UDC *004.056:004.032.26*

DETECTION OF XSS ATTACKS AND SQL INJECTIONS USING CONVOLUTIONAL NEURAL NETWORKS

Qiao X., Kedo E.S

Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Republic of Belarus

Abstract. Cross-site scripting (XSS) and SQL injection (SQLi) remain critical threats to web application security, often bypassing traditional rule-based defenses through obfuscation techniques. This study proposes a lightweight convolutional neural network (CNN) that detects malicious inputs directly from raw HTML or script fragments using an ASCII-based grayscale representation. Evaluated on a public dataset of approximately 27,000 samples, the model achieves 98.39% accuracy with ~1 ms inference latency. The results demonstrate that compact CNN architectures can provide efficient, real-time detection of injection-based attacks in practical deployment environments.

Keywords: cross-site scripting; XSS detection; convolutional neural network; web security; intrusion detection; deep learning; HTML analysis; malicious script detection; cybersecurity.

Introduction

Web applications are frequent targets of injection attacks, among which cross-site scripting (XSS) and SQL injection (SQLi) are the most widespread. XSS allows attackers to execute malicious scripts in a user's browser, while SQLi targets backend databases by injecting crafted queries to manipulate or extract sensitive data [1, 2]. Traditional defenses, including blacklist filtering, signature-based detection, and regular expressions, are widely used in web application firewalls. However, these approaches are often ineffective against modern attack techniques that employ obfuscation, encoding, mixed-case tokens, and nested payload structures. Machine learning provides a more adaptive solution, but classical approaches rely heavily on handcrafted features and struggle to capture complex structural patterns [3]. Deep learning, particularly convolutional neural networks (CNNs), offers a promising alternative by learning directly from raw input data. This paper investigates a compact CNN model that processes ASCII-based representations of input

strings to detect injection attacks. While the primary focus is on XSS, the approach is equally applicable to SQLi patterns, as both attack types exhibit distinctive local structures such as special characters, delimiters, and repeated symbolic sequences.

Main Part

XSS and SQLi remain two of the most common vulnerabilities in modern web applications, and rule-based defenses are increasingly challenged by obfuscated payloads. While XSS enables attackers to execute malicious scripts in a user's browser, SQL injection targets backend databases by manipulating query logic to access or modify sensitive data. This paper presents a compact convolutional neural network for binary injection detection from raw HTML or script fragments. Input strings are transformed into normalized ASCII grayscale matrices, allowing the model to learn structural attack patterns directly from data. Experiments on the public Kaggle XSS dataset of about 27,000 samples show that the proposed CNN reaches 98.39% accuracy, 97.89% precision, 98.40% recall, and 98.14% F1 score on the held-out test set. The model trains in 48 s on an RTX 3080 and supports about 1 ms inference latency per sample, which makes it practical for near-real-time web security monitoring.

Traditional defenses such as blacklist filtering, signature matching, and manually maintained regular expressions are still widely used in web application firewalls. However, these methods are brittle when attackers rely on encoding tricks, mixed-case tokens, nested tags, or event-handler based payloads in XSS, as well as tautologies, UNION-based queries, and comment obfuscation in SQL injection. Classical models depend strongly on handcrafted features and may miss local spatial patterns embedded in HTML strings or query-like inputs. Recent studies have shown that deep learning is particularly effective for malicious content classification because it can learn directly from raw or weakly processed inputs. In both XSS and SQLi domains, convolution is attractive because attack payloads contain short local motifs, including tag delimiters, SQL keywords, logical operators etc.

The experiments use the public Cross-Site Scripting Dataset for Deep Learning published on Kaggle. The corpus contains approximately 27,000 labeled samples, including benign web inputs and malicious XSS payloads from reflected, stored, and DOM-related scenarios. Although the dataset primarily focuses on XSS, the proposed representation and model are generalizable to SQL injection patterns due to their shared reliance on symbolic structure and character-level anomalies. Stratified sampling is applied to split the data into 80% training and 20% testing subsets while preserving class balance. Preprocessing is intentionally simple. Each character in the raw string is converted to its ASCII code, normalized by division with 128, and written

into a fixed 100×100 matrix. Short strings are padded with zeros and long strings are truncated. This representation preserves punctuation density, tag-like shapes, SQL syntax fragments, and recurring payload patterns that are difficult to capture with word-based tokenization. The CNN model contains three convolutional blocks followed by max pooling, dropout, one dense layer, and a sigmoid output unit. ReLU activation is used after each convolution. The network is trained with binary cross-entropy loss and the Adam optimizer. Early stopping is adopted to prevent overfitting, and model quality is evaluated with accuracy, precision, recall, F1 score, and AUC-ROC.

The trained CNN converges rapidly and achieves stable validation behavior without severe overfitting. On the held-out test set, the model correctly classifies 2,362 malicious samples and 2,950 benign samples, with 38 false negatives and 50 false positives. The resulting aggregate scores are 98.39% accuracy, 97.89% precision, 98.40% recall, 98.14% F1 score, and 99.76% AUC-ROC. These results confirm that the image-style ASCII representation is sufficiently expressive for high-quality detection of injection attacks. A closer inspection of the error distribution shows that most misclassified malicious samples are heavily obfuscated strings containing nested encodings, fragmented JavaScript, or complex SQL expressions with unconventional formatting. In contrast, several false positives resemble attack payloads because they contain dense punctuation, encoded content, or debugging snippets. This behavior is acceptable for a screening model because security operations typically prefer a small surplus of alerts over missed attacks.

Computational performance is also favorable. The network contains about 1.3 million parameters, trains in 48 s on an NVIDIA RTX 3080 GPU, and produces inference latency of about 1 ms per sample. Such throughput is compatible with inline or near-inline deployment in a web application firewall, API gateway, or log triage system where fast response is required. The results show that a carefully designed CNN offers an effective trade-off between detection quality and runtime cost. Although more complex hybrid architectures can achieve even higher scores, the CNN baseline remains attractive when computational simplicity, explainability of the processing pipeline, and straightforward deployment are important design constraints.

From a practical perspective, the model can be used as a first-stage detector before deeper traffic inspection. High-confidence benign requests may pass immediately, while suspicious cases, including both XSS and SQLi, can be escalated to stricter filtering or analyst review.

Conclusion

This study demonstrates that a compact CNN combined with an ASCII-based grayscale representation can effectively detect XSS attacks and

SQL injection with high accuracy and low latency. The model captures local structural patterns in raw HTML inputs without relying on manual feature engineering. Its efficiency and strong performance make it suitable for real-time deployment in web security systems.

References

1. Nirmal K., Janet B., Kumar R. It's more than stealing cookies: exploitability of XSS. In: 2018 Second International Conference on Intelligent Computing and Control Systems. IEEE, 2018, pp. 1073-1077.

2. Rathore S., Sharma P. K., Park J. H. XSS Classifier: an efficient XSS attack detection approach based on machine learning classifier on social networking services. Journal of Information Processing Systems, 2017, 13(4), pp. 1014-1028.

3. Wang R., Jia X., Li Q., et al. Machine learning based cross-site scripting detection in online social networks. In: 2014 IEEE International Conference on High Performance Computing and Communications. IEEE, 2014, pp. 787-794.

Information about the authors

Qiao X., master's student, Educational Institution "Belarusian State University of Informatics and Radioelectronics", suoyter@gmail.com.

Kedo E.S., master's student, Educational Institution "Belarusian State University of Informatics and Radioelectronics", katyakkeda@gmail.com.