aforementioned scanning, the optimized a priori algorithm is used to deeply explore the association rules between vulnerability features. The mining basis for strong association rules [15] is that the support must exceed the set minimum threshold, and the confidence must also meet the minimum required threshold. These strong association rules constitute the database of web site vulnerability data mining.

In the process of a priori algorithm mining vulnerability data feature quantity of web site, the definition defined m collection of different vulnerability feature quantity items is I. $I = \{i_1, i_2, i_3, \ldots, i_m\}$, the project is one of them i_k, and the itemset is a collection of items. Define all transaction sets of vulnerability database of website as data set D, a set of non-empty items for a transaction use T to describe, that is, transaction T, multiple project groups constitute a non-empty itemset, that is, a transaction. Transaction Ts two itemsets are X and Y, existing as $X \subseteq T$, $Y \subseteq T$; if X and Y are not empty sets, and at the same time $X \cap Y = \emptyset$, then $X \to Y$ can form a transaction set D association rule.

Support and confidence are the two key variables to measure the strength of the association rule transaction data set D of the percentage of the number of transactions and the total number of the internal coexistence X and Y in the itemset is the support degree, which is used as $\mathrm{support}(X \Rightarrow Y)$ here to express. Transaction data set D internal coexistence X and Y percentage value of the number of transactions and the number of included transactions in the itemset is the confidence level, which is used $\mathrm{confidence}(X \Rightarrow Y)$ to express. Formula (1) and formula (2) are the calculation methods of support and confidence respectively.

$$\mathrm{support}(X \Rightarrow Y) = P(X \cup Y) \tag{1}$$

$$\mathrm{confidence}(X \Rightarrow Y) = P(X \cup Y) \tag{2}$$

To mine strong association rules between website vulnerability data features, it is necessary to ensure that the preset minimum support and confidence thresholds are met. The former is the threshold for evaluating the support and the frequency of itemsets, and the lowest frequency of itemsets is evaluated by using the minimum support, and is evaluated by $\min\_\sup$; the value is between 0 and 1. The lowest reliability of association rules can pass the minimum confidence min_conf to evaluate; the minimum confidence

value is between 0 and 1. The strong association rules obtained from feature mining of website vulnerability data must meet the pre-set minimum support and confidence conditions.

The a priori algorithm generates a large candidate set when extracting website vulnerability features, requiring frequent database scans to calculate support and confidence. The a priori algorithm faces the challenges of high memory usage and long computation time when extracting vulnerability features. To this end, we optimized the algorithm from two dimensions: reducing memory consumption and reducing time complexity.

(1) Interest constraint. The association rule filtering mechanism based on support and confidence has shortcomings: low support data is mistakenly filtered, and confidence calculation does not consider the support of subsequent project sets, resulting in incomplete and unattractive website vulnerability association rules. An interest-assisted a priori algorithm is introduced to construct association rules, and valuable frequent itemsets are deeply filtered based on interest threshold after obtaining frequent itemsets initially. In the rules $X \rightarrow Y$, itemset X and itemset Y's degree of correlation between them is the degree of interest, and the threshold of interest is defined as 1, and formula (3) is itemset X and itemset Y's calculation method of interest.

$$\text{Interest}(X \Rightarrow Y) = \frac{P(XY)}{P(X)P(Y)} \tag{3}$$

Among them, when the item set interest degree is greater than 1, X itemset and Y itemset positively correlate; when the item set interest is less than 1, X itemset and Y itemset negatively correlate. The former represents that if X happens, it will increase Y's probability of occurrence, the latter represents if X happens, it will decrease Y's probability of occurrence. An interest value of exactly 1 indicates that there is no correlation between the two itemsets.

Define the input data set as D, the a priori algorithm based on interest degree is implemented as follows:

Step 1 Input the thresholds of support, confidence threshold and interest.
Step 2 Scan the database D to get all the data that have appeared and regard it as a candidate for frequent 1-itemset. At this moment, $k = 1$, then the frequent 0 itemset is an empty set.
Step 3 Frequent mining k item set.
Step 4 Settings $k = k + 1$, execute step 3.

Step 5 Based on frequent itemsets, rules for extracting the feature quantity of vulnerability data of web sites are constructed, and these rules meet the preset confidence threshold and interest threshold. Finally, the extraction rules of vulnerability data features of web sites that meet the standards are output.

Interest is used in association rule mining to eliminate redundant candidates, reduce unnecessary comparisons, and ensure that the ultimately determined association rules meet both confidence criteria and practical significance.

(2) Support adaptive updating strategy. Longer data often have more feature quantities, so there is a greater possibility of missing data. The traditional a priori algorithm sets the support by a fixed value, and its disadvantage is that frequent itemsets vary with the number of items k. It is difficult to obtain longer scanning data. In order to avoid this drawback, an adaptive updating mechanism of support is proposed, and the support is set to decrease adaptively with the increase in data length, so as to improve the sensitivity of selecting vulnerability data. Improve the a priori algorithm by setting the initial support to $V_0 = \sup$. By adjusting the parameter ε to increase the term and reduce the cost, the number of iterations is calculated based on the k value and the quadratic support formula (4).

$$V_k = \sup * \varepsilon * \exp(-\text{Interest}(X \Rightarrow Y)) \tag{4}$$

As the iteration deepens, the support decreases and gradually stabilizes, effectively retaining low-frequency key information.

2.3 Feature Extraction of Website Vulnerability Under a K-means Clustering Algorithm

The a priori algorithm mainly focuses on frequent itemsets and association rules between transaction data, but it is not comprehensive and accurate for data analysis of website vulnerabilities. Using the K-means clustering algorithm [16–18] to mine website vulnerability data, the classification process divides non fragile and fragile data into main clusters and multiple small clusters, with significant differences in features. The feature information extracted by the a priori algorithm is used as input for K-means clustering for more precise classification. The K-means clustering process is briefly described below.

First, randomly select several initial cluster centers as starting points.

Second, calculate the distance between the web site data mined by each a priori association rule algorithm and the central point, and divide the web site data types. The process is as follows:

$$j_{1,2} = \frac{\text{support}(X \Rightarrow Y)}{\text{confidence}(X \Rightarrow Y)} \sqrt{\sum_{k=1}^{n} V_k (t_{1k} - t_{2k})^2} \tag{5}$$

$$S_i = \arg\min \left| \frac{j_{1,2}}{\mu} \right| \tag{6}$$

where $j_{1,2}$ is the distance measurement between data; t is the eigenvalue, n is the total number of features; k is the number of parameters; S_i is the classification set of web network vulnerability data; and μ is the classification set of web network vulnerability data.

Finally, through the continuous circulation of the above steps, until the calculation result of formula (6) meets the fixed value requirements, a key cluster and a plurality of relatively dispersed clusters are determined. Core clustering, as the focus of mining, has a higher likelihood of becoming vulnerability data as the distance between data points and the core increases; the smaller the distance, the lower the likelihood, thus achieving effective classification of website vulnerability data.

Based on the web network vulnerability data obtained by classification S_i, the vulnerability data features of websites are extracted according to the process shown in Figure 1. The first step is to extract the effective attributes of vulnerability data. Subsequently, segment the text description and remove stop words to reduce redundancy. Step 3, use the enhanced entropy based S-C algorithm to extract vulnerability feature words. Finally, constructing vulnerability word vectors based on feature sets lays the foundation for automatic website vulnerability detection.

Text feature extraction aims to select keywords that efficiently represent vulnerability information from vulnerability descriptions based on evaluation criteria. Given that the analyzed vulnerability data is decomposed into a large vocabulary set, direct vectorization will generate high-dimensional vectors, which seriously affects detection efficiency and accuracy. Therefore, it is necessary to refine feature words from the text and retain vocabulary with high classification contribution as sample features. This move aims to reduce the dimensionality of feature vectors, optimize resource utilization, and improve the performance of classifiers. Based on the analysis of vulnerability text characteristics, we constructed a comprehensive function C to quantify the

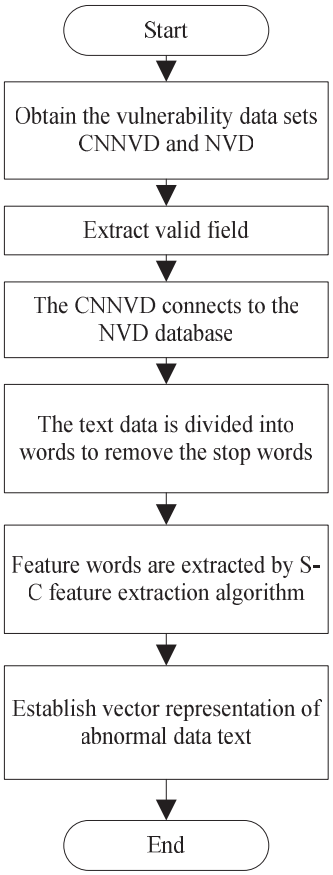


Figure 1 Data preprocessing process.

importance of vocabulary in classification, and combined it with information entropy S, ultimately designing the S-C algorithm to accurately extract the dataset feature set.

The importance of a word to a category is determined in two ways. This reflects the representativeness of vocabulary in specific categories. If a word appears widely and evenly in documents of that category, it is more effective in representing that category. However, if a word appears frequently in most other categories, it cannot represent this kind of anomaly well. On the other hand, it is the importance of words between classes, if the frequency of words in this class is greater than that in all samples. The frequency of occurrence shows that this word can better represent this kind of anomaly, and

a comprehensive function is established through these two aspects C. The set f_t gathers the frequency of the occurrence of word t in different categories. For a specific category i, f_{it} represents the specific frequency of occurrence of word t in that category. It is defined as follows:

Definition 1 (within-class importance *I*): If words t in i's class vulnerability data frequently appears and is evenly distributed on each vulnerability, which means that words and expressions t make i's higher importance within the class, the more important the vulnerability category is. It is more important for the judgment of the vulnerability category i. The formula is as follows:

$$I_{it} = \frac{d_{it}}{S_i/D_i} \tag{7}$$

Among them, D_i represents a category i's number of vulnerability samples and d_{it} represents a category i's number of samples of vulnerabilities of words t.

Definition 2 (inter class importance N): If the frequency of word t distribution in Class i vulnerabilities is greater than the frequency of word t distribution in the total sample, then word t is considered to have more important significance for Class i than other categories. The formula is as follows:

$$E_{it} = \frac{p_{it}}{p_t} \tag{8}$$

Among them, $p_{it} = \frac{f_{it}}{D_i}$ represents that in the category i's average distribution frequency of words t, $p_t = \frac{f_t}{S_i}$ represents the average distribution frequency of word t in the total sample.

Definition 3: Synthesis function C represents the importance of word t for category i, which is obtained by multiplying the importance between classes and the importance within classes. The formula is as follows:

$$C_{it} = E_{it} \times I_{it} \tag{9}$$

Definition 4: Information entropy, as a core concept in information theory, is a ruler for measuring the amount of information. The logic is that the degree of order within the system is inversely proportional to information entropy, that is, the more orderly the system is, the lower the entropy value. On the other hand, as the degree of system chaos increases, information entropy also increases. Therefore, information entropy is not only used to

quantify the amount of information, but also becomes an important indicator for evaluating the ordered state of a system, and its calculation formula can accurately reflect this relationship. The formula is as follows:

$$H = -\sum_{i=1}^{n} \ln p_i \tag{10}$$

Definition 5: Information entropy t of feature word $S(t)$) represents the degree of confusion in the category to which feature word t belongs. The larger $S(t)$, the more difficult it is for the feature word t to distinguish categories.

$$S(t) = -\sum_{i=1}^{n} \frac{C_{it}}{H} \tag{11}$$

Definition 6: The S-C value of a feature word represents the importance value of feature word t for category i calculated using the S-C algorithm. The final web site vulnerability data feature is:

$$Q = S(t) - C_{it} \tag{12}$$

2.4 Automatic Detection of Vulnerabilities Based on Convolutional Neural Networks

In the vulnerability detection of websites, it is often necessary to consider time series and the time–space relationship. A convolutional neural network [19, 20] has the ability to process spatio-temporal information, and can use the sliding window operation of the convolution layer to capture the spatio-temporal relationship between the development process of vulnerabilities and other related events, so as to better identify and analyze vulnerabilities. Therefore, the vulnerability feature data extracted by clustering is used as input, and a convolutional neural network model is introduced. The dynamic window mechanism of convolutional layers is used to capture the spatio-temporal correlation between vulnerability evolution and related events, thereby optimizing the identification and analysis process of vulnerabilities and achieving automated website vulnerability detection based on multi-dimensional correlation data. The specific process is shown in Figure 2.

(1) The 400 $\times$ 100 matrix is extracted using the Q-layer of a convolutional neural network. Each row represents a word in the vulnerability description (a total of 400 words), and each column represents a dimension of the word

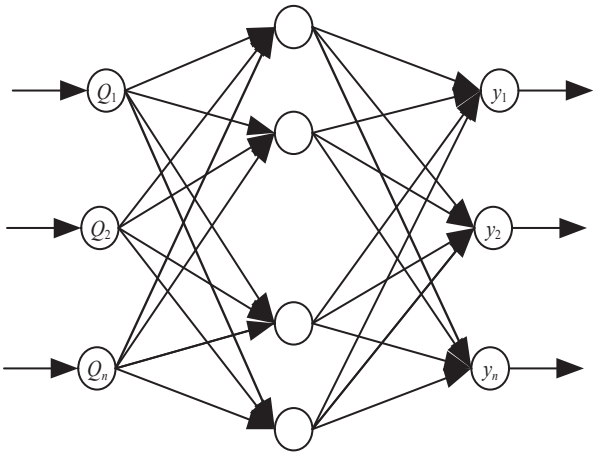


Figure 2 Automatic vulnerability detection based on a convolutional neural network.

vector (a total of 100 dimensions) as the input for the vulnerability feature. The input matrix is convolved with convolution kernel to obtain several characteristic graphs with a column number of 1. Because there are fewer words in each sentence in the vulnerability description, and the convolution kernel with different sizes can better extract the features in the text information, the convolution kernel window is set to 3, 4 and 5 sizes, and the step size is 1.

(2) The pooling layer samples the one-dimensional feature map output in the convolutional neural network layer and extracts the maximum value. The final pooling layer generates a one-dimensional vector output by taking the maximum value of each feature map, which effectively weakens the noise impact that may be introduced by zero padding in the text matrix and improves the robustness of the features.

(3) In the last layer, the text feature vectors obtained after pooling are fully connected, and then sent to the Softmax classifier for distinguishing and detecting website vulnerabilities.

3 Experimental Analysis

To verify the accuracy of the proposed vulnerability automatic detection algorithm, this study selected a website of a normal university in L province as an experimental case, which includes functions such as school introduction, student course selection, and score inquiry. Implementing this

Table 2 Data scanning results

Scanning Plug-in ID	Name	Data Size	Scanning Time	Number of Scans
001	Port	271G	20s	3
002	Dict	2.1T	100s	4
003	HTTP	26G	10s	5
004	SNMP	1.3G	3s	4
005	General	11G	6s	2
006	Overflow	3.6T	130s	6

method automatically detected website vulnerabilities and evaluated their effectiveness in practical applications.

3.1 Experimental Setup

(1) Data set. The school website was scanned to obtain website data, which were classified and stored according to different scanning positions. The data scanning results are shown in Table 2.

From Table 2, we can see that the number of scans for plug-ins is large, the amount of scanned data is large, and the scanning time is short, which proves that this is an efficient data scanning method.

(2) Algorithm parameter setting. In the automatic detection of web site vulnerabilities based on associated data, the main methods of K-means clustering and a convolutional neural network have the following parameters:

Using K-means clustering algorithm, set the number of clusters to 3 and limit the maximum number of iterations to 100.

The convolutional neural network configuration includes: convolutional layers with three 3×3 convolutional kernels; the pooling layer adopts 2×2 maximum pooling; two neurons are configured in the fully connected layer; the learning rate is set to 0.001; the batch processing size is 32.

3.2 Analysis of Results

Implement feature extraction for scanned data and rely on key rules to deeply mine vulnerability information. The mining results are shown intuitively in Figure 3.

Figure 3 clearly shows that the classification method based on data association rule mining can efficiently distinguish between normal and abnormal data, demonstrating significant classification effects and ensuring clear division between the two.

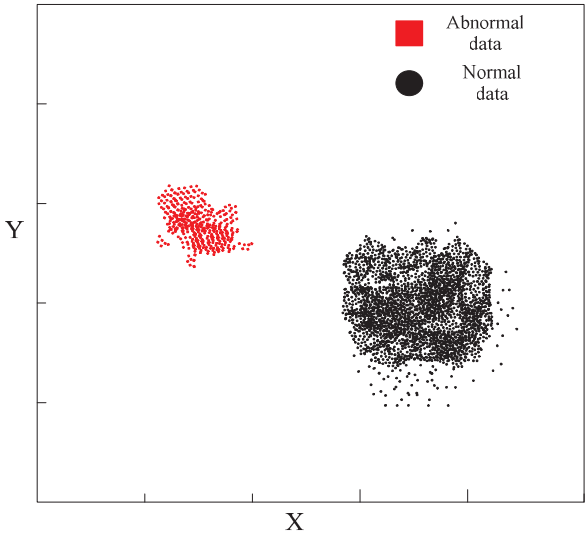


Figure 3 Data mining classification.

Based on the vulnerability dataset extracted from the above anomaly classification, accuracy, recall rate, and F1 comprehensive index are used as evaluation indicators to comprehensively measure the performance of the vulnerability detection method. The definition formulas for each indicator are detailed below.

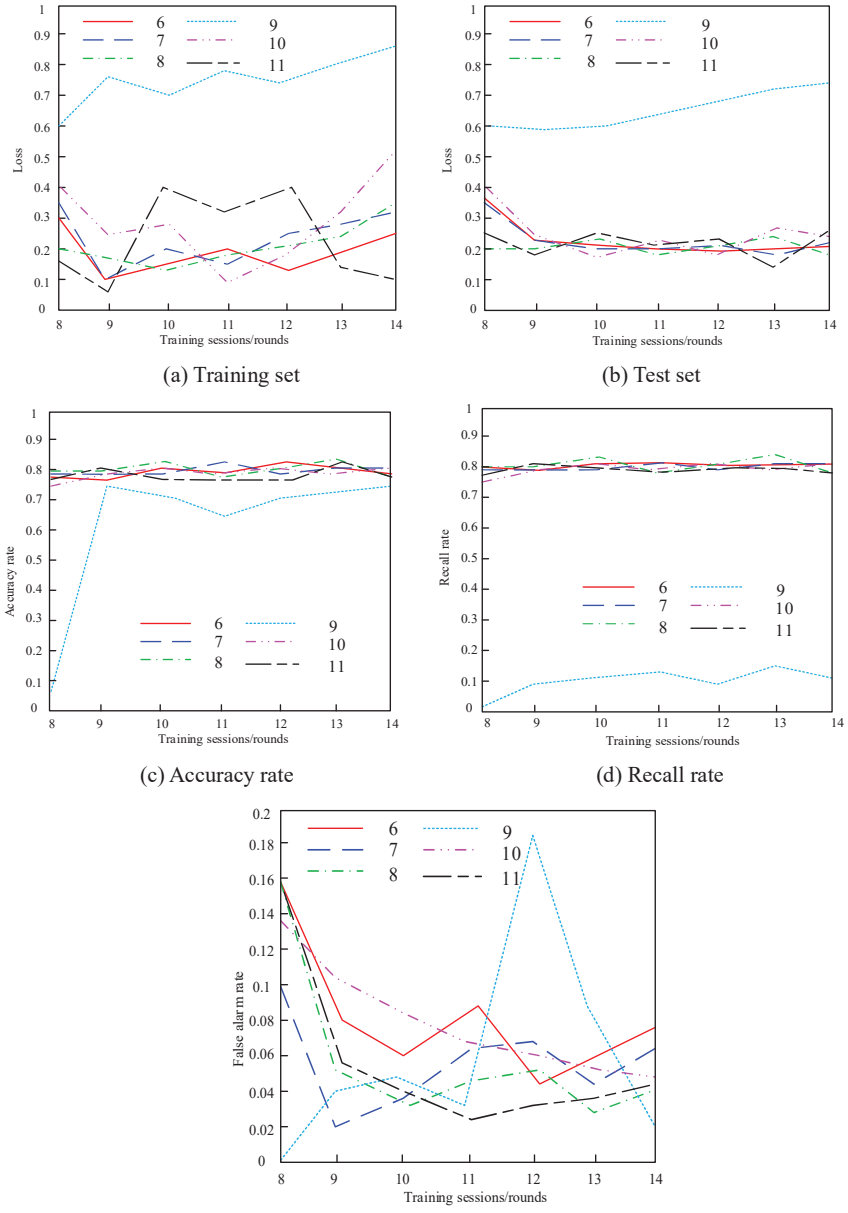
$$\text{precision} = \frac{T_p}{T_p + F_p} \tag{13}$$

$$\text{recall} = \frac{T_p}{T_p + F_N} \tag{14}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{15}$$

In the given formula, T_p represents the true positive example, which is the number of correctly identified vulnerabilities; F_p represents a false positive example, which is the number of vulnerabilities detected incorrectly; and F_N represents false negative examples, referring to the actual number of vulnerabilities that have not been correctly detected.

As the convolution kernel size is set to 2 and the network hierarchy gradually deepens, Figure 4 shows the performance indicators of this method in vulnerability detection.



(a) Training set

(b) Test set

(c) Accuracy rate

(d) Recall rate

(e) False alarm rate

Figure 4 Data when the volume core size is 2.

Table 3 Receptive field sizes of different convolution kernel sizes and network layers

Network Layer Number Convolution Kernel/Layer	2	4	6
6	135	247	382
7	268	544	776
8	543	1012	1554
9	1090	2027	3117
10	2189	4045	6301
11	4302	8316	12086
12	4300	8216	10332

Observing the curve trend in Figure 4, it can be clearly observed that in the training and testing sets, when the number of network layers reaches 9, the loss value reaches its peak. When the number of layers in the training set network increases to 11, the loss significantly decreases to the lowest point. In contrast, in the test set, except for specific layers, the losses caused by other network layers are relatively close and stable. In addition, analysis suggests that excessively long training cycles may lead to overfitting of the model to the training data. From the comparison of curves in Figures 4(c) and 4(d), it can be observed that when the convolution kernel size is fixed and the network has 9 layers, the accuracy and recall of vulnerability detection both show a poor stationary trend. As the number of training sessions increases, these two key indicators significantly decline. Furthermore, the change in false alarm rate shown in Figure 4(e) reveals that only when the network has 9 layers, the false alarm rate significantly improves, while the remaining layers have little impact on the algorithm performance. However, excessive training time may still lead to overfitting issues. Taking all factors into consideration, an 11 layer network is chosen as the optimal configuration.

After determining the number of layers in the network, the next task is to choose the size of the convolution kernel, which should be based on considerations of the receptive field. Specifically, the size of the convolutional kernel needs to match the expected receptive field range. For details, please refer to the corresponding relationship between different convolutional kernel sizes and the receptive field of the network layer in Table 3.

Table 3 shows that for a fixed convolution kernel size, as the number of network layers increases, the receptive field gradually expands, but decreases by the 12th layer, which may be attributed to overfitting. At the same network level, the receptive field area significantly increases with the increase of convolution kernel size, and the growth rate slows down with the increase in convolution kernel size. Specifically, in an 11 layer network, when the

Table 4 Vulnerability detection of the method in this paper

Vulnerability Tag	Attack Time/Min	Vulnerability Detection Result	Number of Attacks
001	2	SQL injection vulnerability	7
002	5	Weak password vulnerability	6
003	6	Directory listing vulnerability	3
004	3	Cross-site scripting vulnerability	2
005	7	Sensitive files and directories leak vulnerabilities	10
006	5	CMS vulnerability	5

convolutional kernel increases from 2 to 4, the receptive field increases by 48%, while when it increases from 4 to 6, the increase decreases to 31%. Based on the above analysis, a convolution kernel size of 4 was ultimately selected.

According to Figure 4 and Table 3, the convolution kernel size is 4. A convolutional neural network with 11 network layers has the best detection effect on vulnerabilities, so they are used as the setting parameters of convolutional neural network.

In order to demonstrate the effectiveness of our method in practical scenarios, we utilized the optimal convolutional neural network configuration obtained from the previous experiments to conduct automated vulnerability detection on the school's official website. The relevant detection results have been summarized in Table 4.

As can be seen from Table 4, using this method to detect vulnerabilities in the official website, we found six kinds of serious vulnerabilities and were attacked at different times and different times. Among them, the maximum number of attacks on sensitive files and directory leaks was 7 minutes and 10 times respectively, followed by directory list vulnerabilities being attacked for 6 minutes and SQL injection vulnerabilities being attacked for 7 times, which is consistent with the records, so we can see the accuracy of this method for vulnerability detection.

We adopted the method proposed in this article and compared it with various methods such as SymFuzzy (from reference [4]), web access control vulnerability detection based on state bias (reference [5]), web vulnerability detection combining stain analysis and symbol execution (reference [6]), network vulnerability detection based on graph convolution (reference [7]), and vulnerability detection using BiLSTM model (reference [8]). For detecting 11 randomly selected vulnerabilities in the CNNVD and NVD vulnerability data sets, the detection results and detection consumption time are shown in Figures 5 and 6.

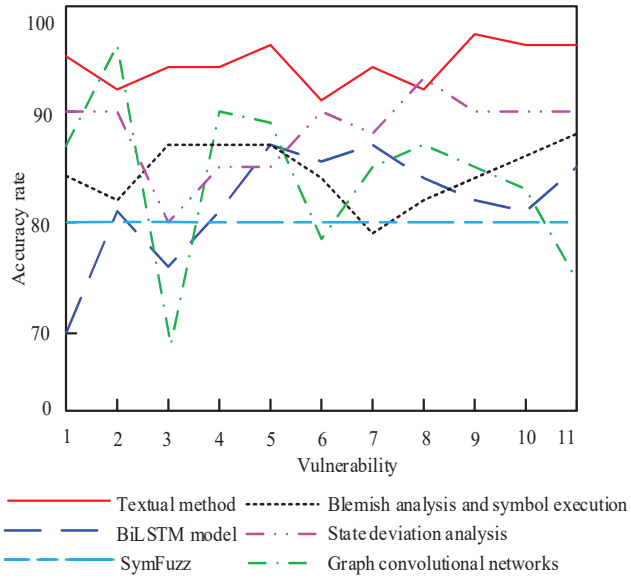


Figure 5 Website vulnerability detection accuracy.

Figure 5 visually illustrates the comparison of various methods in terms of vulnerability detection accuracy. Although the graph convolutional network method and the state deviation analysis method have shown higher accuracy for specific vulnerabilities (such as vulnerability 2), and the latter performs better in detecting vulnerability 8, our method has a detection accuracy of over 90% in detecting 11 types of vulnerabilities, and its average detection accuracy is significantly higher than other comparison methods. The detection accuracy of the network vulnerability detection method based on graph convolution is even lower than 70% for vulnerability 3, and the detection accuracy of the BiLSTM model vulnerability detection method is close to 70 for vulnerability 1 and vulnerability 11, and the detection accuracy of other methods is also uncertain.

As can be seen from Figure 6, although the detection time of this method is longer than the stain analysis and symbol execution method, the relationship graph convolution network method, the state deviation analysis method and the SymFuzz method among the four vulnerabilities 3, 5, 7 and 9, the average detection time of 11 vulnerabilities in this paper is shorter than other methods, and its advantages in detecting vulnerability 10 are far greater than other methods. When exploring web access control vulnerability detection, the method based on state deviation analysis showed a detection time of

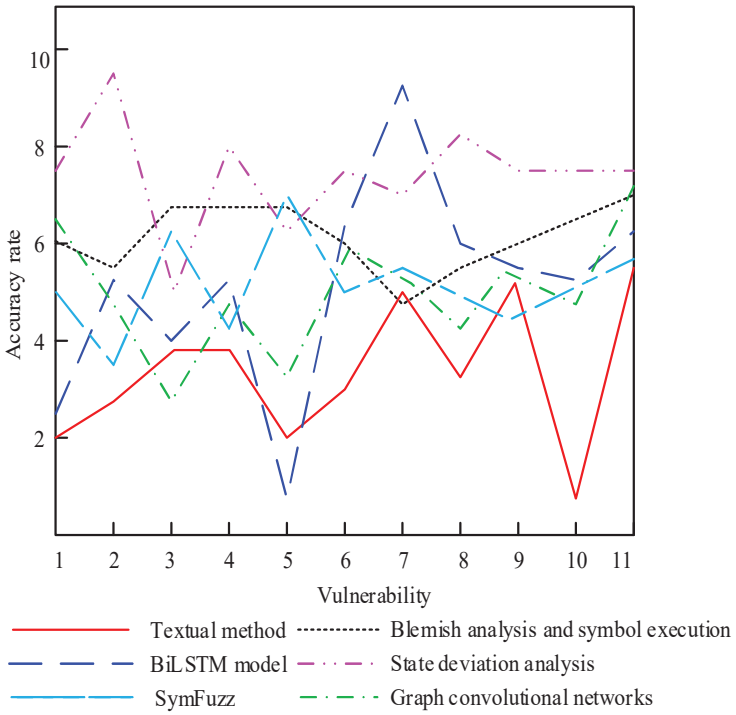


Figure 6 Website vulnerability detection time.

nearly 10 seconds when dealing with specific vulnerability 2, while using the BiLSTM model to handle another type of vulnerability 7, the detection time exceeded 9 seconds. The detection time of other methods for vulnerability is much longer than this method.

In order to further verify the universality of the method proposed in this article, the accuracy, false positive rate, false negative rate, and detection time of this method in detecting web vulnerabilities were evaluated by comparing web websites in different fields, such as e-commerce, finance, education, and healthcare, in order to demonstrate its feasibility and advantages in practical applications. The experimental results of multiple datasets using six methods are shown in Table 5.

According to Table 5, the method proposed in this paper has demonstrated excellent performance in automated vulnerability detection of websites in multiple fields, with generally high detection accuracy, especially in the medical field, reaching up to 95%. At the same time, the false positive rate and false negative rate remain at a low level, with a false positive rate of

Table 5 Analysis of algorithm performance on different datasets

State Deviation Analysis Field	Method	Detection Accuracy/%	False Alarm Rate/%	Leakage Rate/%	Detection Time/ms
Electronic	Textual method	92	3	5	120
commerce	SymFuzz	85	6	9	180
	State deviation analysis	80	8	12	240
	Blemish analysis and symbol execution	88	5	7	300
	Graph convolutional networks	90	4	6	150
	BiLSTM model	87	7	8	200
Finance	Textual method	94	2	4	130
	SymFuzz	88	5	7	190
	State deviation analysis	82	9	11	250
	Blemish analysis and symbol execution	90	4	6	320
	Graph convolutional networks	92	3	5	160
	BiLSTM model	89	6	7	210
Education	Textual method	90	4	6	110
	SymFuzz	84	7	9	170
	State deviation analysis	78	10	14	230
	Blemish analysis and symbol execution	86	6	8	290
	Graph convolutional networks	88	5	7	140
	BiLSTM model	85	8	9	190
Medical care	Textual method	95	1	3	140
	SymFuzz	90	3	5	200
	State deviation analysis	85	7	10	260
	Blemish analysis and symbol execution	92	2	4	330
	Graph convolutional networks	94	2	4	170
	BiLSTM model	91	4	5	220

only 1% and a false negative rate of 3% in the medical field. In contrast, although the SymFuzz, relation graph convolutional network, and BiLSTM model methods also demonstrate good detection accuracy, they are slightly inferior to the method proposed in this paper. However, the detection accuracy

based on state deviation analysis method and based on taint analysis and symbol execution method is relatively low, and the false positive rate and false negative rate are relatively high. In addition, the detection time of the method proposed in this article is relatively short, especially in the fields of e-commerce and education, significantly better than the long detection time based on taint analysis and symbolic execution methods.

In summary, the method proposed in this article demonstrates excellent vulnerability detection performance, which is not only reflected in high precision and efficiency, but also has high stability, ensuring satisfactory results in practical applications.

4 Conclusion

Through the above experiments, we can see that the automatic vulnerability detection method of a website driven by associated data adopted in this paper can successfully detect website vulnerabilities in various situations after multiple meticulous parameter tuning and the convolutional neural network has been optimized. The comparative experiment strongly proves that the proposed method has high accuracy and time efficiency in vulnerability detection, not only verifying its effectiveness, but also successfully opening up a novel and efficient path in the field of website vulnerability detection.

References

[1] Verhaeghe, B., Shatnawi, A., Seriai, A., Etien, A., Anquetil, N., and Derras, M., et al. (2022). From gwt to angular: an experiment report on migrating a legacy web application. IEEE Software, 39(4), 76–83.

[2] Xu, H., Wang, C. R., Berres, A., Laclair, T., and Sanyal, J. (2022). Interactive web application for traffic simulation data management and visualization. Transportation Research Record, 2676(1), 274–292.

[3] Kretschmer, M., Pennekamp, J., and Wehrle, K. (2021). Cookie banners and privacy policies: Measuring the impact of the GDPR on the web. ACM Transactions on the Web (TWEB), 15(4), 1–42.

[4] Li, M., and Huang, H. (2021). SymFuzz: vulnerability detection technology under complex path conditions. Computer Science, 48(5), 25–31.

[5] Ma, Q., Wu, Z., Wang, Y. (2023). Approach of web application access control vulnerability detection based on state deviation analysis. Computer Science, 50(2), 346–352.

[6] Liu, X., Li, Y., Yu, M., Zheng, Y., Yu, J., Guo, Y., Kong, H., and Qiang, W. (2022). Web vulnerability detection based on taint analysis and symbolic execution. Computer Applications and Software, 39(11), 297–303.

[7] Wen, M., Wang R., and Jiang, S. (2022). Source code vulnerability detection based on relational graph convolution network. Journal of Computer Applications, 42(6), 1814–1821.

[8] Gong, K., Zhou, Y., Ding, L., and Wang, Y. (2020). Vulnerability detection using bidirectional long short-term memory networks. Computer Science, 47(5), 295–300.

[9] Anton, S. D. D., Fraunholz, D., Krohmer, D., Reti, D., Schneider, D., and Schotten, H. D. (2021). The global state of security in industrial control systems: an empirical analysis of vulnerabilities around the world. IEEE Internet of Things Journal, 8(24), 17525–17540.

[10] Khalid, F., Abbassi, I. H., Rehman, S., Kamboh, A. M., Hasan, O., and Shafique, M. (2021). Forasec: formal analysis of hardware trojan-based security vulnerabilities in sequential circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(4), 1167–1180.

[11] Hu, X., and Xu, F. (2022). A six-port network based on substrate integrated waveguide coupler with metal strips. IET Microwaves, Antennas & Propagation, 16(1), 18–28.

[12] Choi, W. G., Kim, D., Roh, H., and Park, S. (2020). OurRocks: offloading disk scan directly to GPU in write-optimized database system. IEEE Transactions on Computers, 70(11), 1831–1844.

[13] Zhuo, Z., Cai, T., Zhang, X., and Lv, F. (2021). Long short-term memory on abstract syntax tree for SQL injection detection. IET Software, 15(2), 188–197.

[14] Javed, M. F., Nawaz, W., and Khan, K. U. (2021). Hova-fppm: flexible periodic pattern mining in time series databases using hashed occurrence vectors and apriori approach. Scientific Programming, 2021(1), 1–14.

[15] Zhang, C., Zhao, Y., Zhou, Y., Zhang, X., and Li, T. (2022). A real-time abnormal operation pattern detection method for building energy systems based on association rule bases. Building Simulation, 15(1), 69–81.

[16] Chen, Q., Xu, X., and Chen, S. (2022). Multi-user complaint data stream clustering algorithm based on text mining. Computer Simulation, 39(5), 423–426,498.

[17] Benaimeche, M. A., Yvonnet, J., Bary, B., and He, Q. C. (2022). A k-means clustering machine learning-based multiscale method for anelastic heterogeneous structures with internal variables. International Journal for Numerical Methods in Engineering, 123(9), 2012–2041.
[18] Chen, X., Li, W., and Jiang, Y. (2021). K-means clustering algorithms used in the evaluation of online learners' behaviour. International Journal of Continuing Engineering Education and Life Long Learning, 31(3), 394–404.
[19] Wen, Z., and Zhou M. (2020). Recognition of blowholes and cracks on surface of magnetic tile based on deep learning. Ordnance Material Science and Engineering, 43(6), 106–112.
[20] Rashid, N., Demirel, B. U., and Al Faruque, M. A. (2022). AHAR: Adaptive CNN for energy-efficient human activity recognition in low-power edge devices. IEEE Internet of Things Journal, 9(15), 13041–13051.

Biographies

Xiaoli Li has a bachelor's degree in Electrical Engineering and Automation from South China University of Technology in 2003. Her research interests include network security and digitalization. Work experience: From 2003 to present, Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. Academic situation: 7 academic papers published.

Ling Zhao has a bachelor's degree in Electrical Engineering and Automation from North China Electric Power University in 2019. His research interests include network security and digitalization. Work experience: From 2019 to present, Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. Academic situation: 1 academic papers published.

Haobin Shen has a bachelor's degree from Dalian University of Technology in 2009 and a master's degree from South China Normal University in 2012. His research interests include security of binary systems. Work experience: From 2012 to present, Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. Academic situation: 4 academic papers published.

Hanlin Du has a bachelor's degree from Huazhong University of Science and Technology in 2019. His research interests include network security. Work experience: From 2019 to present, Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. Academic situation: 3 academic papers published, 4 patents.

Zhida Guo has a Masters degree from Sun Yat-sen University of Computer Science in 2013. His research interests include network security. Work experience: From 2013 to present, Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. Academic situation: 6 academic papers published, 7 patents.