Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 6, pp. 1911–1921
DOI : 10.47974/JIOS-2020

A novel deep learning technique for protecting web applications against XSS and SQLI attacks

Sonkarlay J.Y. Weamie *
College of Computer Science and Electronic Engineering
Hunan University
Yuelu District
Changsha 410082
Hunan
China

Vinothkumar Kolluru †
Department of Data Science
Stevens Institute of Technology
1 Castle Point Terrace
Hoboken 07030
NJ
U.S.A.

Yagnesh Challagundla §
Department of Engineering Education
Herbert Wertheim College of Engineering
University of Florida
Gainesville 32611
Florida
U.S.A.

* *ORCID-ID:* **0000-0002-9414-7171**
† *ORCID-ID:* **0009-0006-1713-5720**
§ *ORCID-ID:* **0009-0005-5221-1517**

* *E-mail:* **skweamie@hnu.edu.cn (Corresponding Author)**
† *E-mail:* **vkolluru@stevens.edu**
§ *E-mail:* **yagneshnaidu1234@gmail.com**

© Taru Publications

Thomas T. Tarpeh Jr ‡
School of Computer Science and Engineering
Central South University
Yuelu District
Changsha 410083
Hunan
China

Abstract

The proliferation of emerging technologies has increased vulnerabilities in web applications, making them susceptible to SQL injections and cross-site scripting attacks. Current detection methods often yield high false positive rates. This paper introduces a new technique that improves web application security by decoding and standardizing SQL and XSS_payloads. A hybrid deep learning model was proposed, combining CNN for feature extraction and LSTM for sequential data dependencies. This model achieved impressive results, with accuracies of 99.79% for SQL-XSS Payload, 99.94% for Testbed, and 99.24% for HTTP CSIC 2010 datasets, demonstrating its effectiveness over traditional machine learning methods.

Subject Classification: *68T05, 68N15, 68M10.*

Keywords: *CNNs, Deep learning, LSTM networks, Machine learning, SQLi/XSS.*

1. Introduction

Cross-site scripting and SQL injection attacks remain among the top vulnerabilities that continuously threaten database systems and web applications with increasing sophistication. Even when traditional defense mechanisms such as prepared statements, parameterized queries, and input validation are employed, they provide only limited protection. They are often inadequate in addressing the new tactics used by the adversaries[1]. SQLi types, such as error-based, union-based, and blind injection attacks, exploit database system vulnerabilities to access or modify data without permission [2]. XSS attacks, classified into stored, reflected, DOM-based, and blind XSS, exploit web application vulnerabilities by injecting malicious scripts, compromising user data and disrupting application functionalities [3]. To address these shortcomings, an innovative hybrid CNN-LSTM model is introduced. CNNs are used to extract spatial features of input payloads, and LSTM networks to explore

‡ *ORCID-ID:* `0009-0007-7883-1559`

‡ *E-mail:* `thomastarpeh@gmail.com`

the sequential dependencies in the data set to discover temporal patterns indicative of advanced threats. It detects SQLi, XSS attacks, and protects web applications while reducing false positive rates.

Additionally, this research contributes to developing a novel hybrid model architecture that classifies web requests as benign or malicious, enhances predictive ability over static machine learning models, and proposes a preprocessing framework for better web request data representation. Word2Vec embeddings are applied to convert text data into vector format without sacrificing semantic meaning.

2. Related Work

The reviewed detection methods for SQLi and XSS attacks include traditional algorithms like regular expression matching, SVM, naive Bayes, Random Forests, and decision trees, but these face challenges due to the complexity of URLs [4]. Advanced machine learning techniques have shown promising results, such as fuzzy C-means and neuro-fuzzy systems achieving 98.4% accuracy for SQLi detection [5]. Long short-term memory (LSTM) networks reached 91.53% accuracy in intelligent transportation systems.

In comparison, convolutional neural networks (CNNs) achieved 95.7% and 90.2% accuracy for XSS and SQLi, respectively, with some studies reporting up to 98.5% accuracy for SQLi [6]. Other methods, including residual networks and Graph Convolutional Networks, reported 99.6% accuracy but were unsuitable for JavaScript and HTML [7]. A deep neural network model achieved 96% accuracy, and BERT reached 99.9% on a large dataset [8]. CNNs are effective for analyzing unstructured web

Table 1

Related research by comparison

Model	Accuracy	Precision	Recall	F1-score
CNN [7]	99.5	98.98	1	99.49
SQLNN [10]	96.16	97.28	92.23	94.68
LSTM [11]	98.68	99.85	95.69	97.82
CNN [12]	97.07	-	-	97.51
Neural Network [13]	95	-	-	-
AE-LSTM [14]	87.26	81.22	89.7	81.96
Our Proposed Method	99.94	99.86	99.89	99.87

traffic, while LSTMs excel in sequential data analysis, though both have limitations in real-time applications[9]. This study prefers the combination of CNN and LSTM to prevent SQLi and XSS attacks, as provided in Table 1.

3. Methodology

The dataset utilized in this research undergoes preprocessing before being utilized by the CNN model, after which the features are trained using LSTM models. The training dataset is composed of SQLi/XSS payloads generated with tools such as BurpSuite and the DVWA from the OWASP. Additionally, training is conducted with these payload datasets (SQLI and XSS) alongside other standard datasets, such as the HTTP CSIC2010 dataset. The datasets encompass both benign and malicious payloads in HTTP parameters.

3.1 *Proposed Method and Study Design (Decoding Techniques, Tokenization and Standardization)*

The Word2Vec model uses text classification techniques like generalization, query standardization, and tokenization to process input data and analyze it. It selects features from preprocessed data, which are trained using the CNN-LSTM model. A systematic decoding approach simplifies hyperlinks and removes irrelevant information. The model detects anomalous HTTP request patterns linked to SQL injection attacks. It embeds words post-tokenization and standardization, segmenting repetitive words and mapping them into vector representations. The pooling layer outputs 0 for negative input and Y for positive input, increasing network efficiency as shown in figures 1 to 4.

```
1: GET http://example.com/shop/products.jsp?id=5
  +%27%3B+DROP+TABLE+orders%3B+SELECT+*+FROM+users HTTP/1.1
2: User-Agent: Chrome/90.0.4430.93
3: Pragma: no-cache
4: Cache-control: no-cache
5: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8/plain
6: Accept-Encoding: gzip, deflate, br
```

Figure 1

Example of a malicious SQL query in an HTTP GET request from the CSIC 2010 dataset.

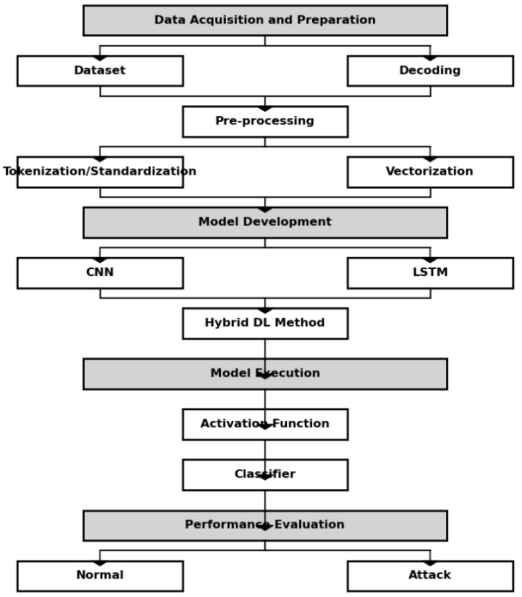


Figure 2

The proposed model architecture for attack detections.

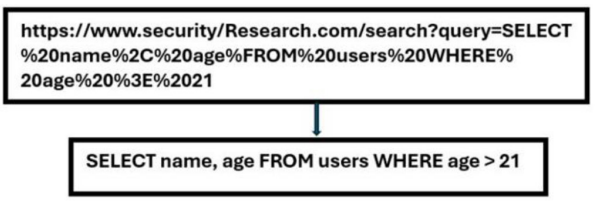


Figure 3

HTTP request URL decoding demo. using an SQL query.

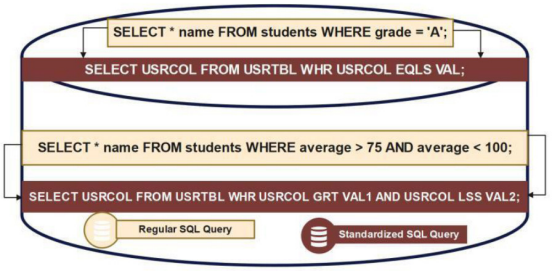


Figure 4

The process of the SQL queries' Standardization.

The functions are mathematically expressed in Equations (1) and (2).

$$f(y) = \max(0, y) \tag{1}$$

$$Zi = CNN(zi) \tag{2}$$

3.2 *LSTM Networks*

LSTM is an RNN architecture that effectively recalls past information using memory blocks. It uses two key gates: an input_gate filters information, and an output_gate decides whether to retain or discard the memory cell's output. Due to adaptive gates, they excel in long-term dependencies and manage cell state updates using three components: multiplying the old state, adding a candidate, and scaling state values. The final output is produced using a sigmoid layer and a hyperbolic tangent function, ensuring values are constrained between -1 and 1. The cell state is multiplied element-wise. The proposed hybrid model architecture and hybrid CNN-LSTM model architecture are shown in Figures 5 and 6. However, the proposed model's parameter settings and

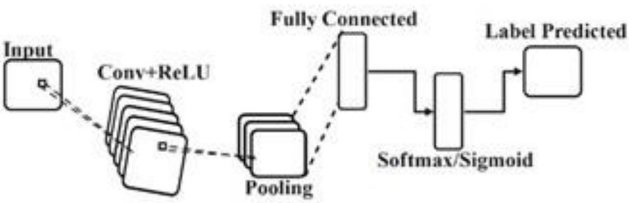


Figure 5

Architecture of CNN.

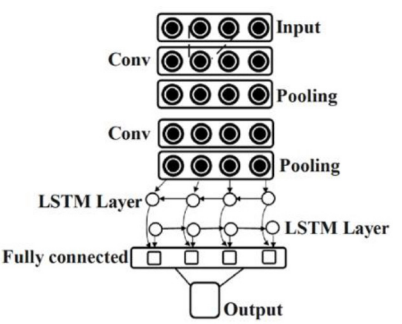


Figure 6

Proposed CNN and LSTM with hybrid model architecture.

structure of each layer parameter and output shape of the CNN-LSTM Architecture are indicated in Tables 2 and Table 3, respectively.

Table 2

The proposed model's parameter settings

Shape Output)	Parameters	Filters	Pool Size	Type (Layer)
[(1000, None)]	0	-	-	Input layer
(1000, 70, None)	4900	-	-	Embedding
(1000, 32, None)	6757	32	-	ConvID
(500, 32, None)	0	-	2	MaxPoolingID
(500, 64, None)	6208	64	-	ConvID
(250, 64, None)	0	-	2	MaxPoolingID
(250, 128, None)	24,704	128	-	ConvID
(125, 128, None)	0	-	-	MaxPolingID
(16,000, None)	0	-	-	Flatten
(64, None)	34,560	-	-	LSTM
(512, None)	8,225,280	-	-	Dense
(512, None)	0	-	-	Dropout
(64, None)	32,832	-	-	Dense
(1, None)	65	-	-	Dense

Table 3

Evaluation of the Proposed structure of each layer parameter and output shape of the CNN-LSTM Architecture.

Value	Hyper-Parameters
Classification function	Sigmoid
Epochs	200/50
Activation Function	ReLU
Optimizer Convolution filters	Adam 128
Fully connected layer	64
Size of the filter's kernel	4

3.3 *Metrics Evaluation*

The evaluation process considers several criteria: accuracy, precision, recall, and F1 score. These metrics quantify the percentage of right

predictions, true positives, and positive instances that are successfully predicted. The formulas for this indicator are shown in Equations (3) to (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 \times \frac{P \times R}{P \times R} \tag{6}$$

4. Discussion and Experimental Results

Our HTTP fraudulent request detection model performs well, as shown in Figures 7 to 9. After 50 training epochs, the model produced relatively accurate results, demonstrating its ability to distinguish between legitimate and malicious queries. For the SQLi-XSS Payload dataset, the model showed an incredible 99.94% accuracy in recognizing and classifying potentially dangerous payloads. The model's consistent reliability and effectiveness in learning SQL injection and XSS vulnerabilities are demonstrated by its ability to generalize well without overfitting.

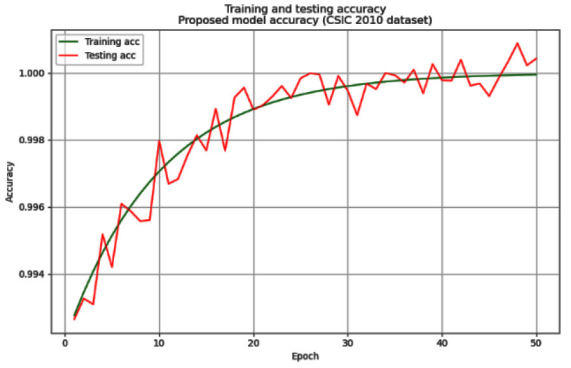


Figure 7

The result of the model training and test accuracy proposed using CSIC 2010 dataset.

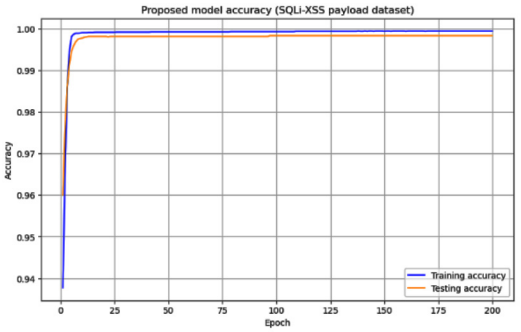


Figure 8

Result of the model's accuracy proposed using SQLi and XSS.

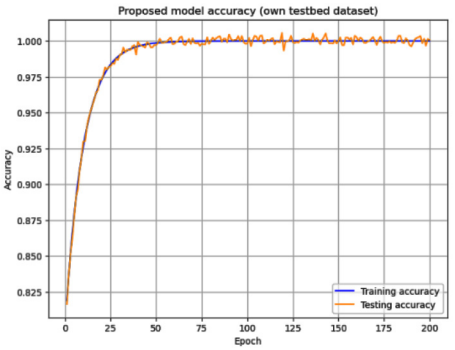


Figure 9

Outcome of the proposed model with our testbed dataset

5. Conclusion

The research utilizes a hybrid CNN and LSTM models to effectively detect SQLi and XSS attacks in web applications. The model exhibits great accuracy and a low false positive rate in experiments, scoring 99.79%, 99. 94% and 99.24% on different datasets. This shows that the hybrid model can secure online applications from these attacks. Future studies may evaluate deeper architectures and approaches for combining CNN and LSTM models with security threat analysis on extra datasets like network meta-data. Detecting malicious URLs and botnets is another promising area for model development. With varied input data and the latest deep learning methods, CNN and LSTM models can improve web application security and counter cyber-attacks.

References

[1] H. Garg, S. Jhunthra, R. Goel, S. Sharma, V. Gupta, R. Sharma, and P. Dass, "Understanding user polarisation regarding COVID-19 vaccines through social network analysis," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 27, no. 8, pp. 2301–2336 (2024).

[2] D. Sree, P. Reddy, G. Reddy, and M. Sumanth, "SQL Injection Attacks: Exploiting Vulnerabilities in Database Systems," pp. 17–23 (2024).

[3] T. K. Mohammed, D. N. Vasundhara, S. H. Mehanoor, E. Sreedevi, P. R. Kumar, C. Manihass, and S. F. Baba, "A novel fusion approach for advancement in crime prediction and forecasting using hybridization of ARIMA and recurrent neural networks," *Journal of Information Systems Engineering and Management,* vol. 10, no. 38, pp. 404–420 (2025).

[4] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative Research on Network Intrusion Detection Methods Based on Machine Learning," *Comput. Secur.,* vol. 121, pp. 102861 (2022).

[5] B. Shilpa, P. R. Kumar, and R. K. Jha, "Spreading factor optimization for interference mitigation in dense indoor LoRa networks," in Proc. IEEE IAS Glob. Conf. Emerg. Technol. (GlobConET), pp. 1–5 (2023).

[6] S. Abaimov and G. Bianchi, "CODDLE: Code-injection detection with deep learning," *IEEE Access,* vol. 7, pp. 128617–128627 (2019).

[7] N. Arivazhagan, K. Somasundaram, D. Vijendra Babu, M. Gomathy Nayagam, R. M. Bommi, G. B. Mohammad, and V. Prabhu Sundramurthy, "Cloud-Internet of Health Things (IOHT) Task Scheduling Using Hybrid Moth Flame Optimization with Deep Neural Network Algorithm for E Healthcare Systems," *Sci. Program.,* Art. no. 4100352 (2022).

[8] P. Dass, V. Gupta, S. Dhingra, R. Arora, P. Katariya, and A. Kumar, "A deep learning model to predict and classify crime rate using tweets," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 8, pp. 2255–2271 (2024).

[9] R. Ranjan, D. Pandey, A. K. Rai, P. Singh, A. Vidyarthi, D. Gupta, and S. N. Mohanty, "A manifold-level hybrid deep learning approach for sentiment classification using an autoregressive model," *Appl. Sci.*, vol. 13, no. 5, pp. 3091 (2023).

[10] M. Hasani and M. Ghods, "Investigating fuzzy topological indices of linear and multi-cyclic Naphtalene," *Journal of Interdisciplinary Mathematics,* vol. 28, no. 5, pp. 1779–1793 (2025).

[11] P. R. Kumar and B. Shilpa, "An IoT-based smart healthcare system with edge intelligence computing," in Reconnoitering the Landscape of Edge Intelligence in Healthcare, Apple Academic Press, pp. 31–46 (2024).

[12] G. B. Mohammad, Shitharth, and P. R. Kumar, "Integrated machine learning model for URL phishing detection," *International Journal of Grid and Distributed Computing*, vol. 14, no. 1, pp. 513–529 (2021).

[13] D. Goyal, A. Kumar, and D. S. Jat, "Special issue on advancements in intelligent systems for information optimization in cyber security: Challenges and innovations (Part-B)," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 5-B, pp. i–viii (2025).

[14] B. Shilpa, P. R. Kumar, and R. K. Jha, "LoRa DL: A deep learning model for enhancing the data transmission over LoRa using autoencoder," *J. Supercomput.,* vol. 79, no. 15, pp. 17079–17097 (2023).

Received December, 2024