**REINFORCING WEB APPLICATION SECURITY: A MODIFIED SCHEME AGAINST SQL INJECTION ATTACKS**

P. Kamakshi Thai 1, Abid Kovuru 2, Som Narendar3, Siddhant Bolenwar **4**

1 Assistant Professor, Department of CSE (AI&ML) ACE Engineering College, Hyderabad, India.

2,3,4 Students of Department of CSE (AI&ML) ACE Engineering College, Hyderabad, India.

# Abstract

SQL injection attacks pose a serious threat to databases as they exploit vulnerabilities in the database layer by injecting SQL codes into user databases. The consequences of a successful attack cause unauthorized access to databases and attackers can gain access to sensitive data. So, to avoid the data breach and unauthorized access we have to detect whether the executed SQL code at the user side is malicious or not. Even though some methods like parameterized queries, escaping characters and input validation are some traditional techniques to detect SQL injection, they have their own limitations. These methods often rely on manual coding practices and may not identify new attacks. As attackers continuously evolve their techniques to attack and gain access to sensitive data there is a need for advanced solutions that can proactively identify and mitigate SQL injection attacks. AI has the capacity to analyze vast amounts of data, detect patterns, and learn from previous attacks. AI brings significant benefits to the prediction of SQL injection attacks. Its ability to detect anomalies, learn from new attack patterns, recognize complex patterns, reduce false positives, provide real-time protection, and scale to handle large applications makes it an indispensable tool. Here we use count vectorizer to create tokens and give these tokens to a Neural Network Algorithm i.e., Multilayer Perceptron to detect the malicious SQL code. By leveraging artificial Intelligence, we can detect and mitigate malicious SQL codes swiftly and accurately to ensure the safety of databases.

 **KEYWORDS:** Artificial Intelligence, Natural Language

 Processing, Prediction, Malicious SQL codes, Machine

 Learning, Threat Detection, SQL Injection, Neural

 Networks, count vectorizer, Multilayer Perceptron.

# Introduction

SQL Injection is a type of cyber-attack that has been around for a long time. It involves injecting malicious SQL code into an application's input fields, which allows attackers to gain unauthorized access to the application's database. This can lead to severe consequences, such as data breaches and system compromises. In recent years, Artificial Intelligence (AI) and machine learning have become popular in various fields, including cybersecurity. The idea of using AI to predict SQL Injection attacks emerged to bolster security measures and counter sophisticated attack techniques. By developing AI models that can analyze application input data, we can identify patterns that indicate the presence of an SQL Injection attack. The traditional methods used to prevent SQL Injection attacks rely on simple rule-based approaches or static pattern matching. However, these methods can sometimes be bypassed by well-crafted attacks. This is where AIbased prediction of SQL Injection attacks becomes essential. We need AI-based prediction because cyber attackers continuously evolve their methods, making it challenging to rely solely on traditional approaches.

AI-powered systems can process large amounts of data, discover hidden patterns, and adapt to new attack techniques, making them more effective in identifying SQL Injection attacks. The significance of AI-based prediction lies in its ability to enhance detection accuracy. AI models can learn from historical attack data and identify even subtle patterns that might go unnoticed by traditional methods. By doing so, they can reduce false positives, which helps minimize disruptions to legitimate user activities. Additionally, AI can serve as a proactive defense mechanism, continuously monitoring and protecting applications from potential threats, including novel and previously unseen SQL Injection attacks. Artificial Intelligence, particularly machine learning, has shown promise in various cybersecurity applications due to its ability to analyze vast amounts of data, detect patterns, and make predictions. Overall, the integration of AI-enabled NLP for predicting malicious SQL code represents a significant advancement in cybersecurity, offering organizations a powerful tool to defend against SQL injection attacks and other database-related threats.

# Literature survey

**Poornima, et al. [1] Proposed that Artificial intelligence (AI) proves its significance in predicting and combating SQL injection attacks.** AI has the capacity to analyse vast amounts of data, detect patterns, and learn from previous attacks, making it an invaluable tool in this context. AI brings significant benefits to the prediction of SQL injection attacks. Its ability to detect anomalies, learn from new attack patterns, recognize complex patterns, reduce false positives, provide real-time protection, and scale to handle large applications makes it an indispensable tool.

**Rattrout, et al. [2] Proposed that DSQLIA model, which enhances the accuracy of machine learning (ML) algorithms in detecting SQL injection attacks.** Traditional rule-based and signature-based methods face limitations in effectively identifying evolving attack techniques. To address this, DSQLIA employs feature engineering and Natural Language Processing (NLP). Relevant features capturing the unique characteristics of SQL injection attacks are identified and created through feature engineering. NLP techniques are applied to analyse the textual content of SQL queries and extract meaningful information for distinguishing between legitimate and malicious queries.

**Peng, et al. [3] First demonstrated that NLP models can be exploited as attack vectors in the wild.** In addition, experiments using four open-source language models verified that straightforward backdoor attacks on Textto-SQL systems achieve a 100% success rate without affecting their performance.

**Jaradat, et al.[4] Proposed that DSQLIA model, which enhances the accuracy of machine learning (ML) algorithms in detecting SQL injection attacks.** Traditional rule-based and signature-based methods face limitations in effectively identifying evolving attack techniques. To address this, DSQLIA employs feature engineering and Natural Language Processing (NLP). Relevant features capturing the unique characteristics of SQL injection attacks are identified and created through feature engineering. NLP techniques are applied to analyse the textual content of SQL queries and extract meaningful information for distinguishing between legitimate and malicious queries. The DSQLIA model evaluates different ML algorithms, including decision trees, support vector machines (SVM), and artificial neural networks (ANN), on a dataset.

**Zhang, et al. [5] explored two specific injection attacks, namely Boolean-based injection and Union-based injection, which use different types of triggers to achieve distinct goals in compromising the parser.** These experimental results demonstrate that both medium-sized models based on fine-tuning and LLM-based parsers using prompting techniques are vulnerable to this type of attack, with attack success rates as high as 99% and 89%, respectively.

**Sharma, et al. [6] Proposed that SQL injection is one of the most dangerous security assaults, causing damage to a company's reputation, financial losses, and the privacy of its clients.** Various classification algorithms can be used to determine whether a particular code is malicious or plain. Some of the neural network and machine learning algorithms are Naive Bayes classifier, LSTM, MLP, and SVM which can be used for the detection of SQL Injection attacks. We compared various algorithms on a common dataset in this study to see how well they performed.

# 3. Proposed Methodology

The proposed system employs a machine learning-based approach to detect SQL injection attacks by analyzing SQL queries for malicious patterns. Initially, a dataset consisting of both benign and malicious SQL queries is collected and preprocessed. The preprocessing step involves cleaning the data and converting the SQL queries into a suitable numerical format using a Count Vectorizer, which transforms textual data into a matrix of token counts. These vectorized queries are then fed into a Multilayer Perceptron (MLP), a type of feedforward artificial neural network, which is trained to classify queries as either normal or malicious. The MLP learns to recognize complex patterns and anomalies within the queries based on the features extracted during vectorization. During the training phase, the model learns from historical attack data, enabling it to generalize and detect previously unseen injection attempts. After training, the model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness. This AI-driven methodology enables real-time, scalable, and adaptive SQL injection detection, offering a proactive solution to safeguard databases from evolving threats.

## 3.1 Advantages of Proposed Methodolgy

## The proposed methodology offers several key advantages.

## By leveraging a Multilayer Perceptron and Count Vectorizer,

##  it provides high accuracy in detecting SQL injection attacks through automated pattern recognition. Unlike traditional techniques, it adapts to new attack methods, reduces false positives, and enables real-time detection. Its scalability makes it suitable for large-scale applications, while its data-driven nature ensures continuous improvement with minimal human intervention.

##  The proposed methodology offers several advantages by combining Count Vectorizer and a Multilayer Perceptron for SQL injection detection. It ensures high accuracy through automated learning of complex patterns and adapts effectively to new and evolving attack techniques. Unlike traditional methods, it reduces false positives, supports real-time monitoring, and scales efficiently for large applications. The model requires minimal manual coding, enabling faster deployment and consistent performance. Additionally, its ability to learn from historical data enhances its detection capability over time, making it a robust and intelligent solution for database security.

##  The model is capable of self-improvement through retraining on new data, enhancing long-term effectiveness. Additionally, it provides consistent and fast threat detection, integrates easily with existing systems, and ensures proactive protection of sensitive data, making it a reliable and intelligent solution for modern database security challenges.

#  4. Architecture



 **5. Algorithm**

**Step 1:** SQLi Dataset The project begins by uploading a SQL Injection (SQLi) dataset that contains SQL statements labeled as either "attack" or "no attack." This dataset is read using pandas, and basic information such as the dataset structure and a preview of the first few rows is displayed.

**Step 2:** Text Preprocessing In this step, data cleaning and preparation are performed. The SQL statements are tokenized, stop words are removed, and other text cleaning steps like stemming or lemmatization might be applied. The dataset is then split into training and testing sets for model training and evaluation.

**Step 3:** Vectorization Here, the SQL statements are transformed into numerical form using the CountVectorizer, which converts the text into token counts, creating feature vectors that represent the sentences. This is necessary for machine learning algorithms to process the textual data.

**Step 4:** Logistic Regression Classifier The Logistic Regression classifier is used as an existing method to detect SQL injection attacks. The training data is used to fit the model, and the test data is used to evaluate its performance. Accuracy and a confusion matrix are generated to understand the classification results.

**Step 5:** Simple Neural Network (Proposed Algorithm) A custom neural network (Multilayer Perceptron) is built with layers designed for SQL injection detection. The network is trained on the tokenized and vectorized SQL statements, and its accuracy is evaluated on the test data. This serves as a proposed improvement over traditional methods.

**Step 6:** Performance Comparison The performance of both the Logistic Regression classifier and the Simple Neural Network is compared. Accuracy scores and confusion matrices are analyzed to determine which model performs better in detecting malicious SQL code.

**Step 7:** Prediction with Trained Model The trained Simple Neural Network model is used to predict whether new SQL statements are attacks or not. The predictions are displayed, classifying each test data sample as either "attack detected" or "no attack detected."

**6. Conclusion**

This study demonstrates an effective AI-driven methodology for the detection of SQL injection attacks by leveraging Count Vectorization for feature extraction and a Multilayer Perceptron (MLP) for classification. The proposed approach addresses the significant limitations of traditional SQL injection prevention techniques, offering enhanced adaptability, high detection accuracy, real-time monitoring capabilities, and scalability for large-scale applications. By learning from historical and evolving attack patterns, the model ensures proactive protection against sophisticated threats while minimizing false positives. Experimental results validate the model’s potential in improving database security through automated, intelligent analysis of SQL queries. Moving forward, integrating advanced deep learning architectures and expanding the dataset with real-world SQL injection scenarios could further enhance detection capabilities. Overall, this research highlights the promising role of artificial intelligence in strengthening cybersecurity frameworks and safeguarding sensitive data against emerging threats.

**7. References**

[1] Poornima, A., Kunusoth Sai Pavan, Shiva Venkata Sai Kumar, Suthari Rithika, and Gadila Vishal. "Artificial Intelligence for Prediction of SQL Injection Attack."

[2] Rattrout, Amjad, Majdi Jaradat, and Rashid Jayousi. "Machine Learning Advancements in SQL Injection Detection: NLP and Feature Engineering Strategies." (2023).

[3] Peng, Xutan, Yipeng Zhang, Jingfeng Yang, and Mark Stevenson. "On the Vulnerabilities of Text-toSQL Models." In 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), pp. 1-12. IEEE, 2023.

[4] Jaradat, Majdi, Amjad Rattrout, and Rashid Jayousi. "Improving ML Accuracy in SQL Injection Detection using NLP and Feature Engineering." (2023).

[5] Zhang, Jinchuan, Yan Zhou, Binyuan Hui, Yaxin Liu, Ziming Li, and Songlin Hu. "Trojansql: Sql injection against natural language interface to database." In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 4344-4359. 2023.

[6] Sharma, Vishal, and Sachin Kumar. "Comparative Study of Machine Learning Algorithms for Prediction of SQL Injections." In Computer Vision and Robotics: Proceedings of CVR 2022, pp. 455-466. Singapore: Springer Nature Singapore, 2023.

[7] Alarfaj, Fawaz Khaled, and Nayeem Ahmad Khan. "Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks." Applied Sciences 13, no. 7 (2023): 4365.