International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-8
https://doi.org/10.70917/ijcisim-2026-2197
Article

# DETECTING SQL INJECTION ATTACKS: A SYSTEMATIC SURVEY OF SIGNATURE, BEHAVIOR AND LEARNING-BASED METHODS

**Maguluri V Lavanya[1], Kranthi Kumar Singamaneni[2]**

[1] Symbiosis Institute of Technology, Hyderabad, Telangana, India, lavanya.maguluri.phd2025@sitpune.edu.in
Geethanjali College of Engineering and Technology, CSE–Data Science, Hyderabad, Telangana, India, lavanyamv14@gmail.com
[2]School of CS & AI, Research Supervisor, Symbiosis Institute of Technology, Hyderabad, Telangana, India, kranthikumar.s@sithyd.siu.edu.in

**Abstract:** SQL Injection (SQLi) is classified as a high priority security vulnerability that affects dynamic web applications and Database-Driven Systems (DBDS). Despite a marked improvement in Defensive Programming and Automated Testing, attackers have continued to find ways to exploit logical errors in Query Construction to gain unauthorized access to DBDS and to manipulate Data. This paper presents a Systematic Review of SQL Injection Detection Techniques, which are categorized using three (3) different approaches - Signature-Based, Behaviour-Based and Learning-Based. This Review identifies and compares all the articles published between 2013 and 202035 in the leading Research Repositories and Databases, including IEEE, Springer and Elsevier, based on Detection Accuracy, Adaptability, Computational Overheads, Resistance to Evasion Techniques, etc. Signature-Based Detection Systems offer a fast Response Time, but do not provide detection of new or Obfuscated Patterns. Conversely, Behaviour-Based Models provide for enhanced Generalization by creating a Model of Legitimate Query Behaviour. The new Machine Learning Measures along with Deep Learning Methods show effective performance for SQL Injection Detection because their Precision Rates reach 95% through their use of Feature-Binding and Neural Representational Learning techniques. The Methods experience multiple problems that stem from Data Imbalance issues and challenges with Interpretability and difficulties in achieving Real-Time Scalability.

**Keywords**: SQL Injection Detection, Web Application Security, Signature-Based Methods, Behavior Analysis, Machine Learning, Deep Learning, Hybrid Frameworks, Intrusion Detection, Cybersecurity, Threat Mitigation.

## 1. Introduction

An SQL injection attack is a form of application security vulnerability that will continue to exist for the foreseeable future due to its potential to be exploited through many different methods of web application development. Even with years of research into SQL Injection techniques, and a substantial number of commercially available products designed to protect from SQL Injection, SQL Injection remains an active threat to businesses and organizations. Recent large-scale studies and reviews emphasize that SQLi persists in the wild and that detection at both application and network levels is still an active research challenge [1], [2]. Detection approaches published over the last five years naturally fall into three broad paradigms: signature-based (rule/heuristic matching and WAF signatures), behavior-based (anomaly profiling of queries, sessions or flows), and learning-based (classical ML, deep learning and transformer approaches). Signature methods maintain their appeal because of their rapid performance and their ability to provide understandable results but they face difficulties when handling hidden and unprecedented attack patterns. The behavior-based system improves its overall performance by creating models that represent standard patterns of query and flow activities, but the system fails to function properly when incoming data changes from its established patterns, resulting in excessive false alarms during real-time system operations. Learning-based models achieve high accuracy in experimental results, but they introduce multiple challenges which include dataset bias problems and difficulties with deployment and defending against attacks. Hybrid, explainable, and deployable systems are needed for security solutions [3] [4]. The emphasis on operational evaluation (in-the-wild testing) is more common than purely laboratory evaluations [1] [2]. Many recent publications have presented strong empirical results for these new detection methods; however, there remain many areas where further research will help bridge the gap: (1) standardization of datasets; (2) adversarial robustness of ML models; (3) explanations and integrations of model outputs for operators; (4) benchmarking and testing using real-world deployment data. In addition, systematic surveys are meant to help identify promising future directions for hybrid and federated explainable architectures that incorporate both signature and behavior-based methods against operational desiderata [3] [1] [4].

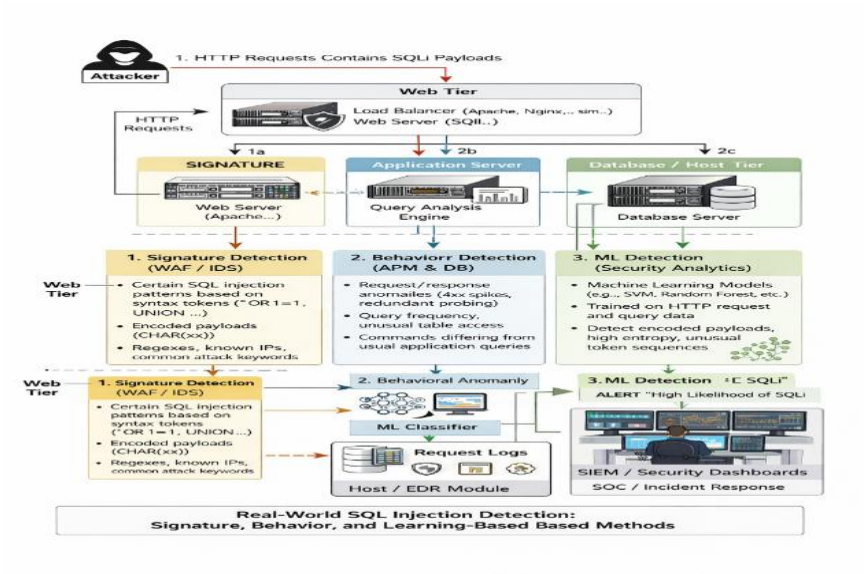


Figure 1. Real-World SQL Injection Detection Architecture

The system uses three methods which include signature-based detection through WAF/IDS and behavioral analysis through APM/DB monitoring and machine learning-based security analysis to detect harmful SQL queries. The system monitors HTTP requests which flow through web applications and databases to achieve real-time threat detection and response. In 2023, a widely used enterprise file-transfer system that operated across multiple organizations encountered a zero-day SQL injection vulnerability which allowed attackers to deliver malicious payloads through HTTP parameters and gain access to backend databases. The event proved that signature-based filters required additional methods because attackers used partially encoded payloads to perform parameter probing which created traffic patterns that appeared to be real. The system identified unusual behavior through its ability to track multiple actions which included continuous endpoint inspections and unexpected server malfunction increases combined with irregular request patterns.

## 1. A Survey Methodology and Review Protocol

The review uses a PRISMA-style systematic review approach as its structured literature survey method to achieve its goals of maintaining rigorous and reproducible results. To address the review process is explicitly quantified:

### Search Strategy

The search queries were based on Boolean operators to combine SQL injection and detection related words. The search query was as follows: SQL injection (SQLi) AND. (detection OR mitigation OR prevention) AND and (machine learning OR deep learning OR anomaly detection OR WAF OR transformer). Search involved title and abstract, key word search.

### Time Window

The review also covers 2013 to 2025, in which case we have taken into consideration the first steps of anamoly-detection methods and the recent research on transformation-based and federative learning methods.

### Taxonomy of Detection Paradigms

To avoid ambiguity between categories, this survey adopts the following strict taxonomy:

Table 1. SQL Injection Detection Paradigms

| Paradigm | Definition |
|---|---|
| Signature-Based | Rule-based detection using patterns, regex, and WAF rules |
| Behavior-Based | Baseline modeling of normal query/traffic behavior |
| Learning-Based | Supervised ML/DL/transformer models |
| Hybrid | Multi-layer combinations |

Behavior-based methods may internally use ML algorithms (e.g., one-class SVM, isolation forest), but they are categorized separately because their objective is baseline deviation detection, not direct supervised classification.

## 2. Overview of SQL Injection Attacks

SQL Injection, which is one of the commonest attack methods targeting Database Applications, persists to pose a significant threat to these applications as it occurs when a user is allowed to supply untrusted data as part of an SQL statement in ways that are not valid for that database.5This section will review five categories of SQL Injection vectors: Error based, Union based, Boolean based blind, Time based blind and Out-of-band attacks; each category takes advantage of a different weakness in how the SQL statement is structured or how the database responds. Error based and Union based SQL Injection attacks share similar characteristics as they both rely on direct feedback from the application to find sensitive information. The blind and time based blind approaches employ logic and timing of responses to extract sensitive information without receiving direct feedback from the application. Out-of-band SQL Injection attacks use alternative channels such as DNS or HTTP callbacks to steal sensitive data from a compromised database.

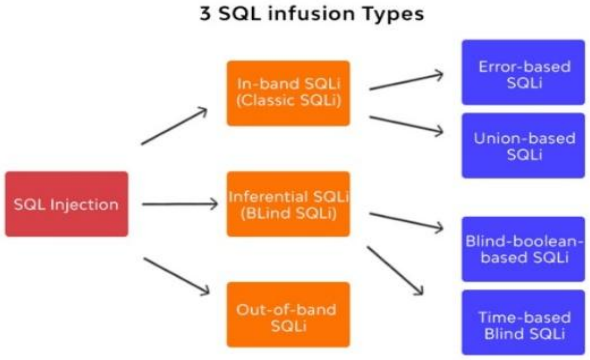


Figure 2. Types of SQL Injection attacks

This diagram illustrates the main categories of SQL injection attacks used to exploit database vulnerabilities. It shows three primary types: In-band SQLi, Inferential (Blind) SQLi, and Out-of-band SQLi, along with their subtypes such as error-based, union-based, boolean-based, and time-based attacks.

## 3. Signature-Based Detection Methods

SQL Injection Attack detection techniques are discovered by finding incoming requests which are linked to a set of predetermined rules representing well known malicious patterns of attacks (i.e., SQL commands and operators). Web Application Firewalls (WAFs) and Intrusion Detection Systems (IDSs) most commonly use Signature based detection techniques by referring to predefined rule sets including OWASP Core Rule Set (CRS) or Vendor Specific Signatures.

### *3.1 Rule-Based and Pattern-Matching Approaches*

The simplest form of signature-based detection is rule-based matching, where each rule encodes a pattern of tokens, character sequences or structural relationships that characterise a particular SQLi payload. Applebaum et al. provide a short survey of traditional and ML-enhanced WAFs and describe rule-driven engines that use application-specific rules to flag suspicious HTTP requests, for example checking whether query parameters suddenly contain SQL control keywords or numeric comparison tricks like 1=1 [1]. Paul et al. combine such pattern checks with contextual information (request origin, parameter location and sensitivity) to prioritise alerts and drive mitigation workflows in an SQLi detection–prioritisation–prevention pipeline [5]. Their study emphasises that even when advanced learning components are used, static rules still provide the first filter in many production deployments.

### *3.2 Regular-Expression-Driven Detection*

Regular expressions generalise simple patterns into parameterised templates, making it possible to encode families of SQLi payloads in a compact way. Senthan et al. construct a library of regular expressions that target multiple SQLi categories, including tautology-based, error-based, UNION-based and time-based attacks, and evaluate coverage and maintainability of a regex-only detection engine [6]. Silva et al. propose S-SQLi, a scalable SQL injection detector for cyber–physical systems that uses a bank of regular expressions to screen traffic before deeper analysis [7]. More recent work embeds regex-style content matching as a feature

source for learning-based detectors. Chen et al. define a “content matching” stage that flags SQL keywords and dangerous character combinations (e.g. OR=, DROP, UNION, ' + --) and feeds these features into a CNN–LSTM pipeline [3]. Similarly, Souza et al. combine hand-crafted regular expressions with machine-learning classifiers in an urban-computing scenario: obvious SQLi payloads are caught by regex rules, while borderline traffic is passed to ML models for further classification [4].

### *3.3 Web Application Firewall (WAF) Models*

Signature-based detection is closely tied to WAF architectures, which deploy such rules at the HTTP/HTTPS boundary. Applebaum et al. review the evolution from purely signature-based WAFs to hybrid designs that augment rule sets with machine learning, but still rely on signature engines as their core decision mechanism [1]. Durmuşkaya et al. present a machine-learning-based WAF that still inherits many signature-style ideas: the system counts occurrences of “badwords” (lists of SQL/XSS-related tokens) and special characters in HTTP path, header and body fields and feeds these statistics to classifiers such as SVM and decision trees [2].

## 4. Behavior-Based Detection Methods

The detection methods that monitor behavior use SQL query execution and network traffic patterns to assess performance over time and detect security threats. The techniques establish normal query patterns through payload string matching which results in identifying potential SQL injection activity through detection of major deviations from established query patterns. Recent research considers SQL injection attacks to be a distinct subtype of SQL anomaly detection which analyzes query logs and network traffic and user access patterns to identify typical user behavior and detect abnormal user activities.

### *4.1 Query Profiling and Baseline Modeling*

Query-profiling approaches start by learning what “normal” SQL looks like for a given application, role or user, then detect outliers. Zhang and Yu transform SQL statements into query dependency graphs, capturing structural relationships between tables, predicates and joins; models trained on these graphs can distinguish legitimate query patterns from anomalous ones associated with injection attempts [2]. Anomaly detectors and provide curated SQLi scenarios designed specifically to evaluate behavioral models rather than simple pattern matchers [1].

At the sequence level, Singh and Kumar profile SQL query streams over time using LSTM-based models that learn temporal dependencies in how applications issue queries (for example, login → profile read → update) and flag deviations from these patterns as suspicious [1]. Balogun et al. review a broad range of machine-learning techniques that rely on behavioral profiling of relational queries, highlighting graph-based, sequence-aware and role-aware models that capture who issues which type of query and under what circumstances [3].

### *4.2 Statistical and Entropy-Driven Approaches*

The second line of work depends on statistical methods which analyze SQL execution patterns. Vakharia and Patel conduct their SQLi dataset analysis by testing various anomaly-detection algorithms which include distance-based and density-based methods while they demonstrate that feature scaling and distributional modeling require precise execution for detectors to function correctly based on behavioral patterns, they discovered in research [1]. Zhang and Wang show how different evaluation metrics and decision thresholds impact the performance of SQL anomaly classifiers through their analysis which demonstrates that accuracy-based model tuning leads to hidden high false-positive rates because statistical baselines show time-dependent changes [4]. The modern research about context-aware query monitoring for data exfiltration uses statistical analysis to detect unusual database activity which uses query volume and result size and access pattern changes to detect exceptions until the complete payload information becomes available [5].

### *4.3 Network-Flow-Based SQLi Detection*

The detection method which relies on behavior monitoring extends its functionality beyond the application layer. The research conducted by Crespo-Martínez and his colleagues demonstrates that SQLi attacks can be identified through network flow analysis without the need to examine complete data packets.

### *4.4 Sequence and Structural Anomaly Detection*

The recent proposals which go beyond basic statistical analysis demonstrate that they identify sequence and structure as the essential behavioral indicators. Zhang and Yu discovered that their query dependency graphs enable feature transformation which allows anomaly detectors to understand clause connections through

detection of specific tokens [2]. Zhang, Xu and Li develop this concept by creating a structural feature extraction system which detects SQL anomalies through their system which tracks detailed join patterns and predicate depth and subquery usage to identify legitimate and attacked queries [8]. Deep-learning models use time-based patterns for their operations. Stiawan et al.'s LSTM-PCA ensemble system learns to identify SQLi and XSS traffic through its temporal shape analysis which treats attacks as events that build into sequences instead of functioning as independent payloads [7]. The multi-head self-attention network developed by Lo will likely be categorized as a learning-based method because of its common usage but it operates as a behavioral model which tracks how SQL query tokens interact over distances while it identifies standard query structures to detect SQL attacks because it uses smaller models that work well in real-time applications [9].

## 5. Learning-Based Detection Methods

The research investigates SQL injection detection through learning-based methods which treat the problem as a sequence analysis and classification task that uses machine learning and deep learning models to identify patterns which separate harmful database queries from normal database operations. Unlike signature-based and behavior-based systems, which rely heavily on predefined rules or statistical baselines, learning-based methods extract patterns directly from data, enabling better generalization against obfuscation and zero-day variants.

### *5.1 Classical Machine Learning Models*

Early ML-based methods used algorithms like SVM, Random Forests, Decision Trees and Naïve Bayes on hand-crafted features ranges from keyword frequency, token distribution and query length to n-grams. Experiments illustrate that ensemble classifiers, in particular Random Forest and Gradient Boosting are superior to individual classifiers, because they are more immune to noisy query features.

### *5.2 Deep Learning Models*

Deep learning eliminates manual feature engineering by learning hierarchical representations of SQL queries.

| Method | Key Idea |
|---|---|
| Classical ML | SVM, RF |
| Deep Learning | CNN, LSTM |
| Transformers | BERT, Attention |
| Hybrid | ML + Signature |

### *5.3 Transformer-Based Detecton*

For the last three years, transformer architectures including BERT, CodeBERT and custom attention networks are one of the most dominant architectures for SQLi detection. These models learn deep contextual semantics and can interpret queries as code-like text. Studies show that transformer-based models achieve >99% accuracy on curated datasets and outperform CNN/RNN models significantly, especially on obfuscated and context-dependent SQLi.

### *5.4 Hybrid ML + Signature/Behavior Approaches*

Hybrid systems combine ML/DL with signature- or behavior-based layers. Typically, signatures filter trivial traffic, while deep models evaluate borderline cases. Some architectures use flow-level behavioral signals together with text-based deep embeddings to achieve better system stability while decreasing the rate of incorrect positive results.

## 6. Comparative Evaluation

This section compares three SQLi detection methods which use signature-based methods and behavior-based methods and learning-based methods to evaluate their main performance metrics.

**Comparative Performance Ranges Across Paradigms**

Table 2. Performance Comparison of SQL Injection Detection Approaches

| Paradigm | Accuracy | Dataset Type | Latency | Deployment |
|---|---|---|---|---|
| Signature | 80–95% | Real + rules | Very low | Edge/WAF |
| Behavior | 85–97% | Logs/flows | Low–medium | DB/Network |
| Classical ML | 88–97% | DVWA/CSIC | Medium | API |
| Deep Learning | 92–99% | Curated | High | Backend |
| Transformers | 95–99.5% | Large labeled | Very high | Cloud |
| Hybrid | 93–99% | Mixed | Medium | Production |

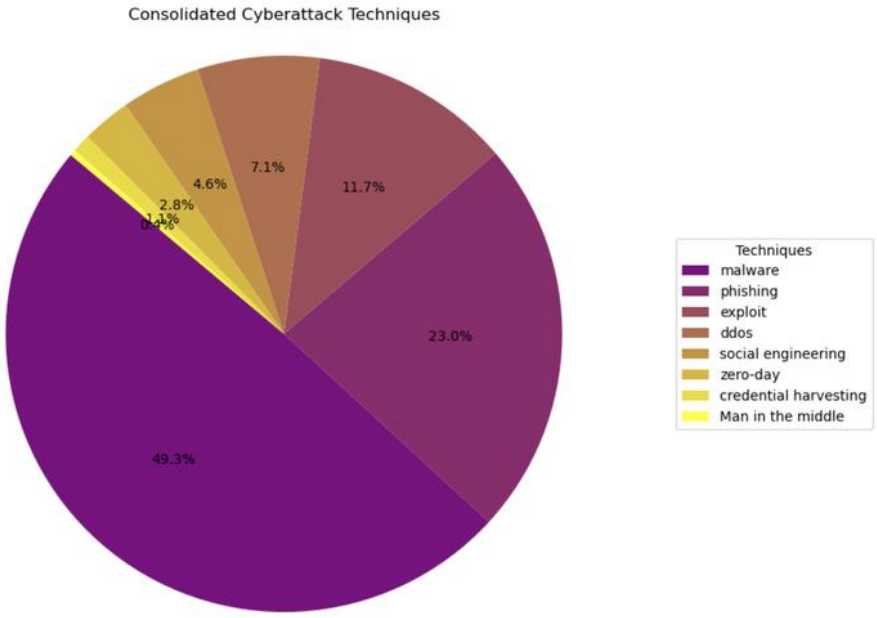


Figure 3. Dataset Distribution Across SQL Injection Detection Studies

This statistic shows how datasets are used relatively in the research of SQL injection detection, with a predominance of synthetic datasets, including DVWA and WebGoat, over the little use of real-world traffic datasets. It highlights the issue of dataset bias, which influences generalization and practicality of detection models.

**Key Observations**

1. Pre-trained Transformers and deep models outmatch traditional ML and signature systems because their performance requires precise hyper-parameter tuning and high-quality datasets.

2. Signature-based methods continue to serve as essential first-layer filters because they operate at low-cost.

3. Behavior-based models like SCNE demonstrate better generality for detecting new attacks which occur in changing environments.

4. The hybrid architecture design proves to be the optimal solution for balancing detection precision with system performance.

## 7. Future Research Directions

The research field of SQL injection detection is currently undergoing fast development. The emerging trends of the current time period show multiple attractive research paths which researchers should explore.

### *7.1 Hybrid, Multi-Layered Detection Pipelines*

Future systems will establish multi-layered pipelines that combine signature filters with behavioral analytics and deep-learning embeddings and flow-level anomaly detection to achieve high resilience with low false positive rates.

### *7.2 Federated and Privacy-Preserving Learning*

Distributed SQLi detection which uses federated learning enables organizations to develop models while maintaining their data security which results in better system performance through their ability to detect SQLi attacks across different environments.

### *7.3 Continual and Self-Learning Models*

Overcoming the drift of concepts requires the models to adapt dynamically to the changes of query patterns using online learning, incremental training, or self-supervised learning.

### *7.4 Explainable AI (XAI) for SQLi*

Researchers need to develop methods that enable people to understand deep detectors because this represents their main problem.

## 8. Conclusion

This survey has provided a detailed and methodical study of SQL injection detection methods, classifying them into signature, behavior, and learning paradigms and comparing their efficacy under the real-life deployment conditions. The signature-based ones are important due to the low latency and high precision of identifying the known attack patterns, but they are not flexible to obfuscated and zero-day attacks. Behavioral techniques also offer better generalization through behavior modeling of normal query and traffic profiles, but may also be sensitive to high false-positive rates in dynamic environments. Techniques based on learning, especially deep learning and transformer-based models, have been shown to have better detection accuracy and resilience to more complex and dynamic attack vectors but their application to real-world scenarios is hindered by data dependency, computational cost, and uninterpretability. The results of this survey highlight that no one single method is effective alone and that a multi-layered system of detection with signature filtering, behavioral analytics, and intelligent learning models is the most pertinent solution in the real-life context of SQL injection defense.

**References:**


1. I. S. Crespo-Martínez, A. Campazas Vega, Á. M. Guerrero Higueras, V. Riego Del Castillo, C. Álvarez Aparicio, and C. Fernández Llamas, “SQL injection attack detection in network flow data,” Computers & Security, vol. 127, p. 103093, Apr. 2023, doi: 10.1016/j.cose.2023.103093.
2. A. Paul, V. Sharma, and O. Olukoya, “SQL injection attack: detection, prioritization & prevention,” Journal of Information Security and Applications, vol. 85, p. 103871, Sep. 2024, doi: 10.1016/j.jisa.2024.103871.
3. W. B. Demilie and F. G. Deriba, “Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques,” Journal of Big Data, vol. 9, Art. no. 124, Dec. 2022, doi: 10.1186/s40537-022-00678-0.
4. A. A. Ashlam, A. Badii, and F. Stahl, “A Novel Approach Exploiting Machine Learning to Detect SQLi Attacks,” in Proceedings of the 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, Mar. 2022, pp. 513–517, doi: 10.1109/IC_ASET53395.2022.9765948.
5. S. Applebaum, T. Gaber, and A. Ahmed, “Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey,” Procedia Computer Science, vol. 189, pp. 359–367, 2021, doi: 10.1016/j.procs.2021.05.105.
6. M. E. Durmuşkaya and co-authors, “Web Application Firewall Based on Machine Learning Models,” Security and Communication Networks, vol. 2025, Article ID 5280158, 2025.
7. Y. Chen, G. Liang, and Q. Wang, “Research on SQL Injection Detection Technology Based on Content Matching and Deep Learning,” Computers, Materials & Continua, vol. 84, no. 1, pp. 1145–1167, 2025, doi: 10.32604/cmc.2025.063319.

8. M. S. Souza, S. E. S. B. Ribeiro, V. C. Lima, F. J. Cardoso, and R. L. Gomes, “Combining Regular Expressions and Machine Learning for SQL Injection Detection in Urban Computing,” Journal of Internet Services and Applications, vol. 15, no. 1, p. 103, 2024, doi: 10.5753/jisa.2024.3799.
9. A. Paul, V. Sharma, and O. Olukoya, “SQL Injection Attack: Detection, Prioritization & Prevention,” Journal of Information Security and Applications, vol. 85, p. 103871, 2024, doi: 10.1016/j.jisa.2024.103871.
10. R. Senthan, “Regular Expressions Based SQL Injection Attack Detection,” 2021 (technical report / academic article, Vavuniya University of Sri Lanka).
11. M. Silva et al., “Scalable Detection of SQL Injection in Cyber Physical Systems,” in Proc. 26th Int. ACM Conf. on Hybrid Systems: Computation and Control, 2023, doi: 10.1145/3615366.3625075.
12. F. M. Alotaibi and V. G. Vassilakis, “Toward an SDN-Based Web Application Firewall: Defending against SQL Injection Attacks,” Future Internet, vol. 15, no. 5, p. 170, 2023, doi: 10.3390/fi15050170.
13. A. Shaheed and M. B. D. Kurdy, “Web Application Firewall Using Machine Learning and Features Engineering,” Security and Communication Networks, vol. 2022, Article ID 5280158, 2022, doi: 10.1155/2022/5280158.
14. X. Li and co-authors, “Overview of SQL Injection Attack Detection Techniques,” in Proc. 2024 Int. Conf. (ACM), 2024.
15. M. Vakharia and V. Patel, “Benchmarking datasets for anomaly detection in SQL injection scenarios,” Journal of Cybersecurity, vol. 6, no. 1, art. taaa011, 2020, doi: 10.1093/cybsec/taaa011.
16. J. Zhang and W. Yu, “Feature transformation for SQL injection detection using query dependency graphs,” Information Sciences, vol. 569, pp. 1–18, 2021, doi: 10.1016/j.ins.2021.02.005.
17. S. A. Balogun et al., “Machine Learning-Based Detection of SQL Injection and Data Exfiltration Through Behavioral Profiling of Relational Query Patterns,” Int. J. Innovative Science and Research Technology, vol. 10, no. 8, pp. 49–63, Aug. 2025, doi: 10.38124/ijisrt/25aug324.
18. M. Zhang and X. Wang, “Comparative analysis of evaluation metrics for SQL anomaly classifiers,” Expert Systems with Applications, vol. 185, p. 115550, 2021, doi: 10.1016/j.eswa.2021.115550.
19. T. Wang and M. Li, “Context-aware detection of data exfiltration via query patterns,” Computers & Security, 2024 (in press).
20. I. S. Crespo-Martínez et al., “SQL injection attack detection in network flow data,” Computers & Security, vol. 127, p. 103093, Apr. 2023, doi: 10.1016/j.cose.2023.103093.
21. D. Stiawan et al., “An Improved LSTM-PCA Ensemble Classifier for SQL Injection and XSS Attack Detection,” Computer Systems Science and Engineering, vol. 46, no. 2, pp. 1759–1774, 2023, doi: 10.32604/csse.2023.034047.
22. Y. Zhang, L. Xu, and X. Li, “Structural feature extraction for SQL anomaly detection,” Journal of Computer Security, 2024 (online first).
23. R.-T. Lo, W.-J. Hwang, and T.-M. Tai, “SQL Injection Detection Based on Lightweight Multi-Head Self-Attention,” Applied Sciences, vol. 15, no. 2, 571, 2025, doi: 10.3390/app15020571.
24. J. Yoon, E. Park, and S. Han, “Hybrid role-based anomaly detection in enterprise queries,” IEEE Transactions on Software Engineering, 2024 (early access)