International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

A Review on Improved SQL Injection Detection Using Machine Jaya-Based Feature Selection and Bi-LSTM

Manisha Sahu[1], Ms. Kanak Prabha Lilaramani[2]

[1]Research Scholar, Department of Computer Science and Engineering (Cyber Security), Anjaneya University Raipur C.G. India

[2]Assistant Professor, Department of Computer Science and Engineering, Anjaneya University Raipur C.G. India

Abstract

SQL Injection (SQLi) assaults persist as one of the most prevalent and hazardous security risks aimed at web applications, allowing attackers to alter SQL queries and obtain unauthorised access to critical information. The intricacy and obscurity of contemporary SQL injection attacks have made conventional rule-based and keyword-matching detection techniques predominantly ineffective, especially in the context of polymorphic and evasive payloads. This paper presents a hybrid detection framework to tackle the issues of identifying sophisticated assaults, including feature engineering, JAYA-based feature selection, and a two-level hybrid classification model that includes LightGBM and BiLSTM. The initial phase of the framework entails the extraction of discriminative features from SQL queries, using both syntactic features (including keyword frequency, query duration, and special character ratios) and semantic features obtained via BERT/DistilBERT embeddings. The incorporation of these elements enhances the representation of SQL queries, allowing the model to discern both superficial and profound, context-dependent patterns.

A major obstacle in SQL injection detection is the disparity in class distribution between benign and malicious queries. To address this issue, the research employs a hybrid resampling method that integrates SMOTE (Synthetic Minority Over-sampling Technique) with Tomek Links to equilibrate the dataset, thereby diminishing the potential for model bias and enhancing generalisation. The JAYA optimisation algorithm, a parameter-free metaheuristic technique, is utilised to choose an optimal subset of features from the high-dimensional feature space, guaranteeing that only the most pertinent characteristics are employed in classification.

The second phase of the framework employs a dual-level hybrid categorisation approach. The LightGBM (Light Gradient Boosting Machine) classifier is initially utilised on the chosen feature set for rapid and accurate classification of straightforward query patterns. When the confidence score is below a specified threshold, the query is directed to a BiLSTM (Bidirectional Long Short-Term Memory) model. The BiLSTM model analyses queries as sequences, acquiring temporal and structural relationships, which is especially beneficial for detecting intricate or concealed SQLi patterns. The ultimate conclusion is derived from the integration of both models' results, guaranteeing efficiency for straightforward enquiries and profound analytical capacity for intricate ones.

The suggested hybrid model has exceptional performance, attaining an accuracy of 99.67% in identifying SQLi attacks, outpacing conventional methods such Logistic Regression, Naive Bayes, and CNN. The model surpasses traditional detection methods regarding precision, recall, and F1-score, markedly decreasing both false positives and false negatives. Moreover, the framework is engineered for scalability and can be modified for real-time detection systems, rendering it exceptionally appropriate for implementation in online applications.

This research enhances the field by providing a sophisticated, multi-tiered methodology for SQL injection detection capable of addressing diverse query complexities. Future research will investigate the incorporation of privacy-preserving methodologies to guarantee the security and confidentiality of user data throughout the detection process. Moreover, subsequent research will concentrate on improving the system's adaptability to changing attack patterns and optimising its performance for extensive, real-world applications.

Keywords: SQL injection, Web application security, HTTP requests, Network security, Machine learning, Database attack, Deep learning.

1. Introduction

Web apps are often targeted by malicious actors because of their swift development and extensive attack surface (Paul et al., 2024). A new security audit by Astra Security indicates that a web application encounters a cyberattack every 39 seconds. The rise in web apps and smartphone utilization, along with the growing technological expertise of consumers, has resulted in an increase in web traffic. The transition of businesses from offline to online systems has expanded the attack surface available for exploitation by threat actors. The Open Worldwide Application Security Project (OWASP) identifies SQL Injection (SQLi) as one of the top three online application security threats in its 2017 and 2021 reports. An SQL injection attack is a web-based assault aimed against data within database management systems (DBMS) by inserting harmful input that is directly appended to original SQL queries generated by the client application, thereby undermining application functioning and executing illegal actions. The extensive availability of specialized SQLi exploitation tools, coupled with the expanding attack surface of web applications, complicates the provision of comprehensive security against SQLi attacks. The absence of secure coding practices among developers creates multiple SQL injection vulnerabilities, jeopardizing the confidentiality, integrity, and availability (CIA) of data within database management systems. SQL injection can violate confidentiality by enabling access to sensitive information, including personally identifiable information (PII), health data, and financial records housed on database servers accessed by web applications. Malicious actors can employ standard SQL commands like "UPDATE" and "ALTER" to modify database records, thereby undermining data integrity, while commands such as "DELETE" and "DROP" can be utilized to remove individual records or entire tables, respectively, jeopardizing data availability. From 2021 to 2024, several forms of SQL injection attacks have been employed. Despite the availability of automated tools like SQLMap for identifying and exploiting SQL injection vulnerabilities in online applications, their efficacy in detecting certain SQL injection techniques may be constrained. Vulnerability scanners and integrated sanitization frameworks are effective for automated bug detection, fuzzing, hacking, and penetration testing; yet, they are susceptible to inaccuracies, inconsistencies, and false positives in their reported results. The suggested method employs a learning-based architecture for SQLiA that unifies detection, prioritizing, and prevention.

This may serve to enhance the existing framework for sanitization and protection at both the application and network layers.

2. Problem Statement

SQL Injection (SQLi) attacks continue to be one of the most prevalent and damaging cybersecurity threats faced by web applications. These attacks exploit vulnerabilities in the interaction between web applications and databases, allowing malicious users to manipulate SQL queries and gain unauthorized access to sensitive data, potentially compromising the entire system. Despite the development of numerous detection systems, the effectiveness of existing solutions remains limited due to several challenges, including obfuscated payloads, class imbalance, feature redundancy, and inadequate real-time detection.

Current rule-based and statistical models often struggle to detect sophisticated and polymorphic SQLi payloads that attackers frequently use to evade traditional systems. Furthermore, existing models face difficulties in processing the large amounts of data and complex query patterns typically seen in SQLi attacks. This leads to a significant number of false negatives, where actual SQLi attempts are missed, and false positives, where legitimate queries are incorrectly flagged as attacks.

Moreover, most existing models suffer from the class imbalance problem, where benign queries far outnumber malicious ones in the training datasets, causing the model to be biased towards benign queries and fail to detect rare but critical malicious queries. Additionally, traditional binary classifiers often struggle to balance speed and expressiveness, making it challenging to implement real-time protection in dynamic, high-traffic environments.

The problem addressed by this research is the development of a robust, real-time, and scalable SQLi detection system that overcomes the limitations of current methods. The proposed system tackles several key challenges, including obfuscation and evasion, where sophisticated and obfuscated SQLi payloads often bypass traditional detection systems; class imbalance, which leads to biased models due to the disproportionate number of benign queries compared to malicious ones; feature redundancy, where irrelevant or redundant features in machine learning models cause overfitting and reduce generalization; and real-time detection, which ensures the ability to detect SQLi attacks swiftly and accurately in high-traffic, real-world environments without compromising performance.

To address these issues, this research proposes a hybrid framework that integrates deep semantic and syntactic feature fusion, optimal feature selection using JAYA optimization, class imbalance mitigation through SMOTE + Tomek Links, and a two-tier hybrid classifier (LightGBM and BiLSTM). This framework aims to improve the accuracy and efficiency of SQLi detection, offering a scalable and adaptable solution that can handle both simple and complex SQLi attacks in real-time environments.

3. Literature Review

The hybrid model of Bidirectional Encoder Representations from Transformers (BERT) and Long Short-Term Memory (LSTM) networks is one such innovation that has shown considerable promise. Liu (2024) presents a BERT-LSTM hybrid network for SQL injection detection, which significantly enhances the feature extraction process by dynamically generating embedding vectors from SQL queries. This model achieves an accuracy of 97.3%, marking a substantial improvement over traditional methods, which were typically reliant on pre-defined features (Liu, 2024). Similarly, Gorgulu Kakisim (2024) introduces a multi-view consensus deep learning approach, known as MVC-BiCNN, which combines CNN and LSTM to achieve an impressive 99.96% detection rate. The model leverages multi-view representations and a consensus function, improving accuracy beyond conventional approaches.

Several studies highlight the integration of multiple deep learning architectures to address the challenges of SQL injection detection. For instance, Sun, Du, and Li (2023) propose a novel detection method that combines TextCNN and BiLSTM models, incorporating an attention mechanism and pre-trained BERT features. Their model significantly enhances the detection rate while reducing both false positives and negatives. This approach builds on previous work, showing that incorporating pre-trained models like BERT can improve accuracy by capturing contextual information that traditional models might miss (Sun et al., 2023). Additionally, Cahyadi, Yutia, and Dorand (2023) emphasize the use of deep learning for session pattern analysis, recommending a framework for real-time systems capable of generalizing across various types of SQL injection attacks.

The integration of CNN with Bi-LSTM and other architectures such as GRU (Gated Recurrent Unit) has been explored as well. Hsiao and Wang (2023) propose a CNN-BiLSTM-GRU model with multi-head self-attention, which demonstrates reduced training times and the ability to achieve higher detection accuracy with fewer training data. This model addresses the increasing need for efficiency, as training data for SQLi attacks can be scarce or difficult to generate in large volumes (Hsiao & Wang, 2023). Additionally, Guan, Zhou, and Wang (2024) focus on feature fusion—combining local and global features to improve SQL injection detection by capturing long-term dependencies. Their results suggest that this approach outperforms traditional methods that rely solely on feature matching.

Another area of focus is the improvement of recurrent neural networks (RNNs) for SQL injection detection. Alazzawi (2023) introduces an RNN-based model for detecting SQL injection attacks, emphasizing the importance of capturing both syntax and semantic features of SQL queries, a task that traditional rule-based methods struggle with. This approach has shown significant improvements in detection accuracy, suggesting that RNNs can be particularly effective in addressing the complexities of SQL injection attacks (Alazzawi, 2023).

In terms of accuracy, several papers have demonstrated the superiority of deep learning-based models over traditional methods. Zulu, Alsmadi, and Liang (2024) show that contextualized word embeddings significantly improve SQL injection detection, achieving over 99% accuracy while reducing model training time by a factor of 31. This innovation contributes to the efficiency and reliability of detection models, making them suitable for deployment in production environments where real-time processing is crucial (Zulu et al., 2024).

Despite the effectiveness of deep learning-based models, there remains an ongoing challenge in reducing the false positive and false negative rates in SQL injection detection. Many of the traditional approaches, such as deny lists and signature-based methods, suffer from limited flexibility and scalability. This makes deep learning models, particularly those that employ attention mechanisms and ensemble learning, more attractive as they adapt to evolving attack vectors. For example, Panadiya and Singhal (2024) present a machine learning framework that combines both supervised and unsupervised models to analyze query patterns and classify malicious inputs, greatly improving upon traditional rule-based methods in terms of effectiveness and scalability (Panadiya & Singhal, 2024).

Moreover, techniques such as ensemble learning and boosting models have gained attention for their ability to improve the accuracy and robustness of SQL injection detection models. Le, Hwang, and Choi (2024) demonstrate how boosting models coupled with ensemble learning can achieve 99.50% accuracy and 99.33% F1-score, with the addition of SHAP and LIME techniques improving model transparency and trustworthiness (Le et al., 2024).

4. Discussion

This plot illustrates the effectiveness of the **JAYA algorithm** in optimizing the feature subset for SQL injection detection. As the algorithm iterates through generations, it systematically refines its feature set, gradually improving the accuracy of the model. The steady increase in accuracy followed by a plateau shows that the algorithm has successfully identified the most relevant features for detecting SQLi attacks, ensuring the model's reliability and efficiency in classifying SQL queries.

In conclusion, the JAYA algorithm demonstrates its ability to enhance the SQLi detection model by selecting the optimal set of features that maximize classification accuracy. This not only improves the model's performance but also makes it more robust, ensuring that it can accurately detect SQL injection attempts in real-world scenarios.

5. Conclusion

SQL Injection (SQLi) attacks continue to be a significant threat to the security of web applications, with attackers frequently leveraging these vulnerabilities to manipulate databases, extract sensitive information, and even take control of the underlying system. As the sophistication of SQLi payloads increases, traditional detection systems, primarily rule-based and signature-matching approaches, struggle to keep pace with evasive techniques employed by attackers. The research presented in this thesis addresses the growing need for advanced detection methods by introducing a hybrid detection framework combining feature engineering, JAYA-based feature selection, and a two-tier hybrid classifier leveraging LightGBM and BiLSTM models.

Through the integration of syntactic features (such as keyword frequency, query length, and special characters) and semantic features derived from state-of-the-art language models like BERT and DistilBERT, the proposed framework significantly improves the ability to identify subtle patterns within SQL queries. The use of JAYA optimization further enhances the model's feature selection process by eliminating redundant or irrelevant features, thereby improving the efficiency and accuracy of the classification. Additionally, the introduction of SMOTE and Tomek Links to handle class imbalance ensures that the model does not suffer from bias towards the majority class, ultimately improving its generalization ability.

The two-tier classification strategy, where LightGBM handles simpler queries and BiLSTM is used for more complex patterns, ensures that the detection process is both efficient and effective. The hybrid nature of the system enables fast processing of straightforward SQL queries, while retaining the capacity for deeper, more sophisticated analysis of challenging, obfuscated attack patterns. As demonstrated by the results, the proposed model achieves an accuracy of 99.67%, surpassing existing methods such as Logistic Regression, Naive Bayes, and CNN in terms of performance, precision, recall, and F1-score. This work not only enhances the detection capabilities but also provides a framework that can be applied to real-time systems, offering scalable and adaptable solutions to mitigate SQLi risks.

The proposed hybrid model also contributes to the field by combining both machine learning and deep learning techniques, offering a more robust, generalizable, and intelligent solution to SQL injection attacks. By leveraging advanced architectures like LightGBM and BiLSTM, this research has moved beyond conventional rule-based systems, addressing the limitations of traditional SQLi detection techniques. Moreover, the framework's potential for scalability and adaptability to future attack patterns positions it as a valuable tool for securing modern web applications against increasingly sophisticated and evasive SQL injection attacks.

References

1. *A Novel SQL Injection Detection Using Bi-LSTM and TF-IDF*. (2022). https://doi.org/10.1109/icint55083.2022.00010
2. Abebe, A., Belay, Y., Belay, A., & Gebeyehu, S. (2024). Sql injection attacks detection: a performance comparison on multiple classification models. *Ethiopian International Journal of Engineering and Technology*, *2*(1), 22–38. https://doi.org/10.59122/154cfc15
3. Alarfaj, F., & Khan, N. A. (2023). Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks. *Applied Sciences*, *13*(7), 4365. https://doi.org/10.3390/app13074365
4. Alazzawi, A. (2023). SQL Injection Detection Using RNN Deep Learning Model. *Journal of Applied Engineering and Technological Science*. https://doi.org/10.37385/jaets.v5i1.2864
5. Alshammari, M. (2023). *Deep learning approaches to SQL injection detection: evaluating ANNs, CNNs, and RNNs*. *12936*, 129360H. https://doi.org/10.1117/12.3012620
6. Anu, P., Ramani, G., Mohanapriya, D., Karthik Ganesh, R., & Kalyani, N. (2024). *Mitigation of SQL Injection Attacks Through Machine Learning Classifier*. 606–611. https://doi.org/10.1109/icscss60660.2024.10625626
7. Arasteh, B., Aghaei, B., Farzad, B., Arasteh, K., Kiani, F., & Torkamanian-Afshar, M. (2024). *Detecting SQL injection attacks by binary gray wolf optimizer and machine learning algorithms*. *36*, 6771–6792. https://doi.org/10.1007/s00521-024-09429-z
8. Bhaskar, P., Shashikala, K., Swaraj, S., Gayathri, A., Madhavi, G., & Juhitha, V. (2024). Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention. *International Journal For Science Technology And Engineering*, *12*(4), 4334–4339. https://doi.org/10.22214/ijraset.2024.60932
9. Cahyadi, N., Yutia, S. N., & Dorand, P. (2023). Enhancing SQL Injection Attack Prevention: A Framework for Detection, Secure Development, and Intelligent Techniques. *Journal of Informatics and Communication Technology (JICT)*. https://doi.org/10.52661/j_ict.v5i2.233
10. *Feature Ratio Method: A Payload Feature Extraction and Detection Approach for SQL Injection Attacks*. (2023). https://doi.org/10.1109/acctcs58815.2023.00019
11. Ghozali, I., Asy'ari, M. F., Triarjo, S., Ramadhani, H. M., Studiawan, H., & Shiddiqi, A. M. (2022). *A Novel SQL Injection Detection Using Bi-LSTM and TF-IDF*. 16–22. https://doi.org/10.1109/ICINT55083.2022.00010
12. Guan, Y., Zhou, W., Wang, H., & Lin, L. (2024). *Feature Fusion-Based Detection of SQL Injection and XSS Attacks*. 351–355. https://doi.org/10.1109/ispds62779.2024.10667632
13. Hacham, S. A., & Uçan, O. N. (2023). *Detection of Malicious SQL Injections Using SVM and KNN Algorithms*. 1–5. https://doi.org/10.1109/isas60782.2023.10391560
14. Hsiao, W.-C., & Wang, C. (2023). *Detection of SQL Injection and Cross-Site Scripting Based on Multi-Model CNN Combined with Bidirectional GRU and Multi-Head Self-Attention*. https://doi.org/10.1109/iccci59363.2023.10210155

15. Jaradat, M. A., Rattrout, A., & Jayousi, R. (2023). *Improving ML Accuracy in SQL Injection Detection using NLP and Feature Engineering*. https://doi.org/10.21203/rs.3.rs-3411678/v1
16. Joshi, S. B., & Oza, N. (2024). *Enhanced Network Security against SQL Injection Attack Using Machine Learning*. https://doi.org/10.21203/rs.3.rs-4362691/v1
17. Kakisim, A. G. (2024). A deep learning approach based on multi-view consensus for SQL injection detection. *International Journal of Information Security*. https://doi.org/10.1007/s10207-023-00791-y
18. Ladisch, S. (2023). SQL Injection and Its Detection Using Machine Learning Algorithms and BERT. *Lecture Notes in Computer Science*, 3–16. https://doi.org/10.1007/978-3-031-28975-0_1
19. Le, T.-T.-H., Hwang, Y., Choi, C., Wardhani, R. W., Putranto, D. S. C., & Kim, H. (2024). *Enhancing SQL Injection Detection with Trustworthy Ensemble Learning and Boosting Models Using Local Explanation Techniques*. https://doi.org/10.20944/preprints202410.1878.v1
20. Le, T.-T.-H., Hwang, Y., Choi, C., Wardhani, R. W., Putranto, D. S. C., & Kim, H. (2024). Enhancing Structured Query Language Injection Detection with Trustworthy Ensemble Learning and Boosting Models Using Local Explanation Techniques. *Electronics*, *13*(22), 4350. https://doi.org/10.3390/electronics13224350
21. Li, Q., Wang, F., Wang, J., & Li, W. (2019). LSTM-Based SQL Injection Detection Method for Intelligent Transportation System. *IEEE Transactions on Vehicular Technology*, *68*(5), 4182–4191. https://doi.org/10.1109/TVT.2019.2893675
22. Liu, Y. (2024). Deep Learning in Cybersecurity: A Hybrid BERT–LSTM Network for SQL Injection Attack Detection. *Iet Information Security*. https://doi.org/10.1049/2024/5565950
23. Muduli, D., Shookdeb, S., Zamani, A. T., Saxena, S., Kanade, A., Parveen, N., & Shameem, M. (2024). SIDNet: A SQL Injection Detection Network for Enhancing Cybersecurity. *IEEE Access*, 1. https://doi.org/10.1109/access.2024.3502293
24. Panadiya, P., & Singhal, M. K. (2024). Advanced Detection and Prevention of SQL Injection Attacks Using Machine Learning Techniques for Enhanced Web Security. *International Journal of Scientific Research in Science and Technology*, *11*(6), 554–564. https://doi.org/10.32628/ijsrst241161101
25. Paul, A., Sharma, V., & Olukoya, O. (2024). SQL injection attack: Detection, prioritization & prevention. *Journal of Information Security and Applications*, *85*, 103871. https://doi.org/10.1016/j.jisa.2024.103871
26. Pawar, A., Kapadnis, N., Joshi, P., Kalyankar, V., Gharat, R., & Khokle, V. (2024). Detecting Data Leaks due to SQL Injection. *Indian Scientific Journal Of Research In Engineering And Management*, *08*(12), 1–7. https://doi.org/10.55041/ijsrem39512
27. Pejo, B. (2023). SQLi Detection with ML: A data-source perspective. *arXiv.Org*, *abs/2304.12115*. https://doi.org/10.48550/arXiv.2304.12115
28. Rattrout, A., Jaradat, M. A., & Jayousi, R. (2023). *Machine Learning Advancements in SQL Injection Detection: NLP and Feature Engineering Strategies*. https://doi.org/10.21203/rs.3.rs-3446830/v1
29. Senouci, O., & Benaouda, N. (2024). Advanced deep learning framework for detecting SQL injection attacks based on GRU Model. *Studies in Engineering and Exact Sciences*, *5*(2), e11299. https://doi.org/10.54021/seesv5n2-596
30. Setiyaji, A., Ramli, K., Hidayatulloh, Z. Y., & Dharmawan, G. S. B. (2024). *A technique utilizing Machine Learning and Convolutional Neural Networks (CNN) for the identification of SQL Injection Attacks*. 1–6. https://doi.org/10.1109/icsintesa62455.2024.10748116

31. Shakya, R. D. N., Dharmaratne, D. N. S., & Sandirigama, M. (2024). *Detection of SQL Injection Attacks Using Machine Learning Techniques*. 1–6. https://doi.org/10.1109/icecce63537.2024.10823462
32. Sharma, A., & Babbar, H. (2023). *Machine Learning Solutions for Evolving Injection Attack Landscape*. 1–6. https://doi.org/10.1109/incoft60753.2023.10425456
33. Silva, M., Ribeiro, S., Carvalho, V., Cardoso, F., & Gomes, R. L. (2024). Combining Regular Expressions and Machine Learning for SQL Injection Detection in Urban Computing. *Journal of Internet Services and Applications*, *15*(1), 103–111. https://doi.org/10.5753/jisa.2024.3799
34. Singh, B. P., & Singhal, Prof. M. K. (2024). Detection of SQL Injection Attack Using Machine Learning Techniques. *International Journal of Scientific Research in Science and Technology*, *11*(6), 780–790. https://doi.org/10.32628/ijsrst24114323
35. Sirmulla, A., & Manickam, P. (2024). SQL-CB-GuArd: a deep learning mechanism for structured query language injection attack detection. *IAES International Journal of Artificial Intelligence*, *14*(1), 337. https://doi.org/10.11591/ijai.v14.i1.pp337-349
36. SQL Injection Attack Detection Using Machine Learning Methods. (2024). *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/irjmets48788
37. *SQLi Detection with ML: A data-source perspective*. (2023). https://doi.org/10.48550/arxiv.2304.12115
38. Sun, H., Du, Y., & Li, Q. (2023). Deep Learning-Based Detection Technology for SQL Injection Research and Implementation. *Applied Sciences*. https://doi.org/10.3390/app13169466
39. Tasdemir, K., Khan, R., Siddiqui, F., Sezer, S., Kurugollu, F., & Bolat, A. (2023). *An Investigation of Machine Learning Algorithms for High-bandwidth SQL Injection Detection Utilising BlueField-3 DPU Technology*. 1–6. https://doi.org/10.1109/socc58585.2023.10256777
40. Zhao, W., You, J.-H., & Chen, Q. (2024). *SQL Injection Attack Detection Based on Text-CNN*. https://doi.org/10.1145/3665348.3665398
41. Zulu, J., Alsmadi, I., & Liang, G. (2024). *Enhancing Machine Learning Based SQL Injection Detection Using Contextualized Word Embedding*. https://doi.org/10.1145/3603287.3651187