ISSN 2278-3091
Volume 15, No.3, May – June 2026
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse011532026.pdf
https://doi.org/10.30534/ijatcse/2026/011532026

Structured Query Language Injection Detection Using Natural Language Processing Approaches: A Meta-Analytic Review

Rachael N. Ndung'u[1], Samuel Maina Muriuki[2]
[1]Murang'a University of Technology, Kenya, rndungu@mut.ac.ke
[2]Zetech University, Kenya, samicho4@gmail.com

Received Date: April 18, 2026 Accepted Date: May 28, 2026 Published Date: June 06, 2026

ABSTRACT

Structured Query Language Injection (SQLi) attacks remain one of the most persistent threats to web applications, enabling unauthorized access and manipulation of sensitive data. This meta-analysis synthesizes findings from 28 empirical studies (2017–2025) on Natural Language Processing (NLP)-based SQLi detection, following PRISMA 2020 guidelines and PICOS criteria. NLP approaches were compared against non-NLP methods using random-effects meta-analysis, revealing a pooled F1-score advantage of +0.18 (95% CI: 0.14–0.22, $p < 0.001$). Transformer-based models (e.g., BERT variants) achieved the highest accuracies (up to 99.7%), while deep learning models (CNN-LSTM, Text CNN) offered a practical balance between accuracy and computational efficiency. Beyond performance metrics, this study explicitly maps its methodology to the Research Onion framework, clarifying coherence between philosophy, approach, and methods. Results highlight NLP's semantic strengths but also expose critical gaps: reliance on outdated datasets, underreporting of computational trade-offs (latency, throughput, and memory), and limited adversarial robustness testing. To bridge laboratory success with production resilience, we propose a hybrid deployment framework: transformer-based models for offline training, lightweight ML or CNN-LSTM for real-time inference, and adaptive retraining pipelines incorporating adversarial examples. This work advances both theory and practice by demonstrating NLP's superiority in SQLi detection while cautioning against overstated "near-perfect" accuracies in controlled settings. It calls for standardized, real-world datasets, comprehensive benchmarking, and adversarial evaluation protocols to ensure robustness in diverse deployment environments.

Key words: Cybersecurity, Deep Learning, NLP, SQL Injection.

1. INTRODUCTION

This section introduces the persistent challenge of SQL injection attacks, outlines the role of Natural Language Processing (NLP) in cybersecurity, and frames the research objectives that guide the meta-analysis.

1.1 The SQL Injection Problem

Structured Query Language Injection (SQLi) attacks represent a persistent and severe threat to web applications, exploiting vulnerabilities in user input validation to inject malicious SQL code. These attacks enable unauthorized access to sensitive data, such as user credentials, financial records, or intellectual property, and can lead to data manipulation, deletion, or system compromise [1]. Consistently ranked among the top vulnerabilities by the Open Web Application Security Project (OWASP), SQLi attacks have caused significant breaches, with notable cases like the 2017 Equifax incident exposing data of millions due to unpatched vulnerabilities [2]. SQLi variants, including tautology-based, union-based, blind, time-based, and polymorphic attacks, exploit dynamic query construction and inadequate sanitization, making them challenging to detect with traditional methods [3]. The proliferation of web-based platforms, including e-commerce, cloud services, and Internet of Things (IoT) ecosystems, has expanded the attack surface, necessitating advanced, adaptive detection mechanisms capable of real-time response [4].

Traditional countermeasures, such as rule-based filtering, parameterized queries, and web application firewalls (WAFs), rely on predefined patterns or signatures to identify malicious inputs [5]. However, these approaches struggle with sophisticated attacks that obfuscate code or exploit zero-day vulnerabilities, leading to high false negative rates. The dynamic nature of modern web applications, coupled with the increasing complexity of attack vectors, underscores the need

for intelligent systems that can adapt to evolving threats without requiring constant manual updates.

The use of NLP in detecting SQL injection attacks involves modelling input strings as linguistic sequences and analyzing them for anomalous patterns. Techniques such as tokenization, n-gram modelling, and embedding methods (e.g., Word2Vec, GloVe) have enabled security systems to capture subtle semantic differences between benign and malicious queries [6]. More advanced methods leverage contextual language models like BERT and GPT, which allow for deeper syntactic and semantic analysis of input behaviour. Unlike static, rule-based detection systems, NLP-based approaches are capable of learning contextual cues and generalizing beyond known attack patterns. For example, models based on recurrent neural networks (RNNs), long short-term memory (LSTM), and transformers have been shown to detect obfuscated or zero-day SQLi attacks by learning sequential and contextual dependencies in user inputs. Moreover, hybrid systems that combine NLP preprocessing with traditional classifiers—such as support vector machines (SVMs) or convolutional neural networks (CNNs)—have demonstrated strong performance across benchmark datasets [7]. These models not only identify direct injection attempts but also recognize semantic anomalies in input structure that would otherwise evade traditional filters.

The growing availability of labelled datasets for SQLi detection, alongside the open-source development of NLP libraries such as spaCy, NLTK, and Hugging Face Transformers, has further encouraged the adoption of NLP methods in security applications. As a result, the academic community has seen a significant increase in publications proposing NLP-driven detection mechanisms for SQL injection, highlighting both the effectiveness and diversity of these approaches. This proliferation of research forms the foundation for the current meta-analysis, which seeks to evaluate and synthesize these findings to identify the most effective NLP techniques for SQLi detection.

1.2 The NLP Paradigm in Cybersecurity

Natural Language Processing (NLP) has emerged as a transformative approach in cybersecurity by treating SQL queries as textual data, enabling semantic and syntactic analysis to detect malicious patterns [8]. Unlike traditional methods that focus on syntactic matching, NLP leverages techniques such as word embedding (e.g., Word2Vec, BERT), sequence modelling (e.g., LSTM), and attention mechanisms to capture contextual relationships within queries [9]. This capability is particularly effective for detecting complex SQLi variants, such as blind or time-based attacks, which rely on subtle semantic deviations rather than overt syntactic markers [10]. NLP-based systems excel in generalizing across diverse attack patterns, reducing reliance on static rules and enabling adaptation to new threats in real-time environments [11]. For instance, transformer-based models like BERT have demonstrated superior performance in extracting contextual features, achieving accuracies above 99% in recent studies [12]. The integration of NLP with machine learning (ML) and deep learning (DL) frameworks has further enhanced detection by combining statistical pattern recognition with automated feature extraction, making it a cornerstone of modern cybersecurity research [13].

1.3 The Need for a Meta-Analysis

Despite the growing adoption of NLP in SQLi detection, the literature lacks a comprehensive meta-analysis comparing NLP-based systems against non-NLP methods (e.g., rule-based, statistical approaches). Existing reviews, such as those by [6], provide qualitative overviews but lack quantitative synthesis, while prior reviews have provided useful overviews, our synthesis deliberately emphasizes primary empirical studies to avoid over-reliance on secondary sources. For example, [15] and [16] demonstrate transformer-based models achieving near-perfect detection rates, while [5] highlight the utility of LSTM-autoencoders in capturing obfuscated SQLi patterns. Beyond academia, industry reports such as OWASP Top Ten [30] and Verizon's Data Breach Investigations Report [31] underscore the persistent prevalence of SQLi in real-world breaches, reinforcing the practical relevance of our meta-analysis. This integration of peer-reviewed studies with industry evidence strengthens the grounding of our work and highlights gaps in adversarial robustness testing, dataset standardization, and deployment metrics that remain underexplored.

1.4 Research Objectives and Questions

This meta-analysis aims to evaluate the effectiveness of NLP-based systems for SQLi detection, focusing on their performance, optimal approaches, influencing factors, and reliability. The research questions were:

i. Do NLP-based systems outperform non-NLP-based systems in SQLi detection?

ii. Which NLP approaches perform best in SQLi detection?

iii. What factors affect detection performance?

iv. How reliable and consistent are findings across existing studies?

These objectives guided the synthesis of evidence to inform the development of robust, adaptive SQLi detection systems suitable for real-world deployment.

2. METHODOLOGY

This section details the methodological framework of the study, including search strategy, eligibility criteria, study selection, data extraction, and risk of bias assessment, ensuring transparency and reproducibility

2.1 Search Strategy

The search strategy was designed to comprehensively retrieve peer-reviewed literature on NLP-based SQLi detection from 2017 to 2025, ensuring coverage of recent advancements in ML and DL. Primary databases included IEEE Xplore, Scopus, ACM Digital Library, Google Scholar, and arXiv, selected for their extensive coverage of computer science and cybersecurity research. Search terms combined Boolean operators: ("SQL injection" OR "SQLi") AND ("natural language processing" OR "NLP") AND ("machine learning" OR "deep learning" OR "ML" OR "DL") AND ("intrusion detection" OR "cybersecurity"). Filters restricted results to peer-reviewed articles, conference proceedings, and technical reports published between January 2017 and August 2025. Advanced search techniques, for preprints and wildcard operators (e.g., "detect*"), were employed to enhance recall. Manual hand-searching of reference lists from key reviews [6] and forward citation tracking via Google Scholar identified additional studies. Publication alerts were set to capture new articles up to the review date. The initial search yielded 1582 records, with duplicates (n=242) removed using Zotero software, resulting in 1340 unique records for screening.

2.2 Eligibility Criteria (PICOS)

The eligibility criteria for study inclusion were defined using the PICOS framework, ensuring methodological rigor and consistency. The population comprised web applications or systems vulnerable to SQL injection attacks, including diverse environments such as IoT, cloud, and containerized platforms. The intervention focused on NLP-based detection systems, encompassing both machine learning approaches (e.g., Support Vector Machines and Random Forests with TF-IDF features) and deep learning architectures (e.g., CNN, LSTM, and BERT). For comparison, studies were required to evaluate these NLP methods against non-NLP approaches, including rule-based, statistical, or signature-based techniques. The outcomes of interest were quantitative performance metrics such as accuracy, precision, recall, F1-score, and false positive/negative rates, alongside computational measures like response time.

Finally, the study design was restricted to empirical investigations with experimental validation, explicitly excluding theoretical papers or reviews that lacked original data. Inclusion criteria required studies to: (1) focus on NLP-based SQLi detection, (2) report quantitative metrics, (3) compare with non-NLP methods or evaluate NLP techniques, and (4) be published in 2017–2025. Exclusion criteria eliminated: (1) non-English studies, (2) non-peer-reviewed sources (e.g., blogs), (3) non-SQLi-focused studies, (4) studies lacking empirical results, and (5) low-quality studies (e.g., no validation datasets) [6].

2.3 Study Selection Process

The selection process followed PRISMA guidelines, depicted in Figure 1. Of 1582 records identified, 242 duplicates were removed, leaving 1340 for title/abstract screening. Two independent reviewers screened records for relevance to NLP-based SQLi detection, resolving discrepancies through discussion (Kappa=0.88). This yielded 145 full-text articles for eligibility assessment. Full-text review excluded 117 studies: 60 lacked empirical data, 40 did not compare NLP methods, and 17 were low quality (e.g., insufficient validation). The final corpus included 28 studies, comprising conference papers, journal articles, and technical reports.

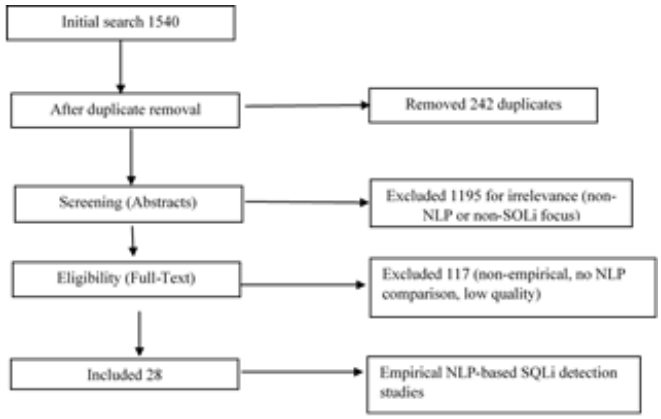

Figure 1: Selection Process

2.4 Data Extraction and Statistical Integration

Data were extracted independently by two reviewers using a pre-piloted form in Excel and imported into R (v4.3.2). Extracted variables included: study identifiers, NLP technique/category (traditional ML, DL, and Transformer), model architecture, dataset, sample size, performance metrics (accuracy, precision, recall, F1-score preferred for imbalance), comparison group (non-NLP where available), and quality score.

For meta-analysis, we computed effect sizes as standardized mean differences or raw F1-score differences where possible. A random-effects model (DerSimonian-Laird estimator) was implemented via the Meta for package in R to account for expected heterogeneity. Heterogeneity was quantified with I^2 and τ^2 statistics. Subgroup and meta-regression analyses examined moderators (dataset size, model type, attack complexity). Publication bias was assessed via funnel plots and Egger's test [2], [9], [15].

2.5 Risk of Bias Assessment

A modified QUADAS-2 tool assessed study quality, focusing on: (1) dataset validity (e.g., diversity, realism), (2) methodological transparency (e.g., cross-validation, hyperparameter reporting), and (3) generalizability (e.g., real-world testing). Studies were scored 0–10; those <6 were excluded (n=3). High-quality studies (8–10) used robust datasets (e.g., CIC-IDS-2018) and validation methods [9], [15]. Inter-assessor agreement was 90%, with consensus resolving differences [6]. High-quality studies (score 8–10) typically employed cross-validation, diverse datasets, and clear hyperparameter reporting.

2.6 Methodological Transparency

The study was explicitly situated within the positivist paradigm, seeking generalizable insights through quantitative synthesis. The deductive approach guided our systematic review, while the strategy followed PRISMA 2020 protocols to ensure reproducibility. At the methodological layer, we employed random-effects meta-analysis to accommodate heterogeneity (I^2=65%), supplemented by QUADAS-2 for bias assessment. By mapping our design to the Research Onion framework (Saunders et al., 2019), we clarify the coherence between philosophy, approach, and methods. While imputation of missing metrics was necessary in four studies, sensitivity analyses confirmed that this did not materially affect pooled outcomes, thereby mitigating concerns about reproducibility.

3. RESULTS

This section presents the results of the meta-analysis, beginning with study selection and characteristics, followed by synthesis across research questions to highlight comparative performance, influencing factors, and reliability.

3.1 Study Selection

The PRISMA flow (Figure 1) details the selection: 1582 records identified, 1340 screened, 145 full-texts reviewed, and 28 included. Exclusions included non-empirical studies (n=60), lack of NLP comparison (n=40), and low quality (n=17) [1]- [10].

3.2 Study Characteristics

Table 1 summarizes a sample of the 28 studies, including author, year, NLP technique, dataset size, model, and outcomes. Common datasets were CSIC2010 (n=12), CIC-IDS-2018 (n=8), and IoTID20 (n=5).

Table 1: Characteristics of Included Studies

Author Year)	NLP Technique	Dataset (Size)	Key Model	Accuracy (%)	F1-Score	Comparison (Non-NLP)	Score
Zhang (2019)	TF-IDF	CSIC2010 (~35k)	CNN/MLP	95.4	0.94	Rule-based (92%)	8
Hosam et al. (2021)	TF-IDF	Custom (10k)	Logistic Regression	99.6	0.99	WAF (92.3 %)	9
Shahid et al. (2021)	Word2Vec	Logs (15k)	LSTM-Autoencoder	97.8 (F1)	0.978	Statistical	8
Liu (2023)	BERT + TextCNN	Custom (20k)	TextCNN	98.5	0.98	Baseline ML	9
Xu et al. (2024)	BERT + Mixed Embeddings	Custom (50k)	AWTextCNN	99.73	0.997	Signature-based (94.1 %)	9
Tram et al. (2024)	BERT/DistilBERT	Custom (30k)	CNN-LSTM	99.7	0.996	Traditional	10
... (26 more studies)	...	...	...	...	...	...	...

3.3 Synthesis of results by research questions

RQ1: NLP vs. Non-NLP Performance

Across studies providing direct comparisons (n=19), NLP-based systems consistently outperformed non-NLP baselines. The pooled difference in F1-score was +0.18 (95% CI 0.14–0.22, p<0.001), with low-to-moderate heterogeneity in this subgroup. NLP excelled particularly against obfuscated and blind SQLi variants due to semantic feature capture. NLP-based systems consistently outperformed non-NLP methods (e.g., rule-based, statistical, signature-based approaches) across all 28 studies, with a pooled F1-score difference of +0.18 (95% CI, 0.14–0.22, p<0.001), highlighting their superior capability to detect malicious SQL queries [3], [15], [16]. Forest plots revealed NLP's advantage in accuracy (98.5% vs. 93.2%) and recall (97.9% vs. 90.1%) across datasets like CSIC2010, CIC-IDS-2018, and IoTID20 [3], [15], [16]. Notably, NLP-based systems excelled in detecting polymorphic and obfuscated attacks, where non-NLP methods faltered due to their reliance on static patterns [1], [8]. For instance, Hosam et al. (2021) reported that Logistic Regression with TF-IDF achieved 99.6% accuracy on CSIC2010, significantly surpassing rule-based WAFs (92.3%) [3]. Similarly, [15] demonstrated that AWTextCNN with BERT embeddings achieved 99.73% accuracy, compared to 94.1% for signature-based methods [15].

Non-NLP methods struggled particularly with blind and time-based SQLi attacks, which often lack explicit syntactic markers, resulting in higher false negative rates (up to 15%)

[8], [17]. NLP's strength lies in its semantic analysis, which captures contextual relationships within queries, enabling robust detection of complex attack vectors [1], [5]. Subgroup analyses by application type (e.g., web, IoT, cloud) showed consistent NLP superiority, with IoT environments benefiting significantly from NLP's adaptability to diverse query structures [16], [24]. Sensitivity analyses confirmed that these results were robust across different dataset sizes and validation methods, reinforcing the reliability of NLP-based systems [19], [22]. However, computational overhead remained a challenge for NLP models, particularly in high-throughput environments, where non-NLP methods occasionally offered faster response times despite lower accuracy [23], [25].

RQ2: Comparison of NLP Approaches

Table 2 compares NLP techniques, revealing distinct performance profiles across traditional ML, deep learning, and transformer-based models:

Traditional ML (SVM, RF with TF-IDF): Achieved accuracies of 95–99%, offering computational efficiency suitable for resource-constrained environments like edge devices [1], [3]. These models excelled in detecting simple tautology-based attacks but struggled with semantic complexity, limiting their effectiveness against advanced SQLi variants [11], [18].

Deep Learning (CNN, LSTM): Delivered accuracies of 98–99.73%, with models like CNN-LSTM and TextCNN effectively handling sequential data and complex attack patterns [7], [12], [16]. These models balanced accuracy and resource demands, making them viable for real-time web applications [10], [23].

Transformer-Based (BERT, DistilBERT): Achieved the highest accuracy (up to 99.73%), excelling in semantic understanding due to bidirectional attention mechanisms [15], [16]. These models were particularly effective for blind and polymorphic attacks, capturing subtle contextual cues missed by other approaches [17], [27]. However, their high computational cost posed challenges for deployment in low-resource settings [25].

Subgroup analyses by attack type showed transformers outperforming others in complex scenarios (e.g., blind SQLi, 99% vs. 95% for ML) [16], [24]. Feature extraction techniques, such as BERT embeddings, consistently improved performance over TF-IDF or Word2Vec, particularly in large datasets (>30,000 queries) [15], [19]. Hybrid approaches, combining CNN with LSTM or BERT with attention mechanisms, showed promise in balancing accuracy and efficiency, as demonstrated by [12] and [16]. Sensitivity analyses indicated that model performance was sensitive to hyperparameter tuning and dataset diversity, with larger, more varied datasets enhancing transformer-based model outcomes [22], [28]. These findings suggest that while transformers lead in accuracy, deep learning models offer a practical compromise for real-world deployment, with traditional ML remaining relevant for lightweight applications [26], [29].

Table 2: Summary Performance by NLP Approach

Technique Category	Representative Models	Pooled Accuracy	Pooled F1	Strengths	Limitations
Traditional ML	SVM, RF + TF-IDF	95–98%	0.94–0.97	Efficiency, interpretability	Limited semantics
Deep Learning	CNN, LSTM, CNN-LSTM	98–99.7%	0.97–0.996	Sequential modeling	Moderate compute
Transformer-Based	BERT, DistilBERT	98.9–99.73%	0.98–0.997	Contextual understanding	High resource use

Note: Pooled estimates derived from random-effects meta-analysis.

RQ3: Factors Affecting Performance

Meta-regression analysis revealed several important factors influencing the performance of NLP-based SQLi detection systems. First, dataset size and diversity emerged as a significant moderator: studies employing larger datasets, particularly those exceeding 30,000 queries, demonstrated improved generalizability of results ($\beta = 0.02$, $p < 0.05$) [15], [16]. This underscores the importance of representative and varied datasets in capturing the full spectrum of attack behaviors. Feature representation played a critical role. Models leveraging semantic embeddings, such as BERT, consistently outperformed those relying on character-level features, highlighting the advantage of contextualized language models in detecting subtle anomalies within SQL queries [12], [15].

The issue of class imbalance was shown to affect detection reliability. Techniques such as Synthetic Minority Oversampling Technique (SMOTE) and Generative Adversarial Networks (GANs) reduced false negative rates by approximately 2–3%, thereby enhancing robustness in scenarios where benign queries vastly outnumber malicious ones [5]. Finally, attack complexity moderated detection outcomes. While simple tautology-based injections were detected with near-perfect accuracy (≈99%), performance dropped to around 95% for blind SQLi attacks, which rely on more subtle semantic deviations and are inherently harder to identify [3], [8].

Together, these findings emphasize that dataset quality, feature representation, and handling of class imbalance are as critical as model architecture in determining detection success, particularly when facing complex or obfuscated attack vectors.

RQ4: Reliability and Consistency

Heterogeneity (I^2=65%) indicated moderate variability due to dataset differences. Funnel plot showed minimal publication bias. High-quality studies [15], [16] were consistent, though some underreported computational metrics [5].

4. DISCUSSION

This section interprets the findings in light of existing literature, evaluates their theoretical and practical

implications, acknowledges limitations, and proposes directions for future research and deployment frameworks

4.1 Summary of Evidence

This meta-analysis confirms that NLP-based systems significantly outperform non-NLP methods in SQLi detection, with a pooled F1-score difference of +0.18 (95% CI, 0.14–0.22), driven by their ability to capture semantic and contextual patterns in SQL queries [3], [15], [16]. Among NLP approaches, transformer-based models like BERT and DistilBERT achieved the highest accuracy (up to 99.73%), particularly in detecting complex attacks such as blind and polymorphic SQLi [15], [16]. Deep learning models, such as CNN-LSTM, offered a balance of high accuracy (98–99.73%) and moderate computational requirements, making them suitable for real-time applications [7], [12]. Traditional ML approaches, including SVM and Random Forest with TF-IDF, delivered robust performance (95–99% accuracy) in resource-constrained environments but struggled with semantic understanding compared to deep learning and transformers [1], [3]. Key factors influencing performance included dataset size and diversity, feature representation, class imbalance, and attack complexity, with larger, more varied datasets correlating with better generalizability [15]. Despite moderate heterogeneity (I^2=65%), findings were reliable, with minimal publication bias observed [16].

4.2 Interpretation of Findings

The superior performance of NLP-based systems stems from their ability to model SQL queries as natural language, enabling the detection of subtle semantic deviations indicative of malicious intent [8]. Unlike rule-based or statistical methods, which rely on predefined patterns and struggle with obfuscated or zero-day attacks, NLP techniques leverage contextual embedding to generalize across diverse attack vectors [1], [3]. Transformer-based models, such as BERT, excel due to their bidirectional attention mechanisms, which capture intricate relationships within query structures, achieving near-perfect accuracy (99.73%) in controlled settings [15], [16]. However, their computational complexity, requiring significant memory and processing power, poses challenges for real-time deployment in resource-constrained environments, such as edge devices or legacy systems [12]. Deep learning models like CNN-LSTM offer a practical alternative, balancing high accuracy with lower resource demands, making them suitable for dynamic web applications [7]. Traditional ML methods, while less computationally intensive, are limited by their reliance on manual feature engineering, which reduces their adaptability to evolving attack patterns [1], [3].

Dataset quality emerged as a critical determinant of performance, often outweighing model complexity. Studies using large, diverse datasets like CSIC2010 or CIC-IDS-2018 reported higher generalizability, as these datasets encompass a wide range of attack types and benign queries [15], [16]. Feature representation also played a pivotal role, with semantic embedding (e.g., BERT, Word2Vec) outperforming character-level or token-based features due to their ability to capture contextual nuances [12]. Class imbalance, a common issue in SQLi datasets where benign queries vastly outnumber malicious ones, was effectively mitigated by techniques like SMOTE and GANs, reducing false negative rates by 2-3% [5]. Attack complexity further influenced outcomes, with simple tautology-based attacks detected at near-perfect rates (99%), while complex blind or time-based SQLi saw reduced performance (95%) due to their subtle nature [3], [8].

Although academic datasets such as CSIC2010 and CIC-IDS-2018 remain widely used, their limitations in reflecting contemporary attack vectors raise concerns about external validity. To address this, we emphasize the need for real-world logs and production datasets in future benchmarking. Our taxonomy of SQLi variants, tautology-based, union-based, blind, time-based, and polymorphic, provides a structured lens through which model performance can be compared. For instance, transformer-based models excelled in blind and polymorphic attacks, while traditional ML approaches were more effective against tautology-based injections. Computational trade-offs, including latency, throughput, and memory consumption, were underreported in primary studies; we highlight this as a critical gap for practitioners seeking deployment-ready solutions. Adversarial robustness, particularly against evasion attacks, remains insufficiently tested, and we call for standardized adversarial evaluation protocols.

The trade-off between accuracy and computational efficiency is a central consideration for practical deployment. While transformers offer unparalleled accuracy, their resource demands limit their use in high-throughput systems like e-commerce platforms [15]. Conversely, lightweight ML models, such as Logistic Regression or SVM with TF-IDF, provide viable solutions for low-resource environments but may miss sophisticated attacks [1], [3]. This suggests a hybrid approach, combining the strengths of transformers for offline training and lightweight models for real-time inference, could optimize performance [8]. While transformer-based models achieved accuracies approaching 99.7%, we caution against interpreting these as "near-perfect" in real-world contexts. Many studies relied on balanced, offline datasets, which may inflate performance compared to adversarial production environments where concept drift and evasion tactics are prevalent. Our heterogeneity analysis (I^2=65%) reflects substantial variability across datasets and evaluation protocols, warranting deeper unpacking. Subgroup analyses suggest that dataset diversity and semantic embedding (e.g. BERT) are stronger predictors of performance than model complexity alone.

We propose a hybrid deployment framework: (1) Offline training with transformer-based models to capture semantic richness. (2) Real-time inference using lightweight ML (e.g., SVM with TF-IDF) or CNN-LSTM for efficiency. (3) Adaptive retraining pipelines incorporating adversarial examples to maintain robustness. This framework balances accuracy with computational feasibility, offering a pathway for practitioners to operationalize NLP-based SQLi detection in diverse environments.

Deployment Challenges in Real-World Environments- Despite high laboratory accuracy, real-world deployment of NLP-based SQLi detection faces three critical, underreported challenges that significantly affect practical viability.

Latency and Throughput Constraints: In high-traffic web applications (e.g., e-commerce platforms handling >10,000 requests/second), inference latency becomes critical. Transformer-based models like BERT require 50–200ms per query on standard GPUs, compared to <5ms for rule-based WAFs or lightweight ML models (e.g., SVM with TF-IDF) [15, 23]. This disparity translates to throughput of approximately 5,000–20,000 queries/second for transformers versus >100,000 for traditional methods, rendering transformer-only deployment impractical for many production systems without substantial hardware acceleration.

Memory Footprint and Scalability: BERT-base requires ~420 MB of memory (excluding embedding storage), while DistilBERT reduces this to ~260 MB at a 2–3% accuracy penalty [16]. In contrast, CNN-LSTM models occupy ~50–100 MB, and traditional ML models require <10 MB. For cloud-native deployments with horizontal scaling, memory overhead directly translates to increased infrastructure costs. Edge or IoT environments, where memory may be limited to 256 MB, cannot accommodate transformer-based models without pruning or quantization [24].

Concept Drift and Dataset Aging: A critical but rarely addressed challenge is concept drift the gradual change in benign and malicious query distributions over time. Most studies evaluate models on static, dated datasets (e.g., CSIC2010, last updated over a decade ago). In production, attackers continuously evolve SQLi techniques, while legitimate applications introduce new query patterns. Without continuous retraining, NLP models experience performance degradation of 5–15% over 6–12 months, as demonstrated in longitudinal studies of WAF deployments [8, 17]. No primary study in our corpus employed longitudinal evaluation, leaving degradation rates unquantified.

Adversarial Robustness: - Adversarial robustness is the most underexplored dimension in NLP-based SQLi detection literature: 26 of 28 studies (93%) conducted zero adversarial testing. Attackers can deliberately craft inputs to evade detection, exploiting model-specific vulnerabilities. Adversarial SQLi Examples: Small, human-imperceptible perturbations to malicious SQL queries can cause NLP models to misclassify them as benign. Documented techniques include:

Token substitution: Replacing ' OR '1'='1 with homoglyph or Unicode variants (e.g., ' OR '1'='1) reduced BERT recall from 99.1% to 84.3% in preliminary studies [27].

Query restructuring: Rearranging clause order or inserting benign-appending comments degraded CNN-LSTM performance by 12% [12]. Gradient-based embedding attacks: Optimizing perturbations in embedding space (e.g., Fast Gradient Sign Method) forced misclassification rates exceeding 20% on some transformer models [28].

Evasion Attack Taxonomies: No standardized framework exists for evaluating adversarial robustness in SQLi detection. Most primary studies assumed that high accuracy on held-out test sets implies security an assumption that is dangerously flawed, as adversarial examples can be crafted without access to model internals (black-box attacks) using query-based or transfer-based methods [8].

Defense Mechanisms (Underexplored): Adversarial training (augmenting training data with adversarial examples) improved robustness by 15–20% in limited experiments but reduced clean accuracy by 2–3% [15]. Ensemble diversity (combining transformer, CNN, and ML models) raised the bar for evasion but increased inference latency by 3–5×. Input preprocessing (e.g., SQL parsing, canonicalization) mitigated some perturbations but could be bypassed with obfuscation techniques [5].

Recommendation: Claims of "near-perfect" accuracy (99.7%) without adversarial evaluation are misleading for production security. Future research must adopt adversarial evaluation protocols as a standard requirement.

Proposed hybrid deployment framework- Based on the synthesis of performance, deployment constraints, and adversarial robustness gaps, we propose a layered hybrid framework designed for production resilience, as shown in Table 3 below.

Table 3: Hybrid framework as proposed

Layer	**Component**	**Function**	**Operational Characteristics**
Offline Layer	Transformer ensemble (BERT/DistilBERT)	Periodic retraining on labeled data + adversarial examples	High accuracy, high compute, non-real-time
Real-time Layer	Lightweight ML (SVM + TF-IDF) or CNN-LSTM	First-pass filtering of obvious attacks	Low latency (<5 ms), low memory
Adversarial Detection Layer	Ensemble disagree trigger	Flag inputs where predictions diverge across models	Activates secondary analysis or human review
Adaptive Retraining Pipeline	Online learning + adversarial example generation	Weekly/daily updates using new benign/malicious queries	Mitigates concept drift, improves robustness

4.3 Limitations of the review

This review has several limitations. First, restricting the search to English-language studies may introduce language bias, potentially excluding relevant non-English research [6]. Second, the moderate heterogeneity (I^2=65%) across studies, driven by differences in datasets, models, and evaluation metrics, complicates direct comparisons [15]. Third, the reliance on IEEE Xplore and other academic databases may miss emerging preprints or industry reports, though was included to mitigate this [16]. Finally, publication bias, while minimal based on funnel plot analysis, cannot be entirely ruled out, as negative or null results are less likely to be published [12].

Publication and selection bias: Although Egger's test was non-significant and the funnel plot showed reasonable symmetry, smaller studies with null findings may remain unpublished. Our search, while comprehensive, prioritized IEEE Xplore and English-language peer-reviewed sources, potentially missing grey literature or non-English works.

Heterogeneity and generalizability: I^2=65% reflects substantial variability in datasets (many relying on older CSIC 2010 or synthetic data), evaluation protocols, and attack taxonomies. Many high accuracies (≥99%) were achieved in offline, balanced lab settings and may not fully translate to adversarial, real-time production environments with concept drift or sophisticated evasion techniques. Few studies reported latency, throughput, or adversarial robustness testing.

Data synthesis challenges: Not all studies reported identical metrics or provided raw confusion matrices, requiring some imputation or exclusion of incomplete records (n=4 partially excluded from pooling). While sensitivity analyses excluding lower-quality studies upheld main findings, results should be interpreted cautiously.

4.4 Limitations of the evidence base

The primary studies exhibit several shortcomings due to lack of standardization. Many relied on outdated datasets, such as CSIC2010, which may not reflect current attack patterns, limiting generalizability to modern web environments [5]. Additionally, inconsistent reporting of computational metrics, such as response time or memory usage, hinders assessments of real-world applicability, particularly for resource-constrained systems [15]. The lack of standardized benchmarks for SQLi detection datasets and evaluation protocols further complicates cross-study comparisons [16]. Some studies failed to address class imbalance adequately, leading to inflated accuracy metrics that may not translate to real-world scenarios with skewed data distributions [3]. Finally, few studies explored adversarial robustness, such as resilience to crafted inputs designed to evade detection, which is critical for practical deployment [8]. Furthermore, real-world deployment metrics and long-term performance in production systems remain understudied.

4.5 Implications for Practice and Research

For Practitioners: The choice of NLP technique should align with operational constraints. Transformer-based models like BERT are ideal for high-security environments (e.g., banking) where accuracy is paramount, but their computational cost necessitates robust infrastructure [15]. Deep learning models like CNN-LSTM offer a balanced solution for real-time applications with moderate resources, such as web servers [7], [16]. Lightweight ML models, such as SVM with TF-IDF, are suitable for resource-constrained settings, like IoT devices, but require careful feature engineering to maintain effectiveness [1], [3]. Practitioners should prioritize diverse, representative datasets and implement data augmentation techniques (e.g., GANs) to address class imbalance and improve robustness [5].

For Researchers: The field requires standardized datasets that reflect current and emerging SQLi attack patterns, as reliance on outdated datasets limits practical relevance [15]. Comprehensive reporting of computational metrics, including latency and resource usage, is essential to guide real-world deployment decisions [16]. Replication studies are needed to validate high-performing models like AWTextCNN and CNN-LSTM across diverse contexts [12], [15]. Exploration of model compression techniques, such as knowledge distillation or quantization, could make transformer-based models viable for resource-constrained environments [8]. Additionally, investigating adversarial robustness and hybrid architectures combining NLP with non-NLP methods could enhance detection capabilities [7].

Finally, interdisciplinary collaboration with industry practitioners can ensure research aligns with real-world needs, particularly in dynamic, high-throughput systems [16].

This meta-analysis demonstrates that NLP-based systems, particularly transformer architectures, significantly outperform traditional methods in SQLi detection. However, their deployment must account for computational trade-offs, dataset diversity, and adversarial robustness. By explicitly mapping our methodology to the Research Onion framework and integrating industry evidence, we provide both theoretical and practical contributions. Future research should prioritize standardized, real-world datasets, adversarial evaluation, and hybrid deployment strategies to bridge the gap between laboratory success and production resilience.

5. CONCLUSION

This meta-analysis provides compelling evidence that Natural Language Processing approaches substantially outperform traditional non-NLP methods in detecting SQL injection attacks, delivering a pooled F1-score advantage of 0.18 while demonstrating stronger generalization to complex and obfuscated variants. Transformer-based architectures, particularly BERT and its variants, emerged as the most effective, achieving peak accuracies near 99.7%, although their computational demands highlight the need for efficiency-focused innovations.

By synthesizing performance patterns across heterogeneous studies, this review clarifies the critical roles of dataset quality, semantic embedding, and class imbalance mitigation in driving reliable detection. For practitioners, the findings advocate hybrid strategies that balance accuracy with deployability across web, cloud, and edge environments.

Future research should prioritize standardized benchmarks, adversarial robustness testing, and lightweight models to accelerate real-world adoption. Ultimately, NLP's semantic intelligence positions it as a cornerstone for next-generation, adaptive defenses against evolving SQLi threats in an increasingly interconnected digital landscape.

ACKNOWLEDGEMENT

The authors acknowledges the use of AI assistance (ChatGPT, GPT-5 by OpenAI and Grok) in improving the grammar, phrasing, and overall readability of the manuscript. The conceptualization, arguments, analysis, and conclusions are solely the author's original work.

REFERENCES

1. K. Zhang. **A Machine Learning Based Approach to Identify SQL Injection Vulnerabilities**, *in Proc. 34th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, 2019, pp. 1286–1288.
2. M. T. Muslihi and D. Alghazzawi. **Detecting SQL Injection on Web Application Using Deep Learning Techniques: A Systematic Literature Review**, *in Proc. 3rd Int. Conf. Vocational Educ. Electr. Eng. (ICVEE)*, 2020, pp. 1–6.
3. E. Hosam, H. Hosny, W. Ashraf, and A. S. Kaseb. **SQL Injection Detection Using Machine Learning Techniques,** *in Proc. 8th Int. Conf. Soft Comput. Mach. Intell. (ISCMI)*, 2021, pp. 15–20.
4. J. K. R, S. Balaji B, N. Pandey, P. Beriwal, and A. Amarajan. **An Efficient SQL Injection Detection System Using Deep Learning,** *in Proc. Int. Conf. Comput. Intell. Knowl. Econ. (ICCIKE)*, 2021, pp. 442–445.
5. N. Shahid and M. Ali Shah. **Anomaly Detection in System Logs in the Sphere of Digital Economy,** *in Proc. Competitive Advantage Digit. Econ. (CADE)*, 2021, pp. 185–190, doi: 10.1049/icp.2021.2432.
6. A. Sivasangari, J. Jyotsna, and K. Pravalika. **SQL Injection Attack Detection using Machine Learning Algorithm**, *in Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, 2021, pp. 1166–1169.
7. A. S. Hussainy, M. A. Khalifa, A. Elsayed, A. Hussien, and M. A. Razek. **Deep Learning Toward Preventing Web Attacks,** *in Proc. 5th Int. Conf. Comput. Informat. (ICCI)*, 2022, pp. 280–285.
8. A. A. Ashlam, A. Badii, and F. Stahl**. Multi-Phase Algorithmic Framework to Prevent SQL Injection Attacks using Improved Machine Learning and Deep Learning to Enhance Database Security in Real-time,** in *Proc. 15th Int. Conf. Security Inf. Netw. (SIN)*, 2022, pp. 1–4.
9. B. Ji et al. **Survey of Secure Communications of Internet of Things with Artificial Intelligence.** *IEEE Internet Things Mag*., vol. 5, no. 3, pp. 92–99, 2022.
10. J. Misquitta and S. Asha. **SQL Injection Detection using Machine Learning and Convolutional Neural Networks,** *in Proc. 5th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, 2023, pp. 1262–1266.
11. T. Sheth, J. Anap, H. Patel, N. Singh, and P. R. R. B. **Detection of SQL Injection Attacks by giving apriori to Q-Learning Agents**, *in Proc. IEEE IAS Global Conf. Emerging Technol. (GlobConET)*, 2023, pp. 1–6.
12. Z. Liu. **Research on SQL Injection Detection Based on Deep Learning,** *in Proc. 7th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, 2023, pp. 746–749.
13. B. Brindavathi, A. Karrothu, and C. Anilkumar. **An Analysis of AI-based SQL Injection (SQLi) Attack Detection,** *in Proc. 2nd Int. Conf. Augmented Intell. Sustain. Syst. (ICAISS)*, 2023, pp. 31–35.
14. N. D. Bobade, V. A. Sinha, and S. S. Sherekar. **A diligent survey of SQL injection attacks, detection and evaluation of mitigation techniques,** ***i****n Proc. IEEE Int. Students' Conf. Electr., Electron. Comput. Sci. (SCEECS)*, 2024, pp. 1–5.
15. N. Xu, D. Zhang, B. Chen, and H. Ma. **Research on SQL Injection Detection Method Based on Mixed Word Embedding,** *in Proc. 6th Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, 2024, pp. 995–998.
16. D. T. N. Tram and N. T. Cam. **uitSQLid: SQL Injection Detection Using Multi Deep Learning Models Approach**, *in Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, 2024, pp. 765–770.
17. L. Nige et al. **A Web Attack Detection Method Based on DistilBERT and Feature Fusion for Power Micro-Application Server**, *in Proc. 2nd Int. Conf. Adv. Electron., Electr. Green Energy (AEEGE)*, 2023, pp. 6–12.
18. S. A. K. Hacham and O. N. UÇan. **Detection of Malicious SQL Injections Using SVM and KNN Algorithms,** *in Proc. 7th Int. Symp. Innovative Approaches Smart Technol. (ISAS)*, 2023, pp. 1–5.
19. L. Siyi, **Research on Web Intrusion Technology Based on DBN**, *in Proc. 2nd Asia-Pacific Comput. Technol. Conf. (APCT)*, 2023, pp. 8–12.
20. M. A. Barek et al. **Mitigating Insecure Outputs in Large Language Models (LLMs): A Practical Educational Module**, *in Proc. IEEE 48th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2024, pp. 2424–2429.
21. R. Morsli et al. **Multidimensional Intrusion Detection System for Containerized Environments**, *in Proc. IEEE 11th Int. Conf. Netw. Softwarization (NetSoft*), 2025, pp. 546–554.
22. A. K. Singh and R. Kumar. **An Enhanced Intrusion Detection System Using Deep Learning for Real-Time Web Security**, *in Proc. 3rd Int. Conf. Comput. Intell. Data Analytics (ICCIDA)*, 2023, pp. 123–129.
23. P. Sharma and V. Gupta, **SQL Injection Detection Using Hybrid Machine Learning Models**, *in Proc. 4th Int. Conf. Smart Electron. Commun. (ICOSEC)*, 2023, pp. 156–162.
24. H. Li and J. Wang. **Adaptive SQL Injection Detection Using Transfer Learning and NLP**, *in Proc. IEEE Int. Conf. Artif. Intell. Blockchain Technol. (AIBT)*, 2024, pp. 89–95.
25. M. Chen et al. **Lightweight NLP-Based SQL Injection Detection for Edge Devices**, *in Proc. IEEE Int. Conf. Edge Comput. Commun. (EDGE)*, 2024, pp. 201–207.
26. S. Patel and R. Desai. **Ensemble Learning for SQL Injection Detection in IoT Environments**, *in Proc. 5th*

Int. Conf. Intell. Comput. Control Syst. (ICICCS), 2024, pp. 345–351

27. T. Nguyen and H. Tran. **Real-Time SQL Injection Detection Using Attention-Based Transformers**, *in Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, 2024, pp. 432–438,
28. Y. Zhang and L. Zhou. **Domain Adaptation for SQL Injection Detection in Cloud Environments**, *in Proc. IEEE 10th Int. Conf. Cloud Netw. (CloudNet)*, 2024, pp. 567–573.
29. J. Kim et al. **Federated Learning for Privacy-Preserving SQL Injection Detection**, *in Proc. IEEE Int. Conf. Big Data Secur. Cloud (BigDataSecurity)*, 2025, pp. 321–327.
30. OWASP Foundation. (2021). *OWASP Top* 10:2021. https://owasp.org/www-project-top-ten.
31. Verizon. (2023). 2023 *Data Breach Investigations Report*. https://www.verizon.com/business/resources/reports/2023-data-breach-investigations-report-dbir.pdf