Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

Online ISSN
3007-3138

Print ISSN
3007-312X

Data Exploration with SQL: A Machine Learning Based End-to-End Prediction and Data Security Framework for the Detection of Attack in Emerging Cloud Computing Databases and Integrated Paradigms: Analysis on Taxonomy, Challenges, and Opportunities

Muhammad Atif Imtiaz[1*]
School of Electrical, Computer and Telecommunications, Engineering, University of Wollongong, NSW 2522, Australia
Department of Electronics Engineering, University of Engineering and Technology Taxila, 47050, Pakistan. Corresponding Author: matif@uow.edu.au

Muhammad Zaryab Ahmed[2]
Department of Computer Science, Faculty of Computer Science & IT Superior, University Lahore, 54000, Pakistan
cs.zaryabahmed@gmail.com

Farwa Khalid[3]
University Of Narowal, Narowal. farvaanvar@gmail.com

Abdullah Moeen Muhammad[4]
Department of Information Technology, Bahria University, Lahore Campus, Techlogix, Pakistan. abmns10@gmail.com

Asad Iqbal[5]
Shark Innovation Labs - Al Sharqi
Dept. of Computer Sciences - National College of Business Administration and Economics (NCBA&E), Lahore.
theasadiqbal.official@gmail.com

Hira Siddique[6]
School of Mathematics and Applied Statistics, University of Wollongong, NSW 2522, Australia. hira@uow.edu.au

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

Abstract

Now-a-day the rising complexity of cyberattacks, traditional-based detection systems often fall due to short in accurately identifying the vulnerabilities. In the last few years, machine learning algorithms have played an important role in detecting SQL injection attacks due to their ability to analyze threats. This paper aims to provide a comprehensive comparative analysis of machine learning algorithms employed in SQL injection detection. In this paper we evaluate the performance across diverse datasets, and metrics to show the accuracy, precision, recall and computational efficiency are examined to detect their strengths and limitations. Additionally, this paper discusses the feature selection, model interpretability, real-time application and challenges in threat detection. These findings provide a clear understanding of the most effective machine learning approaches for enhancing database security, which provide comprehensive guidelines for future research and development. The paper analyzes recent Machine learning studies and explores advanced strategies for mitigating these threats, such as AI-driven anomaly detection, blockchain-based security models, and Zero Trust architectures. The objective is to provide a clear understanding of the risks and actionable insights into building robust, secure database systems. This study offers a comprehensive analysis aimed at helping researchers and practitioners develop effective data security measures, ensuring both resilience and adaptability in an increasingly hostile cyber environment.

Keywords: Database, Security, SQL Injection, Machine learning, Cyber risk, Open data, Systematic review, DBMS, database security threat

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

mitigation, Data protection strategies for DBMS, Cyber threats in database management

Introduction

The cloud computing paradigm is successfully converging as the fifth utility. Traditional techniques like input validation, parameterized queries and web application firewalls, while effective to the extent, struggle to adapt to complex systems to detect the threats. Machine learning provides a comprehensive approach to detect the mitigate SQL injection attacks through pattern recognition, anomaly detection and real-time analysis. SQL injection is one of the most dangerous and persistent attacks for the security of vulnerability targeting the database to manipulate queries and access the unauthorized to steal the sensitive information. This paper provides a comprehensive review of various machine learning algorithms used for SQL injection detection, evaluating their performance, computational efficiency and real-world scenarios, which provide database security and guide future research in this domain. Despite advancements in cybersecurity technologiesIncreasing the reliance on the database application across industries, therefore, needs to develop advanced methods for SQL injection detection [1, 2]. significant gaps remain in fully addressing the risks associated with DBMS. Current research often focuses on isolated threats or specific technologies, resulting in fragmented strategies [3]. Moreover, the dynamic nature of cyber threats and the lack of integrated solutions highlight the need for a holistic approach to database security. This review consolidates findings from multiple studies to bridge these gaps, providing a

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

comprehensive understanding of the challenges and potential solutions [4]. The database server is used for storing and managing the data in the finance, healthcare, e-commerce and government sectors. Now I need to ensure the security of the database which protects the stolen sensitive data [5, 6]. Pervasive threats are the most dangerous for database security SQL injection, a type of cyberattack where malicious SQL statements are inserted into fields to manipulate database queries. These types of attacks can bypass authentication, sensitive data even destroy databases which poses a severe challenge to cybersecurity professionals [7, 8].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

Table 1: Non-Technical Threats in DBMS [9]

Technical threats	Description of threats	Impact and example of threats
Human error	Human mistakes such as accidental disclosures of sensitive information, misdirected emails, and unintentional disclosure of login credentials.	Disclosures of sensitive information, misdirected emails, unintentional disclosure of login credentials
Insider threats	People who have access to sensitive data can intentionally misuse this data for malicious reasons. They are considered as threatening as the outsider threats. Insiders may be disgruntled employees, contractors and business partners. Some tactics that malicious users may use are copying files onto a Universal Serial Bus (USB) drive, emailing sensitive information to a personal account, sharing access credentials with unauthorized individuals, or even planting malware or other hacking tools to facilitate their activities. Insider threats can also be inadvertent negligence by individuals.	Internal threats from employees, lack of awareness and employee negligence
Third-party risks	Some organizations rely on third-party systems or services to manage their data, security vulnerabilities in these third-party systems, or services put enterprise data at risk.	Third-party risks

The Prevalence of SQL injection attacks is a necessity for robust detection to prevention mechanisms. The traditional approaches such as input validation and parameterized queries provide some defense, they adapt to novel and sophisticated attack patterns. These limitations for researchers and practitioners to explore advanced techniques, including the application of machine learning to detect and mitigate SQL injection attacks [10, 11].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

SQL Injection Attack Process

SQLIA is a hacking technique which the attacker adds SQL statements through a web application's input fields or hidden parameters to access to resources. Lack of input validation in web applications causes hacker to be successful. SQL injection is a highly effective and dangerous cyberattack that targets web applications and database systems by exploiting vulnerabilities in user input. This type of attack injects malicious SQL code into input fields, like login forms or search boxes, which are executed by the database server. The attacker manipulates the query, which was not intended by the original developers, which provides authorization access to sensitive data like usernames, passwords, financial records and confidential information [12].

Vol. 3 No. 2 (2025)
Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

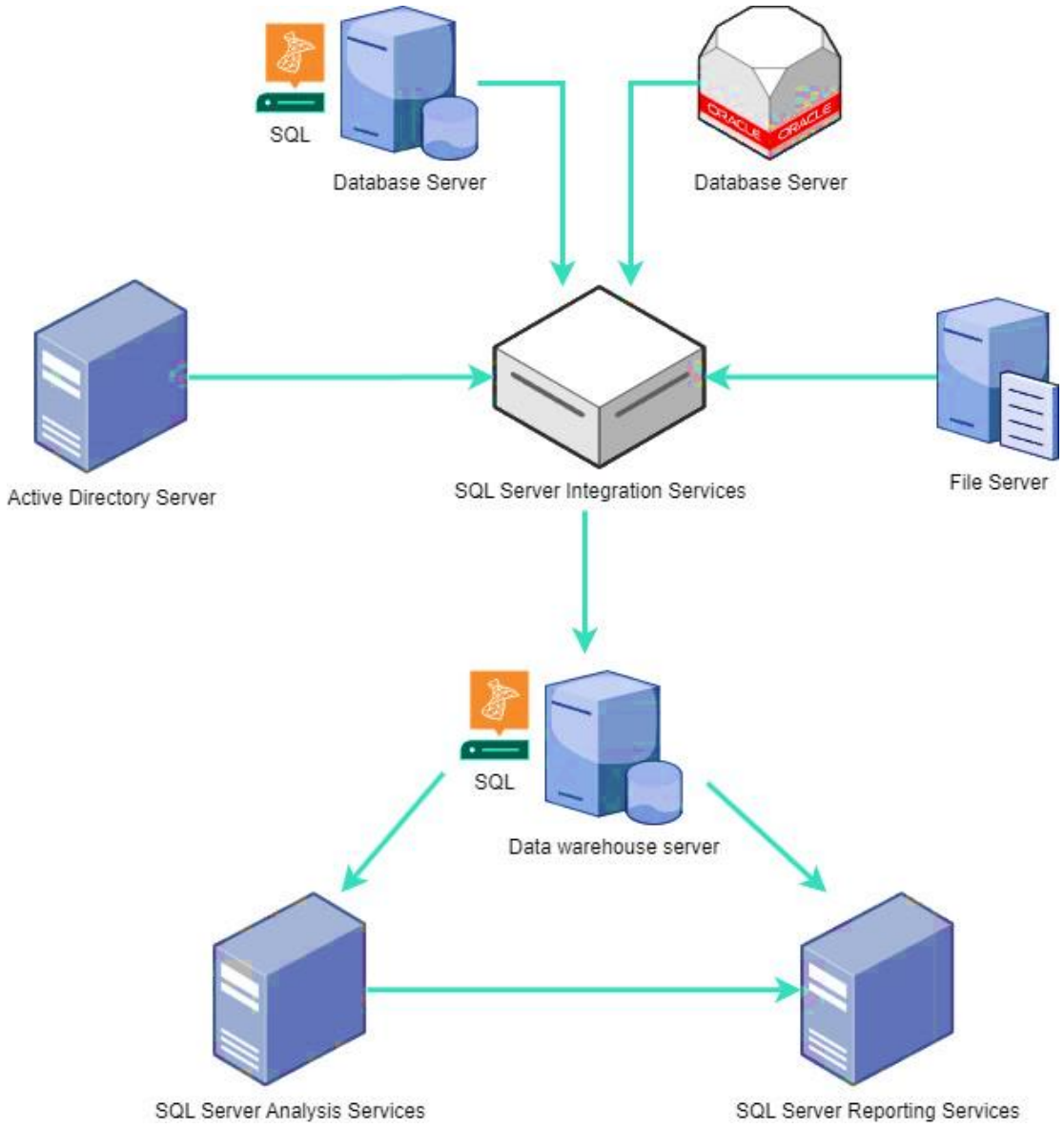

Figure 1: SQL Server Data Warehouse Architecture [13]

SQL (Structured Query Language) is a database language that is used to add, delete, modify, and query data in a relational database. As long as the system uses the database, most of it interacts with the database through SQL statements. The SQL injection attacks can bypass authentication mechanisms, which allows attackers to access

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

the system without privileges of proper authorization. The attackers can use SQL injection to delete and modify data causing severe operational disruptions or financial damage. One of the main reasons SQL injection persists is the threat, which allows attackers with limited technical expertise to exploit poorly secured applications. In advanced security methods, SQL injection plays an important role in exploiting vulnerabilities in web applications, particularly those that do not follow the best practices for input validation and query handling [14].

This makes SQL injection a concern for cybersecurity professionals tasked with protecting sensitive data and ensuring the integrity of database applications [15]. The application of databases across a wide range of industries has made secure by management practices more critical than ever. The rapidly increasing digital platforms in the finance, healthcare, e-commerce and government sectors rely on sophisticated database systems to store and manage the vast amount of sensitive data [16]. The sophistication of attacks increases and their impact on these industries has become more severe. In the financial sector, a successful SQL injection attack can lead to the unauthorized extraction of customer data like account numbers, credit card details and translation histories and other personal information. This data can be stolen on the black market, which is used for identity theft and fraud [17].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

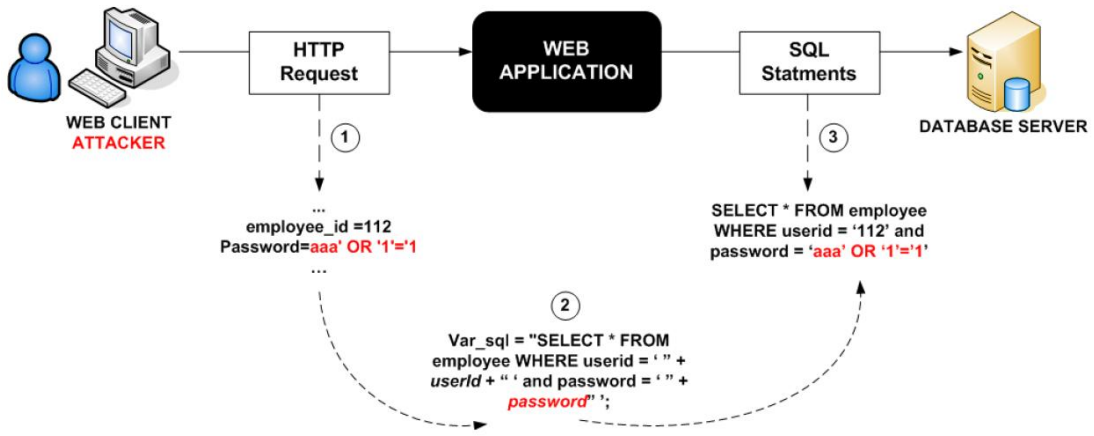

Figure 2: Shows the possible SQL Injection attacks [18]

The financial losses from breaches of data can be substantial, especially if the attacker disrupts the banking operations or online payment system. In the healthcare sector, databases contain sensitive patient information like medical histories, test results and personal identification. In SQL, the UNION operator is used to join two SQL statements or queries [19].

Union Based SQL Injection

Union SQL Injection takes advantage of this feature to make the database return desired results in addition to the intended results.

Error Based SQL Injection

Error-Based SQL Injection Error error-based SQL Injection approach works by passing an invalid input in the query and thereby triggering an error in the database.

Blind SQL Injection

Blind SQL Injection attack is a technique where the malicious user asks questions to the database and decides on further course of action based on the returned answers.

Vol. 3 No. 2 (2025)
Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

Boolean-Based SQL Injection

Boolean-Based SQL Injection Boolean-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result.

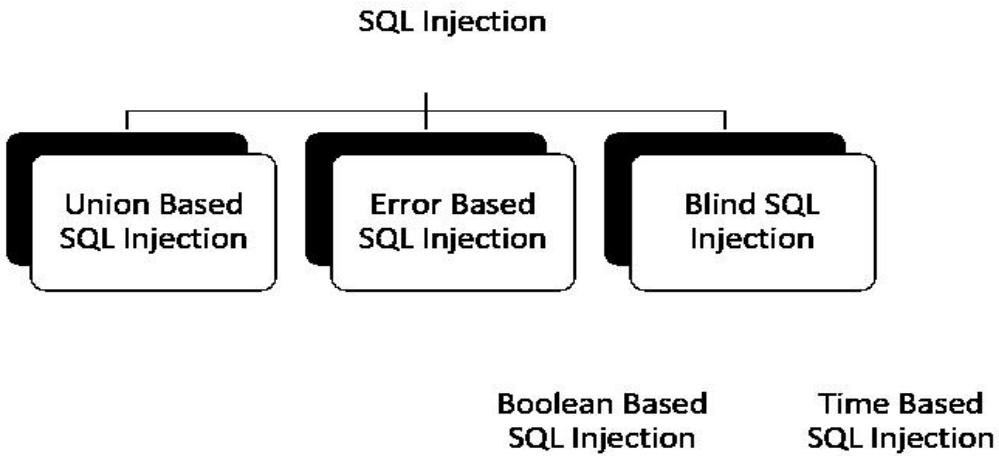

Figure 3: Various Types of SQL Injection [20]

Related Work

The SQL injection attack on the healthcare system exposes private data, which violates patient privacy laws. This is a risk for patients, which leads to costly legal penalties, lawsuits and regulatory sanctions. The e-commerce business faces the challenges of securing customer data and also the transactional information [21]. SQL injection can allow attackers to access the customer profiles, order histories and payment details, leading to direct financial theft. The SQL injection attacks can damage the integrity of product inventories and order systems, including halting sales and delivery issues. When a customer feels that their data is not secure they abandon online platforms

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

which causes the loss. The e-commerce platforms also face the risk of negative publicity and loss as a result of security breaches. In the government sector, SQL injection attacks can steal public service databases like citizens' personal information, tax records, voting data and national security data. The attackers in the government sector, lose trust in government services, which is a risk to national security and even interference with election processes [22]. Governments provide high-profile targets for cyber criminals and adopt advanced defense mechanisms to protect their databases. The objective of the study is analysis of the comprehensive review and comparison of machine learning algorithms for detecting SQL injection attacks. The main objective of this focuses on evaluating the performance of each algorithm.

The study aims to assess the computational efficiency and their practical applicability in real-world scenarios, like handling large-scale data and providing real-time threat detection [23]. We analyze the strengths and limitations of each algorithm, provide valuable insights for future researchers, and enhance data security. The study aims to contribute to the advancement of more robust and adaptive solutions for SQL injection threats. Machine learning plays an important role in cybersecurity to detect threats. This paper focuses on the importance of choosing the right features understanding how the machine learning model makes decisions and applying the real-time to detect the SQL injection attacks. This paper highlights the need to tackle challenges like unbalanced datasets, high computational demands, and finding a balance between accuracy and efficiency. By exploring

Vol. 3 No. 2 (2025)
Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

these areas, this study aims to provide useful insight into database security and create stronger detection systems for the future [24, 25].

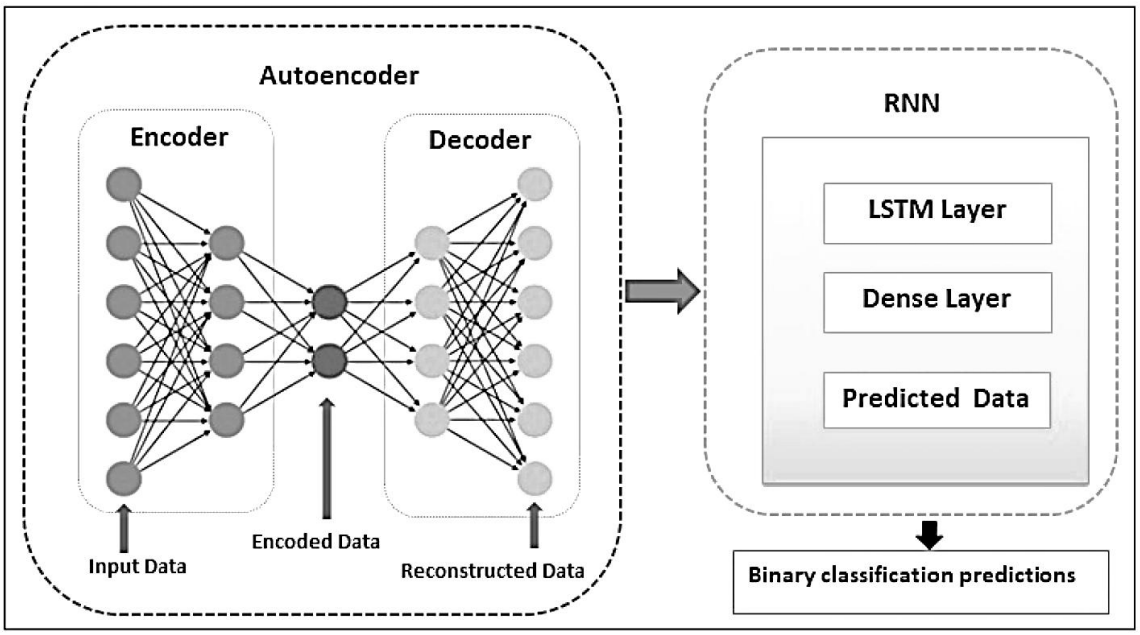

Figure 4: Generalize Deep learning based SQL injection attack detection Architecture [26]

Input validation is the technique where the user input is checked to ensure they do not contain harmful SQL code. The input validation can block some basic attacks, but it is not foolproof. The attackers can sometimes bypass the input validation by using tricks, like encoding the malicious code or using unusual syntax. The input validation does not address the root cause of the vulnerability and can be misconfigured [27]. Parameterized queries are designed to separate SQL commands from user inputs, making the harder for attackers to inject malicious SQL code. This method is more effective than the basic input validation. However, it still relies on developers' correct implementation [28]. If the developer does not use parameterized

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

queries correctly in the SQL in the application, the system remains vulnerable to attacks [29].

SQL Traditional Detection Methods

Traditional security methods are used to protect the database from SQL injection attacks, which help prevent many attacks. These include techniques like input validation, parameterized queries, and web application firewalls. These approaches have significant limitations, especially, when cyberattacks become more sophisticated [30].

Web Application Firewall

Web application firewalls are used to filter and monitor the traffic of websites or applications to block harmful data. Web application firewalls can detect harmful data and block the many types of SQL injection attacks. This relies on predefined patterns of known attacks, which makes them less effective against new or sophisticated attack techniques. Web application firewalls can introduce false positives, blocking the requests that resemble malicious ones, which can disrupt normal operations [31]. One of the biggest issues in the traditional methods is the static. These are designed to block the known attack patterns to detect the new methods of SQL injection. Cybercriminals find new ways to exploit vulnerabilities, and the traditional defenses cannot change tactics. As a result, systems protected by the methods remain at risk, and the attackers adapt and refine strategies over time [32].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

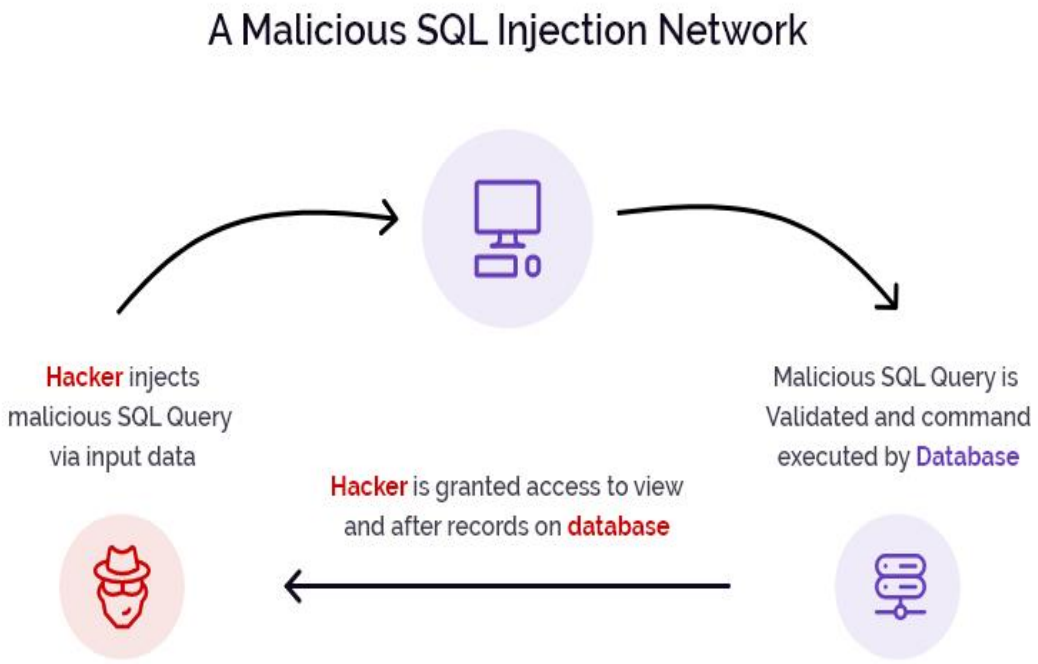

Figure 5: Technical SQL Injection threats [33]

Modern applications become complex, the traditional methods increasing difficulty in detecting sophisticated SQL injection attacks. The attackers used multiple techniques like encoding and advanced payloads to make it harder for traditional systems to identify and block them. These advanced attacks can bypass the protection methods and leave databases exposed [34].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138

Print ISSN
3007-312X

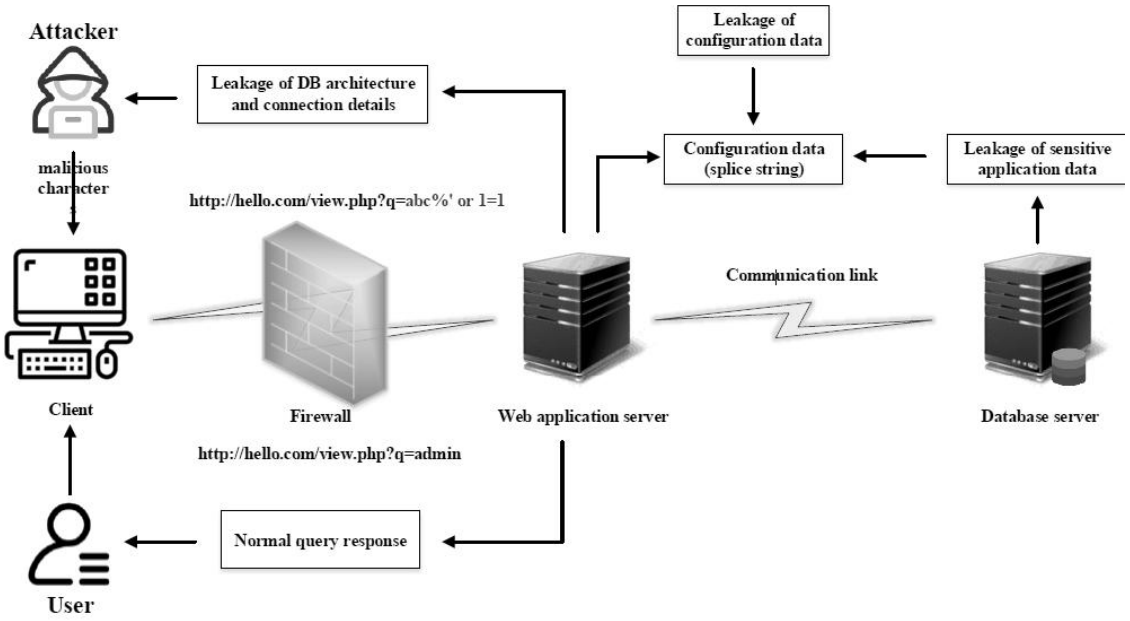

Figure 6: Tradiational SQL Injection Attack System [35]

Data Exploration with SQL Based on Machine Learning Frameworks

In the field of cybersecurity, various machine learning algorithms are used to detect and prevent SQL injection attacks. The main objective of the literature review is to analyze the machine learning algorithms performance and compare them. We analyze the various machine learning algorithms like Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Networks (ANN) and evaluate their unique strengths and limitations. SQL injection attacks have been a serious issue since the 1970s. The OWASP and CWE tools classified these vulnerabilities, which exploit the improper coding practices and generation of SQL queries. Common defenses like input validation, data stored procedures, and pattern matching have some limitations. Machine learning algorithms like Support Vector

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Machines (SVM), have achieved 94% accuracy in detecting SQL injection attacks [36]. The objective is to compare the several machine learning algorithms and techniques explored to identify and prevent SQL injection attacks. We analyze and compare machine learning algorithms like Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Artificial Neural Network (ANN), and Hybrid models of ANN and SVM. The results from our experiments indicate that the Hybrid model outperforms all other techniques in both the training and testing phases. In the training set, the Hybrid model achieved 99.54% accuracy, with a training time of 26.15 seconds. The testing results showed that it maintained a high accuracy of 99.20% and a testing time of 15.33 milliseconds. The other techniques like ANN also performed the high accuracy 99.05% in the training and 98.87 in the testing but required more time compared to the Hybrid Model. SVM, RF, and DT show lower accuracies, especially in the testing phase but still provide good results. Naive Bayes also performed the lowest in accuracy and precision but it had the fastest training and testing times. Overall, our findings suggest that the Hybrid approach, particularly the combination of ANN and SVM, provides the best balance between accuracy and processing time for detecting and preventing SQL injection attacks [37].

Vol. 3 No. 2 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138
Print ISSN
3007-312X

Table 2: Non-technical threats in Databases [38]

Countermeasures	Description
Data encryption	Encryption is the process of transforming data into a coded format to make it unreadable by intruders and difficult to decipher, whether it is during transmission or at rest. It can be applied to many data types, such as emails, files, databases, and other communication channels. Encryption can also prevent insider threats, as insiders who have access to the data will not be able to read it unless they have the required authorization. In addition, it helps organizations to ensure their confidentiality, integrity, and availability by adhering to several data protection regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Ensures the security of users by converting the data with the AES algorithm to the database management system, Message Digest (MD5), and Secure Hash Algorithm (SHA-256) to protect network data transmission. It is cost-effective; investing in the implementation of encryption technology is cheaper than dealing with the consequences of data breaches.
Access control	All DBMS use access control to create user accounts and passwords to prevent unauthorized people from entering the database system and obtaining confidential information. Granting and revoking privileges are methods of enforcing access control. The organization must set policies defined by access control that all contact with the databases must adhere to. It is suggested that web tripwire and login rituals be integrated using Multi-Factor Authentication (MFA). Access control allows organizations to do the following: - Access control allows organizations to implement a layered defense approach to security. - Helps organizations follow protection data regulations. - Prevents insider threats. - Allows organizations to detect and respond to security incidents. Access control systems consist of: - File permissions to create, read, edit, or delete files on the server. - Program permissions are the rights of executing an application program on the server. - Data rights, the rights of retrieving, or updating data in a database. Access control mechanisms: 1. Discretionary Access Control (DAC) 2. Mandatory Access Control (MAC) 3. Role-Based Access Control (RBAC)

In this study, we classify the machine learning algorithms like Support Vector Machine (SVM), Neural Networks (ANN), and Ensemble methods Like Boosted and Bagged Trees have shown good results. The Ensemble Boosted and Bagged Trees achieved 99% accuracy for

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

detecting the SQL injection statements with the overall system accuracy 93.8%. Their classifiers are effective in minimizing errors and ensuring reliable detection of attacks [39, 40]. The result of the four classifiers ANN, Random Forest, Gradient Boosting, and SVM to detect the SQL injection attacks, the ANN achieved the high results 96% accuracy and fastest performance, the other Random Forest, Gradient Boosting, and SVM achieved 93%, 91% and 94% accuracy in the SQL injection attacks [41, 42]. Use machine learning algorithms to identify and respond to unusual database activity in real time, reducing the risk of undetected breaches. Zero Trust Architectures: Implement strict access controls based on the principle of "never trust, always verify," ensuring that no user or device is inherently trusted [43, 44].

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Table 5: Comparative Analysis of Prominent Research Areas Using Databases [45]

Focus Area	Methodology	Findings	Limitations
Big Data and RDBMS Integration	Review of big data integration strategies	Emphasized the need for hybrid database systems integrating relational and non-relational models; highlighted Oracle's Big Data SQL as a robust solution	Lack of practical implementation examples
compares the performance and features of two popular database management systems.	The main differences and features of Microsoft SQL Server and Oracle. Comparing both systems' security and vulnerabilities. Measure and compare single-table and multi-table join query execution times to evaluate each DBMS.	Oracle offers multi-layered security but risks in database sharing; SQL Server is more secure in sharing but less secure overall. SQL Server has better query execution times.	Only Microsoft SQL Server and Oracle are compared in the study. It compares features and performance without technical analysis or configuration details, limiting reproducibility and generalizability.
compares the performance and features of Relational and NoSQL database	The main differences and features of MySQL, PostgreSQL and Microsoft SQL. Comparing both systems' to measure execution times for selecting, updating, and inserting data, scripts were used for benchmarking	This study utilized scripts to measure the execution times of select, update, and insert queries on MySQL, PostgreSQL, and Microsoft SQL Server using datasets of varying sizes (100, 1,000, and 10,000 rows)	Only Microsoft SQL Server and MySQL are compared in the study and residual caching effects, the simplicity of the queries analysed, a dataset very small.

Performance Evaluation of Machine Learning based end-to-end Prediction and Data Security Framework

The dataset of this study comprises 20,000 SQL queries, which are equally divided into benign and malicious queries. The features were extracted on the based query structure, keywords, special characters, and entropy. The preprocessing steps included normalization, handling missing values, and encoding the various categorical features and techniques like TF-IDF and one hot encoding. The balanced and enriched dataset ensures the reliable evaluation of machine learning models for SQL injection detection. In this study, we implemented and optimized the various models of machine learning for detecting SQL injection attacks. Naive Bayes is the fast machine learning model which is based on Bayes' theorem. Which predicts the

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

probability of a query belonging to a certain class like malicious or normal, by looking at the various features of data. It works well when features are independent of each other.

$$P(c|x) = \frac{P(X|C).P(C)}{P(X)} \qquad \text{Eq (1)}$$

P (C|X): The probability of which query belongs to the class C malicious.

P (X|C): This is the likelihood the data of X is given to class C.

P (C): This prior probability of the class C is a common class.

P (X): This is the total probability of the data X.

This model is used for baseline because it works fast and is easy to implement. It works well with simple and structured data. A decision tree is a model that splits data based on maximum information gain. Pruning techniques were applied to reduce the overfitting.

$$Gini\ (t) = 1 - \sum_{i-1}^{k} p_{\frac{2}{i}} \qquad \text{Eq (2)}$$

t. This is a specific node in the decision tree.

k. The classes of malicious queries in the SQL injection detection.

p_i. The proportion of the elements belonging to class *I* in the node *T.*

We optimized the support vector machine (SVM) with a Radial Basis function kernel for non-linear classification. Hyperparameters C regularization parameter and γ kernel coefficient were fine-tuned using the grid search strategy to achieve the optimal performance.

The SVM decision function:

$$f(x) = w^T x + \quad b \qquad \text{Eq (3)}$$

W is the weight of the vector.

X represents the feature of a vector as an input sample.

b is the bias term.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

An ensemble model combining 1,000 decision trees with each tree trained on the bootstrapped samples. The feature important analysis was conducted to optimize the feature selection. A deep neural network with hidden layers, each containing 256 neurons. The dropout and batch normalization were used to prevent overfitting and accelerate convergence.

The activation function:

$$f(x) = max(0, x) \ (ReLU) \qquad \text{Eq (4)}$$

The stacked ensemble combines ANN's nonlinear learning capacity and SVM's decision boundaries. The ANN outputs are fed into the SVM classifier to refine prediction. Hyperparameter tuning was performed for both components.

Euatvalion Metrics based on Machine Learning Based Techniques

The accuracy measures the proportion of the correctly classified instances both true positives and true negatives out of all instances.

The accuracy is defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{Eq (5)}$$

TP: True positives which malicious queries are correctly classified as malicious.

TN: True Negative which benign queries correctly classified as benign.

FP: False positive which benign queries incorrectly classified as malicious.

FN: False Negatives which malicious queries incorrectly classified as benign.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Precision

The precision calculates how many predicted positive instances were positive.

$$Precision = \frac{TP}{TP+FP} \quad \text{Eq (6)}$$

Recall (Sensitivity or True Positive Rate)

Recall measures the model's ability to identify the actual positive instances.

$$Recall = \frac{TP}{TP+FN} \quad \text{Eq (7)}$$

F1-Score

The F1 Score is the harmonic mean of the Precision and Recall, Which provides a single metric to balance both.

$$F1 - Score = 2 . \frac{Precision . Recall}{Precision+Recall} \quad \text{Eq (8)}$$

The duration is required for the model to learn from the training dataset while The time taken to make the predictions on the testing dataset, is critical for real-time applications.

Table 6: Security Countermeasures Effectiveness

Countermeasure	Description	Effectiveness Rating
Machine Learning-Based Detection	Using machine learning algorithms to detect anomalous behaviors indicating attacks.	85%
Blockchain-Based Authentication	Decentralized, tamper-proof authentication to secure sessions and credentials.	75%
Web Application	Filtering and monitoring HTTP	80%

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Firewalls (WAFs)	requests to detect and block malicious activity.	
Secure Coding Practices	Implementing code practices like input sanitization and parameterized queries.	90%
Multi-Factor Authentication (MFA)	Enhancing user authentication by requiring multiple verification factors.	95%

Experimental Setup & Results

The dataset is divided into two subsets with 80% allocated for training and 20% reserved for testing. To ensure robust evaluation and minimize the risk of overfitting, a 10-fold cross-validation technique was employed. These techniques involved partitioning the data set into equal segments, where the nine segments were used for the training and the remaining one for testing. This process was repeated ten times, and each segment served as the testing set once. The performance metrics across all the folds provided a reliability of the model's effectiveness and generalization capability. This experiment focuses on assessing the effectiveness of various machine learning models for detecting and preventing SQL injection attacks. Each algorithm evaluated on the performance metrics, including the training accuracy and testing accuracy, training time and testing time. These results were derived using a comprehensive dataset containing SQL injection and benign queries with 10- fold cross-validation to ensure robustness. This section presents a detailed comparison of the various algorithms and highlights their strengths and limitations in

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

achieving accurate and efficient attacks detection. The Naive Bayes classifier demonstrated the well performance, achieving the training accuracy of 93.45% and in testing accuracy of 92.67%. It was the fastest model to train with training time just 1.25 seconds, and it required only 0.89 milliseconds for testing, which makes it an ideal choice for scenarios that prioritize speed over accuracy. However, the lower accuracy compared to the other models indicates it may not be the best option for the applications for detecting SQL injection attacks.

The Decision Tree model performed better than Naive Bayes, achieving a training accuracy of 97.02 and the testing 96.45% accuracy. While it required a moderate training time of 4.83 seconds and the testing time of 2.12 milliseconds, the Decision Tree model offered a good balance between accuracy and the efficiency of computational. The ability to provide interpretable results makes it a viable choice for understanding the decision- making process in the SQL injection detection system. The Random Forest classifier outperformed the Decision Tree, an ensemble of multiple decision trees, achieving a training accuracy of 98.92% and a testing accuracy of 98.34%. However, the improvement in the accuracy came at the cost of increased computational time with a training time of 15.56 seconds and a testing time is 5.33 milliseconds. The Random forest models showed robust performance and proved to be a reliable choice for SQL injection detection attacks. The support vector machine with the optimized radial basis function kernel exhibited excellent results, achieving a training accuracy of 99.12% and a

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

testing accuracy of 98.95%. The training time for this model was 9.67 seconds with a testing time of 3.01 milliseconds. The SVM model is effectively balanced with the high accuracy and efficiency of computation, which makes it a strong contender for real-time SQL injection detection attacks.

The Artificial Neural Network delivered a near-perfect performance with a training accuracy of 99.78% and a testing accuracy of 99.55%. The training time was the longest among the models at 35.25 seconds. The low testing time of 8.45 milliseconds. The ANN's ability is superior to learn complex patterns, makes it a powerful tool for SQL injection detection.

The Hybrid Model, combining the strengths of ANN and SVM, achieved the best overall results of the study with a training accuracy of 99.93% and a testing accuracy of 99.85%. It shows better performance than the other models. The training time of 28.67 seconds and testing time of 7.12 milliseconds demonstrated a balance between computational efficiency and accuracy. The Hybrid Models' outstanding performance makes it most suitable for detecting SQL injection attacks.

Conclusion

This study evaluated the various machine learning algorithms for detecting SQL injection attacks like Naive Bayes, Decision Tree, Random Forest, SVM, ANN, and the Hybrid Model ANN and SVM. The Hybrid Model achieved the highest training accuracy 99.3% and 99.85% testing with efficient processing times, which makes it the most effective. The ANN and SVM also delivered excellent

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

performance, while Random Forest and Decision Tree provided robustness with lower accuracy. Naive Bayes, though fastest, has the lowest accuracy and the best for speed-critical applications. The Hybrid Model is recommended for real-time SQL injection detection and future work could explore these models expanding datasets to enhance robustness. This article also determined that awareness, knowledge, and behavior are important as cyber threats cause security issues. Some users take fitting action by pursuing cyber threat knowledge while others freely share cyber threat information and experiences also highlighting the dynamic and evolving nature of cybersecurity risks targeting databases. Key threats such as SQL injection, ransomware, insider misuse, and denial-of-service attacks pose significant challenges to organizations.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Choi J , Kim H , Choi C , et al. Efficient Malicious Code Detection Using N-Gram Analysis and SVM[C]// International Conference on Network-based Information Systems. IEEE, 2011

[2] Li X, Qu J, Yao G, Chen J, Shen X. Design and Implementation of an Automatic Scanning Tool of SQL Injection Vulnerability Based on Web Crawler [C]//. International Conference on Security with Intelligent Computing and Big Data Services, 2018, pp. 481-488.

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

[3] Zheng L, Yuan L, Peng X, Zhu G, et al. Research and Implementation of Web Application System Vulnerability Location Technology[C]//. International Conference on Advances in Intelligent Systems and Computing, 2010, pp.937-944.

[4] Akcay S , Kundegorski M E , Willcocks C G , et al. Using Deep Convolutional Neural Network Architectures for Object Classification and Detection within X-ray Baggage Security Imagery[J]. IEEE Transactions on Information Forensics and Security, 2018, 28(9):2203-2215.

[5] Vinayakumar R , , Alazab M , Kp S , et al. Deep Learning Approach for Intelligent Intrusion Detection System[J]. IEEE Access, 2019, 41525-41550.

[6] Zhou T, Sun X, Xia X, Li B, Chen X. Improving defect prediction with deep forest[J]. Information and Software Technology, vol.114, pp.204-216, 2019

[7] Li Q, Wang F, Wang J, Li W, LSTM-Based SQL Injection Detection Method for Intelligent Transportation System[J]. IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4182-4191, 2019

[8] Fang Y, Peng J, Liu L, WOVSQLI: Detection of SQL Injection Behaviors Using Word Vector and LSTM, ICCSP 2018, 170-174

[9] B. Pejo and N. Kapui, "SQLi Detection with ML: A data-source perspective," Apr. 24, 2023, *arXiv*: arXiv:2304.12115. doi: 10.48550/arXiv.2304.12115.

[10] A. A. Ashlam, A. Badii, and F. Stahl, "A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks," in *2022 5th*

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

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.

[11] F. K. Alarfaj and N. A. Khan, "Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks," *Appl. Sci.*, vol. 13, no. 7, p. 4365, Mar. 2023, doi: 10.3390/app13074365.

[12] D. Lu, J. Fei, and L. Liu, "A Semantic Learning-Based SQL Injection Attack Detection Technology," *Electronics*, vol. 12, no. 6, p. 1344, Mar. 2023, doi: 10.3390/electronics12061344.

[13] M. Lodeiro-Santiago, C. Caballero-Gil, and P. Caballero-Gil, "Collaborative SQL-injections detection system with machine learning," Sep. 14, 2022, *arXiv*: arXiv:2209.06553. doi: 10.48550/arXiv.2209.06553.

[14] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review," *J. Cybersecurity Priv.*, vol. 2, no. 4, pp. 764–777, Sep. 2022, doi: 10.3390/jcp2040039.

[15] C. Añasco Loor, K. Morocho, and M. Hallo, "Using Data Mining Techniques for the Detection of SQL Injection Attacks on Database Systems," *Rev. Politécnica*, vol. 51, no. 2, pp. 19–28, May 2023, doi: 10.33333/rp.vol51n2.02.

[16] R. A. Dalimunthe and S. Sahren, "Intrusion detection system and modsecurity for handling sql injection attacks," 2020.

[17] 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.

[18] H. Khan, M. U. Hashmi, Z. Khan, R. Ahmad, A. Saleem,

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

"Performance Evaluation for Secure DES-Algorithm Based Authentication & Counter Measures for Internet Mobile Host Protocol", IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 12, pp. 181-185, July. 2018

[19] Khan, K. Janjua, A. Sikandar, M. W. Qazi, Z. Hameed, "An Efficient Scheduling based cloud computing technique using virtual Machine Resource Allocation for efficient resource utilization of Servers", In 2020 International Conference on Engineering and Emerging Technologies (ICEET), IEEE., pp. 1-7, Apr. 2020

[20] Hassan, H. Khan, I. Uddin, A. Sajid, "Optimal Emerging trends of Deep Learning Technique for Detection based on Convolutional Neural Network", Bulletin of Business and Economics (BBE)., vol. 12, no. 4, pp. 264-273, Nov. 2023

[21] S. K. M. et al., "Privacy-Preserving in Blockchain-based Federated Learning Systems," pp. 1–44, 2024, [Online].Available: http://arxiv.org/abs/2401.03552.

[22] J. Zhang, H. Zhu, F. Wang, J. Zhao, Q. Xu, and H. Li, "Security and Privacy Threats to Federated Learning: Issues,Methods, and Challenges," Secur. Commun. Networks, vol. 2022, 2022, doi: 10.1155/2022/2886795.

[23] W. Si and C. Liu, "Privacy Preservation Learning with Deep Cooperative Method for Multimedia Data Analysis," Secur.Commun. Networks, vol. 2022, no. Iid, 2022, doi: 10.1155/2022/8449987.

[24] Q. Yang et al., "Federated Learning with Privacy-preserving and Model IP-right-protection," Mach. Intell. Res., vol. 20,no. 1, pp. 19–37, 2023, doi: 10.1007/s11633-022-1343-2.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

[25] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learningin medical imaging," Nat. Mach. Intell., vol. 2, no. 6, pp. 305–311, 2020, doi: 10.1038/s42256-020-0186

[26] Fakhar, M. H., Baig, M. Z., Ali, A., Rana, M. T. A., Khan, H., Afzal, W., ... & Albouq, S. (2024). A Deep Learning-based Architecture for Diabetes Detection, Prediction, and Classification. Engineering, Technology & Applied Science Research, 14(5), 17501-17506.

[27] R. Gosselin, L. Vieu, F. Loukil, and A. Benoit, "Privacy and Security in Federated Learning: A Survey," Appl. Sci., vol.12, no. 19, pp. 1–15, 2022, doi: 10.3390/app1219990

[28] F. Houlong, C. Guo, C. Jiang, Y. Ping, and X. Lv, "SDSIOT: An SQL Injection Attack Detection and Stage Identification Method Based on Outbound Traffic".

[29] V. Abdullayev and Dr. A. S. Chauhan, "SQL Injection Attack: Quick View," Mesopotamian J. Cyber Secur., pp. 30–34, Feb. 2023, doi: 10.58496/MJCS/2023/006

[30] Mitropoulos D , Louridas P , Polychronakis M , et al. Defending Against Web Application Attacks: Approaches, Challenges and Implications[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(2):188-203.

[31] Pollack E . Protecting Against SQL Injection: Applications, Performance, and Security in Microsoft SQL Server[M]// Dynamic SQL. 2019.

Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

[32] Maraj A , Rogova E , Jakupi G , et al. Testing Techniques and Analysis of SQL Injection Attacks[C]// International Conference on Knowledge Engineering and Applications (ICKEA) 2017. IEEE, 2017.

[33] Sarhan A A , Farhan S A , Al-Harby F M . Understanding and Discovering SQL Injection Vulnerabilities[J]. 2017.

[34] Das D , Sharma U , Bhattacharyya D K . Defeating SQL injection attack in authentication security: an experimental study[J]. International Journal of Information Security, 2017.

[35] Huang H C , Zhang Z K , Cheng H W , et al. Web Application Security: Threats, Countermeasures, and Pitfalls[J]. Computer, 2017, 50(6):81-85.

[36] Shahriar H , North S , Chen W C . Client-side detection of SQL injection attack[J]. Lecture Notes in Business Information Processing, 2013, 148:512-517.

[37] Mcwhirter P R , Kifayat K , Shi Q , et al. SQL Injection Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence Kernel[J]. Journal of Information Security and Applications, 2018, 40:199-216.

[38] Wang Y, Wang D , Zhao W , et al. Detecting SQL Vulnerability Attack Based on the Dynamic and Static Analysis Technology[C]// IEEE Computer Software & Applications Conference. IEEE Computer Society, 2015, pp. 604-607

[39] J. S K B, P. Anaswara P . Vulnerability detection and prevention of SQL injection[J]. International Journal of Engineering and Technology, 2018, 7(2.31).

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

[40] Gould C, Su Z, Devanbu P . Static Checking of Dynamically Generated Queries in Database Applications[C]// Proceedings of the 26th International Conference on Software Engineering. IEEE Computer Society, 2004.

[41] Naderi-Afooshteh A , Nguyen-Tuong A , Bagheri-Marzijarani M , et al. Joza: Hybrid Taint Inference for Defeating Web Application SQL Injection Attacks[C]// DSN 2015. IEEE, 2015.

[42] Yi W , Zhoujun L , Guo A . Literal Tainting Method for Preventing Code Injection Attack in Web Application[J]. Journal of Computer Research & Development, 2012, 49(11):2414-2423.

[43] Li L , Qi J , Liu N , et al. Static-Based Test Case Dynamic Generation for SQLIVs Detection[C]// International Conference on Broadband & Wireless Computing. IEEE, 2015, pp.173-177.

[44] Appiah B, Opoku-Mensah E, Qin Z. SQL Injection Attack Detection Using Fingerprints and Pattern Matching Technique[C]// International Conference on Software Engineering and Service Science. IEEE, 2017, pp. 583-587.

[45] Shar L K, Tan H B K. Defeating SQL Injection[J]. Computer, 2013, 46(3):69-77.