Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 3, 2143-2148
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i3.5755

Research on API security vulnerability detection and repair mechanism based on deep learning

Hong Zou[1*], Jiafa Zhang[2], Zifeng Zeng[3], Weijie Xu[4], Jiawei Jiang[5]
[1,2,3,4,5]China Southern Power Grid Digital Grid Group Communication Company, Guangzhou 510663, China;
ZH_lwgt01@126.com (H.Z.) zhangjf2@csg.cn (J.Z.) zengzf@csg.cn (Z.Z.) xuwj1@csg.cn (W.X.) jiangjw@csg.cn (J.J.).

Abstract: This paper aims to design a deep learning-based mechanism for detecting and repairing API security vulnerabilities, enabling comprehensive monitoring and automated remediation of API interfaces. The proposed system architecture comprises three main modules: a data acquisition and sensing module for real-time monitoring of API performance indicators, a deep learning module utilizing recurrent neural networks (RNN) to analyze API traffic and identify potential vulnerabilities, and a repair module that develops intelligent repair strategies based on the analysis results. Experimental validation shows that the proposed system significantly outperforms traditional rule-matching and support vector machine (SVM) models in terms of vulnerability detection rate, repair rate, and overall quality assessment, highlighting the effectiveness of deep learning models in API security. The research demonstrates that deep learning approaches can enhance the detection and repair of API vulnerabilities, offering a more effective solution compared to conventional methods. This study provides innovative ideas and methodologies for improving API security, which is crucial for safeguarding applications and systems in an increasingly interconnected digital landscape.

Keywords: *API security vulnerability, Deep learning, Repair mechanism.*

1. Introduction

With the rapid development of Internet technology, API (Application Program Interface), as an important part of modern software architecture, is widely used in the interaction between various applications and services. The popularity of API makes data sharing and function invocation between systems more efficient, but at the same time, it also brings about security risks. API security vulnerabilities, such as authentication loopholes, data leakage, and injection attacks, have become the main target of network attackers [1]. According to relevant research, the number of API security vulnerabilities is increasing year by year, bringing huge security risks to enterprises and users [2]. To cope with these challenges, traditional security detection methods can no longer meet the demands of the rapidly changing network environment and complex attack patterns. Therefore, security detection and remediation mechanisms based on deep learning have gradually become a research hotspot. Deep learning can automatically extract data features, identify potential security threats, and make intelligent decisions based on real-time data, providing new ideas for security protection of APIs [3].

Many research organizations and companies have conducted in-depth exploration in the field of API security. For example, Google, Microsoft and other companies have developed a variety of API security detection tools using deep learning and machine learning technologies [4]. These tools identify anomalous behaviors by analyzing the characteristics of API traffic and provide corresponding security fixing suggestions. Meanwhile, academics have also proposed a variety of deep learning-based models, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), for detecting potential vulnerabilities in API requests. Domestic research on API security is also gradually emerging.

History: Received: 23 January 2025; Revised: 13 March 2025; Accepted: 19 March 2025; Published: 25 March 2025
* Correspondence: ZH_lwgt01@126.com

Many universities and research institutes have begun to pay attention to the detection and repair of API security vulnerabilities, and carry out related research by combining deep learning techniques. Some researchers have proposed a deep learning-based API traffic analysis method to identify potential security threats by comparing the differences between normal and abnormal traffic. In addition, some domestic enterprises are also applying deep learning technology to improve API security in practice [5]. Although the research on API security vulnerability detection and repair mechanism based on deep learning has achieved certain results, there are still some shortcomings. Currently, there are relatively few publicly available datasets for API security vulnerabilities, leading to limitations in the training and validation of deep learning models [6]. The lack of diverse and real-scenario datasets makes the effectiveness and robustness of the models in real-world applications suffer. Deep learning models are often viewed as "black boxes" whose internal decision-making processes cannot be easily explained [7]. This is especially important in the security domain, where security experts need to understand the basis of the model's judgment in order to make sound security decisions. There is still plenty of room for the research of API security vulnerability detection and repair mechanism based on deep learning. In this paper, we conduct an in-depth exploration on the aspects of dataset construction, model interpretation, real-time performance and adaptability to improve the security and reliability of APIs.

2. Deep Learning Based Api Security Vulnerability Detection Design

2.1. Overall System Design

The Deep Learning-based API Security Vulnerability Detection and Repair Mechanism System aims to provide comprehensive security monitoring, vulnerability detection and automatic repair of API interfaces through intelligent means [8]. The overall architecture of the system is divided into three main modules: data acquisition and sensing module, deep learning and data processing module, and API security vulnerability repair module. The modules interact with each other through an efficient data transmission channel to ensure real-time and accurate data.

2.2. API Security Vulnerability Detection Function Module Design

2.2.1. Data Acquisition and Sensing Module

The data acquisition and sensing module is the core component of the entire API security vulnerability detection system, responsible for real-time monitoring and collection of key performance indicators and security status of the API interface. The module is designed to ensure the accuracy, reliability and real-time performance of the data, providing a solid data foundation for subsequent security analysis and vulnerability detection. For the selection of sensors, the module utilizes highly efficient network traffic monitoring tools and Security Information Event Management (SIEM) systems. These tools are known for their high performance, low latency and powerful data analysis capabilities, and can operate stably in a variety of network environments. By integrating multiple types of monitoring tools, the module is able to simultaneously capture a wide range of performance parameters such as API requests, response times, error rates, etc. to comprehensively assess the security status of the API [9]. To ensure efficient and real-time data processing, the module integrates a high-performance processing unit, choosing NVIDIA's Jetson Xavier processor. The processor has powerful graphics processing and parallel computing capabilities, and can support complex data analysis and security detection tasks [10]. This design allows the system to maintain an efficient response speed in the face of a large number of API requests, ensuring rapid processing and analysis of real-time data. Packet capture tools (such as Wireshark or tcpdump) play a crucial role in the data collection process. These tools are responsible for capturing packets from the network and transforming them into an analyzable format. With efficient filtering and parsing mechanisms, the modules are able to capture small changes in API requests with great precision, ensuring data integrity and accuracy. According to the Nyquist sampling theorem, the relationship between the sampling frequency and the highest frequency of the signal is:

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 3: 2143-2148, 2025
DOI: 10.55214/25768484.v9i3.5755

$$f_s \geqslant 2 f_{max} \quad (1)$$

Where: f_s is the sampling frequency, Hz; f_{max} is the highest frequency of the signal. The Nyquist sampling theorem ensures that the sampled signal can completely reconstruct the original data stream, effectively avoiding the problem of information loss. By reasonably setting the sampling frequency and optimizing the layout of the monitoring tools, the data acquisition and sensing module is able to capture the performance signals of the API interface in real-time and accurately. In addition, the module places special emphasis on the ability to detect API security vulnerabilities during the data processing and analysis phase. Through deep learning analysis of the collected high-quality data, the system is able to identify potential security vulnerabilities and abnormal behaviors for effective vulnerability detection and response.

2.2.2. Deep Learning and Data Processing Module

The Deep Learning and Data Processing module is responsible for intelligent analysis and security vulnerability detection of the collected API request and response data. The module is based on NVIDIA's A100 Graphics Processing Unit (GPU) accelerator card with 6,912 CUDA cores and 432 tensor computing cores, with a single-precision floating-point performance of up to 19.5 TFLOPS, which is coupled with High Bandwidth Memory (HBM) to meet the high-performance computing needs of deep learning algorithms in large-scale data processing. With High Bandwidth Memory (HBM), it can meet the high-performance computing needs of deep learning algorithms in large-scale data processing.

At the software level, the module adopts the TensorFlow deep learning framework and combines its powerful model building and training capabilities to build a Recurrent Neural Network (RNN) model to achieve efficient extraction and analysis of API traffic features.RNN, through its time-series processing capabilities, is able to effectively capture the data's temporal sequence features and contextual information. In the case of a given input sequence, the output of RNN can be expressed as:

$$\boldsymbol{Y}_{c'} = \boldsymbol{b}_{c'} + \sum_{c=1}^{C} \boldsymbol{W}_{c',c} * \boldsymbol{X} \quad (2)$$

Where: $\boldsymbol{W}_{c',c} \in \square^{k \times k}$ is the convolutional kernel; $\boldsymbol{b}_{c'} \in \square$ is the bias term; H' and W' are the height of the output feature map and the width of the output feature map. By stacking multiple RNN layers, the module is able to form a hierarchical feature representation that effectively identifies potential security vulnerabilities and anomalous behaviors in API traffic.

2.2.3. API Security Vulnerability Repair Module

API Security Vulnerability Repair Module is responsible for achieving effective repair and management of detected security vulnerabilities, as well as formulating intelligent repair strategies based on the analysis results of the deep learning model. The module adopts a high-performance Intel Xeon processor with multi-core parallel processing capability and a main frequency of up to 3.0 GHz, and supports a variety of high-speed communication interfaces, including Ethernet, PCIe, and USB, etc., to ensure fast data transmission and processing. During the vulnerability repair process, the module introduces a deep learning-based adaptive repair algorithm to achieve intelligent repair of API security vulnerabilities. The algorithm dynamically adjusts the repair strategy by analyzing the output of the deep learning model in real time to improve the accuracy and efficiency of repair. Its basic principle can be expressed as follows:

$$R(s,a)=r+\gamma \max_{a'} R(s',a') \quad (3)$$

where R(s,a) is the expected repair effect of taking action a in states; r is the immediate repair reward; γ is the discount factor; s″ is the next state; and a″ is the optional repair action in states″. By

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 3: 2143-2148, 2025
DOI: 10.55214/25768484.v9i3.5755

constantly updating the value of R, the algorithm is able to learn the optimal repair strategy and thus make more accurate repair decisions when facing complex API vulnerabilities. The module not only realizes real-time monitoring and automated repair of API vulnerabilities, but also continuously optimizes the repair process through the feedback mechanism. By reasonably designing the objective function and constraints of the remediation algorithm, the module can effectively improve the security and stability of the API, and ensure that the final repaired API meets the security standards and best practices. In addition, the module also has a vulnerability repair logging function, so as to facilitate subsequent auditing and analysis, and further enhance the security and traceability of the system.

3. System Experimental Validation

3.1. Experimental Program

A complete set of experimental programs is designed to verify the performance of deep learning-based API security vulnerability detection and repair mechanism. The experiments were conducted in the test lab of a cybersecurity company, and 10 different types of APIs were selected as test objects, covering a wide range of application scenarios and technical specifications. RESTful API is used for data exchange in Web applications, using the HTTP protocol; SOAP API is an XML-based protocol commonly used for interoperability of enterprise-class services; GraphQL API allows the client to specify the structure of the required data, with a high degree of flexibility; WebSocket API is used for real-time communication of bi-directional data transfer; gRPC API is a high-performance remote procedure call framework based on HTTP/2, suitable for microservice architectures; OAuth API is used for user authentication and authorization, commonly used in social media applications; Payment Gateway API is used for online payment processing, such as Stripe or PayPal; Messaging API is used for messaging services such as Twilio or Firebase Cloud Messaging; IoT APIs support data exchange between IoT devices; and Machine Learning APIs provide services for machine learning model calls, such as Google Cloud ML or AWS SageMaker. The training dataset contains API requests and responses, API documentation, and logs of user inputs and outputs, and its data annotation methods include vulnerability type annotation, i.e., marking possible security vulnerabilities (e.g., SQL injection, XSS, authentication flaws, etc.) in each API request and response, and assigning a risk level (high, medium, or low) to each annotated vulnerability, as well as providing the API usage scenarios , invocation frequency and user roles, and other contextual information. The validation dataset, on the other hand, includes real API invocation logs, user feedback and system monitoring data, and its data annotation methods include vulnerability validation, i.e., annotating the existence of vulnerabilities according to the actual running situation and recording the status after fixing them, as well as performance indexes such as the response time and success rate of the API.

During the experiment, the Wireshark network analysis tool was utilized to capture real-time packets of API requests and responses, with the main parameters including request method, status code, response time, and data load, etc. The sampling frequency was set to 1 kHz, and each test cycle lasted for 3 hours. A network traffic monitoring device with a measurement accuracy of ±0.01% was used to record changes in the traffic characteristics of the API to ensure the accuracy and reliability of the data. For data transmission, the experiment uses a wireless communication system based on 5G technology, with uplink and downlink rates of 2 Gb/s and 2.5 Gb/s, respectively, and a latency control of less than 1 ms to ensure real-time data. The experiments are conducted on a high-performance server equipped with NVIDIA A100 GPUs for deep learning model training, and the training dataset contains 10,000 samples, each of which contains API request characteristics and historical vulnerability records.

The experiments were conducted in 5 rounds of iterations, with 200 rounds of training in each round, the batch size was set to 128, and the initial value of the learning rate was 0.001, using Adam optimizer and cross-entropy loss function. In order to verify the effectiveness of the deep learning-based API security vulnerability detection and repair mechanism, the experiments are designed with four typical security vulnerability scenarios, including SQL injection, cross-site scripting attack (XSS), authentication bypass and insecure direct object reference (IDOR). By injecting vulnerability data

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 3: 2143-2148, 2025
DOI: 10.55214/25768484.v9i3.5755

artificially, we compare the detection accuracy of the deep learning model of the system designed in this paper with traditional machine learning models such as Rule-based matching (RBM) and Support Vector Machine (SVM), and record the response time of the deep learning model.

3.2. Analysis of Experimental Results

In the API security vulnerability detection scenario, the performance comparison of the deep learning-based detection and repair mechanism is shown in Table 1, and the effect comparison between the deep learning model and the traditional machine learning model in API vulnerability detection is shown in Table 2.

Table 1.
Performance comparison of deep learning models for API security vulnerability detection

Testing standards	Deep learning model detection acc uracy/%	Rule matching detection accuracy/%	SVM detection accuracy/%	Deep learning response time/ms
Vulnerability detection rate	98.2	85.6	83.4	75
Vulnerability remediation rate	96.7	80.1	78.5	78
Overall quality assessment	97.5	82.0	80.2	76

The data in Table 1 shows that the deep learning model performs well in vulnerability detection rate, vulnerability repair rate and overall quality assessment. Specifically, the deep learning model achieves a vulnerability detection rate of 98.2%, a vulnerability repair rate of 96.7%, and an overall quality assessment of 97.5%. These figures are significantly higher than those of traditional rule matching (85.6% detection rate, 80.1% repair rate, and 82.0% overall quality assessment) and SVM models (83.4% detection rate, 78.5% repair rate, and 80.2% overall quality assessment), suggesting that the deep learning model is more accurate in identifying and distinguishing security vulnerabilities. This advantage makes the security detection and repair process of APIs more reliable and reduces the security risks caused by vulnerabilities. In terms of response time, the response time of the deep learning model is 76 ms, which compared to the response time of rule matching and SVM (75 ms and 78 ms, respectively) is not significantly inferior to the traditional model, but instead shows higher efficiency in processing time (75 ms), which is crucial for security detection that requires fast processing of a large number of API requests.

Table 2.
Performance comparison between rule-based matching and deep learning models for API security vulnerability detection

Detection Methods	Detection accuracy/%	Response time/ms	Final Bug Fix Score
Rule-based matching approach	84.5	120	70
deep learning model	97.5	76	90

From the data in Table 2, the rule-matching-based approach performs less well than the deep learning model in terms of both detection accuracy and response time. Specifically, the detection accuracy of the rule-based matching approach is only 84.5%, while the deep learning model reaches 97.5%, an improvement of nearly 13 percentage points. This significant difference indicates that the deep learning model is more advantageous when dealing with complex API requests and is able to identify potential security vulnerabilities more accurately. In terms of response time, the response time of the rule-based matching approach is 120 milliseconds, which is 44 milliseconds shorter compared to 76 milliseconds for the deep learning model. This fast response capability enables the deep learning model to better adapt to the demands in the dynamically changing network environment and improve the overall security detection efficiency. In addition, the final vulnerability fix score is also an important indicator. The deep learning model's score of 90 is much higher than the rule-based matching method's

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 3: 2143-2148, 2025
DOI: 10.55214/25768484.v9i3.5755

70. This result not only reflects the high accuracy of the deep learning model in the detection process, but also demonstrates its significant advantage in improving the overall security of APIs.

4. Conclusion

The API security vulnerability detection and repair mechanism based on deep learning proposed in this study fully demonstrates the application potential of deep learning technology in the field of network security. By constructing efficient data acquisition and sensing, deep learning analysis and intelligent repair modules, the system is able to monitor the security status of APIs in real time and quickly identify and repair potential security vulnerabilities. Experimental results show that the deep learning model outperforms traditional security detection methods in terms of detection accuracy and response speed, significantly improving the overall security of the API.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] M. S. Alkatheiri, A. O. Aseeri, and Y. Zhuang, "A deep learning method for the security vulnerability study of feed-forward physical unclonable functions," *Arabian Journal for Science and Engineering*, vol. 49, no. 9, pp. 12291-12303, 2024. https://doi.org/10.1007/s13369-023-08643-6

[2] Z. Lai, H. Chen, R. Sun, Y. Zhang, M. Xue, and D. Yuan, "On security weaknesses and vulnerabilities in deep learning systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 01, pp. 1-15, 2024.

[3] V. K. Jain and M. Tripathi, "An integrated deep learning model for Ethereum smart contract vulnerability detection," *International Journal of Information Security*, vol. 23, no. 1, pp. 557-575, 2024. https://doi.org/10.1007/s10207-023-00752-5

[4] E. Zhu, J. Chang, S. Luo, and Z. Ma, "Effective source code vectorization for vulnerability detection using deep learning and attention mechanism," *Available at SSRN 4341595*, pp. 1-29, 2023. https://doi.org/10.2139/ssrn.4341595

[5] Z. Song, J. Xu, K. Li, and Z. Shan, "HCRVD: A vulnerability detection system based on CST-PDG hierarchical code representation learning," *Computers, Materials & Continua*, vol. 79, no. 3, pp. 4573-4601, 2024. https://doi.org/10.32604/cmc.2024.049310

[6] J. Chen, W. Lin, S. Cai, Y. Yin, H. Chen, and D. Towey, "BiTCN_DRSN: An effective software vulnerability detection model based on an improved temporal convolutional network," *Journal of Systems and Software*, vol. 204, p. 111772, 2023. https://doi.org/10.1016/j.jss.2023.111772

[7] S. Wan *et al.*, "Bridging the gap: A study of AI-based vulnerability management between industry and academia," presented at the 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), IEEE., 2024.

[8] Y. Li, Y. Zuo, H. Song, and Z. Lv, "Deep learning in security of internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22133-22146, 2021. https://doi.org/10.1109/JIOT.2021.3106898

[9] Z. Tian, B. Tian, J. Lv, and L. Chen, "Learning and fusing multi-view code representations for function vulnerability detection," *Electronics*, vol. 12, no. 11, p. 2495, 2023. https://doi.org/10.3390/electronics12112495

[10] K. Mittal and P. Khurana Batra, "Graph-ensemble fusion for enhanced IoT intrusion detection: Leveraging GCN and deep learning," *Cluster Computing*, vol. 27, no. 8, pp. 10525-10552, 2024. https://doi.org/10.1007/s10586-024-04404-8