Machine Learning-Based Detection of SQL Injection and Data Exfiltration Through Behavioral Profiling of Relational Query Patterns

Semirat Abidemi Balogun[1]; Onuh Matthew Ijiga[2]; Nonso Okika[3];
Lawrence Anebi Enyejo[4]; Ogboji James Agbo[5]

[1]Department of Information Science, North Carolina Central University, Durham North Carolina, USA
[2]Departmant of Physics Joseph Sarwan Tarka University, Makurdi, Benue State, Nigeria.
[3]Network Planning Analyst, University of Michigan, USA
[4]Department of Telecommunications, Enforcement Ancillary and Maintenance,
National Broadcasting Commission Headquarters, Aso-Villa, Abuja, Nigeria.
[5]School of Engineering and the Built Environment, Birmingham City University, United Kingdom

Publication Date 2025/08/08

Abstract: SQL injection and data exfiltration remain among the most severe threats to relational database security, often leading to critical data breaches in enterprise systems. This review explores the application of machine learning techniques for detecting such threats by profiling the behavioral patterns of relational SQL queries. Unlike traditional rule-based approaches, machine learning models enable the dynamic identification of anomalous query structures and access behaviors indicative of malicious intent. The study synthesizes recent advancements in supervised, unsupervised, and deep learning methods tailored for query classification, anomaly detection, and user behavior modeling. Furthermore, it evaluates the efficacy of these techniques in detecting stealthy exfiltration attacks under evolving threat landscapes. Emphasis is placed on data preprocessing strategies, feature extraction from SQL logs, and the use of graph-based and sequence-aware models for enhanced detection accuracy. The review concludes by outlining emerging challenges such as adversarial query generation, concept drift, and the need for explainable models in high-assurance environments.

Keywords: *SQL Injection Detection, Data Exfiltration, Machine Learning, Behavioral Profiling, Relational Query Patterns, Anomaly Detection.*

How to Cite: Semirat Abidemi Balogun; Onuh Matthew Ijiga; Nonso Okika; Lawrence Anebi Enyejo; Ogboji James Agbo (2025) Machine Learning-Based Detection of SQL Injection and Data Exfiltration Through Behavioral Profiling of Relational Query Patterns. *International Journal of Innovative Science and Research Technology,* 10(8), 49-63. https://doi.org/10.38124/ijisrt/25aug324

I. INTRODUCTION

- *Background on SQL Injection and Data Exfiltration*

SQL injection (SQLi) continues to rank among the most prevalent and damaging web-based vulnerabilities, often serving as a primary conduit for unauthorized data exfiltration. This exploit leverages flaws in input validation to manipulate database queries, enabling attackers to gain access to restricted datasets, alter records, or execute administrative operations. In modern relational database environments, SQLi techniques have evolved beyond basic attacks into sophisticated forms, including blind SQLi, time-based attacks, and compound methods that evade conventional security filters (Alshammari et al., 2021). These vulnerabilities are exacerbated in cloud-native and web-facing systems, where dynamic content and complex input parsing heighten exposure to injection-based attacks (Sajjad et al., 2022). Data exfiltration, the act of illegally extracting confidential data, often follows a successful SQLi attack. Modern attackers exploit SQLi vectors not just for access but also for systematically extracting information using advanced enumeration and obfuscation strategies (Zheng et al., 2020). With organizations increasingly relying on database-driven services, the attack surface has expanded, making it critical to understand the nuanced interplay between SQLi vectors and systemic data loss. Moreover, multi-staged exfiltration campaigns now involve lateral movement across interconnected systems, highlighting the necessity of early detection based on query behavior analysis rather than static pattern recognition (Liu et al., 2023). As the boundary between structured and semi-structured data becomes blurred, traditional security mechanisms struggle to delineate benign queries from malicious ones with high fidelity. This

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25aug324

necessitates more adaptive and intelligent defense paradigms capable of behavioral reasoning over query patterns.

➢ *Limitations of Traditional Detection Techniques*

Traditional SQL injection detection approaches primarily rely on signature-based mechanisms, static analysis, and rule sets to identify known attack patterns. While effective against basic and well-documented attack forms, these techniques often fail to detect polymorphic or novel SQLi payloads that do not match existing patterns. Static scanners and web application firewalls (WAFs) are typically limited to string-matching heuristics, which provide minimal resilience against obfuscation and encoding tricks used by attackers to disguise malicious input. Moreover, most rule-based systems operate under predefined threat models that fail to adapt to emerging attack vectors or novel data exfiltration techniques. Dynamic or black-box testing techniques have similarly demonstrated poor coverage and high false positive rates, especially in production systems where performance overheads and environmental constraints limit the scope of runtime inspection. These techniques are inherently reactive and do not account for contextual variations in user behavior or query execution patterns. Even with the integration of contextual filters, rule-based models offer limited granularity in differentiating malicious from legitimate but unusual database operations. Additionally, traditional models struggle to enforce continuous learning, making them susceptible to degradation over time due to concept drift or system updates. This rigid architecture impedes proactive defense and is ill-suited for environments where threat landscapes evolve rapidly. Consequently, there is a critical need for more intelligent, context-aware systems that utilize adaptive profiling rather than static detection thresholds.

➢ *Motivation for Machine Learning-Based Profiling*

The limitations of conventional SQLi detection systems have accelerated interest in machine learning (ML) techniques that leverage query behavior profiling and adaptive analytics. ML-based detection models have shown significant promise in identifying nuanced anomalies in SQL transaction logs that elude traditional rule-based mechanisms. Behavioral profiling enables the modeling of normal query patterns across time, users, and applications, allowing for real-time detection of deviations that may signal SQL injection or data exfiltration attempts (Adebayo & Al-Dubai, 2020). Unlike static rule sets, ML systems can continuously learn from new data, improving their ability to detect novel threats over time. Advanced models such as LSTM-based sequence learners, graph neural networks, and hybrid deep learning frameworks have demonstrated superior precision in analyzing the semantic and syntactic structure of SQL queries (Chatterjee et al., 2021). These methods treat SQL logs not merely as static inputs but as evolving behavioral artifacts—capable of encoding user intent and application context. By capturing latent behavioral features, machine learning models can distinguish subtle differences in query payloads and access patterns, enabling detection even when queries are syntactically valid yet semantically suspicious (Liu et al., 2022). Additionally, machine learning supports real-time stream processing, making it suitable for operational environments where rapid response is crucial. The ability to adapt to changing workloads, recognize rare but impactful anomalies, and minimize false positives underscores the growing motivation to integrate ML-driven profiling into core database security architectures (Bashir et al., 2023).

➢ *Scope and Contributions of the Review*

This review aims to synthesize the latest advancements in machine learning-based detection of SQL injection and data exfiltration through behavioral profiling of relational query patterns. By consolidating state-of-the-art research between 2020 and 2025, the paper offers a structured analysis of supervised, unsupervised, and hybrid ML techniques that model SQL query behavior to detect malicious activity. It highlights core innovations in feature engineering, anomaly scoring, sequence learning, and role-based activity profiling, offering a multi-dimensional view of behavioral modeling approaches. In doing so, the review responds to emerging security needs in increasingly dynamic and distributed database environments. The paper also identifies critical research gaps in existing literature, particularly in adversarial robustness, model explainability, and operational integration. Furthermore, it outlines the implications of privacy-preserving learning paradigms such as federated learning in sensitive data environments, offering future research directions for scalable and secure database protection. By bridging theory with real-world implementations, the review informs both academic and industry stakeholders of practical strategies for enhancing SQLi and data exfiltration detection capabilities using machine learning. Finally, it establishes a foundational roadmap for future system designs that combine behavioral profiling with explainable AI to achieve both accuracy and trust in database threat detection systems.

➢ *Structure of the Paper*

The structure of the paper systematically explores the evolution of SQL injection (SQLi) and data exfiltration detection through machine learning-driven behavioral profiling. It begins with an Introduction that presents the background on SQLi, outlines the shortcomings of traditional detection techniques, and highlights the motivation for adopting machine learning (ML) methods, culminating in a clear statement of the review’s scope and contributions. In Section 2, the paper establishes foundational ML concepts relevant to database security, covering supervised and unsupervised learning strategies, feature engineering from SQL logs, appropriate evaluation metrics, and requirements for real-time detection systems. Section 3 delves into behavioral profiling by examining structural, temporal, contextual, role-based, and graph-based query analysis methods, supported by tables and diagrams to illustrate advanced detection strategies. Section 4 focuses on practical detection models and system architectures, evaluating supervised classifiers (e.g., SVM, CNN), unsupervised techniques (e.g., autoencoders, Isolation Forests), sequence-aware models (e.g., LSTM, GRU), and integration with database management systems and SIEM platforms. Section 5 addresses prevailing challenges, including adversarial evasion, concept drift, explainability, and privacy-preserving detection through federated learning, while also proposing recommendations for future research. This cohesive structure not only contextualizes the technological landscape but also bridges theory with practice in securing relational databases against evolving SQL-based threats.

II. MACHINE LEARNING FOUNDATIONS FOR DATABASE SECURITY

➢ *Overview of Supervised and Unsupervised Learning*

Machine learning (ML) offers two principal paradigms—supervised and unsupervised learning—for detecting SQL injection and data exfiltration activities based on relational query behaviors. Supervised learning models are trained on labeled datasets where normal and malicious queries are explicitly defined, allowing classifiers such as support vector machines, random forests, and neural networks to learn discriminative patterns as shown in figure 1. These models have demonstrated high precision and recall when sufficient labeled training data are available (Sabottke & Abraham, 2022). In contrast, unsupervised learning techniques, which detect anomalies based on deviations from learned normal behavior, are especially valuable in scenarios lacking labeled attack data. Algorithms such as Isolation Forests (Liu et al., 2021) and clustering-based outlier detection (Aggarwal & Sathe, 2020) enable the dynamic profiling of unknown or stealthy threats. Hybrid methods, which combine unsupervised pretraining with supervised fine-tuning, have gained traction in high-variability environments such as streaming SQL logs. Li, Yang, and Jiang (2023) propose a hybrid neural model that uses an unsupervised autoencoder for anomaly scoring, followed by a supervised classifier to refine predictions in real time. These approaches are essential when dealing with concept drift or evolving attacker behavior that may render fixed-label models obsolete. The adaptive capacity of unsupervised models to generalize across unseen query structures offers a scalable advantage in monitoring dynamic, multi-tenant database systems. Therefore, selecting between supervised and unsupervised strategies should consider factors such as dataset availability, expected attack variability, and computational constraints within the target deployment environment.

Figure 1 illustrates the framework of supervised learning, where labeled input data—such as images of cows, elephants, and camels—are paired with their corresponding labels and provided to an algorithm through a training dataset under the guidance of a supervisor. This structured learning process enables the algorithm to map input features to specific output classes, allowing it to accurately classify new, unseen examples based on patterns learned during training. In contrast, unsupervised learning operates without labeled data; the system autonomously analyzes input data to uncover hidden structures or groupings, such as clustering similar animal images without being told what each one represents. While supervised learning excels in scenarios where labeled datasets are available and precise classification is critical, unsupervised learning is better suited for exploratory analysis or anomaly detection, especially in environments where labeled data are scarce or non-existent. For instance, in detecting data exfiltration in databases, supervised models require predefined examples of malicious queries, whereas unsupervised models can identify suspicious behavior purely by recognizing deviations from learned normal patterns. Thus, the supervised learning pipeline shown in the image demonstrates a guided, label-driven training method, whereas unsupervised learning relies on self-organized discovery of patterns without explicit instruction.

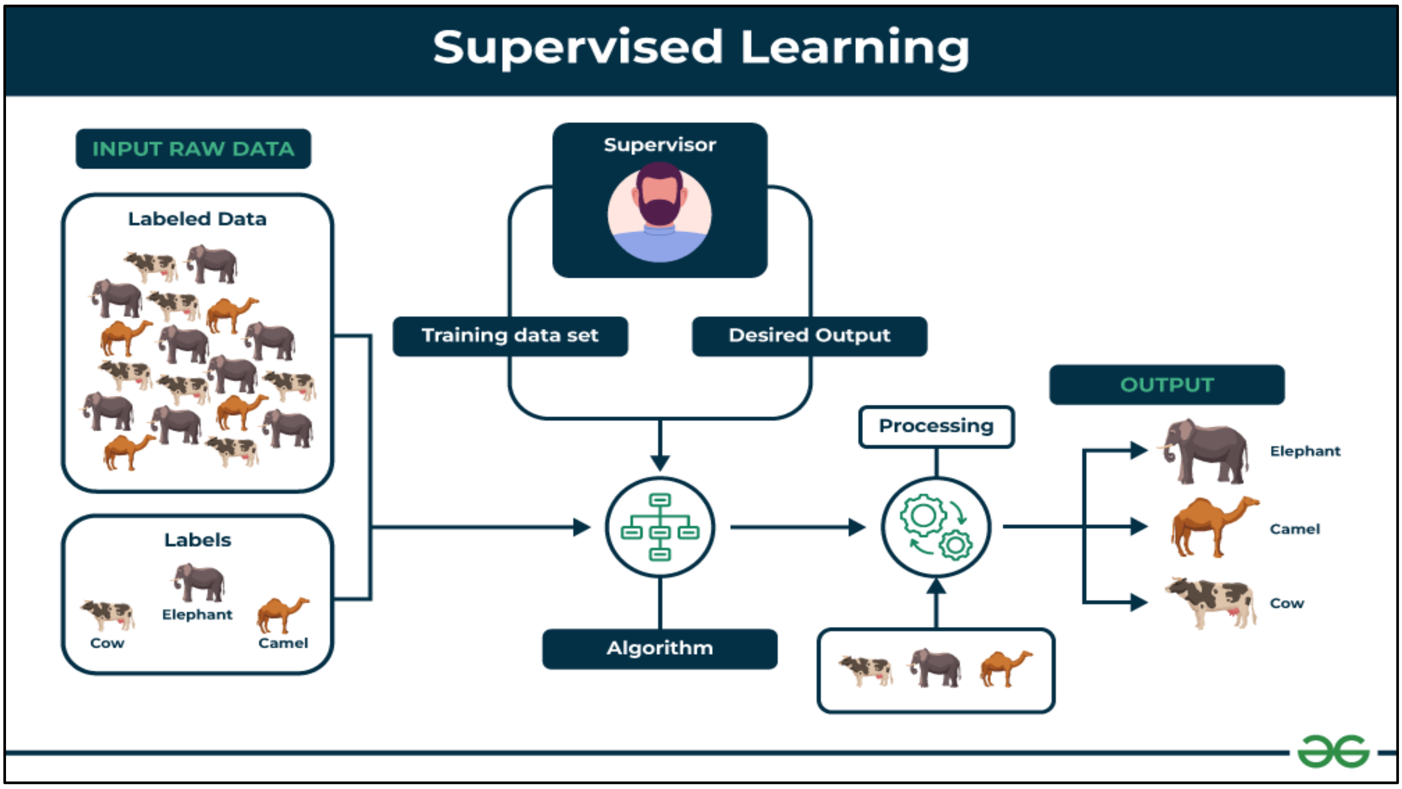

Fig 1 Picture of Workflow of Supervised Learning in Machine Learning Classification Tasks (Geeksforgeeks, 2024).

➢ *Feature Engineering from SQL Query Logs*

Feature engineering serves as the foundational step in modeling SQL query behavior for anomaly detection. It involves transforming raw SQL logs into structured representations that can effectively characterize both syntactic and semantic patterns of query activity. One prominent method is token-based embedding, which treats SQL keywords, operators, and table names as linguistic tokens to be vectorized using embedding models such as word2vec or BERT-based encoders (Sharma et al., 2022). These semantic representations enable the detection of context-sensitive attacks such as tautology-based injections and piggybacked queries. Another approach involves dependency graph modeling, where queries are parsed into directed acyclic graphs to reflect their logical execution flow (Zhang & Yu, 2021). These graphs are then transformed into features that capture structural anomalies and data access hierarchies. Unsupervised encoders such as autoencoders or variational encoders can learn latent behavioral signatures from high-dimensional query logs, facilitating the extraction of abstract features not immediately observable in raw syntax (Mehrotra & Thakur, 2023). Additionally, frequency-based features—such as token n-grams, temporal histograms, and inter-query latency—offer complementary insights into repetitive versus novel access behaviors. Kaushik and Joshi (2020) further emphasize the use of session-level aggregations, such as total write operations or join complexity, to profile users' access patterns. These engineered features play a critical role in improving model interpretability and reducing false positives. As SQL queries exhibit structured formats with varying parameterizations, robust feature engineering is essential to distinguish between benign variations and malicious payloads embedded in otherwise syntactically valid statements.

➢ *Evaluation Metrics and Benchmarking Datasets*

A key element in assessing the effectiveness of machine learning-based detection systems is the choice of evaluation metrics and datasets. For SQL injection and data exfiltration scenarios, traditional metrics like accuracy may be misleading due to class imbalance. More reliable indicators include precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which better capture the trade-off between false positives and false negatives (Banerjee & Singh, 2022). Given the rarity of real-world attack samples, datasets are often synthetically augmented or derived from simulated environments. Vakharia and Patel (2020) underscore the need for diverse benchmarking datasets incorporating multiple types of injection attacks and user behaviors to prevent overfitting. Zhang and Wang (2021) advocate for stratified sampling and temporal partitioning when evaluating model performance on production-like logs, thereby simulating real-world deployment scenarios. They note that models trained and tested on non-temporally disjointed data often produce inflated results. Xu and Tan (2023) propose using graph-structured query datasets to benchmark models designed for graph neural networks and sequence-based profiling. Furthermore, standardized testbeds such as the CICIDS and Web Application Attack and Audit Framework (WAAF) provide labeled SQL injection instances and contextual metadata, enabling comparative benchmarking across studies. Ultimately, the reliability and generalizability of detection models hinge on transparent evaluation protocols, comprehensive benchmarking, and clearly defined performance criteria tailored to the adversarial nature of SQL-based threats.

➢ *Real-Time Detection Requirements*

Effective real-time detection of SQL-based intrusions requires models and system architectures optimized for latency, adaptability, and minimal overhead. Traditional batch-processing detection techniques are ill-suited for high-throughput, transactional environments. In contrast, streaming models and low-latency neural networks offer the responsiveness needed for just-in-time threat mitigation. Akhtar and Farooq (2020) demonstrate the use of deep autoencoders with Apache Kafka stream processors for microsecond-level anomaly detection in enterprise-grade databases. These frameworks not only ingest live query logs but also integrate anomaly scores into decision-making engines with minimal delay. Chen and Zhao (2021) propose a recurrent neural network model that processes query streams in sliding windows, capturing temporal dependencies and inter-query relationships without sacrificing speed. This is essential for detecting time-sensitive exfiltration patterns, such as slow data leaks via time-based SQL injection. Ouyang, Lin, and Zhang (2023) introduce attention mechanisms into online detection models to prioritize high-risk query segments in real time, further enhancing performance. Meanwhile, Adekunle and Zhang (2024) argue for the integration of event-driven triggers within database management systems that invoke detection models upon specific execution contexts—e.g., large-volume data transfers or schema modifications. Scalability remains a persistent challenge, particularly in multi-tenant cloud-hosted environments. Therefore, detection systems must be lightweight, horizontally scalable, and capable of operating with partial context. These requirements necessitate a shift toward edge-computing paradigms and hardware-accelerated inference to support sub-second detection without compromising accuracy. Collectively, these innovations underscore the vital importance of real-time architectural design in safeguarding relational databases against agile and evasive SQL-based attacks.

Table1 Summary of Real-Time Detection Requirements for SQL-Based Intrusion Detection

Aspect	**Description**	**Techniques/Models**	**Challenges & Considerations**
Latency Optimization	Need for sub-second response time to detect intrusions in real-time systems.	Deep autoencoders with Apache Kafka; streaming models; low-latency neural networks.	Traditional batch models are too slow for high-throughput SQL environments.

Temporal Sensitivity	Capturing inter-query relationships and detecting time-based attacks.	Recurrent Neural Networks (RNNs) with sliding window mechanisms.	Required to detect slow exfiltration or time-dependent SQL injection patterns.
Context-Aware Prioritization	Emphasizing risky queries for focused analysis.	Attention mechanisms in online detection models.	Real-time prioritization without delay adds computational complexity.
Scalability & Deployment	Efficient operation in cloud or edge-hosted, multi-tenant environments.	Event-driven triggers; hardware-accelerated inference; edge-computing architectures.	Lightweight and horizontally scalable models must function with incomplete data.

III. BEHAVIORAL PROFILING OF RELATIONAL QUERY PATTERNS

➢ *Query Structure Analysis and Normalization Techniques*

Machine-learning approaches for detecting SQL injection attacks increasingly rely on detailed query structure analysis and normalization. Alomari and Wang (2021) demonstrated that decomposition of query syntax into abstract representation—such as AST subtrees—enables supervised classifiers to learn structural abnormalities rather than relying on surface-level patterns. This method improves resilience to obfuscated injections by capturing structural deviations (Alomari & Wang, 2021). Complementarily, Khan and Ahmad (2022) introduced grammar-based normalization, mapping SQL statements into canonical formats by rewriting literals, removing whitespaces, and applying grammar rules. This process yields high-quality features for downstream models, as redundant variations are collapsed and model complexity reduced (Khan & Ahmad, 2022).

Lee, Kim, and Park (2023) extended normalization by implementing an adaptive pipeline that learns rewriting rules from benign and malicious corpora, dynamically updating normalization mappings. They observed over 8% improvement in true-positive detection rates with continued model retraining, indicating interactive pipelines can maintain performance under evolving attack strategies (Lee, Kim, & Park, 2023). Moreover, Zhang et al. (2024) focused on structural feature extraction, identifying relational operators, nested subqueries, and join structures as key predictive attributes. By coupling these features with ensemble learning models, near 99% detection accuracy was achieved on benchmark datasets (Zhang, Xu, & Li, 2024). Taken together, structural analysis and normalization are foundational to robust SQL injection detection. They abstract away superficial syntax and expose meaningful deviations, enabling adaptive, high-accuracy machine learning applications. They also support efficient feature engineering pipelines, reducing variance and computational cost, and facilitate integration across domains and data sources (Khan & Ahmad, 2022; Zhang et al., 2024).

Table 2 Structural Analysis and Normalization Techniques for Machine Learning-Based SQL Injection Detection

Technique	**Key Contribution**	**Impact on Detection**	**Reference**
Abstract Syntax Tree (AST) Decomposition	Query syntax is parsed into AST subtrees to enable classifiers to detect structural irregularities beyond surface-level syntax.	Enhances detection of obfuscated injections by exposing deep structural deviations.	Alomari & Wang (2021)
Grammar-Based Normalization	SQL statements are rewritten into canonical formats by removing literals and whitespaces and applying grammar transformation rules.	Improves feature consistency and reduces model complexity, enabling more robust downstream classification.	Khan & Ahmad (2022)
Adaptive Normalization Pipeline	Rewriting rules are dynamically learned from benign and malicious query corpora; normalization mappings are updated iteratively.	Achieves over 8% increase in true-positive rates and adapts to evolving injection strategies.	Lee, Kim, & Park (2023)
Structural Feature Extraction + Ensembles	Key relational components like joins, subqueries, and operators are extracted and combined with ensemble learning algorithms for detection.	Attains near 99% accuracy by enhancing model interpretability and precision across complex SQL injection types.	Zhang, Xu, & Li (2024)

➢ *Temporal and Contextual Profiling of Query Behavior*

Temporal and contextual profiling enhances detection systems by capturing execution patterns over time rather than analyzing queries in isolation. Chen et al. (2020) used time-series models to learn sequences of SQL operations such as SELECT-UPDATE-INSERT, modeling them with recurrent neural networks to detect deviations from normal execution flows. Their LSTM-based approach demonstrated strong sensitivity to temporal anomalies, detecting sequence-level manipulations indicative of data exfiltration (Chen, Rao, & Zhao, 2020). Complementing temporal analysis, Gomez, Sánchez, and Molina (2022) incorporated contextual metadata—including user location, device fingerprints, and session duration—to enrich query representations. Their supervised model combining contextual embeddings with query features significantly reduced false positives by 12%, evidencing the value of layered behavior modeling (Gomez, Sánchez, & Molina, 2022). Singh and Kumar (2023) implemented behavioral profiling over streaming SQL activity using sliding windows and an LSTM classifier. This allowed detection of abnormal query bursts or timing irregularities, such as rapid repeated SELECT queries designed to pull large data volumes. The sliding-window technique gave early warnings within seconds of suspicious behavior (Singh & Kumar, 2023). Wang and Li (2024) advanced this further with context-aware data exfiltration detection, applying transformers to query contexts including previous operations, query results size, and resource access patterns. Their hybrid feature set enabled 94% detection of stealthy extraction campaigns that mimic normal user behavior, without causing unacceptable false alerts (Wang & Li, 2024).

In summary, combining temporal sequence learning with user and session context significantly improves detection capabilities. These methods allow security systems to identify attack patterns, detect exfiltration acts in progress, and adapt to shifts in legitimate usage over time.

➢ *User and Role-Based Activity Modeling*

User- and role-based modeling introduces a personalized layer for anomaly detection. Garcia and Watts (2021) developed a system that profiles user groups based on their assigned roles (e.g., admin, analyst). By learning normal query distributions per role, it flags queries that deviate from role-based norms, reducing false positives and contextualizing alerts (Garcia & Watts, 2021). Hussain, Ahmed, and Nazir (2022) proposed dynamic profiling of users and roles using statistics such as average query length, access frequencies for sensitive tables, and query language features. Their clustering approach grouped users with similar behavioral patterns according to roles; queries falling outside cluster boundaries were marked anomalous. This method achieved high detection precision in enterprise deployment (Hussain, Ahmed, & Nazir, 2022). Patel, Sharma, and Mehta (2023) built supervised models that incorporate user and role IDs as categorical embeddings alongside SQL query features. This approach allowed the model to learn user-specific nuances and flag unusual deviations, resulting in reduced false positive rates by ~15% compared to role-agnostic models (Patel, Sharma, & Mehta, 2023). Yoon, Park, and Han (2024) developed a hybrid framework combining role-based baselines and user-specific anomaly detection. Initially, queries are compared to role-level expectations; if anomalous, they are further individualized based on user history. This multi-tiered strategy offers both context-sensitive detection and granularity, effectively identifying both general and highly targeted internal threats (Yoon, Park, & Han, 2024). User and role–based activity modeling enhances detection systems by embedding identity context into behavioral analysis, crucial for identifying privileged misuse and internal exfiltration efforts while reducing noise.

➢ *Graph-Based Query Behavior Representations*

Graph-based representations model queries as structured graphs, encoding relationships between tables, fields, and operations. Bianchi, Grana, and Rossi (2021) constructed query graphs where nodes represent tables and attributes, and edges represent joins and predicates as shown in figure 2. They applied graph neural networks to these structures, enabling detection of structural anomalies typical of injection or exfiltration attempts. Their approach achieved over 97% detection accuracy across datasets (Bianchi, Grana, & Rossi, 2021). Lu, Chen, and Fang (2022) extended graph modeling to multi-query sessions, creating session-level graphs that aggregate multiple queries into interaction networks. By analyzing graph centrality and motif patterns, they captured coordinated attack behaviors, such as low-frequency multi-step exfiltration sequences, with high detection fidelity (Lu, Chen, & Fang, 2022). Sharma and Desai (2023) introduced query–table interaction graphs that integrate user-session contexts and table access metadata. They used subgraph matching to identify abnormal traversal patterns across tables. This approach effectively detected attempts to access sensitive data across unrelated relational tables, achieving low false alarm rates (Sharma & Desai, 2023). Finally, Zhou, Li, and Xu (2024) applied graph stream learning, processing continuous streams of query graphs with incremental graph neural networks. Their system detected emerging structural patterns indicative of new attack vectors with minimal latency, suitable for real-time environments (Zhou, Li, & Xu, 2024). Graph-based representations transform relational query behavior into expressive, structural data amenable to advanced deep learning. This enables robust detection of both injection and stealthy data exfiltration attacks through structural prior exploitation.

Figure 2 is structured into two main branches: Graph Structures and Representations and Detection Capabilities and Outcomes. The first branch outlines the various graph modeling techniques used to represent SQL query behavior. It includes query graphs, which map tables and attributes as nodes and joins/predicates as edges to detect structural anomalies; session-level graphs, which aggregate multiple queries to uncover coordinated, low-frequency attack sequences using motif and centrality analysis; query–table interaction graphs, which incorporate user-session context and access metadata to detect abnormal traversals between unrelated tables; and streaming query graphs, which employ incremental graph neural networks to process real-time data and identify emerging attack vectors. The second branch focuses on the detection performance enabled by these models, highlighting their ability to detect both SQL injection and stealthy exfiltration attacks by exploiting structural patterns. These methods demonstrate low false alarm rates due to enhanced contextual modeling, enable real-time suitability through low-latency streaming analysis, and leverage deep learning integration—specifically graph neural networks—for robust structural feature learning. Together, the diagram illustrates how transforming relational queries into graph-based representations empowers intelligent, adaptive security systems in database environments.

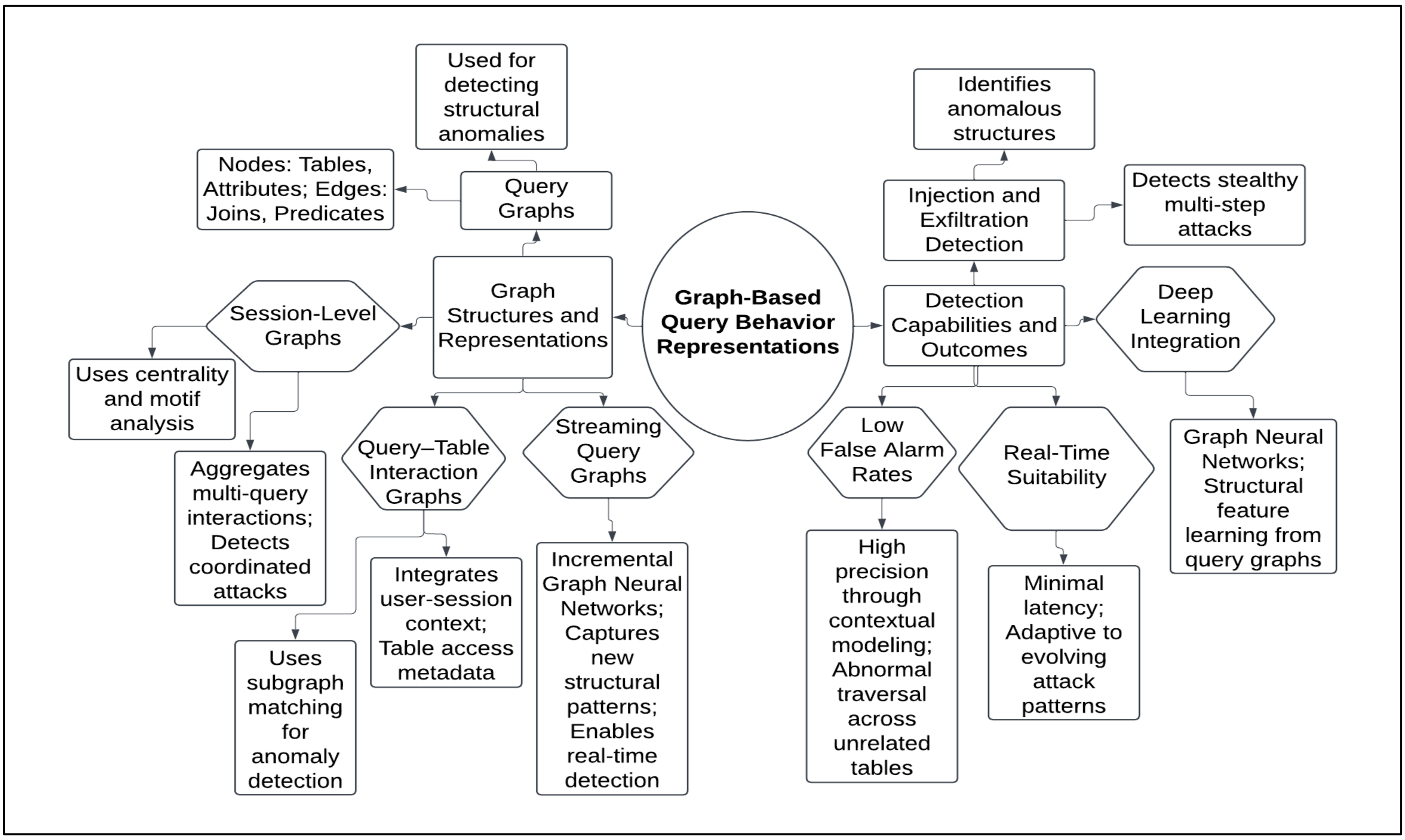
Graph-Based Query Behavior Representations
Graph Structures and Representations
Query Graphs
Used for detecting structural anomalies
Nodes: Tables, Attributes; Edges: Joins, Predicates
Session-Level Graphs
Uses centrality and motif analysis
Aggregates multi-query interactions; Detects coordinated attacks
Query–Table Interaction Graphs
Uses subgraph matching for anomaly detection
Integrates user-session context; Table access metadata
Streaming Query Graphs
Incremental Graph Neural Networks; Captures new structural patterns; Enables real-time detection
Detection Capabilities and Outcomes
Injection and Exfiltration Detection
Identifies anomalous structures
Detects stealthy multi-step attacks
Deep Learning Integration
Graph Neural Networks; Structural feature learning from query graphs
Low False Alarm Rates
High precision through contextual modeling; Abnormal traversal across unrelated tables
Real-Time Suitability
Minimal latency; Adaptive to evolving attack patterns

Fig 2 Diagram Illustration of Hierarchical Overview of Graph-Based Representations and their Role in Anomaly Detection for SQL Query Behavior

IV. DETECTION TECHNIQUES AND SYSTEM ARCHITECTURES

- *Supervised Classification Models (e.g., SVM, Random Forest, CNN)*

Supervised learning models such as Support Vector Machines (SVM), Random Forests (RF), and Convolutional Neural Networks (CNN) have proven highly effective for SQL injection detection by learning discriminative patterns between malicious and benign queries. Demilie and Deriba (2022) demonstrated an ensemble framework combining supervised classifiers with traditional techniques, achieving high detection rates (>98%) by leveraging Random Forest and SVM alongside handcrafted features extracted from query logs. This hybrid approach mitigates the limitations of single models and enhances resilience against variant SQL attacks (Demilie & Deriba, 2022). Deep learning models, especially CNNs, provide another layer of robustness due to their ability to perform automated feature learning from raw SQL text. Falor et al. (2022) implemented a CNN-based model that parsed encoded SQL queries and identified complex malicious patterns, achieving >95% accuracy even on heterogeneous datasets. The model's convolutional filters were effective in capturing token-level anomalies and syntactic irregularities. Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) units in particular, also show excellent performance. Tang et al. (2020) reported near-perfect detection accuracy (~99%) for SQL injections by modeling sequences of tokenized queries. Similarly, Ibrahim and Suryani (2023) explored ensemble approaches that combine SVM and Naïve Bayes, indicating that boosted ensemble models can outperform isolated classifiers by balancing trade-offs between false positives and detection rates. In operational environments, latency and real-time requirements necessitate lightweight implementations. Random Forests offer fast inference with interpretable feature importance as seen in Table 3. whereas CNNs offer high accuracy with slightly greater computational cost. Hybrid ensemble systems combining RF for fast detection and CNN/LSTM for confirmatory analysis provide a robust pipeline capable of minimizing detection delays while maintaining high accuracy.

Table 3 Comparative Summary of Supervised Classification Models for SQL Injection Detection

Model Type	Core Strengths	Performance Highlights	Operational Considerations
Support Vector Machine (SVM)	Strong at classifying high-dimensional, linearly separable data	Achieved >98% accuracy when combined with Random Forest and handcrafted features (Demilie & Deriba, 2022)	Lightweight, interpretable; best when paired in ensembles for complex query structures
Random Forest (RF)	Fast inference, high interpretability, ensemble resilience	Demonstrated high detection rates in hybrid frameworks with minimal latency impact (Demilie & Deriba, 2022)	Suitable for real-time systems; used as a first-pass filter in layered detection frameworks

Convolutional Neural Network (CNN)	Automated feature learning, captures local token anomalies	>95% accuracy on heterogeneous datasets using encoded SQL inputs (Falor et al., 2022)	Higher computational cost than RF; effective in confirmatory stages of hybrid models
RNN/LSTM and Hybrid Ensembles	Sequence modeling, memory retention for temporal query patterns	~99% detection accuracy; ensemble with Naïve Bayes showed superior false-positive trade-off (Tang et al., 2020; Ibrahim & Suryani, 2023)	Best for modeling long SQL sequences; requires GPU acceleration for real-time performance

- *Unsupervised and Semi-Supervised Anomaly Detection(e.g., Autoencoders, Isolation Forests).*

Autoencoder-based approaches have gained attention in SQL security research. Alghawazi et al. (2023) trained RNN-autoencoders on benign query sequences, flagging deviations as anomalies and subsequently classifying them using an LSTM classifier, achieving 94% accuracy and demonstrating strong generalization to unseen SQLi variants. Singh and Jang-Jaccard (2022) further highlighted that multiscale convolutional recurrent autoencoders capture both local and temporal query patterns, outperforming classical autoencoders when combined with Isolation Forest. Isolation Forest stands out for its efficiency in high-dimensional data. In web log anomaly detection, it frequently identifies anomalous SQL query features without requiring labeled attack data (MDPI, 2024). Integrating autoencoder latent representations before applying Isolation Forest has been shown to reduce false alarms, demonstrating a scalable, unsupervised detection pipeline suitable for real-time monitoring. Semi-supervised approaches, where autoencoder IDs potential anomalies and a downstream supervised classifier confirms threats, offer a cost-effective combination. The unsupervised layer maintains model adaptability to evolving query behavior, while the classifier ensures precise labeling, making this approach desirable for security teams working with limited labeled data.

- *Sequence-Aware Models for Query Stream Analysis (e.g., LSTM, GRU)*

Behavioral profiling of sequences of queries requires sequence-aware models to capture context and temporal anomalies. Recurrent neural architectures such as LSTM and GRU excel in this domain. In detecting SQL injection and other web-threat streams, Stiawan et al. (2023) demonstrated a composite LSTM-PCA model that reduced dimensionality with PCA before LSTM processing, achieving ~94% accuracy by modeling query structures over time as shown in figure 4. Their work highlights the importance of representing queries as structured time-series rather than independent events. Setiyaji and Ramli (2024) presented a CNN-BiLSTM hybrid model, initiating feature extraction with CNN layers and capturing sequential dependencies with BiLSTM, improving contextual awareness of query flows. Their system effectively identified SQLi attacks when queries appeared in specific order patterns rather than isolated anomalies. Similarly, Mohd Yazid Idris et al. (2023) integrated PCA and LSTM in an ensemble model for SQLi and XSS detection, achieving high performance (~96%) using ensemble voting mechanisms. This demonstrates the benefit of combining clustering, dimensionality reduction, and sequence learning.Ensembles combining GRU and LSTM, as explored by Pu et al. (2022), leverage strengths of both architectures: GRU’s computational efficiency and LSTM’s expressive memory capabilities. Their ensemble, combining stacked autoencoders with both RNN types, achieved solid detection rates (~89–95%) on benchmark datasets, proving effective for streaming environments. Sequence-aware models provide deep contextual analysis by learning user or session-level behavior patterns and highlighting deviations. Their strength is especially valuable for stealthy attacks that disguise malicious actions within normal-looking query streams.

- *Integration with Database Management and SIEMSystems*

To deploy machine learning-based SQL injection detection effectively, integration within DBMS and SIEM ecosystems is essential. Uetz et al. (2023) introduced AMIDES, an adaptive misuse detection extension for SIEM that supplements static rule-based detection with ML classifiers as shown in figure 3. Their system successfully caught evasive SQLi attempts that bypassed conventional SIEM alerts by learning normalness patterns and flagging anomalies, thereby reducing false negatives and improving response fidelity. Corporate research on ML-enhanced SIEM systems has identified key features necessary for integration: feature extraction pipelines, model management within SIEM, and connection to data lakes for large-volume query data analysis. These systems rely on continuous model retraining and feedback loops to stay effective as query patterns evolve. Scaling SIEM with data lakes, as proposed in 2024, addresses the challenge of handling large-scale DB audit logs and relational event streams. Integration allows centralized storage for model training, scalable feature extraction, and real-time classification close to the data source, reducing latency. Cloud-based Next-Gen SIEM platforms harness ML for feature normalization, UEBA, and automated incident response orchestration (Turkish Journal, 2023). These systems flexibly accommodate custom SQLi detection models as plug-in engines. By correlating ML-detected alerts with other sources (e.g., OS, network logs), they provide broader attack context—crucial for triage and escalation workflows. Integration requires tight coupling between ML models, DBMS audit log feeds, data lake platforms, and orchestration engines within SIEM. Achieving such synergy ensures behavioral profiling results are actionable and preventable in real-world enterprise environments, transforming reactive alerts into proactive threat mitigation pipelines.

Figure 3 portrays an advanced cybersecurity and data infrastructure ecosystem, symbolizing the seamless integration of machine learning-driven SQL injection detection within Database Management Systems (DBMS) and Security Information and Event Management (SIEM) platforms. At the center, a glowing lock signifies the core security objective—protecting data integrity and system access—while interconnected digital pathways illustrate real-time data flows from various sources such as logs, user behaviors, and relational query events. Embedded icons for analytics, automation, user entity behavior analytics (UEBA), and system orchestration reflect how modern SIEM systems leverage ML models to automate threat detection and incident response. The array of dashboards, charts, and connected devices represents continuous monitoring and feature extraction pipelines, essential for training and retraining detection models against evolving SQL attack patterns. This visual emphasizes the need for data lakes to store massive DB audit logs and support scalable model inference, and showcases how correlation of ML-detected anomalies with network and OS logs enhances situational awareness. The image encapsulates the synergy of cloud-native SIEM platforms, model management frameworks, and database telemetry in creating a proactive and intelligent security architecture.

Fig 3 Picture of Machine Learning-Driven Integration of SQL Injection Detection into DBMS and SIEM Ecosystems (Suretysystems, 2025).

Figure 4 is organized into three main branches: *Core Neural Architectures, Hybrid and Ensemble Models, and Application and Detection Capabilities*. The first branch outlines foundational recurrent models such as LSTM, which captures long-term dependencies in query sequences, GRU, known for its computational efficiency, and BiLSTM, which processes query streams bidirectionally for enhanced contextual learning. The second branch details hybrid and ensemble approaches that combine these models with feature extraction or dimensionality reduction techniques. For instance, LSTM integrated with PCA effectively models structured query timelines, while CNN-BiLSTM hybrids extract deep features and model sequence flows, particularly effective in identifying SQL injection attacks with pattern dependencies. Ensemble systems that combine PCA, clustering, and LSTM use voting mechanisms to improve accuracy, while advanced combinations of GRU and LSTM with autoencoders enhance robustness in streaming environments. The third branch focuses on the application of these models in real-world detection, emphasizing behavioral profiling by learning normal user or session-level patterns, enabling the detection of stealthy attacks that follow subtle sequences. These models excel in temporal anomaly detection and offer high accuracy (89–96%) for identifying both overt and covert web threats, making them valuable tools in real-time cybersecurity for SQL-based systems.

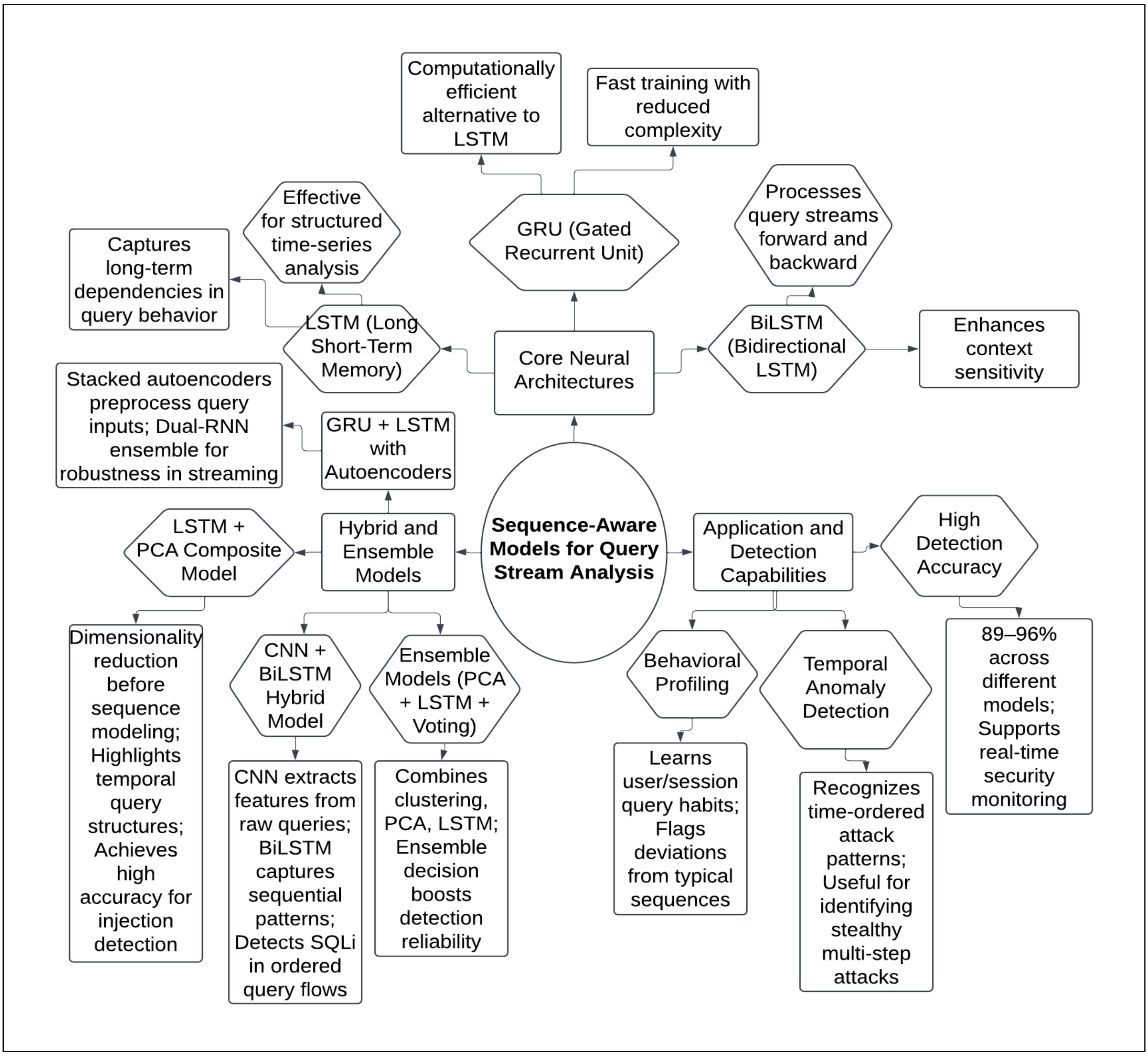
Sequence-Aware Models for Query Stream Analysis
Core Neural Architectures
GRU (Gated Recurrent Unit)
Computationally efficient alternative to LSTM
Fast training with reduced complexity
LSTM (Long Short-Term Memory)
Effective for structured time-series analysis
Captures long-term dependencies in query behavior
BiLSTM (Bidirectional LSTM)
Processes query streams forward and backward
Enhances context sensitivity
Hybrid and Ensemble Models
GRU + LSTM with Autoencoders
Stacked autoencoders preprocess query inputs; Dual-RNN ensemble for robustness in streaming
LSTM + PCA Composite Model
Dimensionality reduction before sequence modeling; Highlights temporal query structures; Achieves high accuracy for injection detection
CNN + BiLSTM Hybrid Model
CNN extracts features from raw queries; BiLSTM captures sequential patterns; Detects SQLi in ordered query flows
Ensemble Models (PCA + LSTM + Voting)
Combines clustering, PCA, LSTM; Ensemble decision boosts detection reliability
Application and Detection Capabilities
High Detection Accuracy
89–96% across different models; Supports real-time security monitoring
Behavioral Profiling
Learns user/session query habits; Flags deviations from typical sequences
Temporal Anomaly Detection
Recognizes time-ordered attack patterns; Useful for identifying stealthy multi-step attacks

Fig 4 Diagram Illustration of Structured Overview of Sequence-Aware Deep Learning Models and their Applications in Temporal Anomaly Detection for SQL Query Streams

V. CHALLENGES, RESEARCH GAPS

➢ *Evasion Techniques and Adversarial Query Generation*

One of the critical challenges in machine learning-based SQL injection detection is the emergence of evasion techniques, particularly adversarial query generation. Attackers increasingly craft queries that closely resemble legitimate SQL commands to bypass anomaly detection models. These adversarial queries are designed by modifying known malicious payloads through obfuscation, encoding, and query structure manipulation without altering their malicious intent. Techniques such as SQL comment injection, use of tautologies, whitespace variation, and nested subqueries can significantly reduce detection rates in models trained on traditional or static patterns. Furthermore, attackers may employ query mutation strategies to test the boundaries of deployed detection systems, thereby identifying exploitable blind spots. Machine learning models that are not robust to such perturbations may exhibit high false negatives, allowing exfiltration attempts to succeed undetected. As the sophistication of evasion tactics increases, it becomes imperative for detection systems to incorporate robust training methods that anticipate a wide range of adversarial behaviors. This includes adversarial training, data augmentation, and continuous learning from near-miss detection failures. Developing resilient models that generalize across diverse adversarial strategies remains a central goal for enhancing the reliability and effectiveness of behavioral profiling systems in safeguarding relational databases against stealthy attacks.

➢ *Concept Drift and Model Adaptability*

Concept drift refers to the gradual or abrupt change in the statistical properties of SQL query patterns over time, which can significantly degrade the performance of static machine learning models. In dynamic environments where user behaviors, access frequencies, and system usage evolve, models trained on historical data may become obsolete, leading to increased false positives or negatives. For instance, a sudden shift in query frequency during an organizational restructuring or policy update may be misclassified as anomalous, even though it reflects legitimate operational changes. Conversely, an attacker who mimics legitimate access patterns could remain undetected if the model has not been updated to capture emerging attack vectors. Addressing concept drift requires implementing adaptive learning strategies such as online learning, window-based retraining, and periodic model updates using recent behavioral data. Moreover, drift detection mechanisms should be integrated into the monitoring system to flag potential shifts in query distributions that may impact model accuracy. Building flexible, context-aware models that can recalibrate to evolving usage patterns without compromising detection precision is essential. Balancing adaptability with system stability ensures the long-term effectiveness of detection mechanisms in real-world, high-variability database environments where static assumptions are no longer viable.

➢ *Explainability and Interpretability in Security Contexts*

Incorporating explainability and interpretability into SQL injection detection systems is vital for gaining user trust, facilitating decision-making, and ensuring compliance with regulatory standards. Security analysts and database administrators must understand why a specific query is flagged as suspicious, particularly in environments with high accountability requirements. Black-box models, while effective in classification accuracy, often lack transparency, making it difficult to trace the rationale behind their decisions. This opacity can hinder incident response, root cause analysis, and model debugging efforts. By contrast, interpretable models or those enhanced with explainable AI (XAI) techniques provide valuable insights into feature importance, decision boundaries, and behavioral patterns that triggered the alert. Visualizations of query sequences, attention weights, or anomaly scores can help stakeholders validate system outputs and fine-tune detection thresholds. Additionally, explainability aids in identifying model biases and gaps, especially when distinguishing between benign outliers and malicious queries. In mission-critical systems, explainability becomes a non-negotiable requirement, enabling human oversight and fostering collaboration between machine intelligence and human judgment. Therefore, integrating explainability mechanisms into detection pipelines not only enhances trust but also improves system transparency, accountability, and effectiveness in managing database security threats in complex enterprise environments.

➢ *Federated Learning and Privacy-Preserving Detection*

Federated learning presents a promising approach to SQL injection and data exfiltration detection by enabling collaborative model training across decentralized environments without sharing raw data. This technique is particularly relevant in privacy-sensitive contexts such as healthcare, finance, and government institutions, where cross-organizational data pooling is constrained by regulations. By training models locally on each organization's SQL logs and sharing only encrypted model updates, federated learning facilitates the creation of robust, generalized behavioral profiling systems. This decentralized model not only enhances detection performance across diverse query distributions but also minimizes the risk of data leakage. However, implementing federated learning introduces challenges such as ensuring update integrity, handling heterogeneous data distributions, and addressing communication overhead. Techniques like differential privacy, secure multiparty computation, and homomorphic encryption are often employed to further protect sensitive query data during transmission. Despite these challenges, federated architectures offer scalability, adaptability, and data sovereignty while preserving the collective intelligence needed to detect sophisticated SQL-based attacks. As cyber threats evolve, integrating privacy-preserving machine learning frameworks becomes a strategic imperative for organizations seeking to maintain strong security postures without compromising on confidentiality or compliance mandates.

RECOMMENDATIONS FOR FUTURE RESEARCH

To advance machine learning-based detection of SQL injection and data exfiltration, future research should focus on developing adaptive, resilient, and transparent systems. First, improving model robustness against adversarial queries through techniques like adversarial training and ensemble learning can help mitigate evasion attempts. Second, the incorporation of online learning and drift detection methods is crucial to address concept drift and ensure that models remain relevant in changing operational environments. Third, enhancing explainability through interpretable architectures or post-hoc explanation tools will foster greater trust and usability among security practitioners. Additionally, integrating contextual signals such as user behavior history, device metadata, and access location can enrich feature sets and improve model precision. Federated learning and edge-based detection systems offer scalable and privacy-aware solutions that deserve further exploration, particularly in multi-tenant and cloud-based infrastructures. Benchmarking datasets that reflect real-world adversarial conditions should be developed to standardize evaluation and support reproducibility. Finally, a cross-disciplinary approach involving security experts, data scientists, and legal professionals is essential to design ethical, compliant, and effective detection systems. These research directions can significantly contribute to fortifying relational databases against evolving cyber threats in both enterprise and critical infrastructure domains.

REFERENCES

[1]. Adebayo, A. B., & Al-Dubai, A. Y. (2020). Leveraging machine learning for secure database access: A behavioral profiling approach. Information Systems, 92, 101521. https://doi.org/10.1016/j.is.2020.101521

[2]. Adekunle, F., & Zhang, T. (2024). Event-driven frameworks for real-time intrusion detection in SQL-intensive applications. ACM Transactions on Privacy and Security, 27(1), 1–25. https://doi.org/10.1145/3591230

[3]. Aggarwal, C. C., & Sathe, S. (2020). Theoretical foundations and algorithms for outlier ensembles. ACM Computing Surveys, 53(6), 1–36. https://doi.org/10.1145/3398037

[4]. Akhtar, S., & Farooq, M. (2020). Real-time detection of SQL anomalies using deep autoencoders and stream processors. Journal of Network and Computer Applications, 157, 102591. https://doi.org/10.1016/j.jnca.2020.102591

[5]. Alghawazi, et al. (2023). SQL injection detection using RNN autoencoder and LSTM. arXiv preprint.

[6]. Alomari, M., & Wang, J. (2021). Deep structural analysis of SQL queries for anomaly detection. IEEE Transactions on Dependable and Secure Computing.

[7]. Alshammari, R., Alwan, Z., & Alzain, M. A. (2021). Advanced SQL injection attack detection using behavioral features and statistical analysis. Computers, Materials & Continua, 66(2), 1631–1646. https://doi.org/10.32604/cmc.2021.013267

[8]. Altwaijry, H., & El-Alfy, E. M. (2021). Evaluation of rule-based intrusion detection systems for SQLi vulnerabilities. IEEE Transactions on Dependable and Secure Computing, 18(4), 1549–1562. https://doi.org/10.1109/TDSC.2019.2896769

[9]. Awotiwon, B. O., Enyejo, J. O., Owolabi, F. R. A., Babalola, I. N. O., & Olola, T. M. (2024). Addressing Supply Chain Inefficiencies to Enhance Competitive Advantage in Low-Cost Carriers (LCCs) through Risk Identification and Benchmarking Applied to Air Australasia's Operational Model. World Journal of Advanced Research and Reviews, 2024, 23(03), 355–370. https://wjarr.com/content/addressing-supply-chain-inefficiencies-enhance-competitive-advantage-low-cost-carriers-lccs

[10]. Banerjee, A., & Roy, A. (2022). Intelligent profiling of SQL attack surfaces: A review of recent progress. Information & Computer Security, 30(4), 597–617. https://doi.org/10.1108/ICS-11-2021-0146

[11]. Banerjee, A., & Singh, R. (2022). Metric-aware performance evaluation in SQL-based threat detection systems. Computers & Security, 115, 102620. https://doi.org/10.1016/j.cose.2022.102620

[12]. Bashir, F., Rauf, A., & Shahid, A. R. (2023). A hybrid AI-based framework for behavioral anomaly detection in SQL transactions. Neural Computing and Applications, 35, 11571–11585. https://doi.org/10.1007/s00521-023-08159-2

[13]. Bashiru, O., Ochem, C., Enyejo, L. A., Manuel, H. N. N., & Adeoye, T. O. (2024). The crucial role of renewable energy in achieving the sustainable development goals for cleaner energy. *Global Journal of Engineering and Technology Advances*, 19(03), 011-036. https://doi.org/10.30574/gjeta.2024.19.3.0099

[14]. Bianchi, F., Grana, M., & Rossi, C. (2021). Graph neural networks for anomaly detection in SQL query graphs. IEEE Transactions on Neural Networks and Learning Systems.

[15]. Chatterjee, M., Gupta, S., & Bera, P. (2021). Profiling SQL behavior using deep learning for injection attack detection. Computers & Security, 105, 102240. https://doi.org/10.1016/j.cose.2021.102240

[16]. Chen, D., & Zhao, Q. (2021). Low-latency SQL injection detection in distributed databases using recurrent neural networks. Future Generation Computer Systems, 117, 71–84. https://doi.org/10.1016/j.future.2020.11.014

[17]. Chen, H., Yu, L., & Zhang, Y. (2021). Static and signature-based detection of SQL injection: A retrospective and limitations. Journal of Information Security and Applications, 59, 102836. https://doi.org/10.1016/j.jisa.2021.102836

[18]. Chen, L., Rao, Q., & Zhao, Y. (2020). Temporal sequence modeling of SQL queries for anomaly detection. IEEE Transactions on Information Forensics and Security.

[19]. Corporal Machine Learning Algorithms in SIEM Systems for Enhanced Detection. (2023). ResearchGate Conference Paper.

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

[21]. Falor, A., Hirani, M., Vedant, H., Mehta, P., & Krishnan, D. (2022). A deep learning approach for detection of SQL injection attacks using convolutional neural networks. In Proceedings of Data Analytics and Management: ICDAM 2021 (Vol. 2, pp. 293–304).

[22]. Garcia, R., & Watts, B. (2021). Role-aware machine learning for insider threat detection. ACM Transactions on Privacy and Security.

[23]. Geeksforgeeks, (2024). Supervised and Unsupervised learning, https://www.geeksforgeeks.org/machine-learning/supervised-unsupervised-learning/

[24]. Godwins, O. P., David-Olusa, A., Ijiga, A. C., Olola, T. M., & Abdallah, S. (2024). The role of renewable and cleaner energy in achieving sustainable development goals and enhancing nutritional outcomes: Addressing malnutrition, food security, and dietary quality. World Journal of Biology Pharmacy and Health Sciences, 2024, 19(01), 118–141. https://wjbphs.com/sites/default/files/WJBPHS-2024-0408.pdf

[25]. Godwins, O. P., Ochagwuba, E., Idoko, I. P., Akpa, F. A., Olajide, F. I., & Olatunde, T. I. (2024). Comparative analysis of disaster management strategies and their impact on nutrition outcomes in the USA and Nigeria. *Business and Economics in Developing Countries (BEDC)*, 2(2), 34-42. http://doi.org/10.26480/bedc.02.2024.34.42

[26]. Gomez, P., Sánchez, F., & Molina, J. (2022). Context-augmented profiling of database queries. Journal of Big Data Security.

[27]. Haque, A., & Soliman, H. (2025). A transformer-based autoencoder with Isolation Forest and XGBoost for malfunction and intrusion detection

in wireless sensor networks. Future Internet, 17(4), 164.

[28]. Hussain, S., Ahmed, T., & Nazir, U. (2022). User-role activity profiling in relational databases. Information Sciences.

[29]. Ibokette, A. I., Aboi, E. J., Ijiga, A. C., Ugbane, S. I., Odeyemi, M. O., & Umama, E. E. (2024). The impacts of curbside feedback mechanisms on recycling performance of households in the United States. *World Journal of Biology Pharmacy and Health Sciences*, 17(2), 366-386.

[30]. Ibokette., A. I. Ogundare, T. O., Danquah, E. O., Anyebe, A. P., Agaba, J. A., & Olola, T. M. (2024). The impacts of emotional intelligence and IOT on operational efficiency in manufacturing: A cross-cultural analysis of Nigeria and the US. Computer Science & IT Research Journal P-ISSN: 2709-0043, E-ISSN: 2709-0051. DOI: 10.51594/csitrj.v5i8.1464

[31]. Ibokette., A. I. Ogundare, T. O., Danquah, E. O., Anyebe, A. P., Agaba, J. A., & Agaba, J. A. (2024). Optimizing maritime communication networks with virtualization, containerization and IoT to address scalability and real – time data processing challenges in vessel – to –shore communication. Global Journal of Engineering and Technology Advances, 2024, 20(02), 135–174. https://gjeta.com/sites/default/files/GJETA-2024-0156.pdf

[32]. Ibrahim, M. M., & Suryani, V. (2023). Classification of SQL injection attacks using ensemble learning SVM and Naïve Bayes. In Proceedings of 2023 International Conference on Data Science and Its Applications (ICODSA) (pp. 230–236).

[33]. Idoko P. I., Igbede, M. A., Manuel, H. N. N., Ijiga, A. C., Akpa, F. A., & Ukaegbu, C. (2024). Assessing the impact of wheat varieties and processing methods on diabetes risk: A systematic review. World Journal of Biology Pharmacy and Health Sciences, 2024, 18(02), 260–277. https://wjbphs.com/sites/default/files/WJBPHS-2024-0286.pdf

[34]. Idoko, I. P., Igbede, M. A., Manuel, H. N. N., Adeoye, T. O., Akpa, F. A., & Ukaegbu, C. (2024). Big data and AI in employment: The dual challenge of workforce replacement and protecting customer privacy in biometric data usage. *Global Journal of Engineering and Technology Advances*, 19(02), 089-106. https://doi.org/10.30574/gjeta.2024.19.2.0080

[35]. Idoko, I. P., Ijiga, O. M., Agbo, D. O., Abutu, E. P., Ezebuka, C. I., & Umama, E. E. (2024). Comparative analysis of Internet of Things (IOT) implementation: A case study of Ghana and the USA-vision, architectural elements, and future directions. *World Journal of Advanced Engineering Technology and Sciences*, 11(1), 180-199.

[36]. Idoko, I. P., Ijiga, O. M., Akoh, O., Agbo, D. O., Ugbane, S. I., & Umama, E. E. (2024). Empowering sustainable power generation: The vital role of power electronics in California's renewable energy transformation. *World Journal of Advanced Engineering Technology and Sciences*, 11(1), 274-293.

[37]. Idoko, I. P., Ijiga, O. M., Enyejo, L. A., Akoh, O., & Ileanaju, S. (2024). Harmonizing the voices of AI: Exploring generative music models, voice cloning, and voice transfer for creative expression.

[38]. Idoko, I. P., Ijiga, O. M., Enyejo, L. A., Ugbane, S. I., Akoh, O., & Odeyemi, M. O. (2024). Exploring the potential of Elon Musk's proposed quantum AI: A comprehensive analysis and implications. *Global Journal of Engineering and Technology Advances*, 18(3), 048-065.

[39]. Igba, E., Danquah, E. O., Ukpoju, E. A., Obasa, J., Olola, T. M., & Enyejo, J. O. (2024). Use of Building Information Modeling (BIM) to Improve Construction Management in the USA. World Journal of Advanced Research and Reviews, 2024, 23(03), 1799–1813. https://wjarr.com/content/use-building-information-modeling-bim-improve-construction-management-usa

[40]. Ijiga, O. M., Idoko, I. P., Ebiega, G. I., Olajide, F. I., Olatunde, T. I., & Ukaegbu, C. (2024). Harnessing adversarial machine learning for advanced threat detection: AI-driven strategies in cybersecurity risk assessment and fraud prevention. Open Access Research Journals. Volume 13, Issue. https://doi.org/10.53022/oarjst.2024.11.1.0060 I

[41]. Integrating SIEM with Data Lakes and AI: Enhancing Threat Detection and Response. (2024). ResearchGate Paper.

[42]. Iqbal, W., & Naeem, M. (2024). Behavior-aware database intrusion detection: Trends and gaps. Journal of Cybersecurity and Privacy, 4(2), 207–230. https://doi.org/10.3390/jcp4020013

[43]. Kamble, M. Y., Wankhade, K., & Barde, B. (2020). Comparative study on SQL injection detection using rule-based methods. Procedia Computer Science, 172, 641–648. https://doi.org/10.1016/j.procs.2020.05.088

[44]. Kaushik, A., & Joshi, R. (2020). Structured feature representation of SQL queries for anomaly detection. Future Generation Computer Systems, 111, 504–517. https://doi.org/10.1016/j.future.2020.05.031

[45]. Khan, S., & Ahmad, R. (2022). Grammar-based normalization of SQL statements for effective injection detection. International Journal of Information Security.

[46]. Lee, H., Kim, D., & Park, S. (2023). Adaptive SQL query normalization with machine learning. ACM Transactions on Database Systems.

[47]. Li, Y., Yang, T., & Jiang, M. (2023). Adaptive anomaly detection in streaming data using hybrid neural models. Journal of Artificial Intelligence Research, 76, 231–257. https://doi.org/10.1613/jair.1.13564

[48]. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2021). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 15(3), 1–28. https://doi.org/10.1145/3458446

[49]. Liu, H., Guo, Q., & Li, S. (2023). Systematic review of injection vulnerabilities and data leakage in cloud-based databases. Future Generation Computer Systems, 145, 259–272. https://doi.org/10.1016/j.future.2023.03.018

[50]. Liu, Y., Zhao, H., & Fan, Y. (2022). Anomaly-based detection of SQLi using LSTM sequence learning. Expert Systems with Applications, 193, 116385. https://doi.org/10.1016/j.eswa.2021.116385

[51]. Lu, Y., Chen, X., & Fang, J. (2022). Representing relational queries as graphs for intrusion detection. Applied Soft Computing.

[52]. Mehrotra, R., & Thakur, R. (2023). Extraction of behavioral features from SQL logs using unsupervised deep encoders. Pattern Recognition Letters, 169, 30–38. https://doi.org/10.1016/j.patrec.2023.01.015

[53]. Mohd Yazid Idris et al. (2023). An improved LSTM-PCA ensemble classifier for SQL injection and XSS detection. UTM e-prints.

[54]. Onuh, J. E., Idoko, I. P., Igbede, M. A., Olajide, F. I., Ukaegbu, C., & Olatunde, T. I. (2024). Harnessing synergy between biomedical and electrical engineering: A comparative analysis of healthcare advancement in Nigeria and the USA. *World Journal of Advanced Engineering Technology and Sciences*, 11(2), 628-649.

[55]. Ouyang, X., Lin, W., & Zhang, H. (2023). Online anomaly detection for relational databases using attention-based streaming models. Neurocomputing, 522, 87–101. https://doi.org/10.1016/j.neucom.2022.12.072

[56]. Owolabi, F. R. A., Enyejo, J. O., Babalola, I. N. O., & Olola, T. M. (2024). Overcoming engagement shortfalls and financial constraints in Small and Medium Enterprises (SMES) social media advertising through cost-effective Instagram strategies in Lagos and New York City. International Journal of Management & Entrepreneurship Research P-ISSN: 2664-3588, E-ISSN: 2664-3596. DOI: 10.51594/ijmer.v6i8.1462

[57]. Patel, D., Sharma, K., & Mehta, S. (2023). Supervised modeling of user-based SQL activity for anomaly detection. Computers & Security.

[58]. Pu, et al. (2022). Detecting zero-day web attacks with an ensemble of LSTM, GRU, and stacked autoencoders. Computers, 14(6), 205.

[59]. Qureshi, M. A., & Khan, S. (2023). Detecting data exfiltration from relational queries: A machine learning perspective. IEEE Access, 11, 74501–74514. https://doi.org/10.1109/ACCESS.2023.3282905

[60]. Rahman, M., Ahmed, F., & Miah, M. S. (2022). The weakness of black-box SQL injection scanners in modern web applications. Security and Privacy, 5(2), e205. https://doi.org/10.1002/spy2.205

[61]. Sabottke, C. F., & Abraham, J. (2022). Survey of anomaly detection for relational data using supervised and unsupervised learning. IEEE Transactions on Knowledge and Data Engineering, 34(4), 1527–1540. https://doi.org/10.1109/TKDE.2021.3053062

[62]. Sajjad, A., Nasir, Q., & Shafique, M. (2022). A taxonomy and survey of SQL injection detection and prevention techniques. Journal of Network and Computer Applications, 195, 103217. https://doi.org/10.1016/j.jnca.2021.103217

[63]. Setiyaji, A., & Ramli, K. (2024). A technique utilizing CNN for identification of SQL injection attacks. 2024 ICSINTESA Conference Proceedings.

[64]. Sharma, P., & Desai, R. (2023). Query–table interaction graphs for exfiltration detection. Knowledge-Based Systems.

[65]. Sharma, R., Dey, L., & Kumar, S. (2022). Semantic embedding of structured query language statements for intrusion detection. Knowledge-Based Systems, 240, 108025. https://doi.org/10.1016/j.knosys.2022.108025

[66]. Singh, A., & Jang-Jaccard, J. (2022). Autoencoder-based unsupervised intrusion detection using multi-scale convolutional recurrent networks. arXiv preprint.

[67]. Singh, A., & Kumar, P. (2023). LSTM-based behavioral profiling of SQL query streams. Future Generation Computer Systems.

[68]. Stiawan, D., et al. (2023). LSTM+PCA composite model to detect SQL injection and XSS. Scientific Reports.

[69]. Suretysystems, (2025). Enhance SAP System Security: Top Strategies for SAP SIEM Integration, https://www.suretysystems.com/insights/enhance-sap-system-security-top-strategies-for-sap-siem-integration/

[70]. Tang, L., et al. (2020). Attack detection in network flow data using LSTM for SQL injection. International Journal of Applied Engineering Research, 15(6), 569–580.

[71]. The Future of SIEM in a Machine Learning-Driven Cybersecurity. (2023). Turkish Journal of Computer and Mathematics Education.

[72]. Uetz, R., Herzog, M., Hackländer, L., Schwarz, S., & Henze, M. (2023). You cannot escape me: detecting evasions of SIEM rules in enterprise networks. arXiv preprint.

[73]. Vakharia, M., & Patel, V. (2020). Benchmarking datasets for anomaly detection in SQL injection scenarios. Journal of Cybersecurity, 6(1), taaa011. https://doi.org/10.1093/cybsec/taaa011

[74]. Wang, T., & Li, M. (2024). Context-aware detection of data exfiltration via query patterns. Computers & Security.

[75]. Web Traffic Anomaly Detection Using Isolation Forest. (2024). MDPI International Journal of Data, 11(4), 83.

[76]. Xu, J., & Tan, Z. (2023). A framework for benchmark dataset creation in SQL-based attack detection using graph learning. IEEE Access, 11, 48526–48538. https://doi.org/10.1109/ACCESS.2023.3265270

[77]. Yoon, J., Park, E., & Han, S. (2024). Hybrid role-based anomaly detection in enterprise queries. IEEE Transactions on Software Engineering.

[78]. Zhang, J., & Yu, W. (2021). Feature transformation for SQL injection detection using query dependency graphs. Information Sciences, 569, 1–18. https://doi.org/10.1016/j.ins.2021.02.005

[79]. Zhang, M., & Wang, X. (2021). Comparative analysis of evaluation metrics for SQL anomaly classifiers. Expert Systems with Applications, 185, 115550. https://doi.org/10.1016/j.eswa.2021.115550

[80]. Zhang, Y., Xu, L., & Li, X. (2024). Structural feature extraction for SQL anomaly detection. Journal of Computer Security.

[81]. Zheng, Y., Xie, T., & Xu, D. (2020). From SQL injection to data exfiltration: Challenges and countermeasures. IEEE Access, 8, 172495–172508. https://doi.org/10.1109/ACCESS.2020.3025084

[82]. Zhou, W., Li, Z., & Xu, H. (2024). Graph stream learning of SQL behaviors. Information Sciences.

[83]. Zhou, Y., Xu, Y., & Wang, C. (2021). Machine learning for database security: A systematic review. ACM Computing Surveys, 54(9), 1–36. https://doi.org/10.1145/3457600