

Research on Deep Learning Driven XSS Vulnerability Detection Technology: From Feature Extraction to Real-time Monitoring

Huang Li*, Amirrudin Kamsin

Faculty of Information Technology, City University Malaysia, Kuala Lumpur 46100, Malaysia.

How to cite this paper: Huang Li, Amirrudin Kamsin. (2025) Research on Deep Learning Driven XSS Vulnerability Detection Technology: From Feature Extraction to Real-time Monitoring. *Future Trends in AI Research*, 2(1), 17-20. DOI: 10.26855/ftair.2025.06.004

Received: November 11, 2025

Accepted: December 21, 2025

Published: January 28, 2026

***Corresponding author:** Huang Li, Faculty of Information Technology, City University Malaysia, Kuala Lumpur 46100, Malaysia.

Abstract

Cross-site scripting (XSS) vulnerabilities, characterized by their stealthy nature and adaptive capabilities, remain the most prevalent security threat in web applications. Conventional rule-based detection methods struggle to address frequent vulnerabilities. This study focuses on deep learning-based XSS detection, adopting a three-phase approach: feature extraction, model construction, and real-time monitoring. We propose multi-dimensional feature extraction methods including word embedding and syntax tree analysis, effectively overcoming limitations of traditional signature engineering in capturing evolving attack patterns. A hybrid neural network model combining CNN and LSTM is developed, where CNN identifies localized character patterns while LSTM analyzes contextual relationships, significantly enhancing detection capabilities against complex XSS scripts. A lightweight real-time monitoring framework utilizing edge computing is designed, achieving latency under 50 milliseconds-meeting stringent requirements for web system protection. Experimental results demonstrate 98.2% accuracy and 97.8% recall rate, with 32% fewer false positives compared to rule-based detection methods. This innovative approach provides an effective and intelligent defense mechanism against XSS vulnerabilities.

Keywords

Deep learning; Vulnerability mining of XSS; Feature extraction; CNN-LSTM model

Introduction

With the proliferation and complexity of web services and applications, Cross-SiteScripting (XSS) has emerged as the most critical security threat for both personal data theft and website stability. According to the 2024 cybersecurity survey report by OWASP, XSS vulnerabilities accounted for 28.7% of all detected weaknesses in web applications, ranking second only to SQL injection attacks. These vulnerabilities exploit malicious code (such as JavaScript or VBScript) embedded in visited pages to steal cookies [1], manipulate interfaces [2], or hijack user login credentials. If not properly addressed, such attacks can lead to personal information leaks, corporate data breaches, and even major cybersecurity incidents.

1. Features of XSS Vulnerability Attack and Limitations of Traditional Detection Technology

2.1 Core features of XSS vulnerability attack

XSS attacks exploit vulnerabilities through “malicious script injection and execution”, characterized by high

stealthiness, broad applicability, and devastating destructive power. Attackers inject malicious code containing JavaScript or VBScript into user input fields, URL variables, or comment sections. When victims access compromised websites, this code executes automatically in the user's browser. XSS attacks can be categorized into three types: storage-based (where malicious code is stored on servers like forum posts), reflection-based (where code is transmitted via URL parameters like search boxes), and DOM-based (where attacks are triggered by manipulating DOM structures without server involvement). To evade detection, attackers often encrypt or modify their code using techniques like URL encoding or Base64 encoding. For instance, they might `<sc<span= ""style= "box-sizing": border-box;">irpt>` compile it into `%3Cscript%3E`, making malicious content appear as legitimate text to increase identification difficulty. Once an attack occurs, it can steal user cookie information, alter website content, and even hijack user devices [1].

2. Key Technologies of XSS Vulnerability Detection Driven by Deep Learning

2.1 Multi-dimensional feature extraction scheme: comprehensive representation from characters to semantics

This project aims to transcend the limitations of traditional “single-character matching” by constructing a three-tier feature system of “character-word-semantics” to achieve comprehensive characterization of XSS attack codes. At the character level, we employ encoding techniques (such as One-Hot encoding and Word2Vec) to convert encoded characters (including special symbols and encoded characters) into vectors, capturing fundamental attack symbol features (`<`, `>`, `script`). Even after URL encoding (e.g., `%3C` corresponding to `<`), original features can be restored through character mapping. At the word level, code is segmented using tokenization tools (e.g., `<script>`, `eval ()` etc.) and combined with pre-trained word vectors like GloVe to assign semantic weights, enabling differentiation between “malicious functions” and “legitimate script keywords”. At the semantic level, syntax analysis methods such as Abstract Syntax Tree (AST) parsing are utilized to examine code logic structures, identifying deep semantic associations like “tag nesting” and “function calls”. For instance, we determine whether combinations of `<script>` and document components constitute malicious data theft, avoiding missed detections caused by isolated feature analysis while providing rich and accurate feature inputs for subsequent model training [3].

2.2 CNN-LSTM hybrid neural network model: taking into account local and contextual features

The CNN-LSTM hybrid neural network model effectively addresses the challenge of detecting XSS attack code that “hides local features and exhibits strong contextual correlations” through the coordinated operation of two modules. The Convolutional Neural Network (CNN) primarily extracts critical features from code snippets, converting input XSS attack code (including encoding distortions and tag nesting) into character-level or word-level vectors. By employing multi-scale convolutional kernels (3×3, 5×5) for sliding scans, it accurately captures local attack features (such as `<script>` tag fragments) and identifies core semantic units even when code is fragmented or distorted (e.g., `<s<script> <s script> cript> cript>`). The Long Short-Term Memory Network (LSTM) utilizes its gating mechanisms (input gate, forget gate, output gate) to capture contextual correlations in code, [4].

2.3 Lightweight real-time monitoring architecture: balance accuracy and efficiency

To address the “conflict between high computational demands and real-time responsiveness” in deep learning model deployment, this project adopts a three-tier design approach of “model compression, parallel computing, and dynamic updates” to achieve precise and efficient XSS vulnerability detection. In terms of model compression, we utilize knowledge extraction and pruning techniques to optimize the CNN-LSTM hybrid model: transferring 98.2% of knowledge from the complex teacher model to a lightweight student model while trimming redundant convolutional kernels and LSTM units (by 35%). This reduces model parameters from 280 MB to 85 MB, triples inference speed, and achieves `<10ms>` per-request detection time for typical server-side scenarios, meeting e-commerce platforms' high-concurrency requirements (e.g., thousands per second). At the parallel computing level, we establish an edge-cloud collaborative detection chain: deploying lightweight models on edge nodes (e.g., web servers) to filter routine requests (with over 90% interception rate for typical XSS attacks), while uploading suspicious complex requests (e.g., multi-layer encoded code) to the cloud for deep analysis using high-precision models [5]. This reduces computational load on edge nodes while preventing false negatives caused by lightweight models. The dynamic update mechanism

collects attack samples in real-time and performs incremental weekly model training. Under user privacy protection, federated learning methods aggregate multi-scenario attack samples (e.g., forum comments, URL parameters), achieving over 95% recognition rate for new-generation variant attacks like AI-generated hidden scripts. Through the test of a government platform, more than 1200 XSS attacks can be intercepted every day, with a false alarm rate of less than 0.8%, while ensuring that the web page loading delay remains unchanged, truly realizing “precise detection without delay, real-time protection without omission” [6].

3. Experimental Verification and Performance Analysis

3.1 Experimental environment and data set

Experimental environment: CPU is IntelXeonGold6330(2.0GHz), GPU is NVIDIAA100 (40GB), memory 128GB, operating system Ubuntu22.04, deep learning framework TensorFlow2.10.

Data set: The open data set XSS-ATTACK-DB and the self-made data set (including 20,000 new mutated XSS samples, such as UTF-7 encoding, DOM type injection samples) were divided into training set, test set and verification set in a ratio of 7:2:1 [7].

3.2 Experimental results and comparative analysis

Table 1. Comparison with traditional detection methods

| test method | precision (%) | recall (%) | Error rate (%) | Detection delay (ms) |
|------------------------------------|---------------|------------|----------------|----------------------|
| Rule matching (OWASPZAP) | 82.3 | 79.5 | 15.1 | 8 |
| Static code analysis (SonarQube) | 85.7 | 81.2 | 12.8 | 300 |
| Deep learning method in this paper | 98.2 | 97.8 | 4.2 | 48 |

4. Conclusions

This paper proposes an XSS detection technique utilizing deep learning methods. While addressing the limitations of traditional feature-based approaches, it enhances detection capabilities for complex deformations through a CNN-LSTM hybrid architecture. Additionally, we implement lightweight/edge computing-based online detection, with experimental data demonstrating superior performance compared to conventional methods. This approach provides an effective solution for addressing XSS vulnerabilities in web systems [8]. Future research can be advanced in three directions: First, adversarial training to improve model robustness against adversarial XSS (i.e., generating blurred code from adversarial samples); Second, federated learning enabling secure data sharing and collaborative modeling of XSS samples among participants to further enhance model versatility; Finally, integrating detection mechanisms into web frameworks through built-in security plugins, forming an integrated “development-detection-defense” approach that drives intelligent evolution of web security mechanisms [9].

References

- [1] Zhang M. Research on automated detection technology of industrial internet XSS vulnerabilities based on deep learning. *Wireless Interconnect Technol.* 2025;22(9):105-108.
- [2] He ZY, He CW, Chen W, et al. Code vulnerability detection based on implicit flow analysis and deep learning. *Comput Eng Des.* 2025;46(7):1951-1958.
- [3] Yang DF, Jiang XW. Research on XSS attack detection based on deep learning. *Jiangsu Commun.* 2023;39(3):94-102.
- [4] Lin YB, Ling J. A method for XSS attack detection based on residual network and GRU. *Comput Eng Appl.* 2022;58(10):101-107.
- [5] Jiang YM, Luo XY, Yu M, et al. A Web attack detection method based on bidirectional long short-term memory neural network. *Inf Countermeas Technol.* 2023;2(1):55-65.
- [6] Pang B. Application of knowledge graph technology in vulnerability detection of computer network links. *Inf Syst Eng.* 2025;(12):61-64.
- [7] Qiu B. Exploring the application of security vulnerability detection technology in computer software. *Cybersecur Technol Appl.*

2025;(10):73-75.

- [8] Liang ZF. Research on security vulnerability detection and protection technology in the big data environment. *Autom Appl.* 2025;66(18):267-269.
- [9] Host. Application of knowledge graph technology in computer network link vulnerability detection. *Cybersecur Informatiz.* 2025;(07):148-150.