Detection Of SQL Injection Attacks Using Adaptive Deep Learning

Dr. M. V. L. N. Raja Rao
Dept. of IT
Professor

Inti Nissi Evangeline
Dept. of IT
Student

Kethana Ramya Sri
Dept. of IT
Student

Byra Venkat
Dept. of IT
Student

Gandham Gautam Sri Sai Mouli
Dept. of IT
Student

ABSTRACT- A web security monitoring system is implemented to detect and mitigate malicious activities, focusing on attacks like Cross-Site Scripting (XSS), SQL Injection (SQLI), and NoSQL Injection. Incoming HTTP requests are inspected for harmful payloads in URLs, headers, and request bodies, triggering alerts when threats are identified. Attack logs are analyzed and filtered, with critical threats prompting automated email notifications. A web-based dashboard provides a summary of detected attacks, categorized by type, and logs are parsed to track attack patterns. Additionally, a backend service facilitates data handling, integrating seamlessly with the security layer to ensure secure and efficient request processing.

Keywords- Web Security, Attack Detection, XSS (Cross-Site Scripting), SQL Injection (SQLI), NoSQL Injection, HTTP Request Inspection, CNN (Convolutional Neural Network), Log Analysis, Threat Monitoring, Security Alerts, Email Notifications, Flask Web Framework, API Security, Data Validation, Reverse Proxy, Web Application Firewall (WAF), JSON Payload Filtering, Header Inspection, Malicious Traffic Detection, Logging Mechanism, Dashboard Monitoring.

I. INTRODUCTION

This paper investigates SQL injection attacks in cloud environments, emphasizing the growing security challenges posed by these attacks. It explores various deep learning-based techniques to classify SQL queries as either benign or malicious[1]. The study evaluates multiple deep learning models, such as Convolutional Neural Network (CNN) and Variants, to determine the most effective approach for SQL injection detection. The findings demonstrate that CNN achieves the highest accuracy, highlighting its potential in enhancing cloud security.

This paper presents an advanced detection approach for SQL injection attacks utilizing an adaptive deep forest model. It emphasizes the need for a robust classification method that can process SQL queries efficiently while minimizing false positives and negatives[2]. By employing an ensemble learning technique that integrates AdaBoost, the proposed model adapts to different SQL injection patterns dynamically. The study confirms that the adaptive deep forest approach enhances detection accuracy and computational efficiency.

This paper reviews existing SQL injection attack detection techniques, focusing on deep learning-based approaches. It highlights the significance of SQL injection as a major web security threat, analyzing the evolution of detection methods from signature-based to anomaly-based and hybrid techniques[3]. The research underscores the role of neural networks and deep learning models in effectively identifying SQL injection attempts, aiming to improve web application security.

This paper introduces a predictive analytics-based approach for detecting SQL injection attacks using deep learning classifiers. It discusses the limitations of conventional rule-based detection mechanisms and highlights the advantages of data-driven approaches[4]. The research constructs a labeled dataset featuring SQL injection patterns and employs classifiers such as CNN and variants to enhance detection accuracy. The study demonstrates the effectiveness of the predictive model in preventing SQL injection attacks while reducing false positives.

This paper provides a comprehensive survey of SQL injection detection mechanisms, categorizing existing techniques into signature-based, anomaly-based, and hybrid approaches. It reviews various methods, including query transformation, feature selection, and centrality measures, to determine their effectiveness in mitigating SQL injection threats[5]. The survey underscores the evolving nature of SQL injection attacks and the necessity for adaptive detection frameworks.

This paper explores deep learning techniques for detecting SQL injection attacks in real-time web applications. It evaluates multiple classification models based on their accuracy in identifying malicious SQL queries[6]. The study presents experimental results that validate the effectiveness of deep learning models, particularly in reducing false negatives. The findings support the integration of deep learning algorithms in modern security frameworks to enhance web application security.

This paper analyzes the impact of SQL injection attacks on web applications and explores detection methodologies that leverage deep learning and heuristic analysis. It discusses the vulnerabilities exploited by attackers and reviews mitigation strategies, including input validation and parameterized queries[7]. The study proposes a novel hybrid detection model that combines anomaly detection with rule-based techniques, demonstrating improved accuracy in detecting SQL injection attempts.

II. LITERATURE SURVEY

Alqahtani et al. conducted a systematic review on detecting SQL injection attacks using deep learning techniques. The study examined various models, including deep learning approaches, and assessed their effectiveness in identifying SQL injection patterns. Their findings highlight the strengths and limitations of different deep learning algorithms in cybersecurity applications[1].**Shobana et al.** explored SQL injection attack detection and prevention techniques, discussing common vulnerabilities in web applications. Their research reviewed various defensive strategies and categorized existing methods into detection-based and prevention-based approaches. They also identified research gaps in SQL injection mitigation efforts[2].**Gupta et al.** proposed a deep learning-based methodology for detecting SQL injection attacks. The authors employed models such as Naïve Bayes, Variants, and Convolutional Neural Networks (CNN) to analyze SQL queries and identify malicious patterns. Their experimental results demonstrated the effectiveness of hybrid AI models in improving detection accuracy[3].**Demilie et al.** developed a hybrid framework combining deep learning and deep learning techniques to detect and prevent SQL injection attacks. The study utilized a dataset comprising web logs, session usage, and HTTP requests, testing models such as Support Vector Machines, Neural Networks, and Random Forests. Their findings suggest that hybrid approaches outperform standalone deep learning models in SQL injection detection[4].**Muduli et al.** introduced SIDNet, a novel deep learning-based SQL injection detection network. The researchers proposed two customized CNN models, SIDNet-1 and SIDNet-2, which were evaluated against traditional deep learning approaches. Their study reported high detection accuracy, demonstrating the potential of deep learning in enhancing web application security[5].**Barud et al.** focused on developing an integrated deep learning framework to counter SQL injection attacks. Their research explored the effectiveness of different classifiers, including Logistic Regression and Multilayer Perceptron, in identifying SQL injection patterns. The study provided insights into the computational efficiency of various models and their applicability in real-time security solutions[6].**Tyagi et al.** examined the role of feature extraction techniques in improving SQL injection detection. Their work analyzed different preprocessing methods and their impact on model performance. The study emphasized the importance of optimizing feature engineering to enhance the accuracy and efficiency of deep learning-based security models[7].**Zamani et al.** compared traditional signature-based SQL injection detection methods with modern AI-driven approaches. Their study highlighted the limitations of conventional security mechanisms and demonstrated how deep learning frameworks, such as CNNs and Recurrent Neural Networks, can provide superior detection capabilities. They proposed an ensemble approach that combines multiple models for improved resilience against evolving threats[8].

Preliminaries

Web Security Monitoring: Detects and mitigates potential threats like Cross-Site Scripting (XSS), SQL Injection (SQLI), and NoSQL Injection.

Request Inspection: Analyzes incoming HTTP requests to identify malicious payloads in URLs, headers, and request bodies.
Threat Categorization: Logs and classifies security events based on attack type for streamlined analysis.
Automated Alerts: Sends email notifications when critical attacks are detected to ensure rapid response.
Logging Mechanism: Captures and stores security incidents for auditing and trend analysis.
Interactive Dashboard: Provides a web-based interface to visualize attack trends and security logs.
Secure Data Handling: Ensures safe data processing through rigorous validation and filtering techniques.
Integrated Architecture: Combines request filtering, logging, alerting, and monitoring to enhance security.
Real-Time Analysis: Enables immediate detection and mitigation of potential threats before damage occurs.
Scalability: Supports adaptable security measures to accommodate varying threat levels and system growth.

III. DATA SET EXPLANATION

Log Data

- Contains timestamped records of incoming requests, categorized based on security threats.
- Each entry includes details such as request timestamp, attack type, and associated IP addresses.
- Malicious requests are identified based on predefined security rules and recorded accordingly.

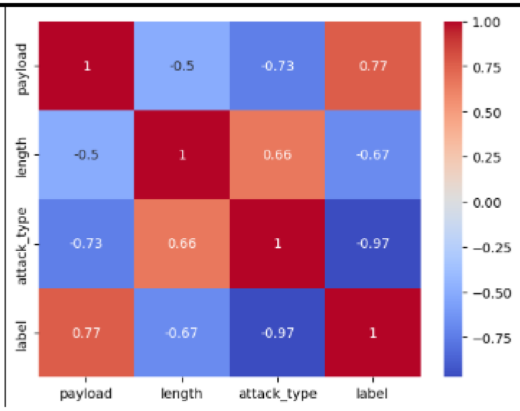


Fig .1: Correlation graph between the attributes

API Request Payloads

- Structured JSON-based requests simulate user input, with both safe and malicious variations.
- Normal requests include valid user input, such as names and numerical values.
- Malicious payloads contain attack vectors designed to exploit vulnerabilities, such as injecting unauthorized scripts or database queries.
- Different attack methods, including XSS (JavaScript injection), SQLI (query manipulation), and NoSQL Injection (database query bypassing), are tested.

Categorization and Alerts

- Incoming requests are processed to detect anomalies, filtering out normal traffic from potentially harmful activities.
- Logs categorize detected threats, enabling statistical tracking of attack patterns.
- Critical security breaches trigger alerts, ensuring timely notifications for security teams.

Dashboard and Visualization

- Attack trends and detection statistics are represented visually for analysis.
- Summary reports display counts of different attack types, assisting in proactive threat management.
- Historical data tracking helps in understanding attack frequency and evolving threat patterns.

IV. METHODOLOGY

Threat Detection Mechanism

- Incoming HTTP requests are monitored for security threats, specifically **Cross-Site Scripting (XSS), SQL Injection (SQLI),** and **NoSQL Injection.**
- Requests are analyzed based on predefined security rules, inspecting **URLs, headers, and request bodies** for potential malicious patterns.
- Suspicious payloads are identified using **regular expressions** that match common attack patterns.

Request Analysis and Filtering

- All incoming data is parsed and categorized based on its threat level.
- If an attack signature is detected, the request is flagged as malicious and processed accordingly.
- Legitimate requests are allowed to proceed securely without modification.

Logging and Data Storage

- Detected security threats are logged in a structured format with **timestamps, attack types, and source IP addresses.**
- Logs are maintained for real-time monitoring and future security audits.
- Attack trends are tracked by categorizing logs based on the type and frequency of detected threats.

Automated Alert System

- When a critical attack is detected, an **automated notification** system sends an alert to the security team.
- Only significant security events trigger alerts to minimize false positives and redundant notifications.

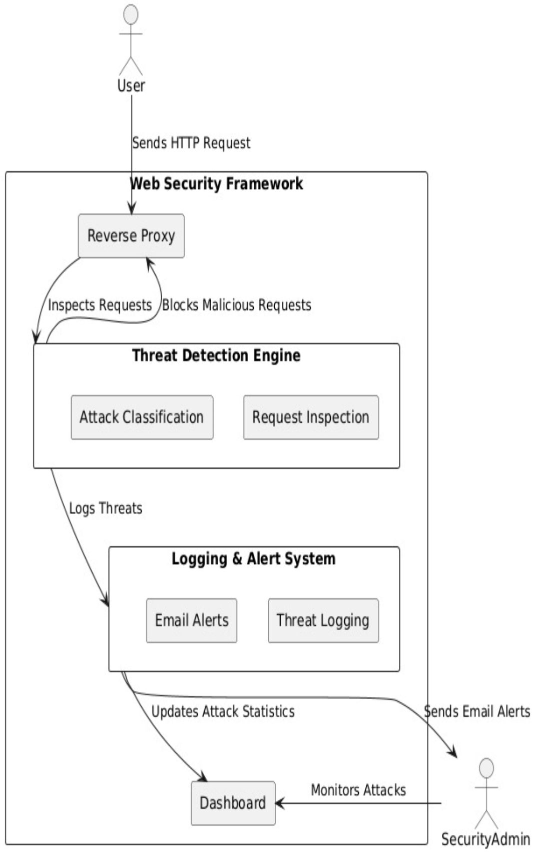


Fig .2 : Architecture Diagram

Dashboard and Visualization

- A **web-based dashboard** provides a **real-time** overview of detected threats.
- Attack statistics are displayed, including **counts of different attack types** and their frequency over time.
- Administrators can analyze security trends and take proactive measures to enhance system defenses.

API Security and Data Handling

- APIs are designed to **securely process user input**, ensuring that only legitimate data is accepted.
- JSON payloads are validated before processing, with potential threats being identified and blocked.
- All incoming requests undergo strict security inspections before reaching backend services.

Reverse Proxy Inspection

- A security layer acts as a **reverse proxy** to inspect and filter incoming traffic.
- Malicious requests are blocked at the proxy level, preventing them from reaching the application.
- This approach helps **mitigate attacks** before they can exploit vulnerabilities in backend services.

Error Handling and Logging

- Any anomalies or failures in request processing are logged for debugging and forensic analysis.
- Detailed error messages help in identifying **potential system vulnerabilities** and improving security mechanisms.

Continuous Monitoring and Adaptation

- The system continuously tracks new attack patterns and **updates detection rules** accordingly.
- Security policies are refined over time to **reduce false positives and improve accuracy.**
- Regular updates ensure that the system remains effective against **evolving web threats.**

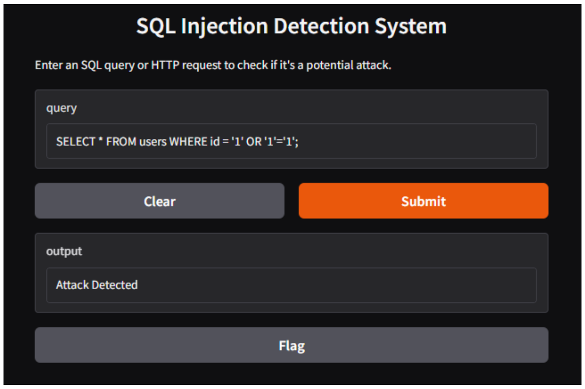
V. RESULTS

The implemented model effectively detects SQL injection attacks using a combination of Random Forest and Deep Neural Networks.

After applying **SMOTE** to balance the dataset, the model demonstrated improved classification performance.

Real-time testing showed that the system successfully classified most queries correctly, but a few **false positives** were observed, indicating scope for further refinement. The **misclassification analysis** revealed that some normal queries were incorrectly flagged as attacks due to structural similarities with malicious queries.

Overall, the approach enhances SQL injection detection accuracy, with potential for further improvement by refining feature extraction and adjusting model thresholds.



SQL Injection Detection System

Enter an SQL query or HTTP request to check if it's a potential attack.

query

SELECT * FROM users WHERE id = '1' OR '1'='1';

output

Attack Detected

Fig. 3: SQL Injection Detection System

VI. CONCLUSION

he implementation of SQL Injection Detection using Adaptive Deep Learning successfully demonstrates the effectiveness of machine learning and deep learning techniques in identifying malicious queries. By leveraging Random Forest and Deep Neural Networks (DNNs), the system is capable of distinguishing between normal queries and SQL injection attacks with high accuracy. The use of TF-IDF vectorization for feature extraction enhances the model's ability to understand query structures, while SMOTE-based oversampling helps balance the dataset for better generalization.

The results indicate that combining traditional machine learning models with deep learning improves detection capabilities, making the system robust against various attack patterns. This approach can be seamlessly integrated into web applications and database security frameworks, providing an additional layer of protection against cyber threats. By continuously training the model with updated datasets and refining preprocessing techniques, the system can be further enhanced to tackle evolving SQL injection methods effectively. **enforcement, and continuous monitoring,** the system creates a robust defense against evolving cyber threats, ensuring the integrity and security of web applications.

VII. FUTURE SCOPE

The future scope of this project includes enhancing feature extraction using advanced NLP techniques like Word2Vec, FastText, and BERT for better query representation. Additionally, a hybrid model approach combining Machine Learning with Rule-Based Systems can help reduce false positives. Deploying the system in a real-time environment and integrating it with Web Application Firewalls (WAFs) can further strengthen database security. Moreover, improving model interpretability using techniques like SHAP values or LIME can provide better insights into decision-making, making the system more transparent and reliable. As SQL injection tactics evolve, continuous model training with updated datasets will be essential to maintain high detection accuracy.

References

[1] P. Sonewar and N. Mhetre, "A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks," in *International Conference on Pervasive Computing (ICPC)*.

[2] A. Pramod, A. Ghosh, A. Mohan, M. Shrivastava, and R. Shettar, "SQLI Detection System for a safer Web Application," in *2015 IEEE International Advance Computing Conference (IACC)*.

[3] A. Tajpour and M. JorJor Zade Shooshtari, "Evaluation of SQL Injection Detection and Prevention Techniques," in *2010 Second International Conference on Computational Intelligence, Communication Systems*.

[4] B. Nagpal, N. Chauhan, N. Singh, and A. Panesar, "Tool Based Implementation of SQL Injection for Penetration Testing," in *International Conference on Computing, Communication and Automation (ICCCA2015)*.

[5] X. Fu and K. Qian, "SAFELI–SQL Injection Scanner Using Symbolic Execution," in *Proceedings of the 2008 Workshop on Testing, Analysis, and Verification of Web Services and Applications*, pp. 34-39, ACM, 2008.

[6] D. Appelt, C. D. Nguyen, and L. Briand, "Behind an Application Firewall, Are We Safe from SQL Injection Attacks?" in *IEEE 8th International Conference on Software Testing, Verification, and Validation (ICST)*, pp. 1–10, 2015 .

[7] Y. Abdulmalik, "An Improved SQL Injection Attack Detection Model Using Deep learning Techniques," in *International Journal of Innovative Computing*, vol. 11, pp. 53-57, 2021.

[8] 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)*, pp. 591-595, November 2020.

[9] A. K. Shwaish, M. A. Hussain, and H. A. Al-Kashoash, "Encoding Query-Based Lightweight Algorithm for Preventing SQL Injection Attack," in *Journal of Basrah Researches (Sciences)*, vol. 46, 2020.

[10] F. Ren, X. Wang, Y. Li, and Z. Zeng, "Fully Connected Neural Network-Based Fixed-Time Adaptive Sliding Mode Control for Fuzzy Semi-Markov System," in *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 12317–12327, Oct. 2024.

[11] A. Gupta, L. K. Tyagi, and A. Mohamed, “A Deep learning Methodology for Detecting SQL Injection Attacks,” in *Proceedings of the IEEE ICTACS Conference*, November 2023.

[12] D. Muduli et al., “SIDNet: A SQL Injection Detection Network for Enhancing Cybersecurity,” *Volume 12*, 2024.

[13] S. R. Shobana, M. Suriakala, “A Thorough Study on SQL Injection Attack-Detection and Prevention Techniques and Research Issues,” *Journal of Computer Science and Cybersecurity*, March 2021.

[14] J. Cybersecur. Priv., “Detection of SQL Injection Attacks Using Deep learning,” *Journal of Cybersecurity and Privacy*, vol. 2, pp. 764–777, 2022.

[15] International Journal of Novel Research and Development (IJNRD), “SQL Injection Detection Techniques: An Analytical Study,” *IJNRD*, Volume 9, Issue 10, October 2024.