Journal of Theoretical and Applied Information Technology
30th April 2025. Vol.103. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING SQL INJECTION (SQLI) MITIGATION BY REMOVING MALICIOUS SQL PARAMETER VALUES USING LONG SHORT-TERM MEMORY (LSTM) NEURAL NETWORKS

[1],NWABUDIKE AUGUSTINE, [1]ABU BAKAR MD. SULTAN, [1],MOHD HAFEEZ OSMAN,[2] KHAIRONI YATIM SHARIF
[1]Faculty of Computer Science and Information Technology University Putra Malaysia, Malaysia.
[2]Department of Computer & Information Sciences, Universiti Teknologi Petronas,Malaysia
Email: [1]augustineeh@gmail.com;[1]abakar@upm.edu.my,
[1]hafeez@upm.edu.my,[2]khaironi@upm.edu.my
*Corresponding author:.abakar@upm.edu.my

ABSTRACT

Web applications are increasingly targeted by cyber attacks. With SQL injection being a significant vulnerability, it caused an estimated USD4 billion global economic loss in 2022. Research has been conducted to explore methods to mitigate the impact of these attacks either by detecting them immediately or preventing them in their tracks. However, conventional approaches such as rule-based or signature-based systems have limitations as they cannot adapt to new or obscure attack patterns. This study explores the potentials of using Long Short-Term Memory (LSTM) neural network-based architectures for sanitization of SQL parameter values from malicious characters and shows that the result demonstrates that LSTM is the top performer, consistently achieving near-perfect accuracy, precision, recall, and F1-scores of 99.66% effectively. In a nutshell, this work has made a very strong and scalable contribution to web application security problems and demonstrates the possibility of SQLi mitigation using LSTM networks. By addressing such a critical gap in the literature, this study is a significant progress in the field of cyber security and robustness against obfuscation of attacks.

Keywords— *SQL Injection, Long Short-Term Memory, Web Application Security, Machine Learning, Cyber Attack*

1. INTRODUCTION

The security of web applications is a critical concern in today's interconnected world [1][2], where numerous digital interactions take place. Despite ongoing advancements in cyber security technologies [3][4], SQL injection (SQLi) remains one of the most persistent threats, as evidenced by the high number of incidents reported annually [5]. The Open Web Application Security Project (OWASP) 2023 report lists that SQL injection is still a high-level risk in web applications and more than 30% of all security intrusions have it as the attack vector [6][7]. The extent of this problem underscores a significant deficiency in contemporary security protocols and the need for necessary evolution in the landscape of security to proactively defend against SQLi assaults [8][9].

The primary challenge in cyber security lies in developing proactive solutions to counter sophisticated attacks. [10][11]. Despite the implementation of various technical defenses, such as parameterized queries, input validation and stored procedures, these measures have not consistently prevented SQL injection (SQLi) attacks [12]. In some cases, SQLi exploits lead to severe consequences, as existing mitigation techniques primarily provide reactive protection detecting malicious input only after it has been submitted or executed leaving vulnerabilities that attackers can still exploit [13]. Moreover, in an environment where attack vectors constantly evolve, traditional signature-based systems and static rule-based methods prove to be ineffective [14].

To address some of these limitations [15]. LSTM neural networks are introduced as a new technique to remove unwanted SQLi parameter values. LSTMs, being a specific category of recurrent neural network (RNN), are strong in learning long-term interdependencies among sequence data, which is sufficient in the analysis of the complex pattern of SQL queries [16][17]. They are especially well-suited to SQL queries because of their special capacity for recognizing sequence features with long dependencies and models which can remove malicious SQL parameters in real time. It is thus preventing the execution of malicious queries before they ever reach the database. This is because SQL execution is a sequential procedure that requires the retention of a complex set of information [18]. This proactive approach to SQLi mitigation could represent a significant breakthrough, improving the accuracy, speed and adaptability of web application defense [19].

Despite the growing body of research in machine learning for cyber security [20][21], the application of LSTM networks for SQLi mitigation is still underexplored [22][23]. The majority of research which frequently makes use of traditional machine learning models like decision trees and support vector machines have focused on detecting SQLi assaults rather than the removal of malicious query [24]. They typically fall short when confronted with novel or evolving attack strategies which are hallmarks of modern SQLi attempts [25]. . Furthermore, existing systems often generate false positives where legitimate queries are flagged as attacks, leading to disruptions in normal application operations [26].

This study examines the increasing complexity of SQLi attacks and the limitations of certain machine learning approaches. [27]. The study is designed to overcome these challenges through exploring state-of-the-art machine learning-based cyber security solutions [28]. Our Approach specifically targets the removal of malicious SQLi parameter values using LSTM, bridging a knowledge gap in real-time mitigation procedures [29]. Unlike existing research that primarily addresses classification models, this paper introduces an adaptive deep-learning approach that proactively eliminates malicious parameters in advance, bridging the gap between detection and prevention [30].

The research focuses on eliminating malicious SQL parameters by utilizing neural networks like LSTM models [31]. Machine learning-based classifiers can flag suspicious queries but fail to log or block the entire query correctly, increasing the likelihood of successful attack [32]. The proposed approach specifically removes the malicious components without altering the original query structure, fostering security without any compromise on functionality [33]. The study proposes a new SQL parameter filtering based on deep learning models in the form of LSTM [34]. The new knowledge involves SQL parameter filtering that goes beyond conventional detection methods by utilizing LSTM-based deep learning models that remove SQL injection attacks. The following research questions were the focus of this investigation.

- Question 1: What extent can Long Short-Term Memory (LSTM) neural networks enhance SQLi mitigation by effectively neutralizing malicious SQL parameter values compared to conventional security approaches?

- Question 2: How do LSTM networks surpass traditional SQL injection defence mechanisms in effectively removing malicious parameter values while improving accuracy, precision, recall, and F1-score and what are the broader implications for modern cyber security frameworks?

The contributions of this work are as follows:

- ✓ SQL Parameter Filtering: It implements an advanced filtering mechanism to prevent the execution of malicious queries, enhancing database security.

- ✓ Comparative Performance Analysis: It evaluates the effectiveness of the proposed LSTM-based approach in comparison to traditional machine learning and rule-based methods, focusing on accuracy, false positive rates and adaptability.

- ✓ Practical Implications: It addresses a critical gap in cyber security by demonstrating how the LSTM model can protect against both known and emerging SQL injection threats, reinforcing proactive security measures.

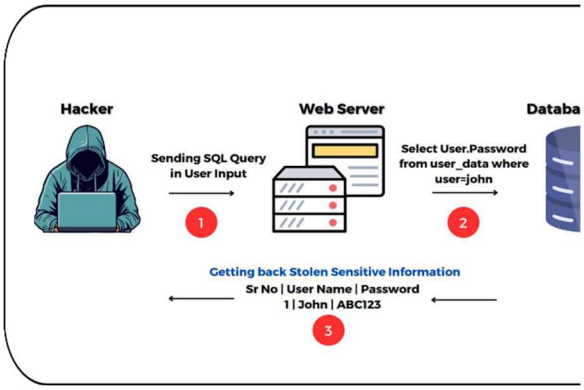

Figure 1: Process of SQL Injection Attack

2. LITERATURE REVIEW

Prior research has mostly used a combination of static and dynamic methods to remove SQL attribute values. However, very few have used machine learning.

1. Current Understanding of SQL Injection Attacks
In some SQL injection attacks, a malicious SQL query can be introduced into database servers using web application input fields [35], [36]. If successful, attackers could erase confidential information, change database records, or pretend to be administrators [37]. Commonly used SQL injection variants are union-based, error-based and blind SQLi, which exploit vulnerabilities in query structure, making the application vulnerable to SQLi [38], [39]. Current research presumed SQLi as the threat of injecting a command that would allow a hacker access to an administrator level in the database. It showed that 84% of targeted SQLi attacks could lead to information leakage of sensitive information in a controlled environment [40]. This indicates the pervasive nature of the threat and the urgency for sophisticated counter measures.

2. Conventional Mitigation Strategies
Web application firewalls (WAFs), parameterized queries and query sanitization are some of the most popular methods for mitigating SQL injection (SQLi) [41]. For instance, we can use parameterized queries to prevent any kind of SQL statement from being run or input validation to ensure that user input follows the forms we specify [42]. WAFs analyze and filter HTTP requests for malicious activity and interference [43]. Traditional defence methods like input validation and WAFs have limitations such as ineffectiveness against polymorphic and the possibility of false positives and negatives with SQLi payloads [44].Traditional static rule defences cannot adapt to these attacks, emphasizing the need for more intelligent solutions to overcome these limitations and enhance their effectiveness [45].

3. Emergence of Machine Learning in SQLi Mitigation

The use of machine learning (ML) for SQLi detection is one of the developments in cyber security. By examining patterns in historical attack data, machine learning models identify anomalous behaviour that may be a sign of SQLi [46]. ML models analyze trends in the past attacking data to detect abnormal behaviour that indicates the possible existence of SQLi. As emphasized by Sun et al[47], this reinforces the notion that machine learning-based systems offer a notable improvement in accuracy over traditional detection approaches [48].

Machine learning approaches have been used for SQL Injection detection, each with its unique positives and negatives [49]. While Support Vector Machines (SVMs) are effective in classifying data as either malicious or benign, they require a lot of feature engineering and can have problems with high-dimensional data [50]. Because LSTM neural networks learn temporal correlations in the input sequence, they have demonstrated approximately 95% accuracy in SQLi detection, outperforming previous methods [51] .

4. Advances in LSTM Neural Networks

Today, industry best practices involve the use of rule-based systems, like Web Application Firewalls (WAFs), which are ill-equipped to handle the rising complexity and frequency of SQL Injection (SQLi) attacks [52]. According to a 2022 report from the Cyber Security & Infrastructure Security Agency (CISA), 68% of organizations use WAFs at the front to defend their web applications [53]. Although ML-based solutions is promising, they also have drawbacks, such as high computational costs, the need for diverse and sizable datasets for efficient training and interpretability issues brought on by their black-box nature [54]. ML-based approaches offer great potentials for addressing the complexity of multi-dimensional time series analyses [55]. The integration has revealed multiple challenges, such as a substantial computational burden, the need for big quantities of heterogeneous multi-dimensional data for training and testing and the opaque nature of deep learning models [56].

The prior work has made extensive use of signature-based detection mechanisms and static rule sets, which are not generalizable against adaptive SQLi attack patterns [57]. Signature-based mechanisms rely on previously known attack patterns and hence, they are of no use against novel and obfuscated SQL injection techniques. Although static rule-based mechanisms offer structured protection, they lack the necessary flexibility to safeguard against newly emerging threats [58]. Our approach integrates LSTM neural networks to learn sequential dependencies in SQL queries so that it enables more powerful and adaptive attack removal. LSTMs are particularly useful because they maintain long-term contextual information, which makes them appropriate for the analysis of SQL query structures.

Although some efforts have been made to filter SQLi attacks using machine learning, little research has focused on removing harmful SQL parameters with neural networks, particularly LSTM models, before execution on the backend database. This study bridges that gap by suggesting a dynamic filtering method that removes harmful parameter values before query execution, providing an extra layer of security to database-driven applications. Additionally, the study compares the effectiveness of LSTM neural networks in SQLi attack prevention and demonstrates that LSTM is more effective than other models with 99.66% accuracy, precision, recall, and F1-score, which means that it is a highly effective solution for SQLi attack prevention.

3. MATERIALS AND METHODS

This section describes the tools and methods used to develop, train and evaluate machine learning and deep learning for SQL injection removal in text-based SQL queries.

3.1 Dataset

The study utilizes a comprehensive dataset from Kaggle.com, consisting of 30,919 data items, to train machine learning models to differentiate between malicious and normal inputs. The dataset is divided into two classes: SQL injection statements (36.64%, 11,330 samples) and non-SQL injection attacks (63.36%, 19,589 samples) illustrated in Table 1, The dataset is divided into 80% for training and 20% for validation [59]. The goal is to develop a robust SQLi mitigation mechanism that can identify and filter harmful SQL parameter values with high accuracy.

Table I. Kaggle _ Dataset

Label	Description	Count	Ratio
1	SQL injection statement	11,330	36.64
0	Non-SQL injection attack	19,589	63.36

3.2 Machine learning library

Python is a widely used and versatile programming language for application development. It is easy to use and learn, allowing quick deployment and integration with various systems. Python is compatible with Windows, Linux, Mac and Symbian and operates independently without a compiler. It features parallel execution libraries for improved system speed and code support for multiple CPUs/GPUs. Scikit-Learn, an open-source machine learning package, was created as a Python extension for the SciPy library.

It allows the use of various machine learning algorithms, including classification and clustering and offers features like integrated feature extraction and model evaluation. Due to its ease of use and pre-resources, Scikit-Learn is often used by researchers for prototyping.

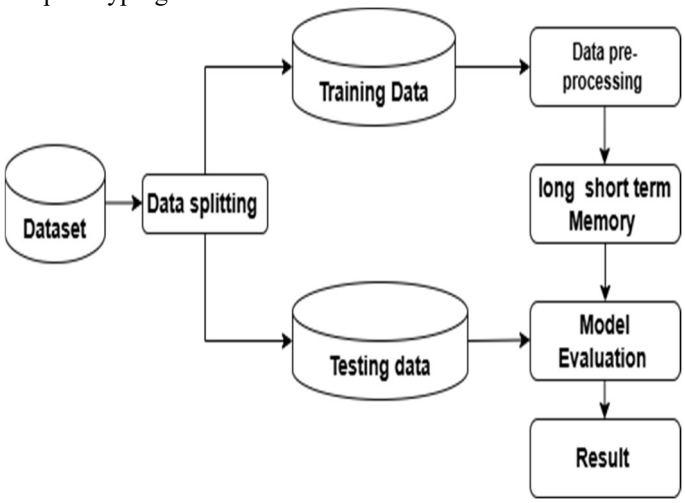

Figure 2: Model Training Workflow

3.3 Methodology

A comprehensive approach to building an LSTM-based model to remove SQL parameter values with the aid of a Python-based model that removes SQL parameter values and renames columns tokenizes text data and changes labels to number formats. The model is divided into sets of exercises and tests to evaluate invisible data. It uses embedding and LSTM layers to capture sequence dependencies inherent in SQL queries. The model architecture

consists of two LSTM layers to investigate time dependency, an embedding layer to convert the input sequence to a fixed-size solid vector and a solid layer with softmax activation functions to determine the output probability for each class. (See figure 2) The model is trained on training data with authentication separation and predicts hygienic outputs for the whole data set. Assessment measures like accuracy, precision, recall, and F1 scores are computed to evaluate the model's performance. In table 2, the result displays the total number of correctly sanitized entries and details for the first 30,814 entries, including the input query and predicted sanitized output.

Table 2: Show Queries and Sanitized Output

S/no	Query	Sanitized Query Output
1	select * from users where id = '1' * (\ ) or 1 = 1 -- 1'	select from users where id '1' or 1 1 1'
2	select * from users where id = 1 or "? (" or 1 = 1 – 1	select from users where id 1 or or 1 1 1
3	select * from users where id = 1 or \.<\ or 1 = 1 – 1	select from users where id 1 or or 1 1 1
4	admin') or ('1' = '1'/*	admin' or '1' '1'
5	select * from users where id = 1 +$ 1 or 1 = 1 – 1	select from users where id 1 1 or 1 1 1
6	select * from information_schema.tables--	select from information schema tables
7	declare @q nvarchar (200) select @q = 0x770061 ...	declare q nvarchar 200 select q
8	") ;waitfor delay '0:0: TIME '--	waitfor delay '0 0 time '
9	admin" or "1" = "1"/*	admin or 1 1
10	)) or pg_sleep (	Orpgsleeptime
11	select * from users where id = 1.&&1 union select 1,version ( ) -- 1	select from users where id 1 1 union select 1 version 1
12	and 1 in (select var from temp) --	and 1 in select var from temp
13	union select * from users where login = char ...	union select from users where login char
14	1 uni/**/on select all from where	1 uni on select all from where
15	: admin') or '1' = '1'#	admin' or '1' '1'
16	select * from information_schema.tables--	select from information schema tables
17	declare @s varchar (22) select @s =	declare s varchar 22 select s
18	;wait for delay '0:0: TIME '--	waitfor delay '0 0 time '
19	x' AND userid IS NULL; --	x' and userid is null
20	x' and members.email is NULL; --	x' and members email is null

3.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an architecture of artificial recurrent neural networks (RNNs) for deep learning [60]. It engages feedback connections so that it can take advantage of temporal dependencies across a sequence of data. [61]. The vanishing or ballooning gradients issue that arises when standard RNNs are trained using sequence data is especially addressed by LSTMs [62]. They are particularly helpful for sequential data tasks like

natural language processing, speech recognition and time series forecasting [63], LSTM networks include a memory cell that can hold information across longer sequences. A memory cell contains three main components, an input gate, a forget gate and an output gate. (See figure 3) These gates manage the information flow in and out of the memory cell [64]. The input gate decides how much input to save while the forget erases previous data, while the output gate decides how much of the memory cell contents to use in computing the hidden state [65]. They are well-suited for tasks that need modelling long-term dependencies like recognizing speech, translating human languages and few others.

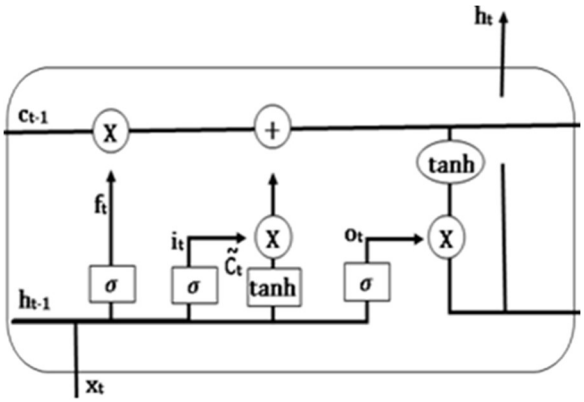

Figure 3: Structure of LSTM[66]

The forget gate unit ft decides what information will be cut from the cell state. This gate will take $ht-1$, and Xt as input for each number in cell state (st) and will output a value between 0 to 1. Ct−1 = "fully reserved" and 0 = "completely discarded.

The following algorithm are derived from Wen et al [67]

$ft = (Wt* [ht-1, 1] + bf)$ (1)

The input gate A is responsible for determining the amount of fresh data that are introduced to the cell state. It includes two steps:

At first, a tanh layer creates the vector or alternate content $Ct-1$ for updating and then subsequently the sigmoid layer determines which information to keep.

$.t = (Wc* [ht-1, xt] + bc)$ (2)

$\tilde{C}t = \tanh(Wt* [ht-1, xt] + bc)$ (3)

The two terms are then multiplied in this step and the resulting matrix is fed to $Ct-1$ for a state update.

$Ct = ft*Ct-1 + it* \tilde{C}t$ (4)

In the end, you have to decide what value you want to bring into existence. This is an abstracted value based on how our cell state needs to change. The sigmoid layer chooses what small part of the cell state to write to. We run that through a tanh (to force the value to be between -1 and 1) and multiply that by what came out of our sigmoid gate. We do not know what we will end up deciding to output.

$.ot = (W0* [ht-1, xt] + b0)$ (5)

$ht = ot* \tanh(Ct)$ (6)

Algorithm for SQL attribute Value

Algorithm Remove Keywords From SQL Query

Input: SQL Query,

Keywords List Output: Processed SQL Query

Step 1: Input SQL Query

SQL Query ← "SELECT * FROM users WHERE name = 'John' AND age = 25"

Step 2: Identify Keywords to Remove Keywords List ← ["name" "age"]

Step 3: Create Regex Pattern for Keywords Keyword Pattern ←

Step 4: Remove Identified Keywords Processed SQL Query ← Remove Keywords using Keyword Pattern from SQL Query

Step 5: Cleanup the Query

Processed SQL Query ← Remove Extra Spaces and Dangling Operators from Processed SQL Query

Step 6: Output Processed Query Return Processed SQL Query End Algorithm

3.5 Decision Tree

Decision trees are a versatile and easily understood method for predictive modelling in machine learning, consisting of internal nodes testing properties, branches representing attribute values and leaf nodes representing a final choice [68], [69]. Decision trees are supervised learning algorithms used for modelling and forecasting results based on data properties, tackling regression and classification issues, making them easy to apply and interpret [70].

3.6 Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning algorithm that performs classification and regression tasks by finding the best hyperplane with the maximum margin. It separates data points into N-dimensional classes, allowing it to be used for text, image, spam, handwriting recognition, gene expression, face detection and outlier detection [71][72]. The algorithm tries to separate the classes by having a maximum margin between the closest points in the two classes [73].

3.7 K-Nearest Neighbour (KNN) Algorithm

The K-NN algorithm is a popular machine learning method that focuses on simplicity and ease of implementation. It supports both numerical and categorical variables, making it suitable for various datasets in classification and regression problems. The algorithm determines the K closest neighbours of an input data point based on their distances. It predicts the input data point's class label based on the most prevalent class label among the K neighbours [74][75].

4. FINDINGS

In this section, the answers to the research questions are discussed in depth. The key conclusions, concepts and how they relate to the objectives of the study are emphasized as each issue is carefully addressed.

4.1 RQ1, the study explores LSTM efficacy in the real-time removal of malicious SQL inputs.

The study compares various SQL removal of malicious SQL parameter values and reveals that the LSTM model outperforms all others in terms of accuracy, precision, recall, and F1 score. It captures sequential patterns with a 99.66% accuracy rate, offering superior robustness for removing malicious SQL parameter values. The Decision Tree model has a 99.21% accuracy rate, while the Support Vector Machine (SVM) has a 95.94% accuracy rate. The K-Nearest Neighbors model has an 85.35% accuracy rate and an F1 score of 84.32%. According to the findings in Table 2, harmful SQL parameters can be filtered out using LSTM networks before they are processed on the backend database.

4.2 RQ2, LSTM networks enhance SQL injection mitigation with superior accuracy, precision, recall, and F1 score.

In this study, we aim to analyze the benefits of the LSTM neural network-based approach for the prevention of SQL injection (SQLi) attacks, which are measured through four performance metrics such as accuracy, precision, recall and F1-score in Table 3. LSTMs are best at learning complex SQL queries with mean behavior and sequential data because they employ semantic analysis. We will demonstrate that LSTMs surpass state-of-the-art algorithms for this task, achieving near-perfect removal rates and significantly lower false positive and negative rates (as evidenced by Figure 4 compared to Figures 5, 6, and 7). The research emphasizes the potentials of LSTM to serve as a reliable, proactive and robust SQLi mitigation solution, thus aiding in the security of databases and web applications.

5. DISCUSSION

5.1 Performance

Table 2: This text explains how SQL queries are filtered and how harmful input injections are removed. The query contains arbitrary SQL imports and vulnerabilities like SQL injection attacks, which can lead to database corruption or unwanted command execution. Identifying dangerous syntax and keywords is crucial for safeguarding the back-end database's integrity, secrecy and reliability which is sanitized to remove the injected logic. This demonstrates the ability of LSTMs or other mitigation methods to recognize and extinguish hostile patterns while maintaining the integrity of the genuine query structure. This improves database security and protects against SQL injection attacks.

The results in Table 3 clearly show the dominance of LSTM in identifying SQL injection (SQLi) with an impressive 99.66% accuracy, precision, recall and F1-score. This is a sign that the model can differentiate very well between complex sequential patterns in SQL queries and can distinguish between malicious and benign inputs. The decision Tree (DT) model, which is also satisfactory with 99.21% accuracy, relies on pre-defined rules that may lead to overfitting and thus become less adaptable in reacting to dynamic attack patterns. Support Vector

Machine (SVM) is satisfactory with 95.94% accuracy, but because it relies on manual extraction of features, it becomes ineffective in dealing with structural changes faced in SQL injection attacks.

Conversely, K-Nearest Neighbors (KNN) does the worst with 85.35% accuracy and an F1-score of 84.32%, indicating poor generalization to different SQLi attack patterns. The relatively high precision of KNN (88.06%) compared to its recall (85.35%) is an indicator that even though it accurately removes a significant number of malicious attacks, it is unable to uncover most actual SQLi threats, thereby making it insecure to employ in actual cyber security activities. The overall outcomes confirm that deep learning methods and more so LSTM, possess a significant advantage over traditional machine learning models with higher accuracy, high adaptability and real-time counter measure availability for SQLi detection.

5.2 Limitation

There are several shortcomings in LSTM model research, for example, LSTM model capability to struggle with SQL injection attacks in real life, dataset diversity dependence, expensive computational cost, potential disruption of the functionality by false positives and false negatives, the model's incapability to maintain pace with dynamic threats, and black-box characteristic, which could limit its adoption in security frameworks. The richness of the dataset makes it efficient, and an unbalanced dataset may result in biases or overfitting. Moreover, the computational cost of LSTMs raises questions about their deployment in real time for high-traffic online applications. While false negatives create security problems, false positives can interact with normal speed to create interference. or explore hybrid approaches to further enhance robustness.

5.3 Threat of Validity

The study takes into account the limitations of SQL parameter value deletion in machine learning models for Kaggle data sets. The internal security threats of SQL parameter value Removal include the non-representativeness of the data, defects in data sets, and measurement test deficiencies that reduce the generalizability and performance of the models by using other patterns, External threats in the case of SQL injection models come in the form of unpredictability and instability of attacks that occur in reality and hence are not even available in train and test datasets. Advanced models like LSTMs are computationally costly and cannot be realistically applied in low-latency and computationally light real-time environments. Construct validity is also a problem, with computational cost and interpretability issues arising from the complexity of the model. Addressing such problems is essential to create solid, efficient, and feasible models for application in production security systems.

Table 3: Model Comparison

Model	Accuracy	Precision	Recall	F1=Score
LSTM	99.66	99.66	99.66	99.66
DT	99.21	99.21	99.21	99.21
SVM	95.94	96.10	95.94	95.90
KNN	85.35	88.06	85.35	84.32

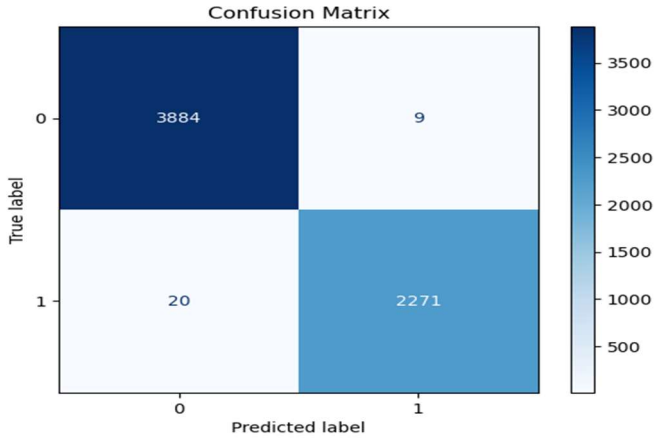

Figure 4 : Confusion Matrix for LSTM

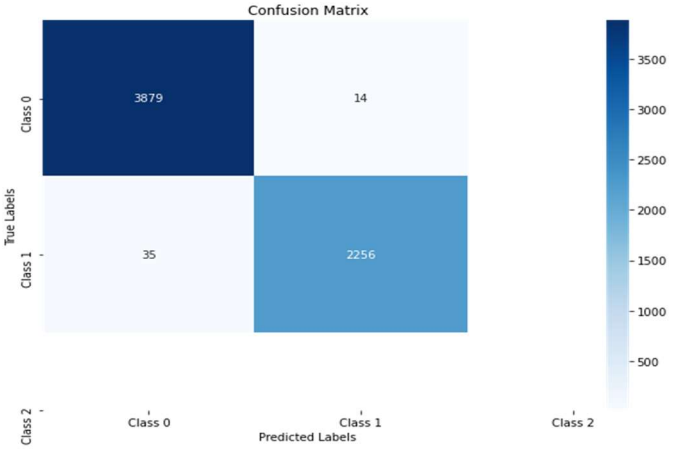

Figure 5: Confusion Matrix for DT

Journal of Theoretical and Applied Information Technology
30th April 2025. Vol.103. No.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

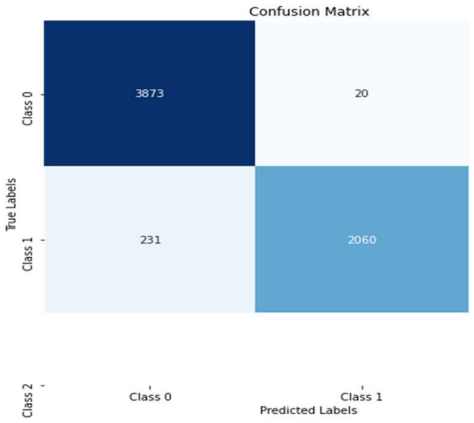

Figure 6: Confusion Matrix for SVM

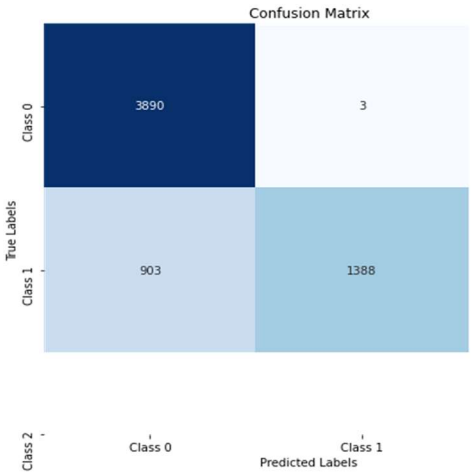

Figure 7: Confusion Matrix for KNN

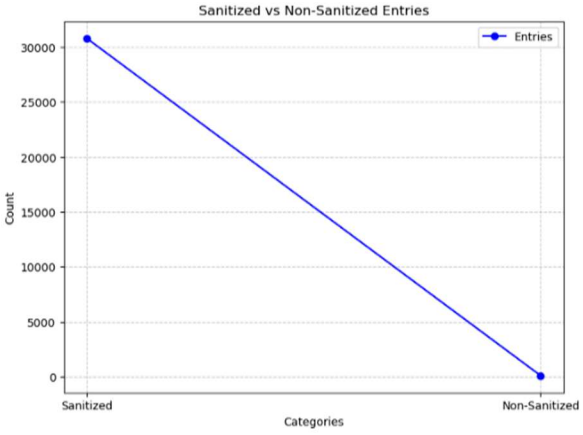

Figure 8 Chart That Shows Sanitizes And Non Sanitizes

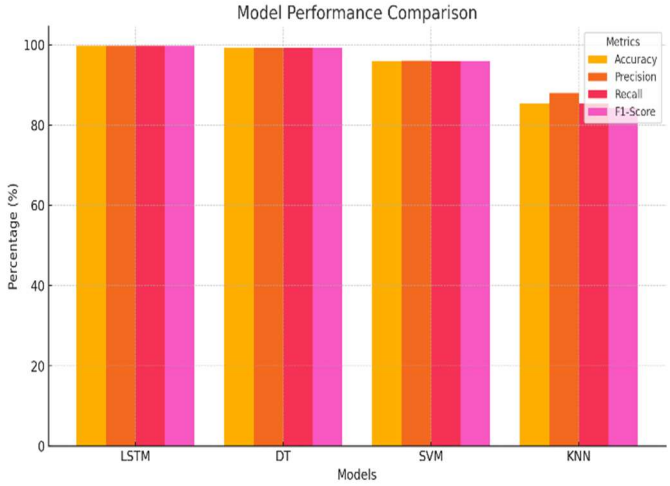

Figure 9: Model Performance

6. CONCLUSION

This research was successful in fulfilling its major objectives through achieving precise answers to RQ1 and RQ2. The research focuses on comparing different approaches for eliminating malicious SQL parameter values and finds that the LSTM model outperforms all others in terms of accuracy, precision, recall and F1-score. The study also examines the strengths of the LSTM neural network-based approach in SQL injection (SQLi) attack prevention. LSTM outperforms other models with 99.66% accuracy, precision, recall and F1-score, demonstrating its better capability in finding malicious SQL patterns and reducing instances of false positives and false negatives. This paper introduces a novel SQL parameter filtering method based on the capabilities of Long Short-Term Memory (LSTM) networks, the suggested LSTM-based filtering method leverages deep learning for dynamic analysis of SQL parameters, and effectively filters and removes SQL injection attacks before they can penetrate a system. This is a tremendous advancement in SQL injection remediation, providing a more adaptive, effective, and robust security solution.

The authors developed a framework using LSTM-based real-time sanitization of SQL parameters to address limitations in conventional SQLi defense mechanisms. They conducted a literature review, developed a methodology for preserving natural SQL query structures, and created a large test dataset. The LSTM model achieved STATE-of-the-art accuracy, precision, recall, and F1-score of 99.66%, demonstrating its excellence over traditional detection methods. The

research highlights the resilience of LSTM against obfuscated attacks and its practical applications in web security. The study contributes to bridging a critical knowledge gap in SQLi mitigation, offering real-time cyber security solutions, and offering future research directions for adaptive threat detection. Further research should focus on integrating this approach with all web application

REFERENCES

[1] A. Attaallah, A. Algarni, and R. Ahmad Khan, 'Managing Security-Risks for Improving Security-Durability of Institutional Web-Applications: Design Perspective', *Comput. Mater. Contin.*, vol. 66, no. 2, pp. 1849–1865, 2021, doi: 10.32604/cmc.2020.013854.

[2] A. Mohammed, J. Alkhathami, H. Alsuwat, and E. Alsuwat, 'Security of Web Applications: Threats, Vulnerabilities, and Protection Methods', *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 8, pp. 167–176, Aug. 2021, doi: 10.22937/IJCSNS.2021.21.8.22.

[3] H. Z. Ashraf, 'Security Concerns: Machine Learning Approaches in the Modern Era and Web Engineering', in *Advances in Web Technologies and Engineering*, I. A. Shah and N. Z. Jhanjhi, Eds., IGI Global, 2024, pp. 241–268. doi: 10.4018/979-8-3693-3703-5.ch012.

[4] R. Kumar, A. Baz, H. Alhakami, W. Alhakami, A. Agrawal, and R. A. Khan, 'A hybrid fuzzy rule-based multi-criteria framework for sustainable-security assessment of web application', *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 2227–2240, Jun. 2021, doi: 10.1016/j.asej.2021.01.003.

[5] A. Ahmad, K. C. Desouza, S. B. Maynard, H. Naseer, and R. L. Baskerville, 'How integration of cyber security management and incident response enables organizational learning', *J. Assoc. Inf. Sci. Technol.*, vol. 71, no. 8, pp. 939–953, Aug. 2020, doi: 10.1002/asi.24311.

[6] A. Khanum, S. Qadir, and S. Jehan, 'OWASP-Based Assessment of Web Application Security', in *2023 18th International Conference on Emerging Technologies (ICET)*, Peshawar, Pakistan: IEEE, Nov. 2023, pp. 240–245. doi: 10.1109/ICET59753.2023.10374730.

[7] A. Fadlil, I. Riadi, and M. A. Mu'min, 'Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework', *Int. J. Eng.*, vol. 37, no. 4, pp. 635–645, 2024, doi: 10.5829/IJE.2024.37.04A.06.

[8] B. S. Lakshmi, D. Kovvuri, H. N. V. Bolisetti, D. S. Chikkala, S. Karri, and G. Yadlapalli, 'A Proactive Approach for Detecting SQL and XSS Injection Attacks', in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India: IEEE, Jun. 2024, pp. 1415–1420. doi: 10.1109/ICAAIC60222.2024.10575599.

[9] S. S. Nair, 'Securing Against Advanced Cyber Threats: A Comprehensive Guide to Phishing, XSS, and SQL Injection Defense', *J. Comput. Sci. Technol. Stud.*, vol. 6, no. 1, pp. 76–93, Jan. 2024, doi: 10.32996/jcsts.2024.6.1.9.

[10] I. Butun, P. Osterberg, and H. Song, 'Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures', *IEEE Commun. Surv. Tutor.*, vol. 22, no. 1, pp. 616–644, 2020, doi: 10.1109/COMST.2019.2953364.

[11] Y. Li and Q. Liu, 'A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments', *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[12] K. Ahmad and M. Karim, 'A Method to Prevent SQL Injection Attack using an Improved Parameterized Stored Procedure', *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120636.

[13] A. Goyal and P. Matta, 'Beyond the Basics: A Study of Advanced Techniques for Detecting and Preventing SQL Injection Attacks', in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India: IEEE, Sep. 2023, pp. 628–631. doi: 10.1109/ICOSEC58147.2023.10276077.

[14] A. Hernández-Rivas, V. Morales-Rocha, and J. P. Sánchez-Solís, '10.1007/978-3-031-69769-2_8', in *Innovative Applications of Artificial Neural Networks to Data Analytics and Signal Processing*, vol. 1171, G. Rivera, W. Pedrycz, J. Moreno-Garcia, and J. P. Sánchez-Solís, Eds., in Studies in Computational Intelligence, vol. 1171. , Cham: Springer Nature Switzerland, 2024, pp. 181–219. doi: 10.1007/978-3-031-69769-2_8.

[15] J. P. Tennant and T. Ross-Hellauer, 'The limitations to our understanding of peer review', *Res. Integr. Peer Rev.*, vol. 5, no. 1,

p. 6, Dec. 2020, doi: 10.1186/s41073-020-00092-1.

[16] M. Alghawazi, D. Alghazzawi, and S. Alarifi, 'Deep Learning Architecture for Detecting SQL Injection Attacks Based on RNN Autoencoder Model', *Mathematics*, vol. 11, no. 15, p. 3286, Jul. 2023, doi: 10.3390/math11153286.

[17] V. V. Meduri, K. Chowdhury, and M. Sarwat, 'Evaluation of Machine Learning Algorithms in Predicting the Next SQL Query from the Future', *ACM Trans. Database Syst.*, vol. 46, no. 1, pp. 1–46, Mar. 2021, doi: 10.1145/3442338.

[18] T. Sheth, J. Anap, H. Patel, N. Singh, and Prof. R. R. B, 'Detection of SQL Injection Attacks by giving apriori to Q-Learning Agents', in *2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, London, United Kingdom: IEEE, May 2023, pp. 1–6. doi: 10.1109/GlobConET56651.2023.10149965.

[19] J. Vinodh and E. G. Mary Kanaga, 'SQL Injection Detection Using Optimized Recurrent Neural Network Integrated with N-Gram: A Comparison', in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, Karaikal, India: IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/IConSCEPT61884.2024.10627822.

[20] G. Apruzzese *et al.*, 'The Role of Machine Learning in Cybersecurity', *Digit. Threats Res. Pract.*, vol. 4, no. 1, pp. 1–38, Mar. 2023, doi: 10.1145/3545574.

[21] 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.

[22] B. Isong, O. Kgote, and A. Abu-Mahfouz, 'Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems', *Electronics*, vol. 13, no. 12, p. 2370, Jun. 2024, doi: 10.3390/electronics13122370.

[23] N. Cahyadi, S. Nurgaida Yutia, and P. Dorand, 'Enhancing SQL Injection Attack Prevention: A Framework for Detection, Secure Development, and Intelligent Techniques', *J. Inform. Commun. Technol. JICT*, vol. 5, no. 2, pp. 138–148, Dec. 2023, doi: 10.52661/j_ict.v5i2.233.

[24] Mohammed A M Oudah and Mohd Fadzli Marhusin, 'SQL Injection Detection using Machine Learning: A Review', *Malays. J. Sci. Health Technol.*, vol. 10, no. 1, pp. 39–49, Apr. 2024, doi: 10.33102/mjosht.v10i1.368.

[25] M. Shahin, M. Maghanaki, A. Hosseinzadeh, and F. F. Chen, 'Advancing Network Security in Industrial IoT: A Deep Dive into AI-Enabled Intrusion Detection Systems', *Adv. Eng. Inform.*, vol. 62, p. 102685, Oct. 2024, doi: 10.1016/j.aei.2024.102685.

[26] T. D. Le, T. Le-Dinh, and S. Uwizeyemungu, 'Search engine optimization poisoning: A cybersecurity threat analysis and mitigation strategies for small and medium-sized enterprises', *Technol. Soc.*, vol. 76, p. 102470, Mar. 2024, doi: 10.1016/j.techsoc.2024.102470.

[27] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, 'Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets', *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, p. e4240, Apr. 2021, doi: 10.1002/ett.4240.

[28] N. Mohamed, 'Securing transportation web applications: An AI-driven approach to detect and mitigate SQL injection attacks', *J. Transp. Secur.*, vol. 17, no. 1, p. 2, Dec. 2024, doi: 10.1007/s12198-023-00269-x.

[29] A. A. Musa, A. Hussaini, W. Liao, F. Liang, and W. Yu, 'Deep Neural Networks for Spatial-Temporal Cyber-Physical Systems: A Survey', *Future Internet*, vol. 15, no. 6, p. 199, May 2023, doi: 10.3390/fi15060199.

[30] J. Malik, R. Muthalagu, and P. M. Pawar, 'A Systematic Review of Adversarial Machine Learning Attacks, Defensive Controls, and Technologies', *IEEE Access*, vol. 12, pp. 99382–99421, 2024, doi: 10.1109/ACCESS.2024.3423323.

[31] C.-J. Su and Y. Li, 'Recurrent neural network based real-time failure detection of storage devices', *Microsyst. Technol.*, vol. 28, no. 2, pp. 621–633, Feb. 2022, doi: 10.1007/s00542-019-04454-8.

[32] A. S. Kokaz and A. Kurnaz Türkben, 'A New Iots Security Framework Using Hybrid Machine Learning Techniques', *IETE J. Res.*, pp. 1–28, Feb. 2025, doi: 10.1080/03772063.2024.2447875.

[33] N. Bhateja, S. Sikka, and A. Malhotra, 'A Review of SQL Injection Attack and Various Detection Approaches', in *Smart and Sustainable Intelligent Systems*, 1st ed., N. Gupta, P. Chatterjee, and T. Choudhury, Eds.,

Wiley, 2021, pp. 481–489. doi: 10.1002/9781119752134.ch34.

[34] Daniel Ajiga, Patrick Azuka Okeleke, Samuel Olaoluwa Folorunsho, and Chinedu Ezeigweneme, 'Designing Cybersecurity Measures for Enterprise Software Applications to Protect Data Integrity', *Comput. Sci. IT Res. J.*, vol. 5, no. 8, pp. 1920–1941, Aug. 2024, doi: 10.51594/csitrj.v5i8.1451.

[35] Mohammed A M Oudah and Mohd Fadzli Marhusin, 'SQL Injection Detection using Machine Learning: A Review', *Malays. J. Sci. Health Technol.*, vol. 10, no. 1, pp. 39–49, Apr. 2024, doi: 10.33102/mjosht.v10i1.368.

[36] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, 'A systematic review of detection and prevention techniques of SQL injection attacks', *Inf. Secur. J. Glob. Perspect.*, vol. 32, no. 4, pp. 252–265, Jul. 2023, doi: 10.1080/19393555.2021.1995537.

[37] P. Sinha, A. K. Rai, and B. Bhushan, 'Information Security threats and attacks with conceivable counteraction', in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Kannur,Kerala, India: IEEE, Jul. 2019, pp. 1208–1213. doi: 10.1109/ICICICT46008.2019.8993384.

[38] S. Alazmi and D. C. De Leon, 'Customizing OWASP ZAP: A Proven Method for Detecting SQL Injection Vulnerabilities', in *2023 IEEE 9th 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)*, New York, NY, USA: IEEE, May 2023, pp. 102–106. doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00028.

[39] 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.

[40] 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.

[41] W. H. Rankothge, M. Randeniya, and V. Samaranayaka, 'Identification and Mitigation Tool for Sql Injection Attacks (SQLIA)', in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, RUPNAGAR, India: IEEE, Nov. 2020, pp. 591–595. doi: 10.1109/ICIIS51140.2020.9342703.

[42] S. Dolatnezhad and M. Amini, 'Preventing SQL Injection Attacks by Automatic Parameterizing Raw Queries Using Lexical and Semantic Analysis Methods', *Sci. Iran.*, vol. 0, no. 0, pp. 0–0, Jan. 2019, doi: 10.24200/sci.2019.21229.

[43] T. Li *et al.*, 'A Survey on Web Application Testing: A Decade of Evolution', 2024, *arXiv*. doi: 10.48550/ARXIV.2412.10476.

[44] J. L. López Delgado and J. A. López Ramos, 'A Comprehensive Survey on Generative AI Solutions in IoT Security', *Electronics*, vol. 13, no. 24, p. 4965, Dec. 2024, doi: 10.3390/electronics13244965.

[45] T. Sureda Riera, J.-R. Bermejo Higuera, J. Bermejo Higuera, J.-J. Martínez Herraiz, and J.-A. Sicilia Montalvo, 'Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review', *Sustainability*, vol. 12, no. 12, p. 4945, Jun. 2020, doi: 10.3390/su12124945.

[46] R. R. Choudhary, S. Verma, and G. Meena, 'Detection of SQL Injection attack Using Machine Learning', in *2021 IEEE International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES)*, Raipur, India: IEEE, Dec. 2021, pp. 1–6. doi: 10.1109/TRIBES52498.2021.9751616.

[47] H. Sun, Y. Du, and Q. Li, 'Deep Learning-Based Detection Technology for SQL Injection Research and Implementation', *Appl. Sci.*, vol. 13, no. 16, p. 9466, Aug. 2023, doi: 10.3390/app13169466.

[48] M. Hasan, Z. Balbahaith, and M. Tarique, 'Detection of SQL Injection Attacks: A Machine Learning Approach', in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, United Arab Emirates: IEEE, Nov. 2019, pp. 1–6. doi: 10.1109/ICECTA48151.2019.8959617.

[49] B. Mondal, A. Banerjee, and S. Gupta, 'review of SQLI detection strategies using machine learning', *Int. J. Health Sci.*, pp. 9663–9676, May 2022, doi: 10.53730/ijhs.v6nS2.7519.

[50] C. Pavithra and M. Saradha, 'A Comprehensive Classification Approach by

Integrating Principal Component Analysis and Support Vector Machines for Advanced Intrusion Detection Systems', *SN Comput. Sci.*, vol. 5, no. 8, p. 996, Oct. 2024, doi: 10.1007/s42979-024-03308-z.

[51] M. Majd and R. Safabakhsh, 'Correlational Convolutional LSTM for human action recognition', *Neurocomputing*, vol. 396, pp. 224–229, Jul. 2020, doi: 10.1016/j.neucom.2018.10.095.

[52] M. B. Islam and G. Governatori, 'RuleRS: a rule-based architecture for decision support systems', *Artif. Intell. Law*, vol. 26, no. 4, pp. 315–344, Dec. 2018, doi: 10.1007/s10506-018-9218-0.

[53] E. Tuyishime, T. C. Balan, P. A. Cotfas, D. T. Cotfas, and A. Rekeraho, 'Enhancing Cloud Security—Proactive Threat Monitoring and Detection Using a SIEM-Based Approach', *Appl. Sci.*, vol. 13, no. 22, p. 12359, Nov. 2023, doi: 10.3390/app132212359.

[54] V. Hassija *et al.*, 'Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence', *Cogn. Comput.*, vol. 16, no. 1, pp. 45–74, Jan. 2024, doi: 10.1007/s12559-023-10179-8.

[55] 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.

[56] A. A. R. Farea *et al.*, 'Injections Attacks Efficient and Secure Techniques Based on Bidirectional Long Short Time Memory Model', *Comput. Mater. Contin.*, vol. 76, no. 3, pp. 3605–3622, 2023, doi: 10.32604/cmc.2023.040121.

[57] W. B. Demilie and F. G. Deriba, 'Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques', *J. Big Data*, vol. 9, no. 1, p. 124, Dec. 2022, doi: 10.1186/s40537-022-00678-0.

[58] P. Nespoli, D. Díaz-López, and F. Gómez Mármol, 'Cyberprotection in IoT environments: A dynamic rule-based solution to defend smart devices', *J. Inf. Secur. Appl.*, vol. 60, p. 102878, Aug. 2021, doi: 10.1016/j.jisa.2021.102878.

[59] V. V. Meduri, K. Chowdhury, and M. Sarwat, 'Evaluation of Machine Learning Algorithms in Predicting the Next SQL Query from the Future', *ACM Trans. Database Syst.*, vol. 46, no. 1, pp. 1–46, Mar. 2021, doi: 10.1145/3442338.

[60] A. Sherstinsky, 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network', *Phys. Nonlinear Phenom.*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[61] 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.

[62] R. Huang *et al.*, 'Well performance prediction based on Long Short-Term Memory (LSTM) neural network', *J. Pet. Sci. Eng.*, vol. 208, p. 109686, Jan. 2022, doi: 10.1016/j.petrol.2021.109686.

[63] Z. Zhuo, T. Cai, X. Zhang, and F. Lv, 'Long short-term memory on abstract syntax tree for SQL injection detection', *IET Softw.*, vol. 15, no. 2, pp. 188–197, Apr. 2021, doi: 10.1049/sfw2.12018.

[64] S. Choi and M. Do, 'Development of the Road Pavement Deterioration Model Based on the Deep Learning Method', *Electronics*, vol. 9, no. 1, p. 3, Dec. 2019, doi: 10.3390/electronics9010003.

[65] D. Stiawan *et al.*, 'An Improved LSTM-PCA Ensemble Classifier for SQL Injection and XSS Attack Detection', *Comput. Syst. Sci. Eng.*, vol. 46, no. 2, pp. 1759–1774, 2023, doi: 10.32604/csse.2023.034047.

[66] P. Wen, C. He, W. Xiong, and J. Liu, 'SQL Injection Detection Technology Based on BiLSTM-Attention', in *2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, Wuhan, China: IEEE, Nov. 2021, pp. 165–170. doi: 10.1109/RCAE53607.2021.9638837.

[67] P. Wen, C. He, W. Xiong, and J. Liu, 'SQL Injection Detection Technology Based on BiLSTM-Attention', in *2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, Wuhan, China: IEEE, Nov. 2021, pp. 165–170. doi: 10.1109/RCAE53607.2021.9638837.

[68] M. Ali, B. Yin, A. Kunar, A. M. Sheikh, and H. Bilal, 'Reduction of Multiplications in Convolutional Neural Networks', in *2020 39th Chinese Control Conference (CCC)*, Shenyang, China: IEEE, Jul. 2020, pp. 7406–7411. doi: 10.23919/CCC50068.2020.9188843.

[69] A. Gondia, A. Siam, W. El-Dakhakhni, and A. H. Nassar, 'Machine Learning Algorithms for Construction Projects Delay Risk Prediction', *J. Constr. Eng. Manag.*, vol. 146, no. 1, p. 04019085, Jan. 2020, doi: 10.1061/(ASCE)CO.1943-7862.0001736.

[70] V. Nasteski, 'An overview of the supervised machine learning methods', *HORIZONS.B*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.

[71] A. H. Farhan and R. F. Hasan, in *Proceedings of Data Analytics and Management*, vol. 572, A. Khanna, Z. Polkowski, and O. Castillo, Eds., in Lecture Notes in Networks and Systems, vol. 572. , Singapore: Springer Nature Singapore, 2023, pp. 631–642. doi: 10.1007/978-981-19-7615-5_52.

[72] M. M. Ibrohim and V. Suryani, 'Classification of SQL Injection Attacks using ensemble learning SVM and Naïve Bayes', in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia: IEEE, Aug. 2023, pp. 230–236. doi: 10.1109/ICoDSA58501.2023.10277436.

[73] Y. Li and B. Zhang, 'Detection of SQL Injection Attacks Based on Improved TFIDF Algorithm', *J. Phys. Conf. Ser.*, vol. 1395, no. 1, p. 012013, Nov. 2019, doi: 10.1088/1742-6596/1395/1/012013.

[74] R. Ravindran, S. Abhishek, and A. T, 'Adaptive Payload Defense: A Cutting-Edge k-NN Framework for Web Security', in *2023 International Conference on Intelligent Computing, Communication & Convergence (ICI3C)*, Bhubaneswar, India: IEEE, Dec. 2023, pp. 486–491. doi: 10.1109/ICI3C60830.2023.00098.

[75] S. Saleem, M. Sheeraz, M. Hanif, and U. Farooq, 'Web Server Attack Detection using Machine Learning', in *2020 International Conference on Cyber Warfare and Security (ICCWS)*, Islamabad, Pakistan: IEEE, Oct. 2020, pp. 1–7. doi: 10.1109/ICCWS48432.2020.9292393.