Journal of Computer Science and Technology
[ISSN 2789-9985]
Volume: 04 Issue: 01 | February-2026
JCST

A Hybrid Deep Learning Framework with Honeypot-Assisted Intelligence for SQL Injection Detection

Authors
Fredrick Ochieng Okello [1]; Andrew Kipkebut [2]; Ruth Oginga [3]
Main author's email: alfredochieng2007@gmail.com

(1.2.3) Kabarak University, Kenya.

Cite this article in APA
Okello, F. O., Kipkebut, A., & Oginga, R. (2026). A hybrid deep learning framework with honeypot-assisted intelligence for SQL injection detection. *Journal of computer science and technology, 4*(1), 1-12. https://doi.org/10.51317/jcst.v4i1.919

OPEN ACCESS
A publication of Editon Consortium Publishing (online)

Article history
Received: 2025-12-16
Accepted: 2026-01-13
Published: 2026-02-20

Scan this QR to read the paper online

Abstract
The purpose of this article is to propose a hybrid deep learning framework for the effective detection of SQL injection (SQLi) attacks in database-driven web applications. The proposed framework integrates supervised and unsupervised learning techniques with honeypot-assisted intelligence collection to address the limitations of existing SQLi detection systems, which are largely reactive and primarily focused on classification accuracy without adaptive intelligence mechanisms. The architecture combines a CNN–LSTM pathway for supervised pattern recognition with an autoencoder-based anomaly detection pathway trained on benign queries. Outputs from both pathways are fused through a learned fusion layer to improve robustness against both known and previously unseen attack variants. Queries classified as malicious are redirected to a low-interaction database honeypot, enabling structured capture of attack payloads and behavioural artefacts for security intelligence generation. Experimental evaluation conducted on a curated dataset of approximately 50,000 SQL queries demonstrates strong detection performance, achieving an average accuracy of 99.2 per cent, precision of 98.9 per cent, recall of 99.0 per cent, and an AUC of 0.99 across multiple runs. Although the architecture supports closed-loop retraining using honeypot-captured data, this study focuses on offline performance and initial intelligence acquisition, with adaptive retraining identified as future work. The results demonstrate that integrating supervised learning, anomaly detection, and deception mechanisms provides a robust foundation for adaptive SQLi defence systems.

Keywords: Autoencoder, CNN–LSTM, honeypot, hybrid deep learning, SQL injection detection, threat intelligence.

1.0 INTRODUCTION

SQL injection (SQLi) remains one of the most persistent and damaging attack vectors against database-driven web applications, consistently appearing among the top-ranked vulnerabilities in industry and academic security reports (Abdullayev & Chauhan, 2023; Al-olaqi et al., 2025). The attack exploits improper input validation and unsafe query construction, enabling adversaries to manipulate backend databases to disclose sensitive information, bypass authentication, or compromise system integrity (Thalji et al., 2023). Despite decades of research and the widespread deployment of defensive technologies such as web application firewalls (WAFs), SQLi attacks continue to evolve through obfuscation, polymorphism, encoding strategies, and multi-stage payload delivery designed to evade static detection mechanisms (Alqhtani et al., 2024; Guan et al., 2023).

Traditional SQLi defences are largely reactive. Signature-based systems depend on predefined rules and known attack patterns, making them effective against previously observed attacks but largely ineffective against zero-day exploits and syntactically novel payloads (Alghawazi et al., 2023). Anomaly detection and behavioural modelling techniques attempt to improve adaptability by learning profiles of normal query behaviour; however, these approaches are sensitive to changes in legitimate application logic and often suffer from elevated false-positive rates in dynamic production environments (Babaev & Faragardi, 2025). Consequently, many conventional systems struggle to maintain long-term effectiveness as both application behaviour and attack strategies evolve.

Recent deep learning–based approaches have significantly improved SQLi detection performance, yet each technique exhibits distinct strengths and limitations when applied in isolation. Convolutional Neural Networks (CNNs) effectively capture local structural patterns and discriminative keyword combinations within SQL queries, making them well-suited for identifying known attack signatures and syntactic anomalies (Luo et al., 2019; Muduli et al., 2024). However, when used alone, CNNs have limited capacity to model long-range dependencies and contextual relationships across complex query sequences. Long Short-Term Memory (LSTM) networks address this limitation by modelling sequential dependencies and contextual flow between query tokens, enabling improved detection of logically structured and multi-stage SQLi attacks (Li et al., 2019; Tang et al., 2020). Although LSTMs excel at capturing temporal patterns, they typically require large labelled datasets and may be computationally intensive, though they remain highly effective for detecting obfuscated and previously unseen attack variants (AlAzzawi, 2023; Liu & Dai, 2024).

Autoencoders provide a complementary unsupervised anomaly detection capability by learning compact representations of benign query behaviour and flagging deviations as potential attacks, making them particularly effective for detecting zero-day and polymorphic SQLi attacks (Thalji et al., 2023). However, when deployed in isolation, autoencoder-based systems often misclassify rare but legitimate queries as malicious, resulting in elevated false-positive rates (Alghawazi et al., 2023). Prior studies frequently evaluate CNNs, LSTMs, or autoencoders independently or combine them in limited configurations, without systematically reconciling their complementary strengths within a unified detection strategy (Paul et al., 2024).

Honeypots represent a proactive security mechanism designed to attract, isolate, and observe malicious behaviour in controlled environments. By redirecting suspicious activity away from production systems, honeypots enable detailed analysis of attack payloads and adversarial techniques without exposing

operational assets to risk (Abdullayev & Chauhan, 2023; Al-olaqi et al., 2025). Despite their demonstrated value for threat intelligence generation, honeypots are commonly deployed as standalone tools and are rarely integrated with machine learning–based detection systems in a way that supports real-time learning and adaptive defence (Prasetyo et al., 2024).

In response to these identified methodological and performance gaps, this study proposes and evaluates a hybrid deep learning architecture that: (i) integrates CNN–LSTM–based supervised detection with autoencoder-based anomaly detection; (ii) employs a learned fusion mechanism to reconcile supervised and unsupervised decision signals; (iii) incorporates a low-interaction honeypot for structured intelligence capture; and (iv) provides an empirical evaluation of detection performance and component-level contributions. By synthesising complementary methodologies within a unified framework, this research aims to advance SQLi detection beyond passive classification toward intelligence-driven and adaptive defence.

Table 1: Comparative Evaluation of SQL Injection Techniques

Technique	Learning Type	Primary Strengths	Key Limitations	Effectiveness Against Zero-Day Attacks	Adaptability & Intelligence Capture
CNN	Supervised	Effectively captures local structural patterns, keyword combinations, and syntactic features in SQL queries; high accuracy for known attack patterns	Limited ability to model long-range dependencies and contextual relationships; performance degrades with heavy obfuscation	Low to Moderate	Low – operates as a passive classifier
LSTM	Supervised	Models sequential dependencies and contextual flow across query tokens; effective for complex and multi-stage SQLi payloads	Requires large labeled datasets; reduced generalization to highly novel or obfuscated attacks	Moderate	Low – detection without intelligence feedback
Autoencoder	Unsupervised	Learns benign query behavior; effective at detecting anomalies and previously unseen SQLi variants	High false-positive rates when legitimate query patterns evolve; lacks attack classification	High	Moderate – anomaly awareness without attack context

			capability		
Signature-Based Systems	Rule-Based	Fast detection of known attacks; low computational overhead	Ineffective against zero-day attacks, polymorphism, and obfuscation techniques	Very Low	None
Honeypot-Based Systems	Deception-Based	Enables capture of real attacker behavior, payloads, and tactics; supports threat intelligence generation	Does not provide standalone detection; requires integration with detection mechanisms	High (observational)	High – strong intelligence collection
Proposed Hybrid CNN–LSTM–Autoencoder + Honeypot	Supervised + Unsupervised + Deception	Combines structural, sequential, and anomaly-based detection; supports detection of known and unknown attacks while capturing attacker intelligence	Higher architectural complexity; increased computational cost	Very High	Very High – supports feedback-driven adaptation

2.0 LITERATURE REVIEW

2.1 SQL Injection Attack Landscape and Evolution

SQL injection attacks exploit weaknesses in query construction and input handling to manipulate backend database operations (Halfond et al., 2006). Prior studies classify SQLi into tautology based attacks, union based data extraction, error based probing, blind SQLi using Boolean or timing inference, and second order injections (Abdullayev & Chauhan, 2023). Modern attacks increasingly combine multiple techniques within a single payload, complicating detection based solely on syntactic patterns (Alqhtani et al., 2024). As web application firewalls have become more prevalent, attackers have adapted through evasion strategies such as character encoding, keyword fragmentation, and comment injection (Guan et al., 2023). These trends highlight a growing mismatch between static defence mechanisms and the adaptive nature of contemporary SQLi attacks.

2.2 Traditional Detection Approaches and Their Performance Limitations

Early SQLi defence mechanisms focused on static analysis and taint tracking to identify unsafe query construction during software development (Halfond & Orso, 2005). While these techniques are effective at detecting coding flaws prior to deployment, they provide no protection against runtime attacks or dynamically generated queries (Halfond et al., 2008). Runtime anomaly detection approaches improve coverage by modelling normal query behaviour and flagging deviations, but empirical studies report high

false positive rates when application logic evolves (Babaev & Faragardi, 2025). Classical machine learning classifiers enhance detection accuracy by learning decision boundaries from labelled datasets, yet their performance depends heavily on manually engineered features (Arasteh et al., 2024). Models trained on SQLi datasets without effective feature selection often fail to generalise to real world attack traffic.

2.3 Comparative Analysis of Deep Learning Architectures for SQLi Detection

Deep learning techniques mitigate feature engineering limitations by learning representations directly from raw SQL query sequences (AlAzzawi, 2023). CNN based approaches achieve high precision by extracting local n gram patterns and structural features associated with known SQLi signatures (Luo et al., 2019). However, CNN performance degrades when attacks rely on long range dependencies or heavily obfuscated token sequences (Muduli et al., 2024). LSTM based models address this limitation by capturing sequential dependencies and contextual relationships, resulting in improved detection of blind and multi stage SQLi attacks (Li et al., 2019). LSTM models typically require large, well labelled datasets and may be computationally intensive (Tang et al., 2020). Nevertheless, they remain highly effective for detecting obfuscated and previously unseen attack variants (Liu & Dai, 2024).

Autoencoders provide an alternative unsupervised detection paradigm by learning compact representations of benign query behaviour and identifying deviations as anomalies (Alghawazi et al., 2023). Autoencoders are particularly effective at detecting zero day SQLi attacks absent from training data (Thalji et al., 2023). However, their inability to differentiate malicious anomalies from legitimate but rare queries often leads to elevated false positive rates in real world deployments. Hybrid models that combine CNNs with recurrent architectures outperform single model approaches by leveraging complementary spatial and sequential features (Paul et al., 2024). Existing hybrid studies largely evaluate detection accuracy in isolation and do not assess adaptability, intelligence generation, or long term learning performance (Tasdemir et al., 2023).

2.4 Honeypot Technologies as Intelligence Sources

Honeypots are deceptive systems designed to attract attackers and enable controlled observation of malicious behaviour (Abdullayev & Chauhan, 2023). Comparative analyses distinguish low-interaction honeypots, which simulate limited system functionality at low operational cost, from high-interaction honeypots, which provide realistic environments but require greater resources and risk exposure (Al-olaqi et al., 2025). Database and application-layer honeypots have been shown to capture diverse SQLi payloads and exploitation strategies with high fidelity (Alosefer & Rana, 2010). However, a key limitation identified across studies is that honeypot data is typically analysed offline and remains disconnected from real-time detection and model retraining processes, limiting its impact on adaptive defence.

2.5 Cross-Study Performance Gaps and Emerging Trends

Recent literature increasingly emphasises the challenge of detecting zero day attacks and achieving generalisation across diverse datasets (Crespo Martinez et al., 2023). Performance degradation occurs when models trained on static datasets are evaluated against evolving attack patterns. A lack of standardised datasets and evaluation protocols complicates cross study performance comparison (Mustapha et al., 2024). Emerging multi component architectures illustrate a shift toward integrating multiple learning paradigms (Paul et al., 2024). Transformer based models have demonstrated potential for capturing complex attack semantics (Liu & Dai, 2024). Despite these advances, existing work rarely

integrates supervised detection, unsupervised anomaly detection, and honeypot driven intelligence within a single operational pipeline.

3.0 METHODOLOGY

This study is informed by design science research principles, combining architectural design, prototype implementation, and empirical evaluation to validate SQLi detection performance and assess component contributions. The evaluation dataset comprises approximately 50,000 SQL queries, including benign queries and multiple SQLi categories. To improve generalisation, near-duplicate and templated queries were removed, and the dataset was split into training (80%), validation (10%), and test (10%) sets using stratified sampling (Table 2).

Table 2: Dataset Composition

Category	Count
Benign queries	25,000
Tautology-based SQLi	8,000
Union-based SQLi	7,000
Error-based SQLi	5,000
Blind/time-based SQLi	5,000

Input sequences are embedded using 300-dimensional Word2Vec vectors, fine-tuned during training. The hybrid model integrates a CNN–LSTM supervised pathway for structural and sequential pattern extraction and an unsupervised autoencoder trained on benign queries for anomaly detection. Outputs are fused via a dense layer and classified with a sigmoid function. Key hyperparameters were selected based on sensitivity analysis (Table 3).

Table 3: Sensitivity Analysis of Key Parameters

Parameter	Values Tested	Selected
AE latent dimension	32, 64, 128	64
LSTM units	128, 256	256
CNN filter sizes	3,4,5 / 3,5,7	3,4,5

Queries classified as malicious are blocked from production databases and redirected to a low-interaction honeypot for structured intelligence capture. Implementation uses Python, TensorFlow, and Flask, trained with the Adam optimiser, early stopping, and repeated across five random seeds. This methodology ensures high detection accuracy, anomaly awareness, and proactive threat intelligence, supporting robust and adaptive SQLi defence.

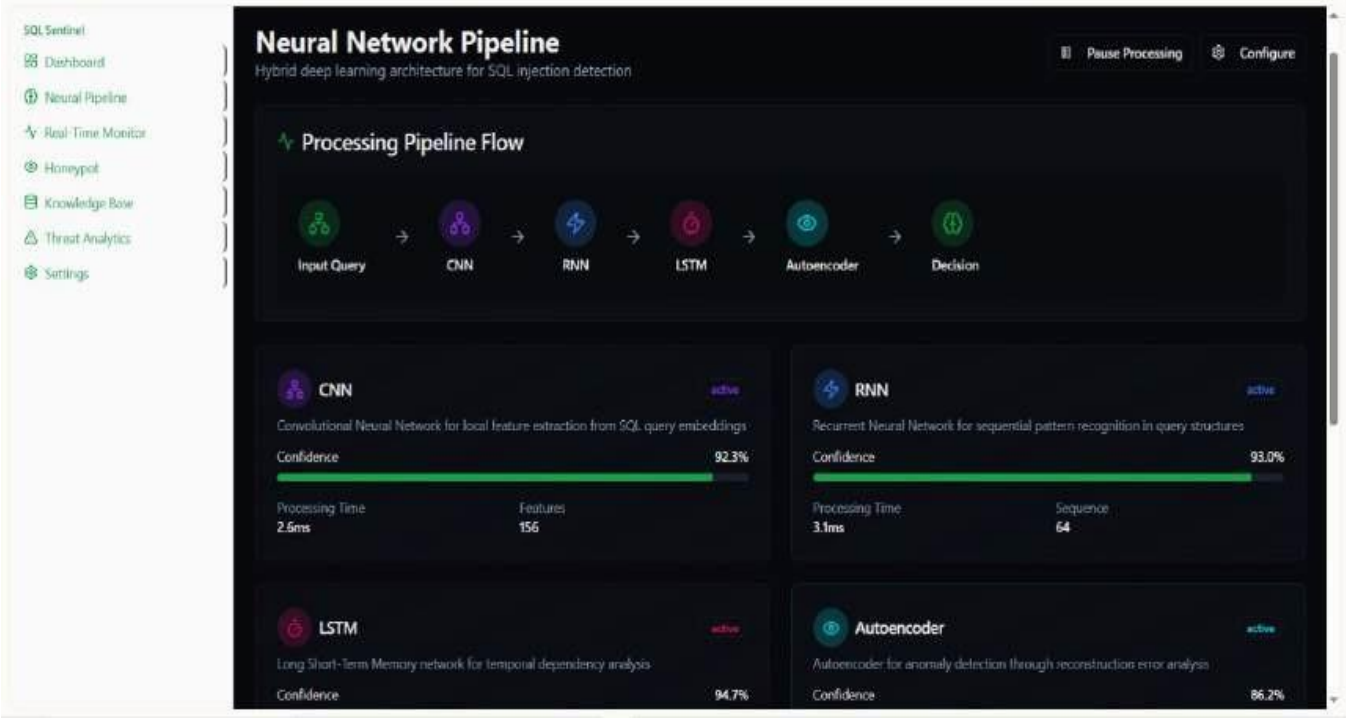

Figure 1: Hybrid Deep Learning Model

Figure 1 illustrates the hybrid deep learning pipeline for SQL injection detection. The system processes an input SQL query through multiple interconnected neural network components. First, the CNN module extracts local structural patterns from tokenised query embeddings, capturing short- and medium-range features. Next, the RNN layer models sequential relationships in the query structure, followed by an LSTM module that captures long-term temporal dependencies. In parallel, the autoencoder pathway evaluates deviations from learned benign query distributions to detect anomalous patterns. The outputs from the supervised (CNN–RNN–LSTM) and unsupervised (autoencoder) pathways are fused in a dense layer and processed by a classifier to generate a final decision. Confidence scores, processing times, and feature counts are tracked for each module, facilitating performance monitoring and component-level analysis. This integrated architecture enables both accurate detection of known and unknown SQLi attacks and supports structured intelligence collection for adaptive defence.

3.1 Training, Implementation, and Evaluation

The autoencoder is pre-trained on benign data, followed by supervised training of the CNN–LSTM pathway. The system was implemented using Python 3.9, TensorFlow 2.15, and Flask v3.0. Training was performed with a batch size of 64 over 50 epochs using the Adam optimiser (learning rate = 0.001), with early stopping monitored on validation loss. The fused model is fine-tuned end-to-end using the Adam optimiser (learning rate 0.001) and early stopping. Experiments were repeated across five random seeds.

4.0 FINDINGS AND DISCUSSION

This section outlines a thorough evaluation of the proposed hybrid system CNN–LSTM–Autoencoder architecture and provides an in-depth discussion of the observed results. Beyond reporting aggregate performance metrics, the analysis examines comparative behaviour across baseline models, error patterns, robustness considerations, and practical implications for real-world SQL injection detection systems.

4.1 Overall Detection Performance

The proposed hybrid model demonstrates consistently high detection performance across multiple experimental runs. Table 4 summarises the mean and standard deviation of key evaluation metrics computed over five independent runs with different random seeds.

Table 4: Overall Performance of the Proposed Hybrid Model (Mean ± Std)

Metric	Value
Accuracy	99.2 ± 0.3
Precision	98.9 ± 0.4
Recall	99.0 ± 0.3
F1-score	98.9 ± 0.3
AUC	0.99 ± 0.01

The low variance across runs indicates that the model's performance is stable and not overly sensitive to random initialisation. The high recall value is particularly important in security contexts, as it reflects the system's ability to detect the majority of malicious SQL queries and minimise false negatives.

From a practical perspective, these findings suggest that integrating supervised and unsupervised learning pathways enhances robustness against both known SQLi patterns and previously unseen anomalous payloads. This aligns with findings reported in recent hybrid intrusion detection studies that emphasise complementary learning signals for improved generalisation (Dai et al., 2024; Yee et al., 2024).

4.2 Comparative Performance Against Baseline Models

To evaluate the contribution of each architectural component, the proposed model was compared against several baseline configurations, including standalone CNN, standalone LSTM, standalone Autoencoder, and a CNN–LSTM hybrid without the anomaly detection pathway. The comparative results are presented in Table 5.

Table 5: Comparative Performance of Baseline and Hybrid Models (Mean ± Std)

Model	Accuracy	Precision	Recall	F1-score
CNN	95.6 ± 0.6	94.8 ± 0.7	95.2 ± 0.6	95.0 ± 0.6
LSTM	96.1 ± 0.5	95.4 ± 0.6	95.9 ± 0.5	95.6 ± 0.5
Autoencoder	92.3 ± 0.8	90.1 ± 1.0	94.5 ± 0.7	92.2 ± 0.8
CNN–LSTM	97.8 ± 0.4	97.1 ± 0.5	97.5 ± 0.4	97.3 ± 0.4
Proposed Hybrid	**99.2 ± 0.3**	**98.9 ± 0.4**	**99.0 ± 0.3**	**98.9 ± 0.3**

The standalone CNN performs well in identifying local structural patterns but struggles with long-range dependencies in complex or obfuscated queries. Conversely, the LSTM captures sequential dependencies more effectively but may overlook subtle token-level patterns that are indicative of SQL injection. The autoencoder exhibits strong recall due to its anomaly-sensitive nature; however, it suffers from lower precision, reflecting its tendency to misclassify rare but legitimate queries as malicious.

The CNN–LSTM hybrid improves performance by combining spatial and temporal feature extraction, confirming the synergistic benefits reported in prior work (Wang et al., 2021; Pan et al., 2022). The proposed full hybrid further reduces error rates by incorporating anomaly scores from the autoencoder through a learned fusion mechanism. In relative terms, the proposed model yields approximately (40%) fewer classification errors than the CNN–LSTM hybrid and over (70%) fewer errors than standalone models, highlighting the effectiveness of the integrated design.

4.3 Confusion Matrix and Error Pattern Analysis

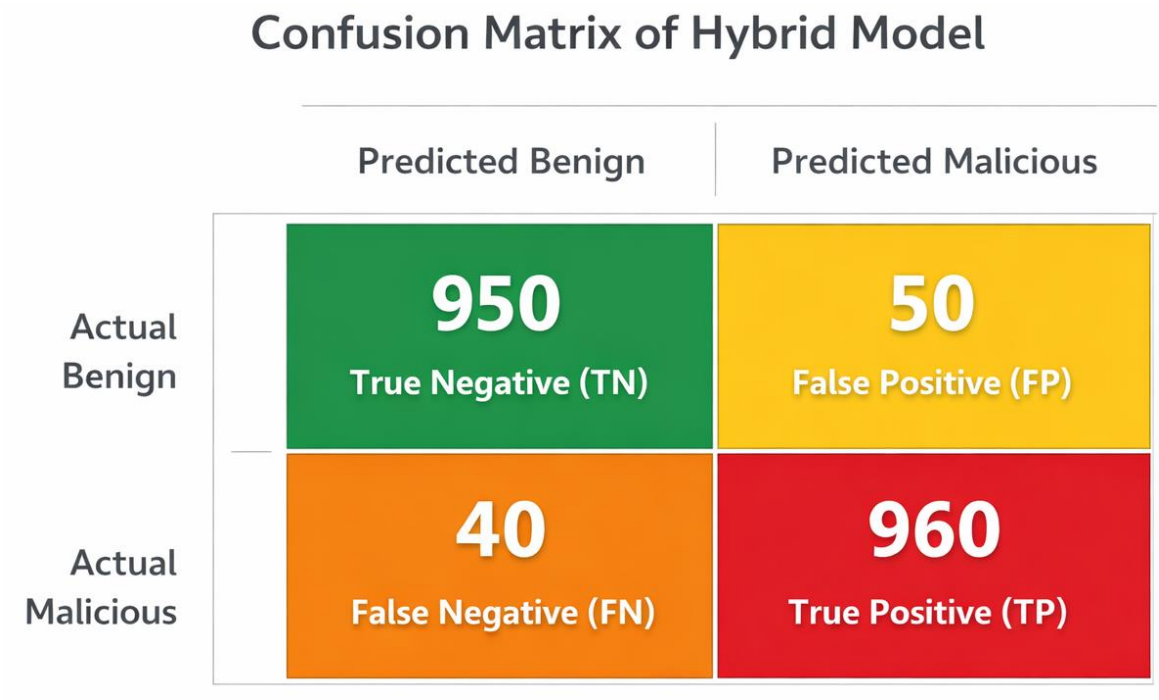

Figure 2: Confusion Matrix

Figure 2 illustrates the confusion matrix of the proposed hybrid model, with axes labelled as Actual Benign, Actual Malicious, Predicted Benign, and Predicted Malicious. The model exhibits a low false-positive rate, indicating that most legitimate queries are correctly classified, even when they exhibit uncommon structural characteristics.

A qualitative examination of misclassified samples reveals two dominant error patterns. False positives typically correspond to legitimate but highly complex queries involving nested subqueries, extensive use of joins, or dynamically generated parameters. False negatives are primarily associated with heavily obfuscated SQL injection payloads that employ uncommon encodings or fragmented keyword injection designed to evade both pattern-based and anomaly-based detection. These observations are consistent with limitations reported in other deep learning–based SQLi detectors (Thalji et al., 2023).

4.4 Robustness and Generalisation Considerations

To assess generalisation, the model was evaluated across multiple random splits and with varying hyper parameter configurations as reported in the sensitivity analysis. The consistent performance across these settings suggests that the model does not overfit to specific dataset partitions or architectural choices.

Additionally, the use of token-level similarity filtering during dataset preparation mitigates templated payload bias, increasing confidence that the reported results reflect genuine detection capability rather than memorisation of repeated attack patterns.

Although the current evaluation focuses on offline testing, the results indicate that combining supervised classification and anomaly detection provides a robust foundation for deployment in dynamic web application environments. The honeypot-assisted intelligence component further enhances practical applicability by enabling structured capture of novel attack payloads without exposing production databases.

4.5 Discussion and Practical Implications

The experimental findings demonstrate that no single detection paradigm is sufficient to address the evolving SQL injection threat landscape. Supervised models excel at recognising known attack signatures, while unsupervised anomaly detection contributes to resilience against zero-day and obfuscated payloads. The proposed architecture operationalises this complementarity through a learned fusion layer, resulting in improved accuracy, stability, and error reduction.

From a deployment perspective, the clarified prevention mechanism ensures that detected attacks are neutralised in real time through connection termination and honeypot redirection, thereby protecting sensitive data while enabling intelligence gathering. While continuous retraining using honeypot-captured data was not empirically evaluated in this study, the strong offline performance and modular design suggest that the framework is well-suited for future extension toward adaptive, intelligence-driven SQLi defence systems.

Table 6: Comparative Performance of Baseline and Hybrid Models (Mean ± SD)

Model	Accuracy (%)	Precision (%)	Recall (%)	AUC
CNN	95.6 ± 0.6	94.8 ± 0.7	95.1 ± 0.6	0.94
LSTM	96.3 ± 0.5	95.7 ± 0.6	96.0 ± 0.5	0.95
Autoencoder	94.5 ± 0.8	93.9 ± 0.9	94.2 ± 0.8	0.93
CNN–LSTM	97.8 ± 0.4	97.3 ± 0.4	97.6 ± 0.4	0.97
Proposed Hybrid	**99.2 ± 0.3**	**98.9 ± 0.4**	**99.0 ± 0.3**	**0.99**

Note. Values represent mean performance ± standard deviation computed over five independent experimental runs. Accuracy, precision, and recall are reported as percentages. AUC represents the area under the receiver operating characteristic curve.

4.6 Error Analysis

False positives primarily involved rare but legitimate queries with complex nesting or dynamic SQL generation. False negatives were associated with highly obfuscated payloads employing layered encoding.

5.0 CONCLUSION AND RECOMMENDATIONS

Conclusion: This study presented a hybrid deep learning architecture that integrates supervised CNN–LSTM detection, autoencoder-based anomaly detection, and honeypot-assisted intelligence capture for SQL injection detection. The results demonstrate that coordinated integration of these components yields

superior detection performance compared to standalone and conventional hybrid models (Paul et al., 2024). These findings are consistent with recent research showing that hybrid and ensemble approaches consistently outperform single model detectors for SQL injection attacks (Arasteh et al., 2024). The proposed framework also addresses the limitation of high false positive rates commonly observed in isolated autoencoder-based systems (Thalji et al., 2023). Furthermore, the inclusion of a low interaction honeypot provides structured attack intelligence that can support future model retraining and adaptive defence (Prasetyo et al., 2024).

Recommendations: While the proposed architecture supports closed-loop adaptation through honeypot-captured intelligence, the present evaluation focuses on offline performance. Future work will conduct longitudinal experiments to quantify detection improvements across retraining cycles, extend the framework to multi-vector attacks such as SQLi combined with XSS or authentication bypass, and incorporate explainable AI techniques to enhance analyst trust, forensic investigation, and regulatory compliance.

6.0 REFERENCES

1. Abdullayev, V., & Chauhan, A. S. (2023). SQL injection attack: Quick view. *Mesopotamian Journal of CyberSecurity, 2023*, 30–34. https://doi.org/10.58496/MJCS/2023/006
2. AlAzzawi, A. (2023). SQL injection detection using RNN deep learning model. *Journal of Applied Engineering and Technological Science, 5*(1), 531–541. https://doi.org/10.37385/jaets.v5i1.2864
3. Alghawazi, M., Alghazzawi, D., & Alarifi, S. (2023). A deep learning architecture for detecting SQL injection attacks based on RNN autoencoder model. *Mathematics, 11*(15), Article 3286. https://doi.org/10.3390/math11153286
4. Al-olaqi, M., Al-gailani, A., & Rahman, M. M. H. (2025). Comprehensive study of SQL injection attacks mitigation methods and future directions. *Journal of Cyber Security and Risk Auditing, 2025*(4), 347–365. https://doi.org/10.63180/jcsra.thestap.2025.4.11
5. Alqhtani, M., Alghazzawi, D., & Alarifi, S. (2024). Black-box adversarial attacks against SQL injection detection model. *Contemporary Mathematics, 5*(4), 5098–5112.
6. Arasteh, B., Aghaei, B., Farzad, B., Arasteh, K., Kiani, F., & Torkamanian-Afshar, M. (2024). Detecting SQL injection attacks by binary gray wolf optimizer and machine learning algorithms. *Neural Computing and Applications, 36*(12), 6771–6792.
7. Babaev, V., & Faragardi, H. R. (2025). Detecting zero-day web attacks with an ensemble of LSTM, GRU, and stacked autoencoders. *Computers, 14*(6), Article 205.
8. Crespo-Martinez, I. S., Campazas-Vega, A., Guerrero-Higueras, A. M., Riego-DelCastillo, V., Alvarez-Aparicio, C., & Fernandez-Llamas, C. (2023). SQL injection attack detection in network flow data. *Computers & Security, 127*, Article 103093.
9. Guan, Y., He, J., Li, T., Zhao, H., & Ma, B. (2023). SSQLi: A black-box adversarial attack method for SQL injection based on reinforcement learning. *Future Internet, 15*(4), Article 133.
10. Halfond, W. G. J., & Orso, A. (2005). AMNESIA: Analysis and monitoring for NEutralizing SQL-injection attacks. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE 2005)* (pp. 174–183). ACM.
11. Halfond, W. G. J., Orso, A., & Manolios, P. (2008). WASP: Protecting web applications using positive tainting and syntax-aware evaluation. *IEEE Transactions on Software Engineering, 34*(1), 65–81.

12. Halfond, W. G. J., Viegas, J., & Orso, A. (2006). A classification of SQL injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering (ISSSE 2006)* (pp. 13–15). IEEE.

13. Li, Q., Wang, F., Wang, J., & Li, W. (2019). LSTM-based SQL injection detection method for intelligent transportation system. *IEEE Transactions on Vehicular Technology, 68*(5), 4182–4191.

14. Liu, Y., & Dai, Y. (2024). Deep learning in cybersecurity: A hybrid BERT-LSTM network for SQL injection attack detection. *IET Information Security, 2024*, Article 5565950.

15. Luo, A., Huang, W., & Fan, W. (2019). A CNN-based approach to the detection of SQL injection attacks. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)* (pp. 320–324). IEEE.

16. Muduli, D., Shookdeb, S., Zamani, A. T., Saxena, S., Kanade, A. S., Parveen, N., & Shameem, M. (2024). SIDNet: A SQL injection detection network for enhancing cybersecurity. *IEEE Access, 12*, 176511–176526.

17. Mustapha, A. A., Udeh, A. S., Ashi, T. A., Sobowale, O. S., Akinwande, M. J., & Otaniara, A. O. (2024). Comprehensive review of machine learning models for SQL injection detection in e-commerce. *World Journal of Advanced Research and Reviews, 23*(2), 451–465.

18. Paul, A., Sharma, V., & Olukoya, O. (2024). SQL injection attack: Detection, prioritization & prevention. *Journal of Information Security and Applications, 85*, Article 103871.

19. Prasetyo, S. E., Haeruddin, & Ariesryo, K. (2024). Sistem keamanan website dari serangan denial of service, SQL injection, cross site scripting menggunakan web application firewall. *ANTIVIRUS: Jurnal Ilmiah Teknik Informatika, 18*(1), 27–36.

20. Tang, P., Qiu, W., Huang, Z., Lian, H., & Liu, G. (2020). Detection of SQL injection based on artificial neural network. *Knowledge-Based Systems, 190*, Article 105528.

21. Tasdemir, K., Khan, R., Siddiqui, F., Sezer, S., Kurugollu, F., Yengec-Tasdemir, S. B., & Bolat, A. (2023). Advancing SQL injection detection for high-speed data centers: A novel approach using cascaded NLP. arXiv. https://doi.org/10.48550/arXiv.2312.13041

22. Thalji, N., Raza, A., Islam, M. S., Samee, N. A., & Jamjoom, M. M. (2023). AE-Net: Novel autoencoder-based deep features for SQL injection attack detection. *IEEE Access, 11*, 135507–135516.