IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15804

Smart SQL Injection Attack Detection Using Hybrid Deep Neural Network

Mrs. T Geetha M.E., MBA., (Ph. D) [*1], Manjuparkavi S[2], Mithunapriya M[3], Navitha G[4], Rajeshwari R[5]

Head of the Department, Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, India[1]

Students, Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, India[2-5]

Abstract: SQL Injection attacks are one of the most critical and frequently occurring security threats in modern web applications. These attacks exploit vulnerabilities in input handling to inject malicious SQL statements, enabling attackers to bypass authentication, retrieve confidential data, modify database contents, or even gain complete control over backend systems. Traditional defense mechanisms such as manual input validation, rule-based filtering, and signature-based intrusion detection systems are limited in their ability to detect complex, obfuscated, and previously unseen attack patterns. This project proposes a **Smart SQL Injection Attack Detection System** using a **Hybrid Deep Neural Network (CNN + LSTM)** to effectively identify both known and unknown SQL injection attacks. Incoming SQL queries are preprocessed through tokenization and encoding, converted to numerical form using TF-IDF, and refined with Chi-Square feature selection. CNN layers capture local attack signatures while LSTM layers model sequential and contextual relationships among query tokens, jointly improving learning capability and detection accuracy. The model is trained and evaluated on benchmark and real-world SQL injection datasets using accuracy, precision, recall, and F1-score. Experimental results show detection accuracy consistently above 95%, with lower false positive rates than traditional and single-model detection techniques, and the system is suitable for real-time deployment in existing web application security infrastructures.

Keywords: SQL Injection, Cybersecurity, Deep Learning, Hybrid Neural Network, CNN, LSTM, Web Security, Attack Detection.

I. INTRODUCTION

The rapid advancement of internet technologies and the widespread adoption of web-based applications have significantly transformed the way information is accessed, processed, and stored in modern society. Today, web applications play a crucial role in various domains such as banking, healthcare, education, e-commerce, and government services, where large volumes of sensitive and confidential data are handled on a daily basis. While this technological evolution has improved efficiency and accessibility, it has also introduced serious security challenges, making web applications prime targets for cyberattacks that exploit design and implementation vulnerabilities.

Among the various types of cyber threats, SQL Injection (SQLi) attacks are considered one of the most critical and commonly exploited vulnerabilities in database-driven web applications. SQL injection occurs when an attacker inserts malicious SQL statements into input fields such as login forms, search boxes, or URL parameters, with the intention of manipulating the backend database. If the application fails to properly validate or sanitize user inputs, these malicious queries are executed by the database, allowing attackers to gain unauthorized access to sensitive information, including data breaches, unauthorized data modification, and complete system compromise.

Traditionally, various techniques have been developed to detect and prevent SQL injection attacks, including rule-based filtering, signature-based detection systems, and web application firewalls. These methods rely on predefined patterns or known attack signatures, and while effective against well-known attack types, they have significant limitations when dealing with modern, obfuscated attack techniques such as encoding, comment insertion, whitespace manipulation, and dynamic query restructuring — often failing to detect unknown or zero-day attacks and generating high false-positive rates.

In recent years, machine learning and deep learning have emerged as promising solutions for addressing complex cybersecurity challenges. Deep Neural Networks (DNNs) automatically extract meaningful features from raw input data, eliminating the need for manual feature engineering, which makes them highly suitable for analyzing complex,

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15804

unstructured data such as SQL queries. Convolutional Neural Networks (CNNs) are particularly effective at extracting local features and identifying keywords, operators, and special-character combinations commonly used in injection attacks, while Long Short-Term Memory (LSTM) networks capture sequential dependencies and contextual relationships between query tokens.

However, a single deep learning model may not capture all aspects of SQL injection attacks: CNNs excel at local patterns but may miss long-term dependencies, while LSTMs are strong in sequence analysis but may not extract fine-grained features. To overcome this, a Hybrid Deep Neural Network (HDNN) that combines CNN and LSTM is proposed, analyzing both structural and sequential characteristics of SQL queries to improve detection accuracy and robustness. The system collects legitimate and malicious SQL queries, preprocesses and normalizes them, extracts features using TF-IDF, applies Chi-Square feature selection, and classifies queries as normal or malicious in real time — aiming for high accuracy, low false positives, and detection of both known and unknown attack patterns.

II. LITERATURE SURVEY

The problem of SQL Injection (SQLi) attacks has been extensively studied due to its critical impact on web application security. Early research focused on traditional detection and prevention techniques such as rule-based filtering, static analysis, and signature-based intrusion detection, providing classifications of SQL injection attacks and countermeasures including input validation and query sanitization. While effective for well-known attack patterns, these approaches lacked the ability to handle newly emerging and obfuscated attacks, requiring continuous manual updates.

As these limitations became evident, researchers explored machine learning techniques such as Support Vector Machines (SVM), Decision Trees, Naive Bayes, and Random Forests, treating SQL injection detection as a classification problem. These models generalized better than rule-based systems but depended heavily on manual feature engineering, which is time-consuming and may not capture complex query structures effectively.

With the advancement of deep learning, researchers shifted toward neural-network-based approaches that automatically learn hierarchical feature representations. CNNs have been applied to detect local patterns such as keywords, operators, and special-character combinations, showing promising results for known attacks but limited ability to capture sequential and contextual dependencies. LSTM networks were subsequently introduced to capture long-term dependencies and token sequences, improving detection of complex and obfuscated attacks, though alone they may not efficiently capture fine-grained local features.

Recent studies have proposed hybrid approaches combining CNN and LSTM models, where the CNN component extracts local features and the LSTM component captures sequential and contextual relationships. Such hybrid approaches, combined with preprocessing and feature-selection techniques like TF-IDF and Chi-Square, have been shown to improve detection accuracy and reduce false positives compared to single-model approaches.

Overall, the literature shows a clear progression from rule-based systems, to machine learning, to deep learning and hybrid architectures. The proposed work builds upon these advancements by implementing a Hybrid Deep Neural Network combining CNN and LSTM to achieve improved accuracy, efficiency, and real-time detection capability for securing web applications against SQL injection attacks.

III. METHODOLOGY

The rapid increase in web-based applications has led to a significant rise in SQL Injection (SQLi) attacks, which pose serious threats to database security. To address this, a smart SQL injection attack detection system using a Hybrid Deep Neural Network (HDNN) is proposed, integrating preprocessing, feature extraction, and CNN-LSTM deep learning models to accurately classify SQL queries as normal or malicious. This section describes the system architecture, data collection, preprocessing, feature extraction, model design, and performance evaluation.

A. System Architecture

The proposed architecture consists of a user input interface, preprocessing module, feature extraction unit, hybrid deep learning model, and classification module. SQL queries are collected and passed through preprocessing, then converted into numerical representations using TF-IDF. These features feed into the hybrid CNN-LSTM network — the CNN layer extracts local patterns while the LSTM layer captures sequential dependencies — and the classification layer

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15804

determines whether the query is legitimate or malicious, ensuring accurate, real-time detection with reduced false positives.

B. SQL Query Data Collection

The dataset consists of both normal and malicious SQL queries, including login authentication queries, search queries, and database transactions. Malicious queries include tautology-based, union-based, error-based, and comment-based injection types. A diverse dataset helps the model generalize; the data is split into training and testing sets for effective evaluation.

C. Preprocessing of SQL Queries

Unnecessary symbols, special characters, and redundant spaces are removed and queries are standardized. Tokenization breaks queries into keywords, operators, and identifiers, and normalization (e.g., lowercase conversion) ensures a consistent format, reducing noise for downstream deep learning analysis.

D. Feature Extraction and Selection

Term Frequency–Inverse Document Frequency (TF-IDF) measures the importance of words in each query, identifying terms that distinguish normal from malicious queries. Chi-Square feature selection further reduces dimensionality and computational complexity while improving detection performance.

E. Hybrid Deep Neural Network Model

The CNN layer extracts local features such as suspicious keywords, operators, and special-character patterns using convolutional filters, while the LSTM layer processes sequential data and captures long-term dependencies between query components. Combining both allows the model to analyze structural and contextual information effectively, improving detection accuracy over either model alone.

F. Model Training and Classification

The hybrid model is trained on labeled datasets using backpropagation and gradient descent to minimize classification error. Incoming queries classified as normal are allowed to execute, while those flagged as malicious are blocked, ensuring secure database operations.

G. Real-Time SQL Injection Detection

Incoming queries are processed through the trained model with minimal delay, continuously monitoring for anomalies. Detected malicious queries are prevented from reaching the database and trigger an alert, enhancing real-time web application security.

H. Performance Evaluation

The system is evaluated on standard datasets using accuracy, precision, recall, and F1-score across different SQL injection attack types. Results demonstrate high accuracy, effective detection of complex and obfuscated attacks, reduced false positives, and low computational overhead, confirming suitability for real-world deployment.

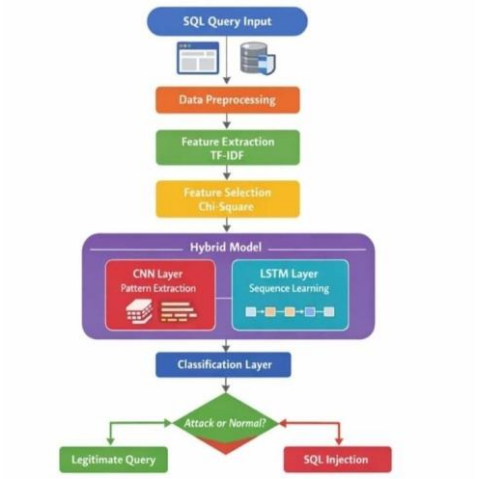

Fig. 1. Architecture Diagram

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15804

IV. RESULT AND DISCUSSION

The proposed smart SQL injection attack detection system using a Hybrid Deep Neural Network (HDNN) was evaluated on a dataset containing both normal and SQL injection queries, including tautology-based and union-based injections. The hybrid CNN-LSTM model achieves high detection accuracy, consistently above 95%, by effectively capturing both structural patterns and sequential relationships within SQL queries, with low false positive and false negative rates that ensure reliable classification without affecting legitimate queries.

In terms of performance, the model provides fast and efficient detection suitable for real-time web applications, with a lightweight architecture that ensures minimal computational overhead while maintaining high accuracy.

The system also shows strong resistance to obfuscated and complex attack patterns, successfully identifying previously unseen SQL injection attempts. Overall, the results confirm that the proposed hybrid approach improves detection accuracy, efficiency, and robustness compared to traditional methods.

V. CONCLUSION

The proposed smart SQL injection attack detection system using a Hybrid Deep Neural Network provides an effective solution for enhancing web application security. By combining CNN and LSTM models, the system successfully analyzes both structural and sequential features of SQL queries, resulting in high detection accuracy and improved reliability, while overcoming the limitations of traditional detection methods by reducing false positives and detecting complex and unknown attack patterns.

The system ensures real-time detection with low computational cost, making it suitable for practical deployment in modern web applications. Overall, the proposed model offers a secure, efficient, and scalable solution for preventing SQL injection attacks. Future improvements can focus on expanding datasets and enhancing model performance for large-scale applications.

REFERENCES

[1]. W. G. Halfond, J. Viegas, and A. Orso, “A classification of SQL-injection attacks and countermeasures,” Proc. IEEE Int. Symp. Secure Software Engineering, pp. 13–15, 2006.
[2]. S. W. Boyd and A. D. Keromytis, “SQLrand: Preventing SQL injection attacks,” Proc. 2nd Applied Cryptography and Network Security Conf., pp. 292–302, 2004.
[3]. M. Valeur, D. Mutz, and G. Vigna, “A learning-based approach to the detection of SQL attacks,” Proc. Conf. Detection of Intrusions and Malware & Vulnerability Assessment, Springer, pp. 123–140, 2005.
[4]. Y. Shin, L. Williams, and T. Xie, “SQLUnitGen: Test case generation for SQL injection detection,” Proc. Int. Symp. Software Reliability Engineering, pp. 13–23, 2006.
[5]. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[6]. F. Chollet, “Deep learning with Python,” Shelter Island, NY, USA: Manning Publications, 2018.
[7]. M. Abadi et al., “TensorFlow: A system for large-scale machine learning,” Proc. 12th USENIX Symp. Operating Systems Design and Implementation, pp. 265–283, 2016.
[8]. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[9]. A. Karpathy, “Deep learning for cybersecurity: Intrusion detection and threat analysis,” IEEE Security & Privacy, vol. 16, no. 5, pp. 56–63, Sep. 2018.
[10]. OWASP Foundation, “SQL Injection,” OWASP Top 10 Web Application Security Risks, 2021.
[11]. S. Kumar and A. Sharma, “Detection of SQL injection attacks using machine learning techniques,” Int. Journal of Computer Applications, vol. 182, no. 35, pp. 15–21, 2019.
[12]. N. Antunes and M. Vieira, “Detecting SQL injection vulnerabilities in web services,” Proc. IEEE Int. Conf. Services Computing, pp. 17–24, 2011.
[13]. H. Shahriar and M. Zulkernine, “Mitigating SQL injection attacks using query transformation and hashing,” Proc. IEEE Int. Conf. Software Maintenance, pp. 111–120, 2012.
[14]. A. Alghamdi, M. Alotaibi, and A. Alzahrani, “SQL injection attack detection using deep learning techniques,” Int. Journal of Advanced Computer Science and Applications, vol. 11, no. 6, pp. 150–157, 2020.
[15]. J. Kim and H. Kim, “Hybrid deep learning approach for web application attack detection,” IEEE Access, vol. 8, pp. 123456–123465, 2020.

