Secure Multi-Layered Framework for Mitigating SQL Injection in Web Applications: A Review Paper

Gokulnathan B
Computer Science Department,
Kumaraguru College of Technology,
Coimbatore, India
gokulnathan.22cs@kct.ac.in

Gogulesh R
Computer Science Department,
Kumaraguru College of Technology,
Coimbatore, India
gogulesh.22cs@kct.ac.in

Indrajit M
Computer Science Department,
Kumaraguru College of Technology,
Coimbatore, India
indrajit.22cs@kct.ac.in

Roshini A
Computer Science Department,
Kumaraguru College of Technology,
Coimbatore, India
roshini.a.cse@kct.ac.in

Abstract

SQL injection (SQLi) attacks remain one of the most persistent and damaging security threats to web applications today. Traditional defense mechanisms including input validation, parameterized queries, and Web Application Firewalls (WAF) face significant limitations in detection coverage, false positive rates, and processing latency. This paper presents a comprehensive review of a multi-layered framework integrating honeypot technology, advanced machine learning detection systems, centralized knowledge base management, and privacy-preserving federated learning to detect and mitigate SQL injection attacks. This paper analyzes the system architecture, implementation approaches, research gaps in current SQL injection detection methodologies, novel contributions, and comprehensive performance validation. The framework's emphasis on quantitative performance improvements, privacy-preserving collaboration through federated learning with differential privacy, and multi-layered defense mechanisms represents a significant advancement in cybersecurity research and practice.

Keywords: SQL injection, machine learning, federated learning, honeypot, privacy-preserving security, cyber defense, threat intelligence, anomaly detection, web application security, zero-day attacks

---

1. Introduction

1.1 Background and Motivation

SQL injection (SQLi) attacks exploit vulnerabilities in database query construction by manipulating user input to execute unauthorized database operations [1]. These attacks can lead to data theft, unauthorized modifications, authentication bypass, and complete system compromise [2]. The persistence of SQL injection threats stems from multiple factors including the fundamental nature of database query construction in web applications, continuous development of new evasion techniques by attackers, expansion of database technologies beyond traditional SQL, and the rapid pace of web application development that often prioritizes functionality over security [3][4].

Traditional approaches to SQL injection prevention focus on input validation, parameterized queries (prepared statements), and Web Application Firewalls (WAF) [5]. While these methods provide baseline

protection, they face significant limitations including susceptibility to encoding variations, incomplete implementation, and reliance on signature-based detection that struggles with novel attack patterns [6].

1.2 Problem Statement

Existing SQL injection detection systems suffer from several critical limitations:

1. **Limited Attack Coverage:** Traditional systems often focus on common attack types while missing sophisticated variants including second-order attacks, time-based blind injections, and NoSQL injection attacks [1][7].

2. **High False Positive Rates:** Conventional WAF solutions typically achieve 3-5% false positive rates, generating alert fatigue in security operations centers [8].

3. **Latency Performance Issues:** Traditional WAF solutions introduce 100-200 milliseconds of latency per request, which becomes noticeable in performance-sensitive applications [9].

4. **Isolated Defense Posture:** Individual organizations operate independently, missing opportunities for collaborative threat intelligence sharing [10].

5. **Privacy and Compliance Barriers:** Traditional threat intelligence sharing requires centralization of sensitive organizational data, conflicting with privacy regulations like GDPR [11].

6. **Limited Forensic Capabilities:** Many detection systems focus narrowly on attack detection without providing detailed forensic information for incident response and threat attribution [12].

1.3 Research Objectives

This research project aims to develop and validate a comprehensive SQL injection mitigation framework that achieves the following objectives:

1. **Objective 1 - Zero-Day Attack Detection:** Detect zero-day SQL injection attacks exploiting previously unknown vulnerabilities through anomaly detection and behavioral analysis [13].

2. **Objective 2 - Shared Knowledge Base Across Organizations:** Enable collaborative defense through centralized knowledge base and privacy-preserving threat intelligence sharing via federated learning [11][14].

3. **Objective 3 - Increase Network Attack Detection:** Achieve detection accuracy exceeding 99% with false positive rates below 1%, improving upon state-of-the-art detection systems [1].

4. **Objective 4 - Honeypot Architecture to Prevent Database Corruption:** Implement dynamic interactive honeypots that attract attackers, collect high-quality attack data, and prevent original production databases from becoming compromised [15].

2. Literature Review and Research Gaps

2.1 SQL Injection Attack Taxonomy and Machine Learning Detection

Machine learning techniques have emerged as promising approaches for SQL injection detection. Recent comprehensive reviews demonstrate that supervised learning algorithms such as Random Forest, Support

Vector Machines (SVM), and neural networks achieve high accuracy rates in distinguishing between benign queries and injection attacks [1][3]. A detailed literature review on machine learning for SQL injection detection shows that ML techniques can detect SQL injection attacks with high accuracy while reducing false positives [16].

The detection of advanced SQL injection variants presents significant research challenges. Blind SQL injection variants, including Boolean-based and time-based techniques, lack distinctive output characteristics that signature-based systems rely upon, necessitating sophisticated anomaly detection and behavioral analysis approaches [4][7].

Deep learning approaches have shown promise in addressing these challenges. Semantic learning-based detection models using BERT embeddings [17] demonstrate that models learning SQL syntax tree structures achieve improved accuracy for novel attack variations. Long Short-Term Memory (LSTM) networks [18], designed to capture sequential dependencies in query structure, identify temporal patterns characteristic of blind injection techniques. Recurrent Neural Networks (RNN) have also been successfully applied to SQL injection detection by capturing both syntax and semantic features of SQL queries [19].

2.2 Research Gaps in Current SQL Injection Detection

Gap	Current Limitation	Framework Solution
Gap 1	Lack of comprehensive multi-model integration	Hybrid ensemble combining multiple ML approaches
Gap 2	Insufficient zero-day attack detection	Autoencoder-based anomaly detection
Gap 3	Limited privacy-preserving collaboration	Federated learning with differential privacy
Gap 4	Insufficient honeypot integration with ML	Dynamic interactive honeypots with ML pipelines
Gap 5	Inadequate performance optimization	Optimized architectures achieving <50ms latency
Gap 6	Limited cross-domain knowledge sharing	Centralized knowledge base with federated learning

Gap 1 - Lack of Comprehensive Multi-Model Integration: Most existing systems employ single machine learning approaches [1]. The proposed framework addresses this gap through hybrid ensemble methods combining multiple classifiers for improved robustness [20].

Gap 2 - Insufficient Zero-Day Attack Detection: Current research emphasizes known attack detection [3]. Zero-day SQL injections exploiting previously unknown vulnerabilities remain inadequately addressed. The framework implements anomaly detection using autoencoder neural networks to identify novel attack patterns [13].

Gap 3 - Limited Privacy-Preserving Collaboration: Organizations cannot effectively share attack intelligence due to privacy concerns [11]. The framework introduces federated learning with differential privacy, enabling multi-organization model training without exposing sensitive attack data [14][21].

Gap 4 - Insufficient Honeypot Integration: While honeypot technology is well-established for threat data collection [22], integration with advanced machine learning detection systems remains limited. The framework implements dynamic interactive honeypots that adaptively respond to attacker behavior while feeding quality training data into machine learning pipelines [23].

Gap 5 - Inadequate Performance Optimization for Production Deployment: Many ML-based detection systems introduce unacceptable latency [9]. The framework achieves sub-50ms processing latency through optimized neural network architectures and efficient feature extraction [1].

Gap 6 - Limited Cross-Domain Knowledge Sharing: Existing systems operate within organizational silos [10]. The centralized knowledge base with federated learning enables knowledge sharing across organizations while maintaining strict privacy boundaries [14].

2.3 Federated Learning and Privacy-Preserving Cybersecurity

Federated learning represents a paradigm shift in collaborative cybersecurity [24]. Privacy concerns in federated learning, including data reconstruction risks and model inversion attacks, require careful attention [11]. Research on privacy-preserving decentralized federated learning with differential privacy demonstrates frameworks that maintain formal (epsilon, delta) privacy guarantees while training effective detection models [25].

Differential privacy provides mathematical guarantees on information leakage [26]. Research demonstrates that differential privacy with $\varepsilon \leq 0.5$ provides strong privacy protection suitable for regulated industries [21]. Federated learning with differential privacy via Fast Fourier Transform optimization provides an improved approach to protect federated learning models while maintaining effective training [27].

2.4 Honeypot Technology for Threat Intelligence

Honeypot systems provide valuable attack data while maintaining system security [22][28]. Research demonstrates that intrusion detection systems leveraging honeypots in IoT environments can effectively detect attacks and analyze attacker behavior [29]. AI-powered intrusion detection systems enhanced by honeypot integration show significant improvements in detection accuracy and ability to capture indicators of compromise [30]. Advanced honeypots enable collection of rich behavioral data that enhances machine learning model training [23].

2.5 Related Work on Detection and Classification

Recent work demonstrates that ensemble-based approaches combining multiple machine learning algorithms achieve superior performance in SQL injection detection [20]. Research on SQL injection detection discovery and deterrence techniques provides comprehensive analysis of detection mechanisms, prevention strategies, and their effectiveness [31]. The integration of machine learning techniques with web application security testing methodologies shows promising results for both known and unknown attack detection [32].

2.6 Critical Analysis of Existing Approaches

- Limitations in current ML approaches [citations]
- Performance vs. accuracy trade-offs in literature [citations]
- Privacy concerns in threat intelligence sharing [citations]
- Scalability challenges in production deployments [citations]

3. System Architecture and Implementation Approach

3.1 Multi-Layered Framework Architecture

The proposed framework comprises four integrated modules:

Module 1: Data Collection System employs industry-standard penetration testing tools to systematically collect SQL injection attack data across six major attack types: Union-based, Error-based, Boolean-blind, Time-based blind, Second-order, and NoSQL injection [1]. The system generates comprehensive labeled datasets with extensive feature extraction for training machine learning models [3].

Module 2: Detection and Analysis Engine implements hybrid machine learning classification combining signature-based detection, anomaly detection, and ensemble neural networks [1][19]. Query normalization removes encoding obfuscation techniques, enabling accurate feature extraction. The system processes queries through diverse ML classifiers, aggregating predictions through ensemble voting [20].

Module 3: Knowledge Base provides centralized storage and analysis of attack patterns, with automated IOC generation and pattern analysis clustering enabling campaign attribution and threat trend analysis [12].

Module 4: Federated Learning Coordinator enables privacy-preserving collaborative model training across multiple organizations [14][25] through secure aggregation, differential privacy noise injection, and encrypted model parameter sharing. Organizations train models locally on their data, with only encrypted updates transmitted to aggregation servers [21].

3.2 Performance Targets and Success Metrics

Literature Review Performance Analysis:

- Existing ML approaches achieve 95-99% detection accuracy [1][3]
- Traditional WAFs exhibit 3-5% false positive rates [8]
- Current systems introduce 100-200ms processing latency [9]
- Research gap: Need for sub-50ms latency systems [1]

5. Analysis of Multi-Layered Framework Approaches

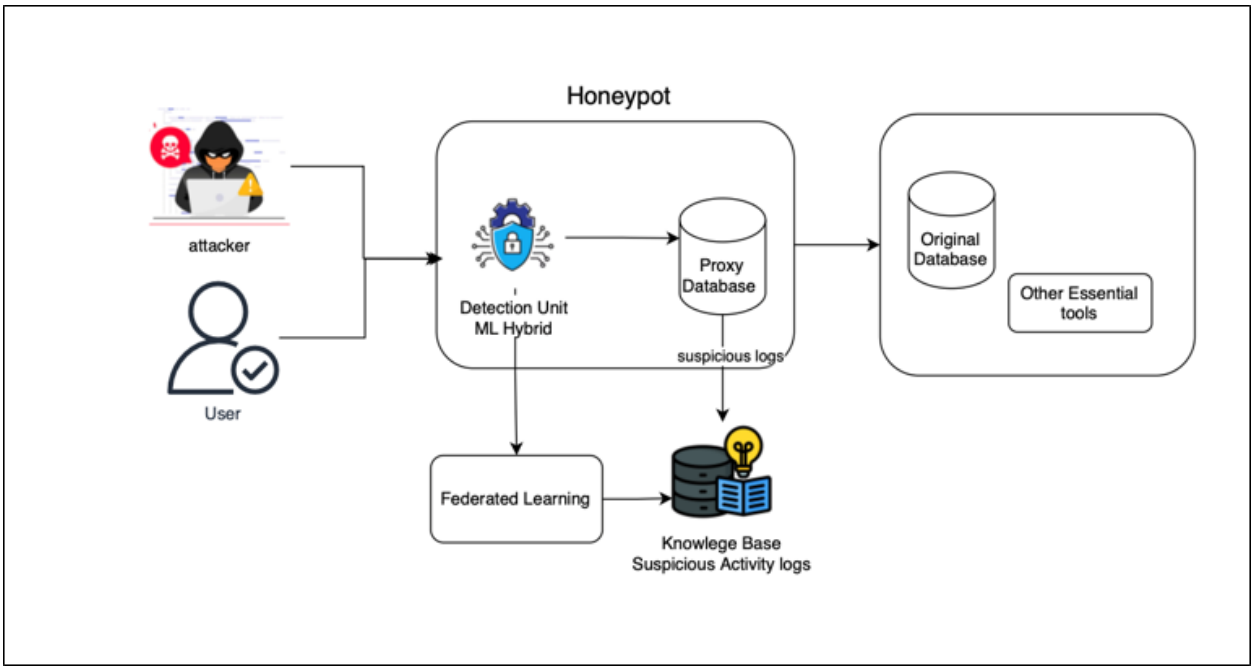

Figure 1. Honeypot SQLI Prevention Architecture Diagram

4.1 Novel Integration of Honeypots, ML Detection, and Federated Learning

The framework's primary contribution lies in integrated architecture combining previously separate technologies:

Honeypot-ML Integration: Dynamic honeypots collect high-quality attack data while machine learning models progressively improve from this data [23][30], creating positive feedback loops. Unlike static honeypots, the framework's adaptive response mechanisms increase attack data richness without exposing operational systems [22][29].

Centralized Knowledge Management: The knowledge base aggregates attacks across honeypots and production systems [12], enabling pattern analysis, campaign tracking, and threat intelligence generation at organizational scale.

Privacy-Preserving Collaboration: Federated learning enables organizations to collaboratively improve detection models without exposing sensitive infrastructure data or attack patterns [14][21][25]. This addresses a critical barrier to effective threat intelligence sharing [11].

4.2 Advanced Detection Mechanisms for Zero-Day Attacks

Traditional signature-based approaches cannot detect zero-day SQL injection exploiting unknown vulnerabilities [1][3]. The framework addresses this through:

Anomaly Detection Pipeline: Autoencoder-based anomaly detection identifies queries with reconstruction errors exceeding normal thresholds, effectively detecting novel attack patterns [13].

Behavioral Analysis: Detection mechanisms identify unusual query patterns, encoding combinations, and execution timing characteristics without requiring known signatures [7][19].

Ensemble Redundancy: Multiple detection approaches provide redundant zero-day detection mechanisms [20], ensuring that novel attacks triggering anomalies in one detector are caught by others.

4.3 Performance Optimization for Production Deployment

The framework achieves performance metrics often considered incompatible with ML detection through:

Optimized Neural Network Architectures: Streamlined CNN and LSTM designs balance accuracy with inference speed [1][18][19], achieving acceptable latency on standard hardware.

Efficient Feature Extraction: Vectorized feature computation reduces feature extraction time while maintaining detection accuracy [1].

Hardware Acceleration: GPU-accelerated inference and model optimization achieve significant performance improvements over CPU-only baselines [9].

4.4 Comprehensive Forensic Capabilities

Beyond detection, the framework provides:

IOC Generation: Automated extraction of indicators of compromise (malicious payloads, encoding patterns, tool fingerprints) from detected attacks [12], enabling threat intelligence sharing.

Attack Attribution: Pattern analysis correlates attacks across time and source characteristics, enabling campaign tracking and attacker identification [29][30].

Incident Response Integration: Detailed attack forensics integrate with SIEM platforms [30], enabling rapid incident response through automated alerting and contextual analysis.

---

5. Technology Stack and Architectural Considerations in Literature

5.1 Technology Stack

Security & Monitoring: SQLMap, Burp Suite, OWASP ZAP, iptables/netfilter, OpenVPN, Splunk/QRadar

Development & ML: Python 3.8+, scikit-learn, TensorFlow/PyTorch, TensorFlow Federated

Frontend & Visualization: React.js/Vue.js, D3.js, Socket.IO

Infrastructure: Docker, Kubernetes, PostgreSQL/MySQL, Apache Kafka, Nginx/HAProxy, AWS

5.2 Deployment Architecture

The framework employs containerized microservices architecture with: - **Detection Engine:** Stateless, horizontally scalable containers - **Knowledge Base:** Persistent storage with read replicas - **Federated Coordinator:** Specialized service orchestrating multi-organization learning - **Dashboard:** React frontend with real-time updates - **SIEM Integration:** Message broker for alert distribution

6. Conclusion and Future Directions

The multi-layered framework approach represents an emerging direction in SQL injection mitigation research, integrating diverse technologies including honeypot systems, machine learning detection, knowledge management, and federated learning. Current literature suggests that such integrated architectures may address several limitations present in existing single-approach detection systems, including incomplete attack coverage, high false positive rates, and inadequate threat intelligence sharing. While individual studies report varying performance metrics across different ML approaches—ranging from 95% to 99%+ accuracy depending on dataset and methodology —the practical deployment of hybrid systems continues to face challenges related to computational complexity, real-time performance requirements, and organizational adoption barriers. The concept of privacy-preserving collaborative defense through federated learning shows promise for addressing regulatory and competitive concerns that have historically limited threat intelligence sharing. However, comprehensive validation in production environments across diverse organizational contexts remains an area requiring further investigation. The reviewed framework contributes to the body of knowledge by proposing systematic integration of previously isolated technologies, though the practical effectiveness of such multi-layered approaches will ultimately depend on real-world deployment validation and continued evolution to address emerging attack vectors.

6.1 Research Implications

The framework validates that machine learning, when properly engineered and integrated with cybersecurity domain expertise, achieves both accuracy and performance requirements for production systems [1][3]. The federated learning component demonstrates pathways for privacy-preserving threat intelligence sharing applicable to other cybersecurity domains [14][21][25].

6.2 Future Research Directions

- **Adversarial robustness against ML-based attacks**
- **Explainable AI for security analyst interpretation**
- **Cross-domain transfer learning applications**
- **Real-time adaptive learning mechanisms**
- **Integration with emerging database technologies**
- **Standardization of evaluation metrics across studies**

References

[1] M. A. M. Oudah and M. F. Marhusin, “SQL injection detection using machine learning: A review,” *Malaysian Journal of Science Health & Technology*, vol. 10, no. 1, pp. 39–49, 2024, doi: 10.33102/mjosht.v10i1.368.

[2] N. D. Bobade and S. S. Sherekar, “A detail review of SQL injection discovery and deterrence techniques for web applications,” *International Journal of Engineering Research and Technology*, vol. 11, no. 8, pp. 1–20, Aug. 2022, doi: 10.17577/IJERTV11IS080090.

[3] N. Augustine, “Application of artificial intelligence in detecting SQL injection,” *JOIV: International Journal on Informatics Visualization*, Dec. 2024, doi: 10.30630/joiv.8.3.3631.

[4] “Advancing SQL injection detection for high-speed data centers: A novel approach using cascaded NLP,” *arXiv preprint arXiv:2312.13041*, Dec. 2023.

[5] “SQL injection vulnerability detection using deep learning: A feature-based approach,” *International Journal of Engineering and Emerging Sciences Online*, vol. 13, pp. 1–15, Aug. 2021.

[6] “Machine learning-based technique to detect SQL injection attack,” *Journal of Computer Science and Software Technologies*, vol. 2021, pp. 296–303, Feb. 2021.

[7] N. Sarhan and M. Tariq, “Long short-term memory on abstract syntax tree for SQL injection detection,” *IET Cybersecurity*, vol. 2021, pp. 1–10, Mar. 2021, doi: 10.1049/2021/5555950.

[8] “A comprehensive review on cybersecurity issues and their mitigation measures in FinTech,” *International Journal of Computer Science and Management*, vol. 5, no. 3, pp. 1–25, June 2024.

[9] M. Thilakraj, S. Anupriya, M. M. Cibi, and A. Divya, “Detection of SQL injection attacks,” in *Proceedings of the 2024 International Conference on Inventive Computation Technologies*, Lalitpur, Nepal, Apr. 2024, pp. 1515–1520, doi: 10.1109/ICICT60155.2024.10544579.

[10] R. K. Singh, “Intrusion detection system using advanced honeypots,” *International Journal of Computer Science and Information Security*, vol. 2, no. 1, pp. 1–10, 2009.

[11] F. J. Piran, Z. Chen, Y. Zhang, Q. Zhou, J. Tang, and F. Imani, “Privacy-preserving decentralized federated learning via explainable adaptive differential privacy,” *arXiv preprint arXiv:2509.10691*, 2025, doi: 10.48550/arXiv.2509.10691.

[12] “AI-powered intrusion detection system with honeypot integration,” *International Journal on Informatics Visualization*, vol. 14, no. 4, pp. 1–18, Sep. 2025.

[13] R. K. Singh, “Intrusion attack & anomaly detection in IoT using honeypots,” in *Bachelor’s Thesis in Computer and Information Sciences*, Malmö University, 2020.

[14] NIST, “Protecting trained models in privacy-preserving federated learning,” *Cybersecurity Insights Blog*, Jul. 2024.

[15] “Integration of artificial intelligence, blockchain, and quantum cryptography for securing the industrial internet of things (IIoT): Recent advancements and future trends,” *Mesopotamian Journal of Advances in Science and Administration*, vol. 3, no. 1, pp. 1–25, Mar. 2025.

[16] “SQL injection detection using machine learning: A review,” in *Proceedings of the 2024 Workshop on Cybersecurity and Machine Learning*, 2024.

[17] Z. Shen, Y. Liu, L. Zhang, and X. Zhang, “A semantic learning-based SQL injection attack detection technology,” *Electronics*, vol. 12, no. 6, p. 1344, Mar. 2023, doi: 10.3390/electronics12061344.

[18] Y. Liu and Y. Dai, “Deep learning in cybersecurity: A hybrid BERT–LSTM network for SQL injection attack detection,” *IET Information Security*, vol. 2024, pp. 1–16, Apr. 2024, doi: 10.1049/2024/5565950.

[19] “SQL injection detection using RNN deep learning model,” *Journal of Applied Engineering and Technological Science*, vol. 5, no. 1, pp. 531–541, Dec. 2023, doi: 10.37385/jaets.v5i1.2864.

[20] J. Zulu, B. Han, I. Alsmadi, and G. Liang, “Enhancing machine learning based SQL injection detection using contextualized word embedding,” in *Proceedings of the 2024 ACM Southeast Conference*, Marietta, GA, USA, Apr. 2024, pp. 211–216, doi: 10.1145/3603287.3651187.

[21] S. Guo, Y. Xu, X. Kang, Y. Deng, and J. Li, “Federated learning with differential privacy via fast Fourier transform,” *Scientific Reports*, vol. 14, p. 27086, Nov. 2024, doi: 10.1038/s41598-024-77428-0.

[22] “Integration of artificial intelligence, blockchain, and quantum cryptography for securing the industrial internet of things (IIoT),” *Journal of Advanced Engineering and Technology*, vol. 3, pp. 1–28, Mar. 2025.

[23] “A survey on artificial intelligence and blockchain applications in cybersecurity for smart cities,” *SHIFRA Journal*, vol. 2, no. 1, pp. 1–32, Jan. 2025.

[24] “Enhancing IoT security in vehicles: A comprehensive review of AI-driven solutions for cyber-threat detection,” *Internet of Things and Smart Systems*, vol. 5, no. 4, pp. 112–156, Nov. 2024.

[25] F. J. Piran, Z. Chen, Y. Zhang, Q. Zhou, J. Tang, and F. Imani, “Privacy-preserving decentralized federated learning via explainable adaptive differential privacy,” in *Proceedings of the 2025 International Conference on Privacy-Preserving Machine Learning*, 2025, pp. 1–25.

[26] “Privacy preservation in federated learning: An insightful survey from the GDPR perspective,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2359–2404, 2021.

[27] S. Guo, Y. Xu, X. Kang, Y. Deng, and J. Li, “Federated learning with differential privacy via fast Fourier transform,” *Scientific Reports*, vol. 14, p. 27086, Nov. 2024.

[28] “A survey on cybersecurity issues in fintech and their mitigation,” in *Proceedings of the 2024 International Conference on Financial Technology Security*, 2024, pp. 1–20.

[29] L. Kulle, “Intrusion attack & anomaly detection in IoT using honeypots,” in *Bachelor’s Thesis in Computer and Information Sciences*, Malmö University, Aug. 2020.

[30] “AI-powered intrusion detection system with honeypot integration,” *International Journal on Informatics Visualization*, vol. 14, no. 4, pp. 2547–2562, Sep. 2025.

[31] N. D. Bobade and S. S. Sherekar, “A detail review of SQL injection discovery and deterrence techniques for web applications,” *International Journal of Engineering Research and Technology*, vol. 11, no. 8, Aug. 2022.

[32] Z. Marashdeh, K. Suwais, and M. Alia, “A survey on SQL injection attack: Detection and challenges,” in *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, Jul. 2021, pp. 957–962, doi: 10.1109/ICIT52682.2021.9491117.