Hindawi
Mobile Information Systems
Volume 2022, Article ID 6207275, 8 pages
https://doi.org/10.1155/2022/6207275

Research Article

Research on SQL Injection Attack and Defense Technology of Power Dispatching Data Network: Based on Data Mining

Jingyuan Sheng

Department of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, Liaoning 110142, China

Correspondence should be addressed to Jingyuan Sheng; z2019240@stu.syuct.edu.cn

Received 18 May 2022; Revised 8 June 2022; Accepted 16 June 2022; Published 30 July 2022

Academic Editor: M. Praveen Kumar Reddy

In the process of SQL injection attack and defense of power dispatching data network, in order to ensure the accuracy of identification and defense, it is often necessary to build a rule base. However, the scale of the temporarily constructed rule base is limited, which is easy to cause false positives and omissions. Based on the traditional defense, data mining technology is introduced to design a SQL injection attack and defense technology for power dispatching data network. First, the SQL injection attack is analyzed, and the attack flow diagram is obtained as the basis of attack identification. Using the ITFIDF algorithm in data mining technology, combined with the dataset distribution diagram of conventional words and sensitive characters, the SQL injection attack is detected. Design the SQL injection attack early warning mechanism, establish the SQL injection attack defense model, and optimize the workflow of the defense model. The performance test results show that compared with the traditional defense technology, the proposed technology has certain advantages in false alarm rate and running time.

1. Introduction

For power companies, power dispatching data network is an important production data information transmission tool in the production and operation process of power enterprises. The main content is the real-time data about power dispatching and related power communication monitoring data in the process of network transmission. Power dispatching data network is an important tool that can effectively improve production safety and dispatching automation, and it is also an important part of the operation [1, 2]. In power enterprises, the security protection of power dispatching data network is very necessary for the safe operation of power dispatching data network and economic and stable operation. With the increasing importance of it, the related security risks are also increasing. According to the report of a power enterprise, almost all dispatching data network applications have suffered from different forms of attacks from the network, of which more than 60% are SQL injection attacks. In recent years, the number of SQL injection attacks is on the rise. Therefore, for power companies, it is of certain practical significance to analyze the SQL injection attacks on power dispatching data network and obtain high-performance defense technologies [3–5].

In the power dispatching data network, generally, the web application is designed based on ASP.NET language. In the process of network design, there is no verification test of user data input, resulting in some security vulnerabilities injected by SQL in the network database. In this case, due to the external threats that can be encountered, the wind and fire wall designed on the website cannot be identified. At this time, the SQL injection attacker can change the data and parameters in the database in the network system without being recognized as illegal by the system so as to cause harm to the statements in the database system [6–8]. In the traditional defense method, in the process of identifying and defending the SQL injection attack of power dispatching data network, it is necessary to build a rule base. However, the size of the constructed rule base is limited, which is easy to cause false positives and missing positives [9, 10].

This paper is divided into four parts: introduction, attack and defense technology of power dispatching data network, experimental results and discussion, and conclusion. This paper introduces data mining technology and designs a SQL injection attack and defense technology for the power dispatching data network. This paper intends to analyze the SQL injection attacks, obtain the common SQL attack flow diagram, and take it as the basis of attack identification. In the process of inspection, the ITFIDF algorithm in data mining technology is used to detect SQL injection attack, and finally the SQL injection attack defense model is established to realize the research of defense technology.

2. Attack and Defense Technology of Power Dispatching Data Network

2.1. Analysis and Identification of SQL Injection Attacks. For the power dispatching data network, the SQL injection attacks are attacks on the SQL statements in the network using SQL syntax. The attacker modifies the SQL statement by maliciously tampering with the content, illegally accesses the database, and performs relevant operations. For the power dispatching data network, it is very destructive. If you want to identify and defend against SQL injection attacks, you must first understand the process of SQL injection attacks [11, 12]. The attacker constructs special user input based on SQL syntax. After it is transmitted to the application in the scheduling data network, the attacker will construct the corresponding SQL statement according to the dynamic parameters. After the obtained SQL statement is transferred to the database, SQL injection attack is generated accordingly. The attack process is shown in Figure 1:

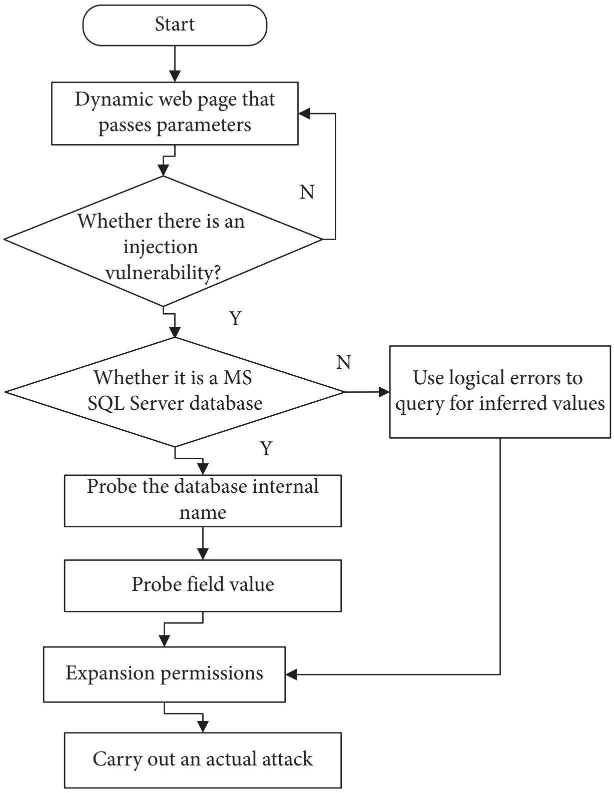

Start
Dynamic web page that passes parameters
Whether there is an injection vulnerability?
N
Y
Whether it is a MS SQL Server database
N
Use logical errors to query for inferred values
Y
Probe the database internal name
Probe field value
Expansion permissions
Carry out an actual attack

Figure 1: Schematic diagram of attack flow.

In the process shown in the figure above, the attacker will first find the SQL injection point and judge the type of database in the network. For different types of databases, attackers will take different methods to expand their privileges. When the attacker really obtains the administrator's privileges, he can carry out the attack. After understanding the process and principle of SQL injection attack, it lays a foundation for subsequent identification and defense.

2.2. SQL Injection Attack Detection Based on Data Mining. Next the SQL injection attack detection based on data mining will be introduced. The application of power dispatching data network is expanding, and the network access behavior will be more and more. For the power dispatching data network and the power dispatching data Internet, their server processes hundreds of Internet applications every day, and these applications are often mixed with some malicious attacks [13]. Many of these malicious intrusions are SQL attacks, while some intrusions have no obvious intrusion characteristics in appearance after careful construction. Therefore, it must be verified at a deeper level in order to achieve the effectiveness protection effect of source code data. In the process of data scheduling, the code is extracted from the network application and statically analyzed. Then in this process, the address of SQL injection point and the statement model of static SQL are automatically obtained. At this time, get the AOP program, realize the dynamic acquisition of SQL, get the real-time dynamic SQL statement to be executed, compare the logical structure of SQL prediction, and complete the verification. In this process, this paper uses the ITFIDF algorithm, which is a new weighting technology combining data mining and information retrieval. Therefore, this paper uses data mining techniques to help defend against SQL attacks. In this algorithm, we can evaluate the importance of a word segmentation in the file. In the detection of this algorithm, first analyze the distribution of regular words and sensitive characters in the SQL statement dataset, as shown in Figure 2.

In the process of text vectorization of dataset using this algorithm, the distribution of sensitive characters t_i is relatively scattered and keeps an uneven state. In general, it occurs more frequently in the set of attack classes. For the statements of normal classes, the ability to distinguish sensitive characters is stronger.

$$m + k \approx x + y. \tag{1}$$

At this time, it exists as

$$\begin{aligned} \mathrm{idf}_t &= \log\left(\frac{a}{x+y}\right), \\ \mathrm{idf}_{t_i} &= \log\left(\frac{a}{m+k}\right). \end{aligned} \tag{2}$$

In the above formula, idf_t represents the inverse document frequency of conventional words in the SQL

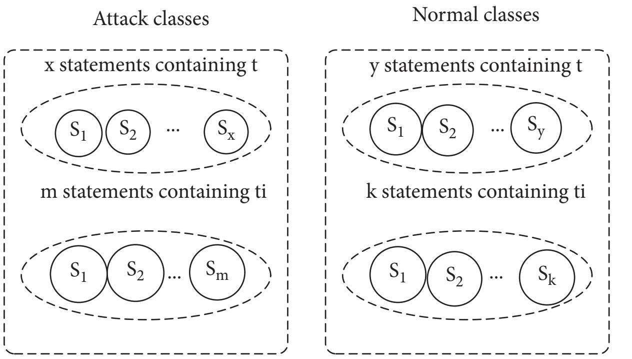

Attack classes
Normal classes
x statements containing t
S1 S2 ··· Sx
m statements containing ti
S1 S2 ··· Sm
y statements containing t
S1 S2 ··· Sy
k statements containing ti
S1 S2 ··· Sk

Figure 2: Dataset distribution diagram of conventional words and sensitive characters.

statement dataset, and idf_{t_i} represents the inverse document frequency of sensitive characters in the SQL statement dataset, and then it can get

$$\mathrm{idf}_t \approx \mathrm{idf}_{t_i}. \quad (3)$$

In the algorithm designed in this paper, in the calculation of inverse document frequency, not only the total amount of data in SQL dataset is considered but also the relationship between the number of statements containing word segmentation is verified. However, there is a lack of comprehensive analysis of the distribution of the word segmentation in the same category dataset, resulting in the weight of the calculated inverse document frequency being not accurate enough [14]. In the improvement process, when calculating the inverse document frequency of word segmentation, the parameter of the distribution number of relevant words in the set is added. The calculation formula is shown as follows:

$$P-idf_t = \log\frac{m*D}{\left|\left\{j\colon t_i \in d_j\right\}\right|}. \quad (4)$$

In the above formula, D represents the total number of statements in the dataset, and $P-\mathrm{idf}$ represents the improved inverse document frequency. The inverse document frequency represents the number of relative SQL statements of the word segmentation in the SQL statement set where the word segmentation is located. The word frequency calculation formula of word segmentation is

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \quad (5)$$

In the above formula, $n_{i,j}$ represents the number of times a participle appears in the SQL statement, and $\sum_k n_{k,j}$ represents the total number of participles in the selected SQL statement. The inverse document frequency calculation formula of the word segmentation is

$$\mathrm{idf}_i = \log\frac{\mathrm{d}}{\left|\left\{j\colon t_i \in d_j\right\}\right|}. \quad (6)$$

Bring the above formula into the calculation to obtain the weight $(P-W\mathrm{idf}_{i,j})$ calculation formula of the injected attack statement, as shown in formula (5):

$$P-W\mathrm{idf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}\log\frac{m*\mathrm{d}}{\left|\left\{j\colon t_i \in \mathrm{d}_j\right\}\right|}. \quad (7)$$

In the calculation of the above formula, the interference between the number of different characters can be effectively avoided, and there is no need to re-establish the rule base. The final weight can effectively represent the weight of the word segmentation in identifying SQL injection attack statements. According to the comparison of weights, it can effectively identify SQL injection attacks.

2.3. Establish the SQL Injection Attack Defense Model. After the completion of SQL attacks, it is necessary to identify a wide variety of applications. The defense model proposed in this paper is a web application for the interaction between the power dispatching data network platform and the rule database, and its most significant feature is the large scale of data [2]. In this case, an early warning mechanism of SQL injection attack is established by using data mining technology, as shown in Figure 3:

In the figure above, the historical data and detection data are input into the data input interface, which is associated with the tool data mining module and early warning module. In the user input interface, after parameter mining and early warning data constraints, the output data is reinput into the attack data mining module. The mining results are input into the functional early warning module after passing through the risk factor database, and the final early warning information is the output.

Under the support of the above early warning mechanism, the established defense model should first filter the requests with obvious attack characteristics. For some carefully disguised attack requests, the syntax structure is compared to achieve defense. The overall workflow of the defense model is shown in Figure 4.

The defense model established in this paper is mainly deployed and implemented in the server of power dispatching data network. Intercept offensive HTTP requests in the GET or POST environment, obtain the host IP address, and match and filter the IP address with the existing IP address in the blacklist. If the match is successful, the request can be directly listed as an SQL injection attack. If the matching fails, the sent request will be filtered. At this time, if the match is successful, it indicates that the request is an SQL attack request. If the number of attacks reaches the specified limit, you need to maintain the IP blacklist and add new attack IP data. If it does not match, take the next defensive measure. Parameterize the SQL statement and convert it into XML format. Select the corresponding main file according to the SQL statement generated during the operation of the network application. After matching, judge whether it is an attack type. In the above operations, the request determined as an attack type will be discarded directly, and the internal

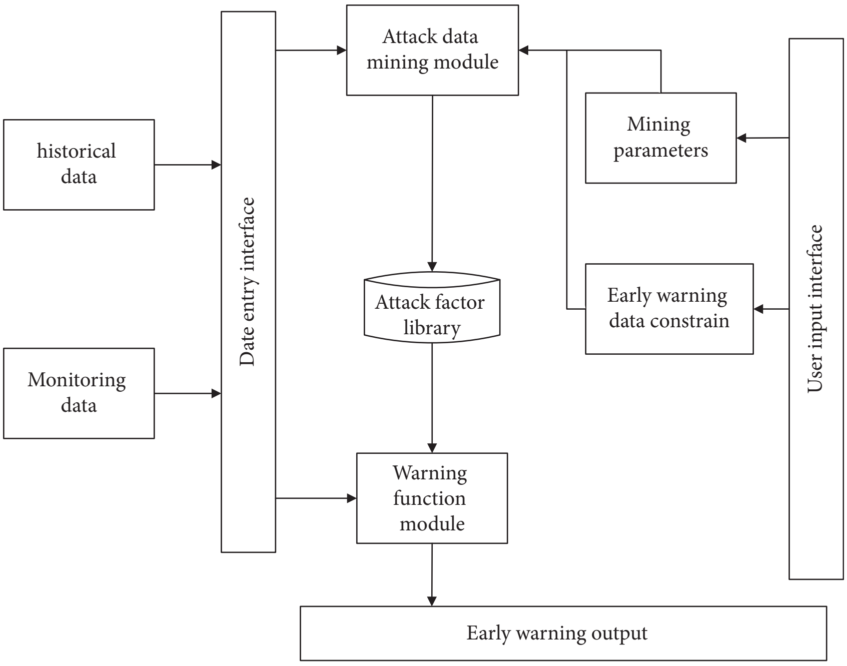


Figure 3: Early warning mechanism of SQL injection attack based on data mining.

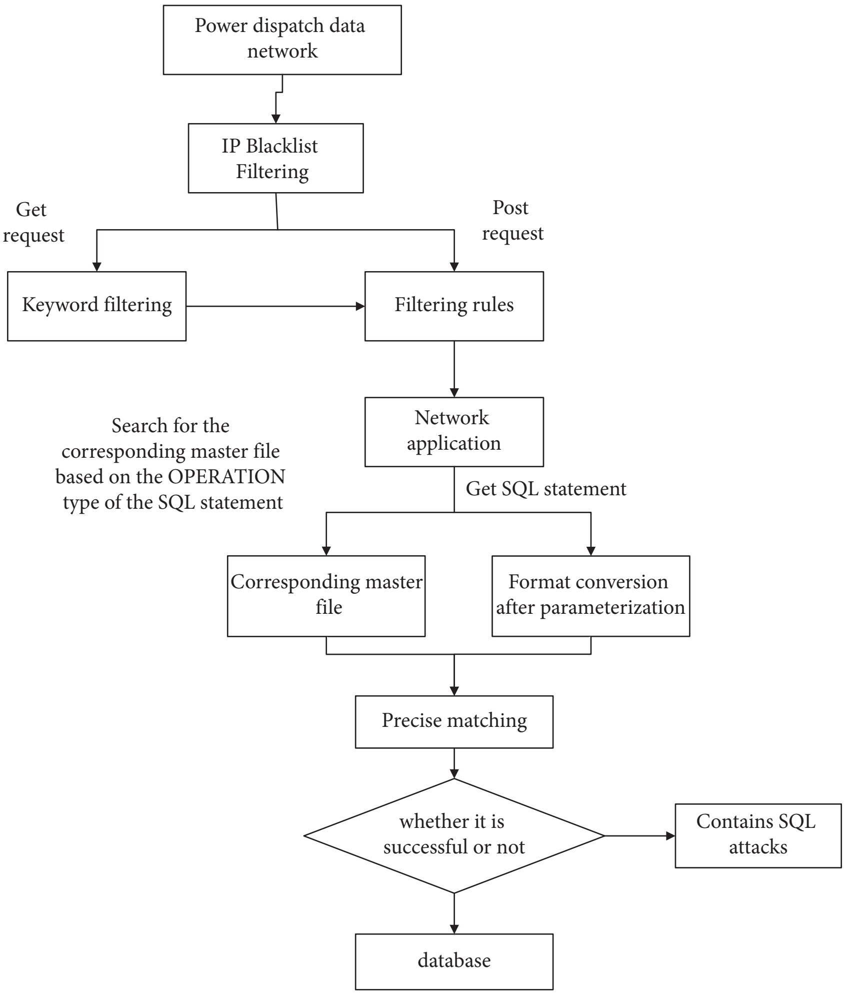

Power dispatch data network
IP Blacklist Filtering
Get request
Post request
Keyword filtering
Filtering rules
Network application
Search for the corresponding master file based on the OPERATION type of the SQL statement
Get SQL statement
Corresponding master file
Format conversion after parameterization
Precise matching
whether it is successful or not
Contains SQL attacks
database

Figure 4: Workflow of the defense model.

database will be upgraded at the same time. The abnormal response is also directly shielded to realize online database maintenance and data update. So far, the research and design of SQL injection attack and defense technology of power dispatching data network based on data mining have been completed.

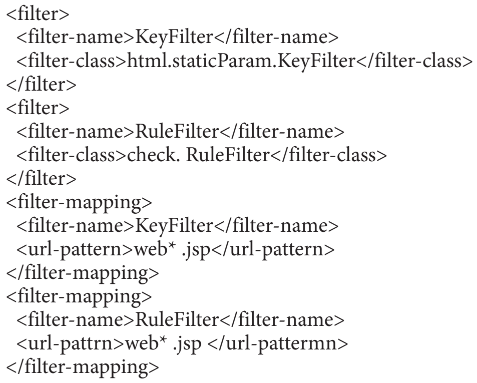
3. Experimental Results and Discussion

In order to verify the effectiveness of SQL injection attack and defense technology of power dispatching data network based on data mining, relevant experiments need to be designed in this chapter. In the experiment, the effectiveness and performance of the design method are verified. The relevant SQL attack traffic set required in the experiment is shown in Table 1:

Table 1: Experimental SQL attack traffic set.

Flow type	Number of attack traffic	Overall proportion (%)
Multimedia	653115	80.90
INT	132125	16.38
P2P	1565	0.19
WWW	1236	0.15
ATTACK	586	0.07
GAMES	4550	0.06
BULK	1265	0.16
MALL	1692	0.21
SERVICE	4589	0.57
COM	5633	0.70
BULK	899	0.11

In the hardware setting in this experiment, the CPU server processor used is 7643, 48 cores and 96 threads 2.3 g of EPYC 7003 series. The software adopts Windows 10 operating system and installs relevant SQL injection tools, including SqlMap and Pangolin. The test environment is set under DVWA, and C# programming is adopted. The SQL attack traffic set used in the experiment is the data in the above table, and the legal data is intercepted from the normal traffic in the power dispatching network. In the experiment, the input filtering module is deployed through the filter program. The specific configuration is shown in Figure 5:



```
<filter>
  <filter-name>KeyFilter</filter-name>
  <filter-class>html.staticParam.KeyFilter</filter-class>
</filter>
<filter>
  <filter-name>RuleFilter</filter-name>
  <filter-class>check. RuleFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>KeyFilter</filter-name>
  <url-pattern>web* .jsp</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>RuleFilter</filter-name>
  <url-pattrn>web* .jsp </url-patternn>
</filter-mapping>
```

Figure 5: Experimental deployment and configuration.

In the figure above, the request and response of related programs to HTTP are set in the file. In the experiment, the performance of this method is evaluated and measured by the false positive rate and the false negative rate. Among them, the false positive rate indicates that a normal request is treated as an attack request and filtered out in the test. The number of such normal requests' accounts for the proportion of the total number of all test requests. The false negative rate indicates that SQL attack traffic is released as a normal request in the test without filtering. The number of such attack requests accounts for the proportion of the total number of all test requests. In addition, it is also necessary to calculate the average response time of all requests to determine the processing speed of defense technology. Under the above experimental environment and judgment methods, the false positive rate, false negative rate, and running time of defense technology are tested.

3.1. False Positive Rate Detection. In order to verify the effectiveness of the SQL injection attack and defense technology of power dispatching data network based on data mining in solving the problem of false positives, the method designed in this paper is compared with the traditional keyword filtering method and the filtering method based on the SQL syntax tree. In the experiment, different attack data are selected to test the power dispatching data network. In the experiment, the number of normal traffic and attack traffic in each attack accounts for 50%, respectively. In addition, the defense technologies to be tested are deployed on the scheduling data network, and the identification and defense of SQL injection attacks are carried out. In the experiment, select the experimental data to construct the user's request to access the data network, count the number of false positives for normal requests under each method, and calculate the false positives rate of each method. The experimental data under each method are shown in Table 2:

According to the data in the above table, the false alarm rate under three methods can be calculated, and the results are shown in Figure 6:

It can be seen from the results in the figure above that when the filtering method based on SQL syntax tree is adopted, the false positive rate changes little with the increase of the number of requests and is basically maintained at about 10%. When using the traditional keyword filtering method, when the number of requests is small, the false positive rate is low. With the increase of the number of requests, the false positive rate also increases, and finally reaches about 8%. In the method designed in this paper, the performance of false positive rate is relatively stable, which is about 1%. Compared with the two traditional methods, the method designed in this paper has certain advantages in the false alarm rate, and the false alarm rate is significantly reduced, which can effectively solve the problem of false alarm in practical application.

3.2. False Alarm Rate Detection. In the detection of false negative rate, in order to verify the performance advantages of this method, this method is compared with the traditional keyword filtering method and the filtering method based on

Table 2: Statistical quantity of different methods under false positive rate detection.

Total number of requests	Actual number of SQL injection attacks	Number of false positives in this method	Number of false positives of the keyword filtering method	Number of false positives of the filtering method based on the SQL syntax tree
200	100	1	6	18
400	200	3	15	38
600	300	7	19	58
1000	500	10	36	98
2000	1000	21	62	189
5000	2500	53	151	476
10000	5000	97	296	952
20000	10000	189	558	186
50000	25000	521	3552	476
100000	50000	1210	7362	896
200000	100000	1983	19320	1993

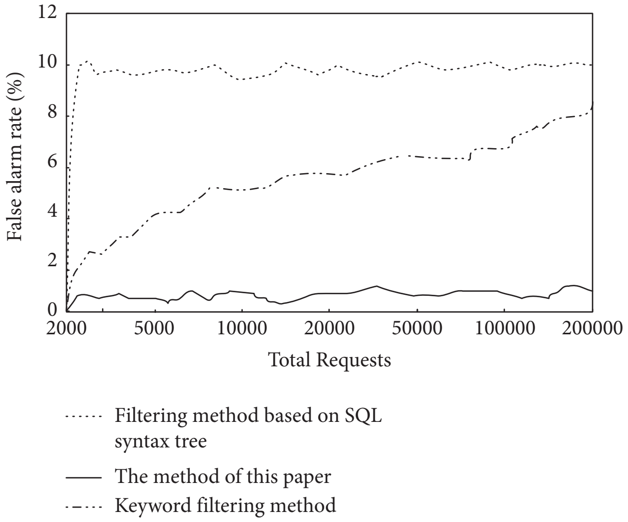


Figure 6: Performance test results of false positive rate.

the SQL syntax tree. The experimental process is similar to that in 3.1. In the experiment, the experimental data are selected to construct the user's request to access the data network. And count the number of false positives for normal requests under each method, and calculate the false positives rate of each method. The experimental data under each method are shown in Table 3:

According to the data in the above table, the false alarm rate under the three methods can be calculated, and the results are shown in Figure 7:

It can be seen from the experimental results in the figure above that when defending against SQL injection attacks, the false alarm rate of the three methods increases first and then decreases. However, it can be seen from the image that the highest false alarm rate of the defense technology designed in this paper is about 1.5%, and then slowly decreases to 0.5%. The false alarm rate of the two traditional methods is more than 6%. In contrast, the defense method designed in this paper only makes judgment in deep data mining, which reduces the rate of missing reports.

3.3. Run Time Detection.

In the above experimental steps, the average response time of each request of the three methods is counted. The response time includes the total time of analyzing statements, returning results after executing database operations and closing links. During the experiment, it is necessary to ensure that the power dispatching data network and computer in the test need to focus on the test without dealing with other work. This can eliminate external interference to the greatest extent and ensure the accuracy of test results. The test results obtained are shown in Table 4:

The experimental results in the above table show that when the power dispatching data network uses the traditional defense technology to identify the SQL injection attack, the average response time of each request is between 0.4 and 0.6 ms. In the process of processing requests, the average response time of the method designed in this paper is about 0.20 ms. This shows that the designed process improves response time by 50% to 66.66%. The significant improvement of response time not only improves the efficiency of

Table 3: Statistical quantity of different methods under the detection of missing report rate.

Total number of requests	Actual number of SQL injection attacks	Number of missing reports of this method	Number of missing reports of the keyword filtering method	Number of missing reports of the filtering method based on the SQL syntax tree
200	100	3	4	5
400	200	5	16	9
600	300	6	19	15
1000	500	8	25	28
2000	1000	15	35	51
5000	2500	20	43	113
10000	5000	49	80	232
20000	10000	89	325	405
50000	25000	570	1356	1536
100000	50000	895	2236	2142
200000	100000	1124	5515	5823

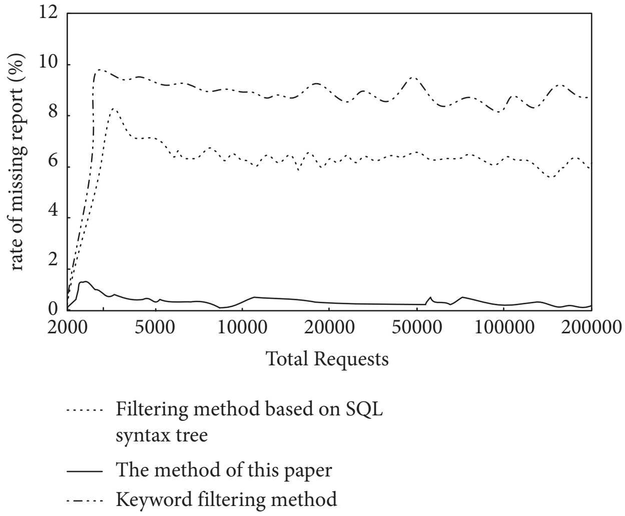


Figure 7: Performance test results of false alarm rate.

Table 4: Statistical results of processing time of different methods.

Total requests	Average response time per request of this method (ms)	Average response time per request of keyword filtering method (ms)	Average response time per request of filtering method based on SQL syntax tree (ms)
200	0.26	0.42	0.51
400	0.25	0.50	0.52
600	0.22	0.51	0.53
1000	0.14	0.45	0.53
2000	0.28	0.52	0.56
5000	0.12	0.56	0.55
10000	0.23	0.42	0.57
20000	0.23	0.42	0.56
50000	0.25	0.54	0.60
100000	0.15	0.45	0.62
200000	0.24	0.52	0.61

judging attacks but also indirectly improves the defense capability of data networks.

To sum up, comparing the SQL injection attack and defense technology of power dispatching data network based on data mining designed in this paper with the traditional defense technology, this technology has certain advantages in false positive rate, false negative rate, and running time. It verifies the effectiveness of this paper.

4. Conclusion

SQL injection attack is ancient script attack vulnerability, and it is also the most common network attack at present. The development of the network makes the spread of SQL injection attack wider and wider. For the internal production data information transmission network of power companies, it cannot be spared. Once the internal communication network of power companies is attacked, it will seriously affect the safe, economic and stable operation of power enterprises. SQL injection attack itself has a certain concealment, high success rate, and great destructiveness. Based on the traditional SQL injection attack and defense technology and with the working characteristics of power dispatching data network, it designs a SQL injection attack and defense technology. Through the analysis of SQL injection attack, the attack flow diagram is obtained as the basis of attack identification. Using the ITFIDF algorithm in data mining technology, combined with the dataset distribution diagram of conventional words and sensitive characters, the SQL injection attack is detected. Design the SQL injection attack early warning mechanism, establish the SQL injection attack defense model, and optimize the workflow of the defense model. After completing the design of the overall defense technology, the test results of this technology are better than the two traditional technologies.

Although the defense technology designed in this paper has achieved certain results, due to the limitations of experience and other reasons, the current technology still has room to be improved. In terms of defense, it still needs in-depth practice, and the defense interface needs to be further improved. SQL injection attack is only one aspect of network security. With the expansion of power company dispatching data network, new attacks will continue to appear. Strengthening the security management of power dispatching data network and regularly evaluating the network security are also the next research direction.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] Y. C. Huang, R. Ma, and L. Q. Ma, “False data injection attack and defense method on load frequency control,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2910–2919, 2021.

[2] M. Anwar, A. Rahman, Z. Galib, and S. M. Galib, “Bi-level poisoning attack model and countermeasure for appliance consumption data of smart homes,” *Energies*, vol. 14, no. 13, p. 3887, 2021.

[3] K. Q. Sun, W. Qiu, W. X. Yao, S. T. You, H. Yin, and Y. L. Liu, “Frequency injection based HVDC attack-defense control via squeeze-excitation double CNN,” *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5305–5316, 2021.

[4] D. R. Geethakumari and G. Geethakumari, “A framework for the identification of suspicious packets to detect anti-forensic attacks in the cloud environment,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2385–2398, 2020.

[5] H. F. Gu, J. N. Zhang, T. Liu et al., “DIAVA: a traffic-based framework for detection of SQL injection attacks and vulnerability analysis of leaked data,” *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 188–202, 2020.

[6] J. J. Q. Yu, “Sybil attack identification for crowdsourced navigation: a self-supervised deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4622–4634, 2021.

[7] K. Lee and J. H. Lee, “Battery draining attack and defense against power saving wireless LAN devices,” *Sensors*, vol. 20, no. 7, p. 2043, 2020.

[8] G. Y. Liu, B. H. Zhong, and X. J. Zhong, “wrgEpidemic Analysis of Wireless Rechargeable Sensor Networks Based on an Attack-Defense Game Model,” *Sensors*, vol. 21, no. 2, p. 594, 2021.

[9] C. Zeng, C. Y. Liu, H. F. Chen, and J. Chen, “Adversarial hiding deception strategy and network optimization method for heterogeneous network defense,” *Electronics*, vol. 10, no. 21, p. 2614, 2021.

[10] S. J. Xu, Y. Hu, and R. Q. Y. Hu, “Edge intelligence assisted gateway defense in cyber security,” *IEEE Network*, vol. 34, no. 4, pp. 14–19, 2020.

[11] L. C. Sejaphala and M. Velempini, “The design of a defense mechanism to mitigate sinkhole attack in software defined wireless sensor cognitive radio networks,” *Wireless Personal Communications*, vol. 113, no. 2, pp. 977–993, 2020.

[12] Y. G. Liu, S. B. Gao, J. Shi, X. G. Wei, and Z. Han, “Sequential-mining-based vulnerable branches identification for the transmission network under continuous load redistribution attacks,” *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5151–5160, 2020.

[13] B. Piereck, M. Oliveira-Lima, A. M. Benko-Iseppon et al., “LAITOR4HPC: a text mining pipeline based on HPC for building interaction networks,” *BMC Bioinformatics*, vol. 21, no. 1, p. 365, 2020.

[14] T. Zhao, C. . h. Zhang, Y. J. A. Zhang, and J. Angela, “Distributed AC-DC optimal power dispatch of VSC-based energy routers in smart microgrids,” *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4457–4470, 2021.