JOIV : Int. J. Inform. Visualization, 8(4) - December 2024 2131-2138

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Application of Artificial Intelligence in Detecting SQL Injection Attacks

Nwabudike Augustine [a,c], Abu Bakar Md. Sultan [a,*], Mohd Hafeez Osman [a], Khaironi Yatim Sharif [b]

[a] Faculty of Computer Science and Information Technology University Putra Malaysia, Malaysia
[b] Department of Computer and Information Science Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
[c] Department of Computer Science, Delta State Polytechnic Ogwashi Uku Nigeria

Corresponding author: *abakar@upm.edu.my

***Abstract*—SQL injection attacks rank among the most significant threats to data security. While AI and machine learning have advanced considerably, their application in cybersecurity remains relatively undeveloped. This work mainly aims to solve the IT-related challenge of insufficient knowledge bases and tools for security practitioners to monitor and mitigate SQL Injection attacks with AI/ML techniques. The study uses a mixed-methods approach to evaluate how well different AI and ML algorithms identify SQL injection attacks by combining algorithmic evaluation with empirical investigation. Datasets of well-known SQL injection attack patterns and AI/ML models intended for cybersecurity anomaly detection are among the resources underexplored; these findings show the potential for boosting detection capabilities by deploying ML and AI-based security solutions; specific algorithms have demonstrated success rates of up to 80% in detecting SQL injections. Despite this promising performance, around 75% of survey participants acknowledged a decrease in harmful content, with a similar number highlighting increased efficiency in their roles as security researchers or incident responders. Nevertheless, the tool's adoption among cybersecurity professionals remains under 30%. This underscores a gap between the capabilities these technologies offer and their current level of adoption among professionals. This will help lay the groundwork for future work in identifying the best solutions and providing potential approaches to incorporating AI/ML into cybersecurity frameworks. The implications of this study indicate that adopting robust defenses against SQL injection and other cyber threats could increase many folds if we continue to research and implement AI ML. technologies.**

Keywords: SQL injection; machine learning; artificial intelligence; cybersecurity; SQL injection attack.

Manuscript received 7 Jan. 2024; revised 18 Aug. 2024; accepted 22 Oct. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

In the current digital era, the increasing prevalence of cyberattacks has become a critical concern for organizations worldwide [1]. Among these, SQL injection attacks stand out due to their simplicity and devastating impact, with over 65% of web application vulnerabilities attributed to this form of attack [2], [3], [4]. This alarming statistic underscores the urgency of developing advanced methods to identify and fix SQL injection attacks. The use of Machine Learning (ML) and Artificial Intelligence (AI) in this field offers hope, promising significant improvements in detection and prevention. However, despite their potential, these technologies remain underutilized [5]. The growing vulnerability of web applications to SQL injection attacks poses risks of data breaches and unauthorized access. Cybersecurity professionals lack comprehensive strategies to detect and mitigate these attacks using ML and AI technologies, and many organizations lack information on their integration.

In a cyberattack known as an SQL injection (SQLi) assault, malicious SQL statements are inserted into an input field for execution. These manipulate SQL queries to take advantage of flaws in software applications [6]. When successful, SQLi attacks can bypass authentication and authorization mechanisms, allowing attackers to gain control of the database and private information [7].

There are several types of SQL injection attacks, which include Classic SQL Injection, which involves directly inserting or "injecting" malicious SQL code into a query [8], [9]. Blind SQL Injection is when an attacker asks true or false inquiries to obtain information while sending instructions to the database without directly seeing the results.[10], Other types of SQL injection attacks include error-based SQL Injection. This method uses the database server's error messages to collect data about the structure of the database

[11]; union-based SQL Injection is a technique that combines the output of two or more SELECT queries using the UNION SQL operator to produce a single result that is returned as part of the HTTP response. SQL injection also comes in the form of time-based SQL injection attacks, which are inferential attacks where the attacker uses time delays to determine if certain queries are true or false [12] (see Figure 1), according to the Open Web Application Security Project (OWASP), SQL injection ranks among the top ten critical web application security threats [13]. Machine learning and AI are emerging as effective and accurate tools for detecting these evolving threats, with studies indicating they can improve detection rates by up to 80% [14].

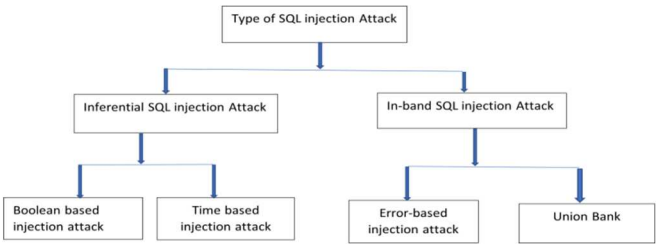

Fig. 1 Types of SQL Injection Attack

However, the rate at which ML and AI techniques are adopted, notwithstanding their potential in the industry, remains below 30% [15]. This gap highlights the need for further research and development to integrate these advanced technologies into practical cybersecurity frameworks. This study seeks to bridge the gap in literature and practice by investigating the application of artificial intelligence (AI) and machine learning (ML) to detect SQL injection attacks. The research objectives are the following:

a. Reviewing Existing Literature: This includes presenting a comprehensive assessment of the relevant literature on SQL injection attacks and the ML and AI systems used for their assessment.
b. Identifying Effective Algorithms: Explore the various predictions to be made and ego-centric behavioral patterns observed over different time periods to identify interesting patterns in such ML/AI solutions.
c. Developing a Framework: Generate a model that visualizes how the corps can utilize neural networks and machine learning to enhance the existing model in cybersecurity.

The research questions for this paper are the following:

a. Research Question 1 (RQ1): What factors enhance the effectiveness of traditional SQL injection detection methods when integrated with machine learning and artificial intelligence algorithms?
b. Research Question 2 (RQ2): What are the common practical challenges of applying machine learning and AI techniques for detecting and preventing SQL injections within current cybersecurity systems, and what are the best practices for overcoming these challenges?

II. Materials and Method

The increasing sophistication of cyberattacks demands equally sophisticated defense mechanisms. SQL injection attacks are a prevalent cyber threat, posing serious risks to web application security. The Open Web Application Security Project (OWASP) identifies them as one of the ten most critical vulnerabilities in web applications [16], accounting for over 65% of web application threats [17]. A crucial issue is highlighted by the increasing assault of web applications to more complex SQL injection attacks; conventional detection techniques are insufficient to counter these sophisticated attacks. Rule-based systems, signature-based detection, and anomaly-based procedures are examples of traditional methods that frequently fall behind the attackers' quickly changing strategies [18]. The limitations of conventional security become starkly evident when fraudsters create increasingly sophisticated and disguised attack techniques, making web applications more vulnerable to compromise and exploitation [19]. To protect online application security in the face of this growing problem, more robust, intelligent, and adaptable detection systems must be developed and integrated [20]. Utilizing artificial intelligence and machine learning skills presents a viable way to address this problem [21]. The research aims to explore existing approaches to detecting attacks using SQL injections and assess the possibilities of ML and AI methods in enhancing detection capabilities.

A. Existing Approaches to Detecting SQL Injection Attack

Rule-based detection systems, a traditional method for identifying SQL injection threats, use predefined rules and patterns to recognize suspicious activities. These systems are designed to recognize particular types of SQL injection attempts using known attack signatures and heuristics. While effective for detecting familiar attack patterns, rule-based detection is limited when spotting new or disguised threats. For example, these systems might miss SQL injection attempts that utilize advanced evasion techniques or novel attack strategies [22]. One of the primary challenges with rule-based detection is the need for continuous updates to the rule set to keep up with emerging threats [23]. Additionally, rule-based systems can produce a large number of false positives, causing security personnel to become weary of alert analysts.

Signature-based detection is a cybersecurity method that uses predefined signatures or patterns associated with known attack techniques to identify SQL injection attacks. These signatures are crafted based on a database of attack methods, including common vulnerabilities and cybercriminal tactics. The detection system scans incoming traffic and compares it against these signatures to identify potential attacks. This method is effective for detecting well-known SQL injection techniques but may be less effective against novel or sophisticated attacks that deviate from known patterns [24]. However, like rule-based detection, signature-based systems have limitations in detecting novel or obfuscated attacks. Attackers can bypass signature-based detection by slightly modifying their attack patterns or using evasion techniques. Additionally, maintaining signature databases requires continuous updates to address new threats, which can be resource-intensive for organizations.

Anomaly-based detection systems use machine learning and statistical models to find departures from the norm [25]. These systems establish a baseline of regular activity for web applications and monitor incoming traffic for deviations that may indicate a potential SQL injection attack. Anomaly-based detection can effectively identify previously unknown attack

vectors and sophisticated evasion techniques [26]. Anomaly-based detection is effective in detecting zero-day threats and exploiting previously unknown vulnerabilities. However, it can generate false positives due to legitimate user behavior variations, and its effectiveness relies on the accuracy of the baseline model, which can be challenging to maintain.

B. Machine Learning Techniques for Detecting SQL Injection Attacks

Machine Learning and AI techniques, including supervised, unsupervised, and deep learning, are utilized to detect SQL injection attacks. Standard algorithms used for SQL injection detection include decision trees, support vector machines, and random forests. Decision trees are effective at capturing non-linear relationships but are prone to overfitting. SVM is well-suited for handling high-dimensional spaces, while random forests combine multiple predictions to enhance accuracy, though they may sacrifice interpretability. Studies have demonstrated that supervised learning algorithms can achieve high detection accuracy for SQL injection attacks for example, research by [27] indicated that a random forest classifier achieved an accuracy of 99.8% in detecting SQL injection attacks, However, the performance of supervised learning models depends heavily on the quality and representativeness of the training data. Additionally, these models may struggle to detect novel attack patterns that are not present in the training dataset.

Conversely, Unsupervised learning algorithms aim to identify patterns and structures in data without relying on labeled datasets or prior knowledge of the outcomes. In the context of SQL injection detection, unsupervised learning can detect anomalous SQL queries that deviate from normal behavior. Clustering algorithms and anomaly detection methods commonly use unsupervised learning techniques to identify SQL injection attacks [28]. Based on their similarity, clustering algorithms organize data points into groups. Clustering can be used to find clusters of related SQL queries in SQL injection detection, with outliers perhaps pointing to malicious activity. K-means, DBSCAN, and hierarchical clustering are examples of common clustering techniques. Data points that substantially depart from the typical patterns are found using anomaly detection algorithms. Techniques like isolation forests, one-class SVM, and autoencoders can be used to identify SQL injection attacks. Unsupervised learning algorithms can produce false positives and be sensitive to hyperparameters, yet they can identify novel attack patterns without labeled data [29]. Deep Learning approaches, including CNN, RNN, and autoencoders, are machine learning techniques used to model complex data connections and patterns, promising in detecting SQL injection attacks and used for image recognition [30]. They effectively detect SQL injection by capturing local patterns in SQL queries and identifying malicious sequences [31]. Recurrent Neural Networks (RNNs) use sequential data to model and identify malicious sequences.

C. Contrasting Views and Criticisms

ML and AI are widely used to detect SQL injection attacks, but some researchers and practitioners question their effectiveness due to their lack of interpretability. Security analysts often need models that offer transparency in decision-making, but black-box models are frequently employed. Deep neural networks, in particular, can be hard to interpret, making it challenging to have confidence in their predictions. [32] ML and AI models heavily rely on the availability of high-quality, labeled training data. In many cases, obtaining such data can be challenging due to privacy concerns, data sensitivity, and the lack of publicly available datasets. Additionally, the performance of these models can degrade significantly when applied to real-world scenarios that differ from the training environment [33]. ML and AI models are also susceptible to adversarial assaults, which use input data manipulation to trick the model in the context of temporal patterns. Autoencoders, unsupervised neural networks, detect anomalies by compressing and reconstructing data. Deep learning approaches have shown effectiveness in detecting SQL injection attacks—for instance, a study by [34]. The CNN-based model achieved 99.50% accuracy in detecting SQL injection attacks but requires large training volumes, computational resources, and interpretability challenges for security analysts.

In SQL injection detection, attackers can craft queries to bypass ML-based detection systems. This raises concerns about the robustness and reliability of ML models in adversarial environments [35]. Training and deploying ML and AI models demand a lot of processing power and profound learning models. Organizations with limited resources may find implementing and maintaining these systems challenging [36]. Additionally, deploying ML models in real-time detection systems can introduce latency, affecting the performance of web applications [37]. Finally, ethical questions about using AI and ML in cybersecurity include data privacy, bias, and fairness. Ensuring that ML models are trained on unbiased and representative data is crucial to avoid discriminatory outcomes. Additionally, the deployment of ML-based detection systems must adhere to privacy regulations and ethical guidelines to protect user data [38].

D. Gaps in Literature and Contribution to Knowledge

Despite the advancements in SQL injection detection techniques, several gaps remain in the existing literature (Table I).

TABLE I
GAPS IN LITERATURE AND RESEARCH NEEDS FOR SQL INJECTION DETECTION

Gap	Current state	Research needs
Comparative studies of ML and AI Algorithm [39]	Focus on a single or a limited set of algorithms	Comprehensive comparative studies and various metrics
Integration into Practical Cybersecurity Frameworks [40]	Demonstrations in controlled or simulated environments	Practical guidelines and best practices for real-world deployment, addressing integration challenge
Detection of novel and evolving SQL injection attacks [41]	Effective detection of known attack patterns struggles with new techniques	Development of adaptive and self-learning emerging threats

E. Challenges and Limitations in Detecting SQL Injection Attacks Using ML and Deep Learning (DL) and AI

SQL injection attacks significantly threaten web application security, accounting for over 65% of vulnerabilities [42]. Although AI and Machine Learning have shown promise in detecting such attacks, they face challenges and limitations. This section highlights gaps in the literature and the need for more advanced detection systems to address these evolving sophistications. According to the Open Web Application Security Project [43], injection remains a top threat, underscoring the need for continuous advancements in detection technologies [44]. Studies show that despite the advancements in ML and AI, traditional detection systems still fail to detect up to 25% of sophisticated SQL injection attacks [45]. Web applications' vulnerability to SQL injection attacks necessitates ML and AI integration in security measures, as traditional detection methods are insufficient, it requires further research (Tables II and III, and Fig. 2).

TABLE II
GAPS IN LITERATURE AND RESEARCH NEEDS FOR SQL INJECTION DETECTION

Aspect	Current State	Challenges/Gaps	Research Needs
Prevalence and Impact [46], [47]	SQL injection accounts for over 65% of web vulnerabilities	Challenges in the Adoption of Detection Mechanisms:	Focus on performance metrics and practical usability
Data Availability and Quality [48]	Crucial for training robust ML models.	Scarcity of high-quality, labeled datasets due to privacy concerns and data sensitivity	Development of representative and diverse datasets
Model Interpretability [49]	High detection accuracy but low interpretability of "black box "models	A lack of transparency undermines trust and limits the ability of security analysts to take effective action.	The creation of interpretable models capable of delivering clear and transparent explanations
Robustness to Adversarial Attacks [50]	Specially crafted inputs can deceive models	Susceptibility to adversarial attacks poses significant challenges to robustness	Study of adversarial training methods, defensive distillation techniques, and strategies for strengthening model resilience
False Positives and False Negatives [51]	Critical balance between minimizing false positives and negatives	A high rate of false positives overwhelms teams, while high false negatives raise concerns about model robustness	Techniques for optimizing and balancing to reduce both false positives and false negatives.
Computational Resource Requirements [52]	Training and deploying ML models requires substantial resources	Organizations with limited resources might face difficulties, and real-time detection can lead to increased latency	Efficient resource management and optimization strategies for practical deployment
Evolving Threat Landscape [53]	Attackers are constantly developing new methods to evade detection.	Static models might be ineffective at identifying new types of attacks.	Developing adaptive models capable of real-time learning and updating based on new data.
Adaptive and Self-Learning Models [54]	Existing models are often static	Need for continuous updates to maintain effectiveness against new threats	Research on self-learning models that can adapt to emerging attack patterns.

TABLE III
AN ANALYSIS OF VARIOUS ML/DL ALGORITHM

Author	Year	Method	Limitation	Areas of further research	Performance Metric
[55]	2024	SVM	The system's reliance on high-quality data preprocessing may hinder its robustness in scenarios with noisy or incomplete data	further reducing false positives without compromising the detection rate	Accuracy 81%
[56]	2024	KNN	This study's time complexity analysis focuses on model training, excluding the computational cost of pre-training the Roberta embedding model	Investigating methods to reduce the computational overhead of contextualized embeddings, making them more suitable for real-world applications.	Accuracy 99.65
[57]	2024	random forest	The study's dataset, sourced from public repositories like Kaggle and GitHub, may not accurately represent all real-world SQL injection attack variations, potentially limiting its generalizability.	The dataset will be expanded to encompass a wider range of SQL injection attack types and real-world data to enhance the model's robustness and generalizability	Accuracy 99
[58]	2024	CCBA	The evaluation was conducted on multiple datasets, but the models' robustness against real-world scenarios with diverse and evolving attack patterns was not fully explored.	The study explores methods for detecting advanced attack types such as zero-day attacks, flood attacks, DDoS attacks, and evolving malware strategies...	Accuracy (99.3) %
[59]	2024	AdaBoost	The study's generalizability could have been compromised if the dataset used for training and evaluation did not include a diverse range of SQL injection types and real-world variations.	Expanding the dataset to include diverse SQL injection attacks, query structures, and obfuscation techniques could enhance the model's robustness and generalizability.	Accuracy 99
[41]	2024	BERT–LSTM	Integrating BERT and LSTM may increase the model's complexity, potentially requiring more computational resources and longer training times, which could limit its real-time application feasibility.	Further studies should focus on developing adaptive techniques that allow the model to update and learn from new SQL injection attacks dynamically.	Accuracy 99.19%

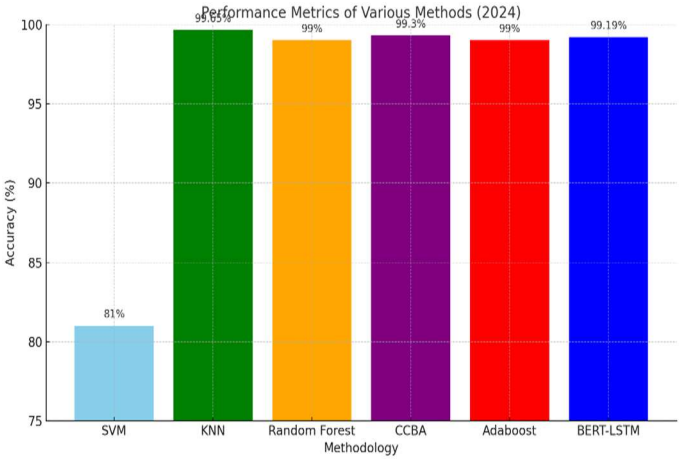

Fig. 2 Performance of different model methodologies over the years

TABLE IV
A COMPREHENSIVE UNDERSTANDING OF HOW ML AND AI TECHNIQUES WERE APPLIED TO DETECT SQL INJECTION ATTACK

No	Tuple	Description
1	ML Techniques	Decision Trees: This uses a hierarchical structure to classify SQL queries by some specific attributes and thus can easily detect anomalies Random Forests: This technique aggregates the predictions of many decision trees to increase detection accuracy and reduce overfitting. Support Vector Machines (SVM): It creates a decision boundary to separate legitimate and malicious SQL queries, useful when working with higher dimensions of features K-Means Clustering: This groups similar SQL queries, with the outliers being flagged as a potential attack (which are useful in detecting new or coming trends of attacks that have not been seen before. Isolation Forest: It finds anomalies separating observations; pointing to the query needing fewer splits in trees will be one, which is why it fits well for identifying very rare SQL Injection attack
2	DL Techniques	Convolutional Neural Networks (CNNs): CNNs were traditionally employed in image recognition; however, we can treat SQL queries as sequences and use them to identify complex attack signatures. Recurrent Neural Networks (RNNs) Processing sequences of SQL queries that reflect the temporal dependency and how query patterns are changing over time. Long-Short-Term Memory (LSTM) Networks: LSTM networks are a special kind of RNN that can learn long-term dependencies with their input sequence and are helpful in detecting fine-grained SQL Injection Techniques.
3	Data Collection & Preparations	Data Sourcing: Extract versatile datasets containing web application logs, database interactions and synthetic SQL Injection attacks. Feature Extraction: This involves determining useful features like query length, SQL keywords and abnormal characters or sequences. Preprocessing: Tokenization, normalization, and encoding of raw SQL queries into a structured format to feed data for the training process.
4	Feature Engineering	Query Length: Running count of the length of SQL queries allowing query length to act as a feature useful in spotting abnormally long queries here and there used, often caused by attacks. SQL Keywords: Monitor keywords such as SELECT, DROP, UNION, to alert you when suspicious queries are being executed., Detect special characters and strings, frequently used in SQL Injection attack
5	Handling Imbalanced Data	Frequency Analysis: to monitor query execution and detect anomalies based on the frequency of the queries along with some common patterns, Resampling Techniques: Using methods like SMOTE (Synthetic Minority Over-sampling Technique) to replicate the Attack samples
6	Performance Metric	Performance metrics are accuracy, precision, recall and F1-score,
7	Comparison of ML & DL	DL Models: known to be "black boxes," providing less explanation on decision-making, but techniques like LIME (Local Interpretable Model-agnostic Explanations) can aid in understanding intricate models: Simpler and faster to train, easier for interpretability as suitable in most cases of straightforward detection tasks.
8	Comparison of ML & DL	DL Models: They are very good at capturing high-level abstractions and relationships. They require large data and computational demands but are recommended for sophisticated or future attack detection. Resource Considerations: ML models are less weight-bearing, while DL models require more computation power and memory utilization, which can affect deployment in a resource-constrained environment.
9	Deployment Best Practices	Real-time Analysis: Models should be able to handle and analyze SQL queries in real-time without slowing down the system. They should also regularly update models with new data to keep up with cybercriminals' changing tactics and Adversarial train models to make them robust against clever evasion attacks from attackers.

III. RESULTS AND DISCUSSION

This section provides a detailed discussion of the responses to the research questions. Each question is addressed systematically, highlighting the key findings, insights, and relevance to the study's objectives.

A. RQ1: Factors enhancing SQL injection with AI and ML

In question research 1, the use of sophisticated machine learning and artificial intelligence algorithms can enhance the detection of SQL injection threats. These techniques include random forests, support vector machines (SVM), convolutional neural networks (CNN), and ensemble methods, as shown in Table 3. Each has its strengths in understanding complex patterns and anomalies for strong detection mechanisms. Random Forests have an accuracy of 99%, combining multiple predictions from different decision trees to minimize overfitting. Support Vector Machines (SVMs) have an accuracy of *81% but* are slow to change with attacking patterns. K-Nearest Neighbors (KNN) achieves a sensitivity of 99.65%, but time constraints for training require real-time embedding techniques.

Convolutional Neural Networks (CNNs) are useful for features inherent in SQL queries but rely on the timeliness and quality of the dataset. RNNs and LSTMs have successfully dealt with sequential SQL commands, with BERT-LSTM Models showing 99.19% accuracy. Hybrid and combined models take advantage of individual model features to increase detection. Feature engineering is crucial in ML and DL models, including query length, SQL keywords, and special characters. Techniques like tokenization and normalization improve data quality, but challenges like imbalanced datasets, noise features, and ever-changing attack vectors remain. Methods like Synthetic Minority Over-

sampling (SMOTE) and cost-sensitive learning can help improve model training on minority attack samples.

B. RQ2: Challenges in Using ML and AI for SQL Injection Detection

Research Question 2's discussion of SQL injection attack detection and prevention within the current cybersecurity framework raises additional concerns about the adversary's resilience, the quality of data, and the interpretability of deep learning models, which are crucial for their effectiveness. Table IV focuses on key points of best practice to solve these problems.

Attackers exploit model vulnerabilities through evasion strategies, and sophisticated feature engineering and dynamic learning systems can strengthen models like Random Forests and SVMs. Convulsive class imbalance issues are a significant concern in applying ML and DL algorithms practically. Frequency analysis or resampling can help ensure identical distributions for classes. LIME (Local Interpretable Model-agnostic Explanations) helps understand these models' interpretability. However, high computational requirements and significant resources are limitations. Decision Trees or Isolation Forests are suitable for resource-constrained environments. System durability against attacks is achieved through frequent model adjustments and hybrid approach integrations. Techniques like bagging and stacking can improve the ability to detect SQL injection attacks by combining the strengths of different models.

IV. Conclusion

Recent studies have proven the potential of Artificial Intelligence (AI) and Machine Learning (ML) in improving the detection of SQL injection attacks, mostly in areas where traditional approaches fall short. The analysis identifies the benefits and drawbacks of each strategy by assessing several different algorithms, including supervised, unsupervised, and deep learning techniques. Supervised learning models like random forests and support vector machines possess a keen sense of intricate attack patterns. When using top-notch data for training, the availability and representation of tagged datasets place limitations on them. Unsupervised techniques such as anomaly detection and clustering allow the identification of new risks without the need for labeled data. While deep learning techniques that demand a large number of labeled datasets and substantial processing resources, such as recurrent and convolution neural networks, offer promise,

Additionally, this study tackles key challenges such as mitigating adversarial attack risks, improving the transparency of machine learning models, striking a balance between false positives and false negatives, and addressing latency issues in real-time detection systems. Despite these difficulties, there is a big chance to strengthen cybersecurity defenses by incorporating ML and AI into SQL injection detection procedures. Comparative research and useful suggestions are required to improve detection frameworks and guarantee their efficacy in practical settings. The study highlights the potential of Machine Learning and Artificial Intelligence to enhance the effectiveness of traditional SQL injection detection methods, improving accuracy and adaptability in cybersecurity. It also underscores the need to address practical challenges such as adversary strength, data availability, and model transparency. Future research should focus on developing robust models that integrate seamlessly into cybersecurity systems while adopting best practices to overcome these barriers for more effective threat mitigation.

Acknowledgment

We thank the Department of Software Engineering, Faculty of Computer Science and Information, for their support.

Reference

[1] A. S. P. Boggs *et al.*, 'National Institute of Standards and Technology environmental scan 2023 : societal and technology landscape to inform science and technology research', National Institute of Standards and Technology (U.S.), Gaithersburg, MD, NIST IR 8482, Aug. 2023. doi:10.6028/NIST.IR.8482.

[2] F. Faisal Fadlalla and H. T. Elshoush, 'Input Validation Vulnerabilities in Web Applications: Systematic Review, Classification, and Analysis of the Current State-of-the-Art', *IEEE Access*, vol. 11, pp. 40128–40161, 2023, doi: 10.1109/access.2023.3266385.

[3] U. Farooq, 'Ensemble Machine Learning Approaches for Detection of SQL Injection Attack', *Teh. Glas.*, vol. 15, no. 1, pp. 112–120, Mar. 2021, doi: 10.31803/tg-20210205101347.

[4] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, 'Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks', in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA: IEEE, May 2016, pp. 582–597. doi:10.1109/SP.2016.41.

[5] S. A. K. Hacham and O. N. UÇan, 'Detection of Malicious SQL Injections Using SVM and KNN Algorithms', in *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, Istanbul, Turkiye: IEEE, Nov. 2023, pp. 1–5. doi: 10.1109/ISAS60782.2023.10391560.

[6] H. Bahruddin, V. Suryani, and A. A. Wardana, 'Adversary Simulation of Structured Query Language (SQL) Injection Attack Using Genetic Algorithm for Web Application Firewalls (WAF) Bypass', in *Intelligent Systems and Applications*, vol. 823, K. Arai, Ed., in Lecture Notes in Networks and Systems, vol. 823. , Cham: Springer Nature Switzerland, 2024, pp. 656–669. doi: 10.1007/978-3-031-47724-9_43.

[7] E. G. H. Grata *et al.*, 'Artificial Intelligence for Threat Anomaly Detection Using Graph Databases – A Semantic Outlook', in *Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection*, 1st ed., S. Mahajan, M. Khurana, and V. V. Estrela, Eds., Wiley, 2024, pp. 249–278. doi: 10.1002/9781394196470.ch13.

[8] M. Hossain Hadi and K. Hashim Al-Saedi, 'Adaptive Hybrid Learning for Websites Vulnerability prediction', *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 16, no. 1, Mar. 2024, doi: 10.29304/jqcsm.2024.16.11433.

[9] J. Zulu, B. Han, I. Alsmadi, and G. Liang, 'Enhancing Machine Learning Based SQL Injection Detection Using Contextualized Word Embedding', in *Proceedings of the 2024 ACM Southeast Conference on ZZZ*, Marietta GA USA: ACM, Apr. 2024, pp. 211–216. doi:10.1145/3603287.3651187.

[10] S. Kum, S. Oh, J. Yeom, and J. Moon, 'Optimization of Edge Resources for Deep Learning Application with Batch and Model Management', *Sensors*, vol. 22, no. 17, p. 6717, Sep. 2022, doi:10.3390/s22176717.

[11] A. Odeh and A. Abu Taleb, 'Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection', *Appl. Sci.*, vol. 13, no. 21, p. 11985, Nov. 2023, doi: 10.3390/app132111985.

[12] D. Dasgupta, Z. Akhtar, and S. Sen, 'Machine learning in cybersecurity: a comprehensive survey', *J. Def. Model. Simul. Appl. Methodol. Technol.*, vol. 19, no. 1, pp. 57–106, Jan. 2022, doi:10.1177/1548512920951275.

[13] Z. Marashdeh, K. Suwais, and M. Alia, 'A Survey on SQL Injection Attack: Detection and Challenges', in *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan: IEEE, Jul. 2021, pp. 957–962. doi: 10.1109/ICIT52682.2021.9491117.

[14] B. Zhang, R. Ren, J. Liu, M. Jiang, J. Ren, and J. Li, 'SQLPsdem: A Proxy-Based Mechanism Towards Detecting, Locating and Preventing Second-Order SQL Injections', *IEEE Trans. Softw. Eng.*, vol. 50, no. 7, pp. 1807–1826, Jul. 2024, doi: 10.1109/TSE.2024.3400404.

[15] S. O. Abioye *et al.*, 'Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges', *J.*

Build. Eng., vol. 44, p. 103299, Dec. 2021, doi:10.1016/j.jobe.2021.103299.

[16] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, 'A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions', *Appl. Sci.*, vol. 12, no. 8, p. 4077, Apr. 2022, doi: 10.3390/app12084077.

[17] S. Chakraborty, S. K. Pandey, S. Maity, and L. Dey, 'Detection and Classification of Novel Attacks and Anomaly in IoT Network using Rule based Deep Learning Model', *SN Comput. Sci.*, vol. 5, no. 8, p. 1056, Nov. 2024, doi: 10.1007/s42979-024-03429-5.

[18] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, 'Cyber-physical systems security: Limitations, issues and future trends', *Microprocess. Microsyst.*, vol. 77, p. 103201, Sep. 2020, doi: 10.1016/j.micpro.2020.103201.

[19] C. Turner, R. Jeremiah, D. Richards, and A. Joseph, 'A Rule Status Monitoring Algorithm for Rule-Based Intrusion Detection and Prevention Systems', *Procedia Comput. Sci.*, vol. 95, pp. 361–368, 2016, doi: 10.1016/j.procs.2016.09.346.

[20] A. K. Tyagi and P. Chahal, 'Artificial Intelligence and Machine Learning Algorithms':, in *Advances in Computer and Electrical Engineering*, R. Kashyap and A. V. S. Kumar, Eds., IGI Global, 2020, pp. 188–219. doi: 10.4018/978-1-7998-0182-5.ch008.

[21] S. K. Shandilya, A. Datta, Y. Kartik, and A. Nagar, 'Role of Artificial Intelligence and Machine Learning', in *Digital Resilience: Navigating Disruption and Safeguarding Data Privacy*, in EAI/Springer Innovations in Communication and Computing. , Cham: Springer Nature Switzerland, 2024, pp. 313–399. doi: 10.1007/978-3-031-53290-0_6.

[22] M. Alqhtani, D. Alghazzawi, and S. Alarifi, 'Black-Box Adversarial Attacks Against SQL Injection Detection Model', *Contemp. Math.*, pp. 5098–5112, Nov. 2024, doi: 10.37256/cm.5420245292.

[23] M. W. A. Ashraf, A. R. Singh, A. Pandian, R. S. Rathore, M. Bajaj, and I. Zaitsev, 'A hybrid approach using support vector machine rule-based system: detecting cyber threats in internet of things', *Sci. Rep.*, vol. 14, no. 1, p. 27058, Nov. 2024, doi: 10.1038/s41598-024-78976-1.

[24] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, 'Network Intrusion Detection for IoT Security Based on Learning Techniques', *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[25] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv, 2019, arXiv:1901.03407. [Online]. Available: https://arxiv.org/abs/1901.03407.

[26] S. Sharma, P. Zavarsky, and S. Butakov, 'Machine Learning based Intrusion Detection System for Web-Based Attacks', in *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, Baltimore, MD, USA: IEEE, May 2020, pp. 227–230. doi:10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048.

[27] M. S. Darweesh *et al.*, 'Random Forest-Based NIDS: Advancing Network Threat Detection', Aug. 07, 2024, *In Review*. doi:10.21203/rs.3.rs-4737281/v1.

[28] M. Homaei, Ó. Mogollón-Gutiérrez, J. C. Sancho, M. Ávila, and A. Caro, 'A review of digital twins and their application in cybersecurity based on artificial intelligence', *Artif. Intell. Rev.*, vol. 57, no. 8, p. 201, Jul. 2024, doi: 10.1007/s10462-024-10805-3.

[29] Ö. Kasim, 'An ensemble classification-based approach to detect attack level of SQL injections', *J. Inf. Secur. Appl.*, vol. 59, p. 102852, Jun. 2021, doi: 10.1016/j.jisa.2021.102852.

[30] A. Kumar, S. Dutta, and P. Pranav, 'Analysis of SQL injection attacks in the cloud and in WEB applications', *Secur. Priv.*, vol. 7, no. 3, p. e370, May 2024, doi: 10.1002/spy2.370.

[31] P. S. Muhuri, P. Chatterjee, X. Yuan, K. Roy, and A. Esterline, 'Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks', *Information*, vol. 11, no. 5, p. 243, May 2020, doi: 10.3390/info11050243.

[32] K. Dwivedi, A. Agrawal, A. Bhatia, and K. Tiwari, "A novel classification of attacks on blockchain layers: Vulnerabilities, attacks, mitigations, and research directions," arXiv, 2024, arXiv:2404.18090.

[33] F. O. Okello, D. Kaburu, and N. G. John, 'Automation-Based User Input Sql Injection Detection and Prevention Framework', *Comput. Inf. Sci.*, vol. 16, no. 2, p. 51, May 2023, doi: 10.5539/cis.v16n2p51.

[34] A. Luo, W. Huang, and W. Fan, 'A CNN-based Approach to the Detection of SQL Injection Attacks', in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, Beijing, China: IEEE, Jun. 2019, pp. 320–324. doi:10.1109/ICIS46139.2019.8940196.

[35] A. Paleyes, R.-G. Urma, and N. D. Lawrence, 'Challenges in Deploying Machine Learning: A Survey of Case Studies', *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–29, Jul. 2023, doi:10.1145/3533378.

[36] M. F. Gholami, F. Daneshgar, G. Beydoun, and F. Rabhi, 'Challenges in migrating legacy software systems to the cloud — an empirical study', *Inf. Syst.*, vol. 67, pp. 100–113, Jul. 2017, doi:10.1016/j.is.2017.03.008.

[37] Y. Liu and Y. Dai, 'Deep Learning in Cybersecurity: A Hybrid BERT–LSTM Network for SQL Injection Attack Detection', *IET Inf. Secur.*, vol. 2024, pp. 1–16, Apr. 2024, doi: 10.1049/2024/5565950.

[38] D. Pessach and E. Shmueli, 'A Review on Fairness in Machine Learning', *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–44, Mar. 2023, doi: 10.1145/3494672.

[39] H. Chen and M. A. Babar, 'Security for Machine Learning-based Software Systems: A Survey of Threats, Practices, and Challenges', *ACM Comput. Surv.*, vol. 56, no. 6, pp. 1–38, Jun. 2024, doi:10.1145/3638531.

[40] Y. Yuan, Y. Lu, K. Zhu, H. Huang, L. Yu, and J. Zhao, 'A Static Detection Method for SQL Injection Vulnerability Based on Program Transformation', *Appl. Sci.*, vol. 13, no. 21, p. 11763, Oct. 2023, doi:10.3390/app132111763.

[41] Y. Liu and Y. Dai, 'Deep Learning in Cybersecurity: A Hybrid BERT–LSTM Network for SQL Injection Attack Detection', *IET Inf. Secur.*, vol. 2024, pp. 1–16, Apr. 2024, doi: 10.1049/2024/5565950.

[42] F. Q. Kareem *et al.*, 'SQL Injection Attacks Prevention System Technology: Review', *Asian J. Res. Comput. Sci.*, pp. 13–32, Jul. 2021, doi: 10.9734/ajrcos/2021/v10i330242.

[43] Z. Marashdeh, K. Suwais, and M. Alia, 'A Survey on SQL Injection Attack: Detection and Challenges', in *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan: IEEE, Jul. 2021, pp. 957–962. doi: 10.1109/ICIT52682.2021.9491117.

[44] Z. C. S. S. Hlaing and M. Khaing, 'A Detection and Prevention Technique on SQL Injection Attacks', in *2020 IEEE Conference on Computer Applications(ICCA)*, Yangon, Myanmar: IEEE, Feb. 2020, pp. 1–6. doi: 10.1109/ICCA49400.2020.9022833.

[45] S. Kumar, M. Mahajan, and S. Batra, 'A Recent Study of Machine Learning Based Techniques for the Detection of Cyber-Attacks on Web Applications', in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Gautam Buddha Nagar, India: IEEE, Sep. 2023, pp. 153–158. doi:10.1109/IC3I59117.2023.10397832.

[46] D. Mitropoulos, P. Louridas, M. Polychronakis, and A. D. Keromytis, 'Defending Against Web Application Attacks: Approaches, Challenges and Implications', *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 188–203, Mar. 2019, doi:10.1109/TDSC.2017.2665620.

[47] A. Abebe, Y. Belay, A. Belay, and S. Gebeyehu, 'SQL Injection Attacks Detection: A Performance Comparison on Multiple Classification Models', *Ethiop. Int. J. Eng. Technol.*, vol. 2, no. 1, pp. 22–38, Jul. 2024, doi: 10.59122/154CFC15.

[48] F. U. Rehman, S. Umbreen, and M. Rehman, 'MetaCDP: Metamorphic Testing for Quality Assurance of Containerized Data Pipelines', in *2024 IEEE Cloud Summit*, Washington, DC, USA: IEEE, Jun. 2024, pp. 135–142. doi: 10.1109/Cloud-Summit61220.2024.00029.

[49] J. R. Dora, L. Hluchý, and K. Nemoga, 'Ontology for Blind SQL Injection', *Comput. Inform.*, vol. 42, no. 2, pp. 480–500, 2023, doi:10.31577/cai_2023_2_480.

[50] B. Montaruli, G. Floris, C. Scano, L. Demetrio, A. Valenza, L. Compagna, D. Ariu, L. Piras, D. Balzarotti, and B. Biggio, "ModSec-AdvLearn: Countering Adversarial SQL Injections with Robust Machine Learning," arXiv, 2024, arXiv:2308.04964. [Online]. Available: https://arxiv.org/abs/2308.04964.

[51] A. A. Ashlam, A. Badii, and F. Stahl, 'A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks', in *2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, Hammamet, Tunisia: IEEE, Mar. 2022, pp. 513–517. doi:10.1109/IC_ASET53395.2022.9765948.

[52] A. Kumar, S. Dutta, and P. Pranav, 'Analysis of SQL injection attacks in the cloud and in WEB applications', *Secur. Priv.*, vol. 7, no. 3, p. e370, May 2024, doi: 10.1002/spy2.370.

[53] Q. Li, W. Li, J. Wang, and M. Cheng, 'A SQL Injection Detection Method Based on Adaptive Deep Forest', *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[54] S. Sharma, P. Zavarsky, and S. Butakov, 'Machine Learning based Intrusion Detection System for Web-Based Attacks', in *2020 IEEE 6th*

Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Baltimore, MD, USA: IEEE, May 2020, pp. 227–230. doi:10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048.

[55] M. Thilakraj, S. Anupriya, M. M. Cibi, and A. Divya, 'Detection of SQL Injection Attacks', in *2024 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal: IEEE, Apr. 2024, pp. 1515–1520. doi: 10.1109/ICICT60155.2024.10544579.

[56] J. Zulu, B. Han, I. Alsmadi, and G. Liang, 'Enhancing Machine Learning Based SQL Injection Detection Using Contextualized Word Embedding', in *Proceedings of the 2024 ACM Southeast Conference on ZZZ*, Marietta GA USA: ACM, Apr. 2024, pp. 211–216. doi:10.1145/3603287.3651187.

[57] E. Peralta-Garcia, J. Quevedo-Monsalbe, V. Tuesta-Monteza, and J. Arcila-Diaz, 'Detecting Structured Query Language Injections in Web Microservices Using Machine Learning', *Informatics*, vol. 11, no. 2, p. 15, Apr. 2024, doi: 10.3390/informatics11020015.

[58] X. Wang, J. Zhai, and H. Yang, 'Detecting command injection attacks in web applications based on novel deep learning methods', *Sci. Rep.*, vol. 14, no. 1, p. 25487, Oct. 2024, doi: 10.1038/s41598-024-74350-3.

[59] A. Odeh and A. A. Taleb, 'Ensemble learning techniques against structured query language injection attacks', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 35, no. 2, p. 1004, Aug. 2024, doi:10.11591/ijeecs.v35.i2.pp1004-1012.