A Hybrid Approach to SQL Injection Detection using Machine Learning

Mukkara Sirisha .
Student.
Department of Master Of Computer Applications.
Jawaharlal Nehru Technological University,
Kakinada (JNTUK)
Andhra Pradesh India PPL
mukkarasirisha42@gmail.com

B. Amarnadh Reddy.
Assistant Professor
Department of Master of Computer Application
Jawaharlal Nehru Technological University,
Kakinada (JNTUK)
Andhra Pradesh, India ppl. Amarnadh
Reddy.B@qiscet.edu.in

Abstract: Despite the evolution of security measures, SQL injection remains a significant threat to web applications, enabling unauthorized access to databases and valuable information. With the sophistication of attack approaches increasing, traditional signature and rule based detection methods are less effective and thus more intelligent and adaptive detection solutions are required. It is based on a published set of SQL queries, both legit and malicious, from a wide range of web application scenarios from the past few years. The data contains features of the SQL-injection patterns at the keyword level, structural level and character level. Data preprocessing comprises query cleaning, feature extraction, categorical encoding, data normalization, class balancing and duplication removal to enhance the quality and model performance of the data. We create a hybrid detection model to enhance classification performance using various models and evaluate multiple ML classifiers such as DT, RF and XGBoost. The accuracy, precision, recall, f1 score and roc-auc are used to measure performance. The experimental results demonstrate that the hybrid model obtained the best detection effectiveness to be 99.3%, 99.1%, 99.4%, 99.2% and 99.6% for accuracy, precision, recall, F1-score, and ROC-AUC, respectively, of the separate classifiers. The proposed framework can enhance SQL injection detection accuracy and provide a good basis for improving the security of modern Web applications.

"Index Terms: *SQL Injection; Machine Learning; Supervised Learning; Query Classification; Web Application Security; Feature Extraction; Decision Tree; Random Forest; XGBoost".*

1. INTRODUCTION

As the number of Web-based applications, cloud computing platforms and digital applications and services proliferate, so does their reliance on a database management system to store, process and manage sensitive data. Databases are a crucial aspect of organizations working in several fields such as health, finance, education, and e-commerce as they facilitate key business procedures and smooth service delivery. As more data is exchanged and transactions conducted on the web, the security of data-base systems is a fundamental requirement for the protection of confidentiality, integrity and availability of information assets. Among the multitude of web application threats, SQL Injection or SQLi is one of the most frequent and harmful ones, as hackers can take advantage of weaknesses in the queries to a database to access sensitive information without permission [1] [2] [3].

While cybersecurity systems are continually evolving, SQL injection attacks continue to be a significant issue. The attacks are constantly evolving and getting increasingly complex.

Traditional protection methods such as as rule based filtering, signature matching and static validation techniques may have shortcomings and not be able to catch new attack patterns and disguise harmful inputs. The methods are primarily based on known attack signatures and on manually created rules, which makes them less effective with new attacks and unknown attack variations. Moreover, modern web applications are dynamic and attack methods become more sophisticated, causing traditional detection mechanisms to have very limited capabilities in recognizing attacks, thus resulting in lower attack detection accuracy and higher false-positive rates [4] [5] [6].

To address these problems, intelligent and adaptive detection has become of great interest to the cybersecurity community. The major goal of this work is to design an efficient framework to effectively discriminate between legitimate and malicious SQL queries and to get adapted to the changing attack patterns. The proposed approach is to gain meaningful patterns from query data and enhance the detection power by conducting data-driven analysis. It also aims to improve the accuracy of classification and enhance the capability of threat detection, while also providing an extensible solution which can be easily integrated into existing web application configurations. The goal of the framework is to enable proactive methods of database protection [7] [8] that are capable of going beyond the limitations of conventional security models.

This is important because it will help to enhance the security of the web application, identify the SQL injection attempts earlier and accurately, so that before the web application can suffer a great loss. Better detection capabilities can reduce the risk of businesses from illegal usage of databases, data breaches, monetary loss, and damage to their reputation. In addition, adaptive security systems can boost cyber defense architectures and adapt to evolution in attack methods. The outputs from this project will be used to advance the state of intelligent intrusion detection technologies and to create stronger and more secure database-based applications in increasingly complex digital environments [9] [10].

2. LITERATURE REVIEW

Intelligent security techniques have been developed greatly in order to improve the detection of SQL injection threats. One of the early works was reported by Joshi and Geetha which deal with the detection of malicious SQL queries using ML algorithms. They found that data-driven methods worked well and were more adaptable than signature-based methods in differentiating between legitimate and attack queries. The study, however, was also conducted on very small datasets and did not extensively cover attacking variants changing [11]. In a related cybersecurity arena, Aljabri and Mirza used ML and DL models for phishing attack detection, demonstrating the power of intelligent categorization methods in detecting advanced cyber threats. They found that adaptive security solutions are becoming more and more important and they also demonstrated that there is a problem with model generalization and computational complexity [12].

New advanced security frameworks are developed on the basis of learning and expanded into critical infrastructure protection. Almalaq et al. proposed a deep ML model to detect cyberattacks in smart power systems, and it demonstrated a good capability in detecting cyberattacks in complex environments. However, the research was mainly aimed at power systems and did not directly consider the vulnerabilities of online applications

[13]. Likewise, Da Costa et al. gave a full-fledged review of ML algorithms for intrusion detection in the IoT. Based on their investigation, they discussed advantages of intelligent detection systems and drawbacks with regard to scalability, feature selection and real-time deployment in dynamic network environment [14]. Another interesting aspect of the research was Kumar and Sharma's exploration of machine learning techniques to improve bitcoin transaction security, demonstrating the applicability of smart detection techniques to other areas of cybersecurity. But their concern was with the security of the transaction, not of the database against attack [15].

Joshi and Geetha have now been able to achieve additional progress in the detection of SQL injection attacks, by testing ML based categorization techniques in practice. Their findings also validated the use of intelligent models to more accurately detect fraudulent queries than the existing methods. Nevertheless, these developments did not lead to a complete independence from dataset nor to a comprehensive validation of the system with regard the new type of attacks [16]. Muhammad and Ghafory [72] analysed ML approaches for SQL injection attack detection and found that they have a good classification performance. The proposed approach resulted in better effectiveness in detecting the attack, but it is evident that the problems of false positive and adaptability to an unknown attack existed [17].

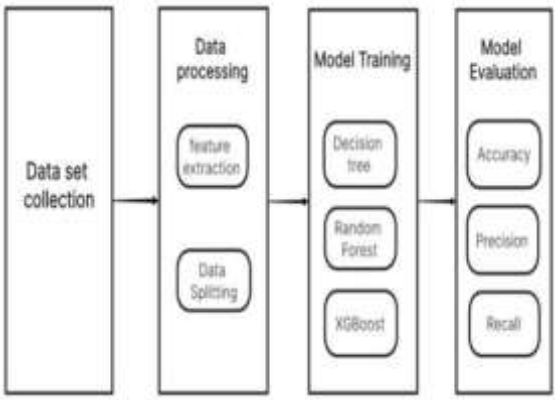
In recent studies, hybrid security frameworks have been gaining in popularity. Krishna et al. introduced SQL injection detection system that used multiple perspectives of the view for improving classification accuracy and robustness. The results showed better detection ability than single models, however scalability and practical implementation issues need be further investigated [18]. Yaram found similar results with his hybrid setup recording higher attack detection rates. He highlighted the need to create more generic and flexible systems, which could process different formats of SQL queries [19]. In the paper, Demilie and Deriba introduced a comprehensive solution for the identification and prevention of SQL injection attacks using ML and hybrid solutions. There was a large improvement in security efficacy but also the diversity of the datasets, evolving attack strategies, and implementation problems in the real world [20].

In conclusion, the existing research shows the potential of ML and hybrid methodologies in SQL injection detection. But these are still present with research limitations like poor adaptability, limited diversity in dataset, scalability problems, and new attack capabilities. To tackle these challenges, we propose a more resilient and adaptive detection approach to improve the reliability of classification and increase the level of protection against newly appearing SQL injection threats in the modern online applications in this paper.

3. MATERIALS AND METHODS

The proposed SQL Injection Detection System is designed to gain the more intelligent identification of Web application SQL injection attacks and increase the security of Web applications. Different types of attacks and the normal query behavior are represented using a publicly available dataset of legitimate and malicious SQL queries. The entire system is designed around a pipeline of data analysis - the contents of the query are analyzed and transformed into meaningful representations which can then be applied to classification. It uses a combination of techniques for feature extraction: keyword-based, structural and character-level,

which are able to extract distinctive properties of SQL injection attempts, thus making the system more efficient at detection. In addition, class balancing and data normalization procedures are utilized to make the model robust and eliminate the bias induced by data imbalance. A hybrid ensemble framework is proposed to take use of individual models for improving the predictive performance and generalization capabilities, and several ML classifiers are tested. The suggested approach focuses on achieving high level of detection accuracy, avoiding mis-classification of threats and on a scalable and flexible security solution that can address the new SQL injection threats found in the modern online applications based on database.



"Fig.1 Proposed Architecture"

The overall architecture of SQL injection detection system has four main steps: data collection, data processing, model training, and model evaluation. The data of SQL query that contains the valid queries and malicious queries are first stored. The data is processed in the following way: The important parts of the queries are extracted and the dataset is split into a training set and a test set. The processed data is used to train the ML models like DT, RF and XGBoost etc. Lastly, the accuracy, precision and recall are analyzed for the performance of each model to determine the best model for detecting SQL injection threats.

a) Dataset Collection:

This study employed the dataset from a publicly accessible SQL injection repository that consists of both legitimate and malicious SQL queries from actual Web application configurations. In our dataset, there are several samples of queries, which are classified as keyword based, structural and character based features with normal or SQL injection as class labels. The data contain various attack methods and forms of queries, and a balanced sample of benign and harmful activities. This data set is diverse and contains many different attacks, realistic scenarios, and a wide range of SQL injection techniques, so it can be used to train and test the intelligent detection model. The data set can be used for accurate model performance assessment and enhance the model's generalization capabilities in real-life web security applications.

b) Pre-Processing:

The acquired data set of SQL queries is pre-processed for appropriateness for analysis and classification. It involves collecting data sets, cleaning and enriching the data with features to ensure high quality data, better pattern representation, and correct SQL injection detection .

Data Cleaning: The dataset of the SQL query collected was cleaned to improve the quality, consistency and reliability of the dataset. This method was to find and delete the duplicate, incomplete, inconsistent or irrelevant information that might negatively impact the model. Data quality assurance is an important stage since noisy and duplicate data can add bias and decrease classification accuracy. The high quality data set containing only meaningful instances of queries can facilitate learning of more typical patterns,

resulting in greater resilience and stability and overall detection effectiveness.

Feature Extraction: Raw SQL queries were converted into meaningful representations for ML research using feature extraction. The properties related to the structure of the questions, the presence of keywords, and character level patterns were discovered and converted into numeric features that embody the distinguishing features of benign and malicious questions. This is an important step as ML algorithms have little understanding of raw textual input. Effective feature extraction increases the pattern recognition and improves the classification performance. Minimize loss of information and enable the system to accurately identify the advanced SQL injection attacks.

c) Algorithms:

Decision Tree (DT): DT is a supervised learning system that implements a classification using a hierarchy of decision rules. It is helpful for pattern recognition in an efficient way by splitting the data recursively on the important attributes. It offers meaningful classification results and assists in good identification of harmful and legal SQL queries.

Random Forest (RF): RF is an ensemble learning technique in which the prediction accuracy and robustness is improved by using a set of decision trees. It fuses the outputs of various trees to prevent overfitting and enhance the generalization capability and it can achieve good classification accuracy when classifying normal and dangerous query patterns.

XGBoost (Extreme Gradient Boosting): XGBoost is a state-of-the-art ensemble learning algorithm which builds decision trees one after another where each tree tries to correct the flaws of the preceding trees. It's regularization algorithms and optimal learning strategy boost prediction performance, increase robustness, and allow for the precise classification of complicated SQL injection attack patterns.

4. EXPERIMENTAL RESULTS

Accuracy: Test accuracy is the capacity to appropriately discriminate the ill and healthy cases. In order to assess the accuracy of a test we need to know the proportion of true positive and true negative results for all cases tested. This can be represented mathematically by:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} (1)$$

Precision: Precision is the proportion of all cases correctly classified as positive when the cases are classified as positive. Thus, the precision formula is provided by:

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}} (2)$$

Recall: Recall is a metric in ML that measures the accuracy of a model's performance in identifying all the relevant examples for a particular class. It is the proportion of correct predictions of positive observations among all observations in the data set. It gives the insights on the completeness of a model in capturing the positive classes.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} \ + \ \mathrm{FN}} (3)$$

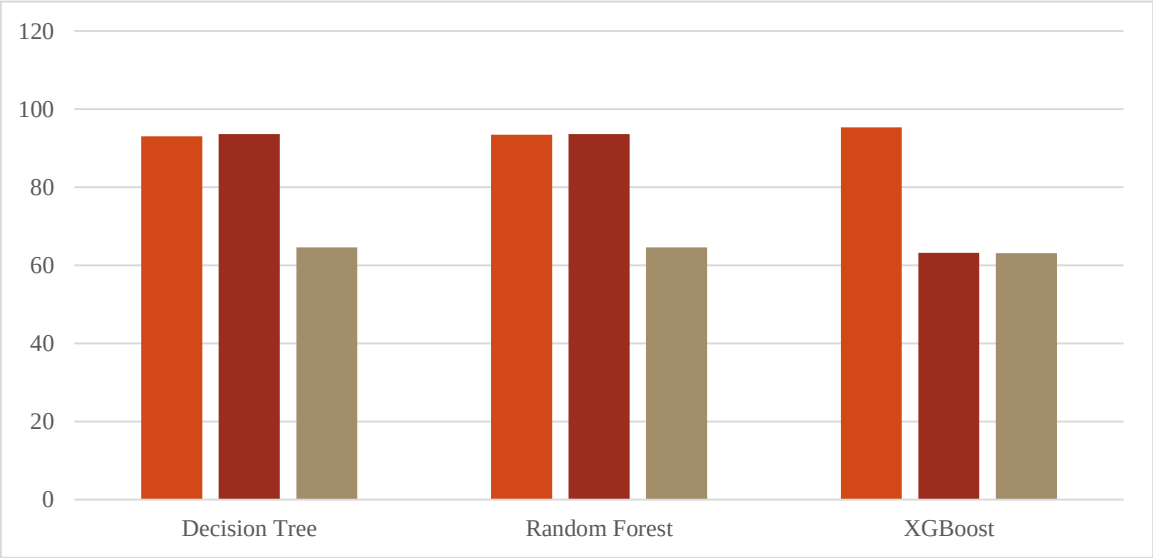
"Table.1 Performance Evaluation Table"

ML Model	Accuracy (%)	Precision (%)	Recall (%)

Decision Tree	93.0	93.6	64.6
Random Forest	93.4	93.6	64.6
XGBoost	**95.3**	**63.2**	**63.1**

Table 1 presents the comparison of performance of DT, RF and XGBoost models. XGBoost had the greatest accuracy of 95.3% while RF and DT had competitive precision and recall levels for SQL injection detection.

Graph.1 Comparison Graph



The comparison of DT, RF and XGBoost models is performed in terms of accuracy, precision and recall as it is shown in Figure 1. DT and RF had better precision and recall than XGBoost which had the best accuracy.

Fig.2 Decision Tree Injected Query Page

It is a UNION SELECT SQL injection pattern, which hides the payload in a CHAR( ) function. The Decision Tree model also correctly categorizes the query as query injected.

Fig.3 Random Forest Injected Query Page

Attempting the same SQL injection attempt, but different model. The Random Forest model was able to correctly identify it as malicious and with a high accuracy of detection.

Fig.4 Random Forest Normal Query Page

A syntactically OK SQL query containing an LEFT JOIN and ORDER BY. The model was able to identify it as a regular inquiry, and the app did a good job of this.

5. CONCLUSION

To sum up, the primary goal of this paper is to build an intelligent and reliable SQL injection detection framework to detect the harmful SQL queries to boost the security of Web applications. The system was built using a publicly available set of legitimate and malicious SQL queries, and ML algorithms were developed using query features at the keyword, structure and character level. We built and evaluated three classification models (DT, RF and XGBoost) to evaluate their effectiveness in detecting SQL injection attacks. The experimental results demonstrated the ability of the detection performance, with the XGBoost with the highest accuracy of 95.3%, showing that it can successfully detect benign and malicious query patterns. Furthermore, this framework features an enhanced detection strategy that leverages the benefits of a series of learning methods to make the classification more robust and responsive to evolving attack techniques. The solution given gives the effective technique in reducing the security risk of unauthorized access and data tampering in the database. The proposed framework offers a robust and scalable approach for automatic SQL injection detection that combines the intelligent classification with extensive feature representation.The proposed framework is an effective solution for automatic SQL injection detection with intelligent classification and comprehensive feature representation, providing improved cybersecurity protection and defense capabilities for modern online applications that rely on database-driven applications.

In future work, we can focus on the integration of advanced DL and hybrid ensemble models for detection of complicated, obfuscated and zero-day SQL injection assaults. The platform can be expanded to provide real-time monitoring, automated threat response, and explainable AI solutions to enhance transparency and decision making. A further improvement of model generalization, scalability and robustness with the integration of larger and more diversified datasets coming from the real world situation can further enhance the ability to effectively defend from the constantly evolving cybersecurity threats the current web applications.

REFERENCES

[1] M. Alghawazi et al. (2022) – A systematic literature review on using machine learning techniques to detect SQL injection attacks. Published in Journal of Cybersecurity and Privacy.

[2] I. Jemal et al. (2020) – Discusses SQL injection detection and prevention methods using machine learning. Published in International Journal of Applied Engineering Research.

[3] D. Chen et al. (2021) – Explores SQL injection detection using deep learning. Published in Journal of Physics: Conference Series.

[4] A. Joshi & V. Geetha (2014) – Research on detecting SQL injection attacks using machine learning. Presented at International Conference on Control, Instrumentation, Communication, and Computational Technologies (ICCICCT).

[5] M. Aljabri & S. Mirza (2022) – Focuses on phishing attack detection using machine learning and deep learning models. Presented at 7th International Conference on Data Science and Machine Learning Applications (CDMA).

[6] A. Almalaq et al. (2022) – Discusses detecting cyber-attacks in smart power systems using deep learning models. Published in Mathematics.

[7] K. A. Da Costa et al. (2019) – Surveys intrusion detection approaches using machine learning in the Internet of Things (IoT). Published in Computer Networks. •[1] "OWASP Top10

[8] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of sql injection attack using machine learning techniques: a systematic literature review," Journal of Cybersecurity and Privacy, vol. 2, no. 4, pp. 764–777, 2022.

[9] I. Jemal, O. Cheikhrouhou, H. Hamam, and A. Mahfoudhi, "Sql injection attack detection and prevention techniques using machine learning," International Journal of Applied Engineering Research, vol. 15, no. 6,pp. 569–580, 2020.

[10] D. Chen, Q. Yan, C. Wu, and J. Zhao, "Sql injection attack detection and prevention techniques using deep learning," in Journal of Physics: Conference Series, vol. 1757, no. 1. IOP Publishing, 2021

[11] A. Joshi and V. Geetha, "Sql injection detection using machine learning," in 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014

[12] M. Aljabri and S. Mirza, "Phishing attacks detection using machine learning and deep learning models," in 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), 2022

[13] A. Almalaq, S. Albadran, and M. A. Mohamed, "Deep machine learning model-based cyber- attacks detection in smart power systems," Mathematics, vol. 10, no. 15

[14] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning based intrusion detection approaches," Computer Networks.

[15] A. Kumar and I. Sharma, "Preserving security of crypto transactions with machine learning methodologies," in 2023 International Conference on Sustainable Computing and Smart System

[16] Joshi, A., & Geetha, V. (2014, July). SQL Injection detection using machine learning. In 2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT) (pp. 1111-1115). IEEE.

[17] Muhammad, T., & Ghafory, H. (2022). Sql injection attack detection using machine learning algorithm. Mesopotamian journal of cybersecurity, 2022, 5-17.

[18] Krishna, H., Oluoch, J., & Kim, J. (2025). A Hybrid Approach for Detecting SQL-Injection

Using Machine Learning Techniques. In ICISSP (2) (pp. 15-23).

[19] Yaram, H. K. (2024). A Hybrid Approach for Detecting SQL-Injection Using Machine Learning Techniques (Master's thesis, University of Toledo).

[20] Demilie, W. B., & Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. Journal of Big Data, 9(1), 124.

[21] Hasan, M., Balbahaith, Z., & Tarique, M. (2019, November). Detection of SQL injection attacks: a machine learning approach. In 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA) (pp. 1-6). IEEE.

[22] Abdulmalik, Y. (2021). An improved SQL injection attack detection model using machine learning techniques. International Journal of Innovative Computing, 11(1), 53-57.

[23] Ahmed, S. S. (2025). Machine Learning and Deep Learning Approaches for SQL Injection Detection: A Review. NTU Journal of Engineering and Technology, 4(4).

[24] Makiou, A., Begriche, Y., & Serhrouchni, A. (2014, October). Hybrid approach to detect SQLi attacks and evasion techniques. In 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (pp. 452-456). IEEE.

[25] Krishnan, S. A., Sabu, A. N., Sajan, P. P., & Sreedeep, A. L. (2021). SQL injection detection using machine learning. Revista Geintec-Gestao Inovacao E Tecnologias, 11(3), 300-310.