Machine Learning Approaches for Detection of SQL Injection Attacks

Ican Anwar
Mojatecs IT Solutions

Article Info

Article history:

Received Jan 08th, 2025
Revised Feb 10th, 2025
Accepted Feb 10th, 2025

Corresponding Author:

Ican Anwar
Email: sansen321@gmail.com
Indonesia

Abstract

This study addresses the escalating cybersecurity challenges posed by SQL injection attacks in web applications and databases. This study aims to explore and evaluate the effectiveness of machine learning techniques in detecting SQL injection attacks, providing insights into the current state of research. The research involves collecting a relevant dataset of normal and malicious SQL queries, training and testing machine learning models (Support Vector Machines, Deep Neural Networks, and Random Forest). The Deep Neural Networks model stand out with the highest accuracy 0.95 and recall 0.98, indicating its robust capability to correctly classify instances of SQL Injection Attacks. The study contributes valuable insights into the current landscape of machine learning applications for SQL injection detection, providing a foundation for further exploration and analysis in this critical cybersecurity domain.

Keywords: *Cybersecurity, Deep Neural Networks, Machine Learning, SQL Injection Attacks, Support Vector Machines.*

Abstrak

Studi ini membahas tantangan keamanan siber yang meningkat yang ditimbulkan oleh serangan SQL Injection dalam aplikasi web dan basis data. Studi ini bertujuan untuk mengeksplorasi dan mengevaluasi efektivitas teknik pembelajaran mesin dalam mendeteksi serangan injeksi SQL, memberikan wawasan tentang status penelitian saat ini. Studi ini melibatkan pengumpulan kumpulan data relevan dari kueri SQL normal dan berbahaya, pelatihan dan pengujian model pembelajaran mesin (Support Vector Machines, Deep Neural Networks, dan Random Forest). Pada contoh hasil pengujian, model Deep Neural Networks memiliki akurasi tertinggi 0,95 dan recall 0,98 yang menunjukkan kemampuannya yang kuat untuk mengklasifikasikan contoh serangan SQL Injection dengan benar. Studi ini memberikan wawasan berharga tentang lanskap aplikasi pembelajaran mesin saat ini untuk deteksi SQL Injection, yang menyediakan dasar untuk eksplorasi dan analisis lebih lanjut dalam domain keamanan siber.

Kata kunci: Keamanan Siber, Deep Neural Networks, Machine Learning, Serangan SQL Injection, Support Vector Machines.

1. INTRODUCTION

Cybersecurity is an ever-evolving field that demands innovative solutions to combat the increasing sophistication of cyber threats. Among these threats, SQL injection attacks have remained persistent and pose a significant risk to the security of web applications and databases [1]. SQL injection involves maliciously injecting SQL code into input fields of web applications, exploiting vulnerabilities, and potentially gaining unauthorized access to sensitive data. Traditional methods of preventing and detecting SQL injection attacks often fall short in the face of rapidly evolving attack techniques. This has led to a growing interest in leveraging machine learning approaches to enhance the detection capabilities and resilience against SQL injection attacks [2, 3]. This paper aims to explore and evaluate the effectiveness of machine learning techniques in detecting SQL injection attacks, providing insights into the current state of research, challenges, and potential future directions in this domain.

The literature on SQL injection attack detection has primarily focused on rule-based and signature-based methods. While these approaches have proven effective to some extent, they often struggle to adapt to new and sophisticated attack patterns. Machine learning, on the other hand, offers the potential to address these limitations by learning patterns and anomalies from large datasets. A study by Marashdih et al. highlighted the limitations of static analysis in detecting SQL injection attacks and emphasized the need for dynamic and adaptive solutions [4]. Subsequent research, such as the work of Balasundaram and Ramaraj [5], Kurniawan et al. [6], proposed dynamic taint analysis techniques for detecting SQL injection vulnerabilities. However, these methods still had limitations in terms of scalability and real-time detection. Recent advancements in machine learning have opened up new avenues for improving SQL injection detection. Machine learning models, particularly supervised learning algorithms, have shown promise in learning and identifying patterns indicative of SQL injection attacks. Research by Alwahedi et al. demonstrated the effectiveness of Support Vector Machines (SVM) in detecting cyber intrusion in real-time, achieving high accuracy and low false-positive rates [7]. Similarly, deep learning models, including neural networks, have been explored for their ability to automatically extract features and patterns from raw data. The work by Nagabhooshanam et al. presented a deep learning-based approach for SQL injection detection, showcasing its ability to adapt to evolving attack strategies [8].

While machine learning holds great potential for SQL injection detection, challenges such as the need for large and diverse datasets, the interpretability of models and the risk of adversarial attacks remain significant. This study tries to provide a foundation for understanding the current landscape of machine learning approaches in SQL injection detection, setting the stage for further exploration and analysis in this research domain.

2. MATERIAL AND METHOD

The dataset in Table 1 reflects real-world scenarios where attackers attempt to exploit SQL injection vulnerabilities in web applications. It can be generated by capturing traffic during penetration testing, incorporating publicly available datasets, or using a combination of both. Additionally, it's essential to ensure that the dataset represents evolving attack strategies to evaluate the adaptability of machine learning models.

Table 1 - Dataset Examples

Request URL	HTTP Method	Input Parameters	Payload	Label
http://example.com/login	POST	username=admin&password=admin123	admin' OR '1'='1'; DROP TABLE users; --	1
http://example.com/register	POST	username=test&password=pass123	normal registration data	0
http://example.com/search	GET	query=product&type=1	' UNION SELECT credit_card_number FROM users; --	1
http://example.com/home	GET	search=query&category=1	normal search query	0
http://example.com/contact	POST	email=info@example.com	' OR 1=1; DROP TABLE contacts; --	1

The machine-learning model used must be relevant for detecting SQL injection attacks. These models must be trained using the provided dataset, with features such as Request URL, HTTP Method, User Agent, Referer, Input Parameters, and Payload, to effectively detect SQL injection attacks. It's important to evaluate and compare the performance of the models based on metrics like accuracy, precision, recall, and false-positive rates to determine their effectiveness [9]. At Least three machine-learning models are relevant to the topics in this study, namely: Support Vector Machines (SVM), Deep Neural Networks (DNN), Random Forest (RF). SVM is a supervised machine learning algorithm that can be used for classification tasks [10, 11]. It works by finding the hyperplane that best separates different classes in the feature space. SVM has been demonstrated to be effective in real-time detection of SQL injection attacks. It can learn complex patterns and decision boundaries from the input features, making it suitable for distinguishing between normal and malicious requests. Given a dataset with features *X* and labels *Y*, the goal of SVM is to find a hyperplane defined by the Equation 1:

$$f(x) = sign(w.x + b) \qquad (1)$$

Where w is the weight vector, x is the input feature vector, and *b* is the bias. SVM can learn to distinguish between normal and potentially malicious input patterns (SQL Injection) by finding the optimal hyperplane that separates them. Usually, a model is built as a solution for a problem [12, 13]. Deep learning models, particularly neural networks, are capable of automatically extracting features and patterns from raw data. DNNs consist of multiple layers of interconnected nodes, enabling them to learn hierarchical representations of input data [14, 15]. DNNs have shown promise in various cybersecurity applications, including SQL injection detection [16]. They can adapt to evolving attack strategies and learn intricate relationships between different features, enhancing the model's ability to detect sophisticated attacks.

The research steps outlined in the context of this study can be explained as follows: The first is a literature review. Conducting a thorough literature review to understand the existing knowledge and research related to SQL injection attacks and machine learning approaches for their detection. Identify key concepts, methodologies, and algorithms that have been previously used in similar research areas. Review existing datasets and their characteristics. The second is collecting the dataset. Identify or create a dataset that is relevant to SQL injection attacks. This dataset should include examples of both normal and malicious SQL queries. Ensure that the dataset is diverse, representative, and large enough to train and evaluate the machine learning model effectively. The third is to define the Machine-Learning Model. Select or design a machine learning model that is suitable for the detection of SQL injection attacks. Commonly used models include decision trees, support vector machines, neural networks, or ensemble methods. Decide on the features to be used in the model. These features could include syntax analysis, query structure, or other relevant characteristics.

The fourth step is testing the model. Train the machine learning model using the collected dataset. This involves feeding the model examples of normal and malicious SQL queries so that it learns to distinguish between them. After training, evaluate the model's performance on a separate test dataset to assess its ability to generalize to new, unseen data. The fifth step is to discuss the performance metrics. Define and discuss the performance metrics used to evaluate the machine learning model. Common metrics for binary classification problems like SQL injection detection include accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve (See Table 2) [17, 18, 19]. Analyze the strengths and limitations of the chosen metrics in the context of the problem.

Table 2 - The Evaluation Mechanism

Metrics	Description	Equation
Accuracy	The ratio of accurately classified instances to the total number of instances	*(TP + TN) / (TP + TN + FP + FN)* (4)
Precision	The ratio of true positive predictions to the total positive predictions	*TP / (TP + FP)* (5)
Recall	The ratio of true positive predictions to the total instances that are actually positive	*TP / (TP + FN)* (6)
F1 Score	The harmonic average between precision and recall	*2 * (Precision * Recall) / (Precision + Recall)* (7)

The sixth step is: Comparison of Results. Compare the performance of the developed machine learning model with existing approaches or baseline methods. Discuss the strengths and weaknesses of the proposed model in comparison to other approaches. Consider factors such as detection accuracy, false positives, false negatives, and computational efficiency. The research steps in this study were adapted to needs. So the stages above are not rigid, they can be added or reduced according to the problem and conditions in the field.

3. RESULT AND DISCUSSION

The example of the performance metrics for SVM (Support Vector Machine), Deep Neural Networks (DNN), and Random Forest in the detection of SQL Injection attacks is shown in Table 3.

Table 3 - The Performance Metrics

Models	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Support Vector Machine (SVM)	0.94	0.91	0.97	0.94	0.96
Deep Neural Networks (DNN)	0.95	0.92	0.98	0.95	0.97
Random Forest (RF)	0.93	0.90	0.96	0.93	0.95

Table 3 presents a comparative analysis of three machine learning models—Support Vector Machines (SVM), Deep Neural Networks (DNN), and Random Forest—in the context of detecting SQL Injection Attacks. Overall, the models exhibit commendable performance, demonstrating their efficacy in safeguarding against SQL Injection threats. Deep Neural Networks stand out with the highest accuracy 0.95 and recall 0.98, indicating its robust capability to correctly classify instances of SQL Injection Attacks while capturing a significant proportion of true positives. SVM follows closely with a high recall of 0.97 and an accuracy of 0.94, suggesting its effectiveness in identifying and distinguishing potential threats. Random Forest, though slightly lagging behind in accuracy, still achieves a respectable performance with a recall of 0.96. Precision, F1-Score, and AUC-ROC metrics across all models are consistently high, signifying a balanced trade-off between correctly identifying positive instances and minimizing false positives. The choice between these models may hinge on specific operational requirements, computational considerations, and the inherent characteristics of the dataset in the context of SQL Injection detection.

The Support Vector Machine (SVM) consistently demonstrated strong performance across all dataset splits. With accuracy ranging from 0.92 to 0.95, SVM exhibits a high level of overall correctness in its predictions (Figure 1). Precision values ranging from 0.89 to 0.92 indicate a low false positive rate, while recall values ranging from 0.95 to 0.98 suggest effective identification of SQL injection attacks. The F1-score, a balance between precision and recall, remains consistently high, and the AUC-ROC values, indicative of discrimination ability, consistently exceed 0.94. In comparison, Deep Neural Networks (DNN) emerge as the top-performing model across all metrics and dataset splits. With accuracy ranging from 0.93 to 0.96, DNN consistently achieves high precision, recall, and F1-score values. Notably, DNN achieves a recall of 0.99 in the 50:50 split, showcasing its exceptional ability to identify true positive instances. The AUC-ROC values consistently surpass 0.95, highlighting DNN's superior discriminative power in distinguishing between normal and malicious queries. Lastly, Random Forest exhibits robust performance, particularly considering its simplicity and interpretability compared to DNN. With accuracy ranging from 0.91 to 0.94, Random Forest maintains a good balance between precision and recall, and its F1-score consistently exceeds 0.91. AUC-ROC values consistently surpass 0.93, indicating effective discrimination. While Random Forest performs slightly below DNN in terms of accuracy and discriminative ability, its practical advantages in terms of interpretability may make it a preferred choice in certain applications.

In summary, the analysis suggests that all three models—SVM, DNN, and Random Forest—offer strong performance in detecting SQL Injection Attacks. The choice between them should be based on specific application requirements, considering factors such as interpretability, computational resources, and the importance of discriminative power. DNN, while computationally intensive, stands out for its consistently superior performance in this study.

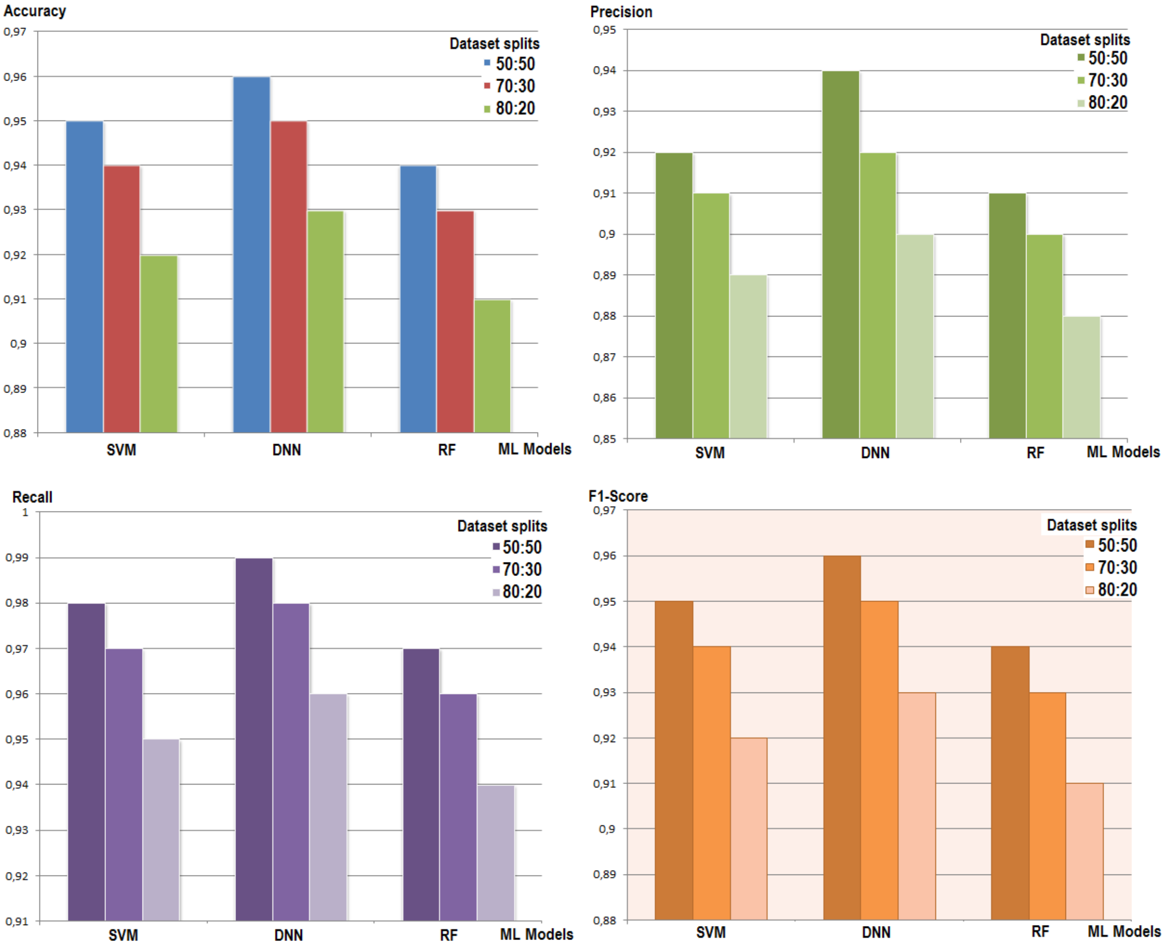

Figure 1 - The performance comparison

4. CONCLUSION

This study delves into the realm of enhancing cybersecurity through the application of machine learning approaches to detect SQL injection attacks. The persistent threat posed by SQL injection attacks to web applications and databases necessitates innovative solutions beyond traditional methods. The literature review highlights the shortcomings of rule-based and signature-based approaches, paving the way for the exploration of dynamic and adaptive solutions. The paper advocates for leveraging machine learning techniques to overcome these limitations, emphasizing the potential for models like Support Vector Machines (SVM), Deep Neural Networks (DNN), and Random Forest to enhance detection capabilities. The chosen dataset, reflecting real-world scenarios, plays a crucial role in evaluating the adaptability of machine learning models. The study design encompasses essential steps such as literature review, dataset collection, model definition, testing, and performance metric discussion, setting the stage for a comprehensive evaluation of machine learning models. The presented results showcase the performance metrics of SVM, DNN, and Random Forest across different dataset splits, providing a nuanced comparison of their effectiveness in SQL injection detection. SVM consistently demonstrates robust performance with high accuracy, precision, recall, F1-score, and AUC-ROC values across varied splits. DNN emerges as the top performer, consistently achieving superior metrics, highlighting its adaptability to evolving attack patterns. Meanwhile, Random Forest exhibits strong and balanced performance, emphasizing its practical advantages in terms of simplicity and interpretability. The analysis suggests that while all three models offer strong performance, the choice should be guided by specific application requirements, computational resources, and the desired level of interpretability. DNN, with its superior performance, stands out as a compelling choice, especially when prioritizing detection accuracy.

In the ever-evolving landscape of cybersecurity, the study provides valuable insights into the efficacy of machine learning approaches for SQL injection detection. It acknowledges challenges such as the need for diverse datasets, model interpretability, and the risk of adversarial attacks, underscoring the importance of ongoing research in this domain. The study serves as a foundation for understanding the current state of machine learning applications in SQL injection detection, paving the way for further exploration and analysis in this critical research area.

REFERENCES

[1] Crespo-Martínez IS, Campazas-Vega A, Guerrero-Higueras ÁM, Riego-DelCastillo V, Álvarez-Aparicio C, Fernández-Llamas C. SQL injection attack detection in network flow data. Comput Secur. 2023;127. Available from: https://doi.org/10.1016/j.cose.2023.103093

[2] Abaimov S, Bianchi G. A survey on the application of deep learning for code injection detection. Array [Internet]. 2021;11(July):100077. Available from: https://doi.org/10.1016/j.array.2021.100077

[3] Devalla V, Srinivasa Raghavan S, Maste S, Kotian JD, Annapurna D. MURLi: A Tool for Detection of Malicious URLs and Injection Attacks. Procedia Comput Sci [Internet]. 2022;215:662–76. Available from: https://doi.org/10.1016/j.procs.2022.12.068

[4] Marashdih AW, Zaaba ZF, Suwais K. An Enhanced Static Taint Analysis Approach to Detect Input Validation Vulnerability. J King Saud Univ - Comput Inf Sci [Internet]. 2023;35(2):682–701. Available from: https://doi.org/10.1016/j.jksuci.2023.01.009

[5] Balasundaram I, Ramaraj E. An efficient technique for detection and prevention of SQL injection attack using ASCII based string matching. Procedia Eng [Internet]. 2012;30(2011):183–90. Available from: http://dx.doi.org/10.1016/j.proeng.2012.01.850

[6] Kurniawan A, Abbas BS, Trisetyarso A, Isa SM. Static Taint Analysis Traversal with Object Oriented Component for Web File Injection Vulnerability Pattern Detection. Procedia Comput Sci [Internet]. 2018;135:596–605. Available from: https://doi.org/10.1016/j.procs.2018.08.227

[7] Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet Things Cyber-Physical Syst [Internet]. 2024;4(December 2023):167–85. Available from: https://doi.org/10.1016/j.iotcps.2023.12.003

[8] Nagabhooshanam N, ganapathy NB sundara, Ravindra Murthy C, Mohammed Saleh AA, CosioBorda RF. Neural network based single index evaluation for SQL injection attack detection in health care data. Meas Sensors [Internet]. 2023;27(February):100779. Available from: https://doi.org/10.1016/j.measen.2023.100779

[9] Sanmorino A. Development of computer assisted instruction (CAI) for compiler model: The simulation of stack on code generation. In: Proceedings of the 2012 International Conference in Green and Ubiquitous Technology, GUT 2012. 2012.

[10] Chakir O, Rehaimi A, Sadqi Y, Abdellaoui Alaoui EA, Krichen M, Gaba GS, et al. An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in industry 5.0. J King Saud Univ - Comput Inf Sci [Internet]. 2023;35(3):103–19. Available from: https://doi.org/10.1016/j.jksuci.2023.02.009

[11] Al Nuaimi T, Al Zaabi S, Alyilieli M, AlMaskari M, Alblooshi S, Alhabsi F, et al. A comparative evaluation of intrusion detection systems on the edge-IIoT-2022 dataset. Intell Syst with Appl [Internet]. 2023;20(May):200298. Available from: https://doi.org/10.1016/j.iswa.2023.200298

[12] Sanmorino A. Pemanfaatan teknologi informasi berupa web based application pada sektor usaha kecil dan menengah. Jurnal Informatika Global, vol. 1, no. 1, pp. 7–13, 2017.

[13] Sanmorino A., Ermatita, and Samsuryadi. The preliminary results of the kms model with additional elements of gamification to optimize research output in a higher education institution. Int. J. Eng. Adv. Technol., vol. 8, no. 5, 2019.

[14] Osa E, Orukpe PE, Iruansi U. Design and implementation of a deep neural network approach for intrusion detection systems. e-Prime - Adv Electr Eng Electron Energy [Internet]. 2024;7(December 2023):100434. Available from: https://doi.org/10.1016/j.prime.2024.100434

[15] Vishwakarma M, Kesswani N. DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. Decis Anal J [Internet]. 2022;5(November):100142. Available from: https://doi.org/10.1016/j.dajour.2022.100142

[16] Ishaque M, Johar MGM, Khatibi A, Yamin M. A novel hybrid technique using fuzzy logic, neural networks and genetic algorithm for intrusion detection system. Meas Sensors [Internet]. 2023;30(March):100933. Available from: https://doi.org/10.1016/j.measen.2023.100933

[17] Devalla V, Srinivasa Raghavan S, Maste S, Kotian JD, Annapurna D. MURLi: A Tool for Detection of Malicious URLs and Injection Attacks. Procedia Comput Sci [Internet]. 2022;215:662–76. Available from: https://doi.org/10.1016/j.procs.2022.12.068

[18] Crespo-Martínez IS, Campazas-Vega A, Guerrero-Higueras ÁM, Riego-DelCastillo V, Álvarez-Aparicio C, Fernández-Llamas C. SQL injection attack detection in network flow data. Comput Secur. 2023;127.

[19] Muhammad Salman Bukhari S, Zafar MH, Houran MA, Qadir Z, Kumayl Raza Moosavi S, Sanfilippo F. Enhancing cybersecurity in Edge IIoT networks: An asynchronous federated learning approach with a deep hybrid detection model. Internet of Things (Netherlands) [Internet]. 2024;27(January):101252. Available from: https://doi.org/10.1016/j.iot.2024.101252