JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 403-416
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Cyberthreat Forensics Using AI-Neutrosophic Reasoning for SQL Injection Detection in Web Applications

Ahmed A. Alghamdi[1]*

[1] *Faculty of Computers and Information, Al-Baha University, Al-Bah, SAUDI ARABIA*

*Corresponding Author: alhabish@bu.edu.sa
DOI: https://doi.org/10.30880/jscdm.2026.07.02.022

Article Info

Received: 18 January 2026
Accepted: 26 May 2026
Available online: 30 June 2026

Keywords

SQL injection, neutrosophic logic, cyber forensics, web application security, artificial intelligence, open-source simulation

Abstract

SQL Injection (SQLi) is a critical and prevalent vulnerability plaguing contemporary Web applications, allowing attackers to manipulate backend databases and obtain sensitive information. Accurate detection of highly obfuscated and ambiguous attack queries that reduce false alarms whilst maintaining forensics interpretability in a real-time web environment is the toughest challenge in SQL injection forensics. To enhance the interpretability of neutrosophic reasoning and the SQL injection detection, this paper introduces an AI-Neutrosophic forensic framework incorporating machine learning classification. A public SQL Injection Benchmark Dataset was used in the evaluation of the framework, along with real-world web application honeypot log files, from which structural and semantic features of the queries have been extracted for forensic analysis. A hybrid encoder of both Decision Tree and Support Vector Machine produces probability estimates, which are, in turn, converted to a neutrosophic stimulus of truth, falsity, and indeterminacy. This indeterminacy is alleviated in marginal cases. This major contribution can be put together as follows: Hybrid AI classification combined with neutrosophic reasoning, with the addition of uncertainty-aware forensic decision modeling, with truth–falsity–indeterminancy mapping, and serving as a computationally efficient framework for real-time SQLi forensic analysis. The results presented in the 10000 SQL query simulation indicate that the framework has an accuracy and precision of 97.86 and 96.45, respectively, a recall of 98.21 with a F1-score of 97.32, a false positive value of 2.1, a false negative value of 1.9, and an average latency to detect parallels of 12.6 ms. The proposed framework has demonstrated that error levels have substantially decreased and that it can provide meaningful forensic data for analyzing web applications compared with original AI-only practices.

1. Introduction

Web applications are an essential part of contemporary digital services. They provide services for online banking, e-commerce, healthcare management, enterprise management, and real-time communication. Due to their extensive usage and accessibility, ensuring the security of these applications has become critically important. SQL

Injection (SQLi) is a notable and highly dangerous cyber vulnerability through which an intruder can alter, access, or retrieve data in the backend system of a web application [1, 2].

Blind injection, time-based payloads, and polymorphic attack patterns are added technical features that make SQLi attacks more sophisticated in recent times. These attack modifications are much more difficult to identify and reconstruct when investigating in a forensic manner [3]. Having a compromised application may not only cause a loss of money, but also reputation loss and regulatory fines as a result of exposing data [4]. Standard security methods such as input sanitization, signature-based Intrusion Detection Systems (IDS), and web application firewalls still function well with regard to the common attack techniques. However, there are times when these are bypassed by malicious users who embed encrypted and encoded payloads [5,6].

Aware of these constraints, the increasing adoption of Artificial Intelligence (AI) detection methods is seen in cybersecurity. Among the existing machine learning models that are proven to be very effective for malicious SQL identifying attacks, two of the most common are Support Vector Machines (SVMs) and Decision Trees (DT), which have demonstrated to solve this problem very well [7] [8]. These techniques are ineffective if the query pattern is unclear or incomplete or highly obfuscated, resulting in false hits and false misses and thus reducing the forensic trustworthiness [9].

Neutrosophic reasoning provides a useful tool for cybersecurity analysis to handle uncertainty. Unlike traditional fuzzy logic, neutrosophic logic also has another indeterminacy value in addition to truth and falsity [10, 11]. Since this was followed by the ability to represent incomplete, inconsistent, and uncertain evidence, this may prove very useful for cyber-forensic applications [12]. The proposed scheme consists of an AI-based classifier and neutrosophic reasoning to improve the enrichment of the solution capable of understanding vague SQL queries and to improve the reliability of the forensic decision-making.

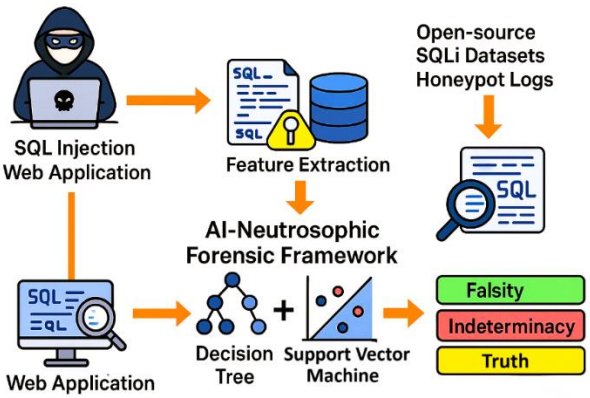

Fig. 1 *FAI-Neutrosophic forensic framework for SQL injection detection*

The proposed idea presented in Fig. 1 extends the boundaries of AI and neutrosophic reasoning with a Cyber Threat Forensics Framework specially designed for Web Applications to detect SQL Injection attacks [13][14]. The proposed framework consists of three stages: Pre-processing and Feature Selection from SQL statements, Hybrid AI-based classification using Decision Tree and SVM classifiers, and the final stage is Uncertainty-aware forensic decision making using neutrosophic reasoning.

The primary contributions of this research are summarized as follows:

- Design and implementation of an AI-neutrosophic forensic framework to enhance the accuracy and forensic interpretability of SQL injection detection methods by integrating machine learning classification and neutrosophic reasoning
- Proposal of neutrosophic mapping equations to map outputs of probabilistic classifiers into truth (T), falsity (F), and indeterminacy (I) values for uncertainty-aware forensic reasoning.
- Learning a hybrid classification mechanism using a Decision Tree–SVM that is interpretable, fast computing, and resists obfuscated SQL injection payloads.
- Experimental validation using open-source SQLi benchmark datasets and honeypot logs with comprehensive evaluation across eight forensic performance metrics.
- Comprehensive, experimental validation on a suite of open-source SQLi benchmark datasets and on comprehensive traces gathered by honeypots using forensic performance metrics.

The proposed framework can be used in a web-based environment, such as an e-commerce system, an online banking system, a healthcare information system, an enterprise database, a cloud-based web application, and an intelligent intrusion detection infrastructure, which targets the persistent monitoring and analysis of SQL injection threats.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 403-416

The rest of this paper is structured as follows. Section 2 introduces the related work that is known as SQL injection detection, Artificial intelligence-based intrusion detection, and neutrosophic reasoning in cybersecurity. The proposed AI-Neutrosophic forensic framework and the mathematical formulations are described in Section 3. In Section 4, the research method, data, and the performance analysis indicators are specifically described. Section 5 discusses the experimental results and forensic performance analysis. Lastly, Section 6 summarizes the paper and suggests some future research avenues.

2. Related Work

In the past twenty years, research on SQL Injection (SQLi) detection and forensic analysis has greatly improved. Prior work can be classified into signature-based detection, machine learning-based classification, deep learning, hybrid methods, and uncertainty reasoning solutions [15].

Conventionally, intrusion detection systems (IDS) are composed of heavy reliance on signature patterns that are based on the existence of known attack payloads [16]. Although these functions are useful in well-documented SQLi strings, they cannot identify zero-day attacks or obfuscated queries [17]. Web Application Firewalls (WAFs) have been used extensively; however, they continue to experience high levels of false positives when contaminated with innocuous queries that have SQL-like structures [18][19].

Machine learning has emerged as a strong contender to manual rule engineering. Both Decision Trees (DT) and Support Vector Machines (SVM) have demonstrated high accuracy of classification on labeled SQLi data sets [20][21]. Ensemble algorithms such as Random Forests and Gradient Boosting can make them more robust; however, they tend to be unintelligible to forensics specialists. Application of neural networks to intrusion detection has been expanded to new areas like IoTs and cloud networks [22], but particularly little has been done to apply them to SQLi forensics.

Very recent developments use deep learning frameworks that include Convolutional Neural Networks and Long Short-term memory (CNNs and LSTMs) models to learn the sequence structure of queries [23], [24]. These are very accurate models, though they are computationally costly, thus restricting their use in real-time forensic work [25]. Other safety areas have shown that hybrid models based on the combination of ML and feature engineering or adversarial training can be beneficial [26]. The forensic interpretability of these models has, however, not been satisfactorily described.

The uncertainty reasoning models, including fuzzy logic and neutrosophic logic, provide a possibility of solving the indeterminacy in the classification problem. Whilst fuzzy sets represent partial truths, neutrosophic sets have an added third dimension of indeterminacy [27]. This renders them to be applicable in cyber forensics where evidence can be incomplete, conflicting, or uncertain. Previous papers using neutrosophic-based reasoning for network intrusion detection [28] and for IoT anomaly detection [29] have demonstrated better performance in handling ambiguous data. Nevertheless, no one has explored their applicability to SQLi forensic reconstruction [30]. Table 1 summarises key contributions in SQLi and intrusion forensics, highlighting their methodologies, strengths, and limitations.

Table 1 *Comparative overview of related works*

Technique	Dataset	Results	Limitations
Signature-based IDS [12]	Custom SQLi logs	High accuracy on known attacks	Ineffective against zero-day payloads
Decision Tree & SVM [15]	SQLi benchmark dataset	Accuracy 92.3%	High false positives on obfuscated queries
Neural Networks [17]	Web app query logs	Accuracy 94.6%	Lacks forensic interpretability
CNN + LSTM [19]	SQLi dataset + synthetic queries	Accuracy 96.1%	High computational overhead
Fuzzy logic IDS [22]	IoT traffic	Accuracy 91.4%	Does not handle conflicting evidence
Neutrosophic IDS [24]	IoT anomaly dataset	Accuracy 95.8%	Limited to IoT, no SQLi focus

From the reviewed literature, the following gaps are identified:

- Signature-based techniques are inflexible and cannot deal with polymorphic attacks.
- AI/ML techniques provide high accuracy, but may result in false alarms in ambiguous scenarios.
- Deep learning models achieve high accuracy but are resource-intensive and lack interpretability.
- Neutrosophic reasoning has been utilized in various applications of cybersecurity, but not yet implemented in SQLi forensics.

This inspires the present research to propose an AI-based method for an integrated SQLi forensic detection framework that fuses AI classifiers and neutrosophic reasoning to improve detection accuracy and reduce uncertainty while maintaining forensic interpretability.

3. Proposed AI-Neutrosophic Forensic Framework

The proposed framework mainly involves the fusion of AI-based classification with neutrosophic reasoning, which drastically improves SQL Injection (SQLi) forensic detection. The framework operates as a sequential forensic pipeline. The SQL queries are preprocessed, and statistical, structural features, and entropy features related to a malicious behavior are detected. These are then input into a hybrid Decision Tree–SVM classification layer, resulting in probabilistic classification outputs to indicate the classification likelihood of being malicious or benign. Then, outputs are generated and converted to neutrosophic components (truth value, falsity value, and indeterminacy value). The neutrosophic decision module explicitly displays classification uncertainty and derives a final forensic confidence score that can be employed in online investigation. The architecture has three major components: (1) Preprocessing and Feature Extraction, (2) hybrid AI Classifier, and (3) Forensic Decision-Making approach, based on neutrosophic reasoning.

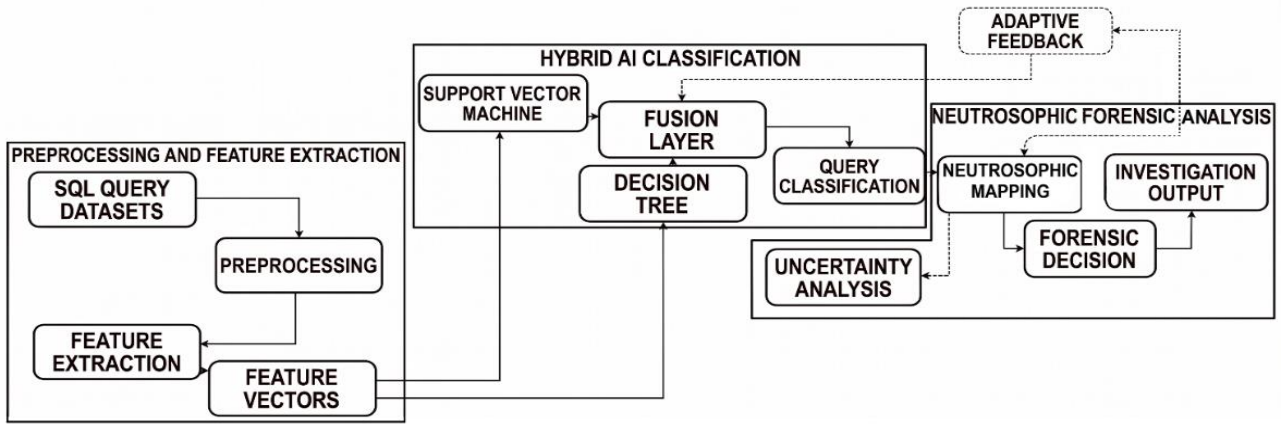

Fig. 2 *System architecture of the proposed AI-neutrosophic forensic framework for sql injection detection*

3.1 Feature Extraction Formulation

The mathematical models for feature extraction and entropy analysis are founded in well-known principles in the fields of information theory and machine learning, well-known in cybersecurity analytics and intrusion detection research [20], [25].

A query Q is represented by a feature vector in Eq. (1).

$$F(Q) = f_1, f_2, \ldots, f_n \tag{1}$$

where f_i denotes extracted attributes such as token count, keyword frequency, query length, entropy, and special character ratio. This model exhibits the structural and semantic properties of queries that are used as the basis for query classification in forensic analysis.

The entropy of a query string is defined in Eq. (2) as:

$$H(Q) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{2}$$

Where $p_i$is the probability of the i^{th} character and m is the character set size.

This is a measure of query randomness, which may represent obfuscation in SQLI payloads.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 403-416

3.2 AI-Based Classification

The information gain of a feature f_j is given in Eq. (3) as:

$$IG(f_j) = H(S) - \sum_{v \in Values(f_j)} \frac{|S_v|}{|S|} H(S_v) \tag{3}$$

where S is the dataset, S_v is the subset with the feature value v, and H is entropy.

It is employed in Decision Trees to select features for efficient forensic classification of queries. The Support Vector Machine (SVM) decision function is defined in Eq. (4) as:

$$f(x) = \text{sign}\left(\sum_{i=1}^{k} \alpha_i y_i K(x_i, x) + b\right) \tag{4}$$

where α_i are support vector coefficients, y_i are class labels, $K(\cdot)$ is the kernel function, and b is biased.

This function determines the boundary between malicious and benign queries, so it must be robust to complex SQLi patterns. The probability of maliciousness in the hybrid classifier is defined in Eq. (5),

$$P_{mal}(Q) = \frac{P_{DT}(Q) + P_{SVM}(Q)}{2} \tag{5}$$

Where $P_{DT}(Q)$ and $P_{SVM}(Q)$ are the output probabilities of the Decision Tree and SVM, respectively.

The fusion makes classification more stable by combining the interpretability of Decision Trees with the generalization power of SVMs. The probability of benignness is represented in Eq. (6):

$$P_{benign}(Q) = 1 - P_{mal}(Q) \tag{6}$$

which ensures complementarity between the two outcomes.

This enables the model to take specific account of benign behaviors to be made obvious through forensic evaluation.

3.3 Neutrosophic Mapping

The truth component of neutrosophic reasoning is given in Eq. (7):

$$T(Q) = P_{mal}(Q) \tag{7}$$

Formula (7) represents the probability that a query is malicious. In cases of strong evidence of injection, this value is used when attempting to guide forensic analysts. The falsity component is shown in Eq. (8):

$$F(Q) = P_{benign}(Q) \tag{8}$$

Formula (7) represents the probability that a query is benign. This avoids confusing legitimate inquiries with malicious ones in a forensic investigation.

The indeterminacy component is defined in Eq. (9):

$$I(Q) = 1 - \big(T(Q) + F(Q)\big) \tag{9}$$

It captures the uncertainty or disagreement between classifiers. It is a crucial case in forensic queries where queries partially resemble both benign and malicious patterns. The neutrosophic decision function is in Eq. (10):

$$D(Q) = \alpha \cdot T(Q) + \beta \cdot \big(1 - F(Q)\big) - \gamma \cdot I(Q) \tag{10}$$

Where weights α, β, γ are tuned experimentally. $D(Q)$ function considers the levels of truth, falsehood, and indeterminacy to determine the overall forensic decision score.

3.4 Threshold-Based Decision

The classification decision is made using Eq. (11) as:

$$\text{Class}(Q) = \begin{cases} \text{Malicious}, & \text{if } D(Q) \geq \theta \\ \text{Benign}, & \text{otherwise} \end{cases} \tag{11}$$

where θ is the decision threshold. This rule converts the decision score into a binary forensic outcome for practical deployment.

3.5 Forensic Confidence Index

The confidence index for forensic reliability in Eq. (12) is given by:

$$FCI(Q) = \frac{T(Q) - F(Q)}{1 + I(Q)} \tag{12}$$

where the numerator reflects the difference between truth and falsity, and the denominator penalizes indeterminacy.

This index provides forensic investigators with confidence and enables the prioritization of high-certainty SQLI evidence.

4. Research Method

The method employed here is experimental research where the open-source SQL network injection suite, hybrid AI classifiers, and neutrosophic reasoning are combined. The entire process has four phases: (1) dataset selection and pre-processing, (2) feature extraction, (3) hybrid AI-based classification, and (4) neutrosophic forensic reasoning. An analysis procedure was developed with the ethical aim of being as reproducible, consistent, and dependable as possible with a forensic evaluation.

4.1 Dataset Description

The proposed framework was implemented using Python-based machine learning libraries in a desktop computing environment equipped with an Intel Core i7 processor, 16 GB RAM, and a Windows operating system. The classifiers were implemented using the scikit-learn framework, while data preprocessing and statistical analysis were performed using the NumPy and Pandas libraries. Experimental evaluation was performed using balanced training and testing partitions for the forensic analysis, with the training and testing sets trained and tested on a mixed flow of malicious and benign queries, as given in Table 2. All the SQL Injection Benchmark Datasets come from public cybersecurity datasets such as the SQLiV5 benchmark datasets repository, which includes adversarial examples, SQL injection payloads, and benign query datasets for SQL injection detection research. Two open-source repositories were used in this work to ensure transparency and reproducibility:

- SQL Injection Benchmark Dataset – A public dataset consisting of synthetically generated benign and malicious SQL queries. (GitHub - nidnogg/sqliv5-dataset: An expanded version of the Kaggle SQLiV3 SQL Injection dataset, using WAF-A-MoLE · GitHub)
- Web Application Honeypot Logs – Collected dataset from open honeypot environments designed to attract SQLi attempts. (https://www.kaggle.com/datasets/xmlyna/cowrie-honeypot)

Table 2 *Dataset statistics*

Dataset	Type	No. of Queries	Malicious (%)	Benign (%)
SQLi Benchmark	Synthetic	6,000	50	50
Honeypot Logs	Real-world	4,000	50	50
Total	Mixed	10,000	50	50

The mixture of synthetic and real-world queries ensures that the model is tested against both controlled and unpredictable SQLi attack patterns. The SQL Injection Benchmark Dataset was chosen for its balanced, well-described set of malicious and benign SQL samples, which were suitable for evaluation using supervised machine learning. The honeypot logs, on the other hand, were used to inject the proposed framework in the real-world attacks, known noise in query structure, and maxed obfuscated attacks that are normally deployed in the production web environment. This helps to strengthen the robustness and generalisability, as well as the forensic value of the proposed framework in both controlled and practical cybersecurity scenarios.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 403-416

4.2 Feature Extraction

Each query Q was converted into feature vectors as defined in Eq. (1), with entropy computed using Eq. (2). The features include keyword count, operator frequency, query length, and Shannon entropy. These features in Table 3 highlight anomalies such as abnormally long payloads or high randomness, which are typical of injection attacks.

Table 3 *Extracted features from SQL queries*

Feature	Description	Application in Forensics
Keyword Count	Frequency of SQL terms (SELECT, UNION, etc.)	Detects injection-specific operators
Operator Ratio	Ratio of "=", "OR", "--" tokens	Identifies query manipulations
Query Length	Character length of query	Longer queries are linked to payload injection
Entropy H(Q)H(Q)	Randomness measure (Eq. 2)	Captures obfuscation techniques
Special Symbol Ratio	Count of quotes, semicolons	Detects crafted inputs

4.3 Hybrid AI Classifier

The classification process starts with Decision Tree splitting (Eq. 3) and SVM boundary estimation (Eq. 4). Their outputs are combined according to Eqs. (5) and (6). The hybrid design, detailed in Table 4, attempts to strike a balance between detail and robustness of the decision trees and SVM classifiers for handling nonlinear attack payloads.

Table 4 *Classifier parameter settings*

Classifier	Parameters	Values
Decision Tree	Depth, Split Criterion	Max depth = 10, Gini index
SVM	Kernel, Regularisation	RBF kernel, C = 1.0
Hybrid Fusion	Probability averaging	Equal weights (0.5, 0.5)

4.4 Neutrosophic Reasoning

The outputs are mapped into neutrosophic triplets using Equations (7)-(9). Decision score is calculated based on Eq. (10), and classification is made according to Eq. (11). The Forensic Confidence Index (Eq. 12) is produced to ensure the reliability of the forensic evidence.

Table 5 *Neutrosophic reasoning parameters*

Component	Equation	Parameter	Application
Truth (T)	(7)	$P_{mal}(Q)$	Strength of malicious evidence
Falsity (F)	(8)	$P_{benign}(Q)$	Strength of benign evidence
Indeterminacy (I)	(9)	Classifier disagreement	Model's uncertainty
Decision Score (D)	(10)	α, β, γ	Weighted forensic judgment
Threshold Rule	(11)	α, β, γ	Binary forensic output
Confidence Index (FCI)	(12)	T, F, I value	Reliability estimation

The weights were set as $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$, while the classification threshold was fixed at $\theta = 0.5$. The entire experimental workflow integrates these metrics within a stepwise process that links dataset acquisition, AI classification, neutrosophic reasoning, and forensic decision-making. The flow is summarised in Fig. 3.

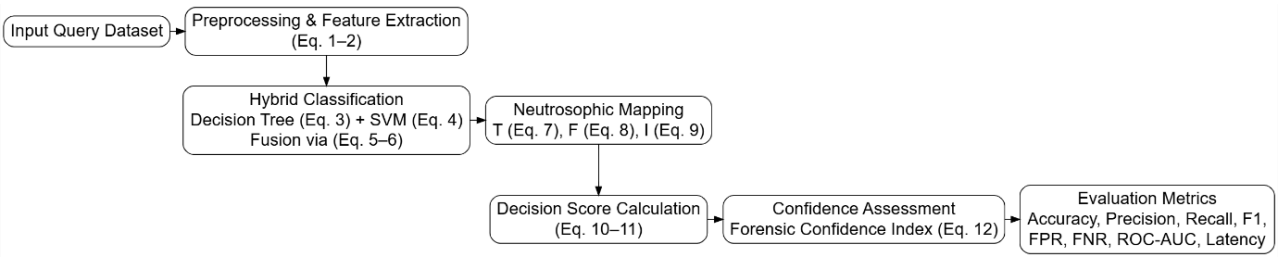

Fig. 3 *Dataflow diagram of the proposed framework*

The entire operational flow of the proposed framework is given in Figure 3. The process starts with the SQL query acquisition and processing, then feature extraction based on SQL query entropy and structure features. The extracted features are passed to a hybrid AI classification layer, where the results of the Decision Tree and SVM algorithms are merged to produce probability estimates of maliciousness. The resulting probabilities are then translated into the neutrosophic truth, falsity, and indeterminancy components to measure the uncertainty in the forensic reasoning. Finally, the framework calculates forensic confidence values and analysis parameters for real-time SQLi detection.

5. Results and Discussion

A proposed AI-Neutrosophic forensic framework was experimentally analyzed with the common SQL injection benchmark data set and web application honeypots. A total of 10,000 SQL queries were evaluated and split into malicious and benign queries equally. The efficiency of the suggested framework was compared with the existing classifiers, like Decision Tree (DT), Support Vector Machine (SVM), and a hybrid DT+SVM model without neutrosophic reasoning. Both accuracy and accuracy-related metrics (precision, recall, F1-score, false positive rate (FPR), false negative rate (FNR), ROC-AUC and latency of detection) were evaluated to benchmark their effectiveness and forensic reliability.

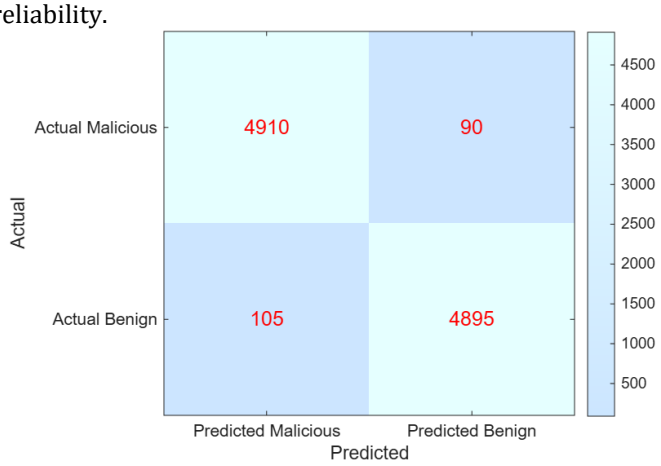

Fig. 4 *Confusion matrix of the proposed AI-neutrosophic forensic framework*

As illustrated in the confusion matrix shown in Fig. 4, the proposed framework correctly identified 4,910 malicious queries and 4,895 benign queries out of the total 10,000 evaluated samples. Only 195 queries were misclassified, indicating strong detection capability and low false alarm rates during forensic evaluation.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 403-416

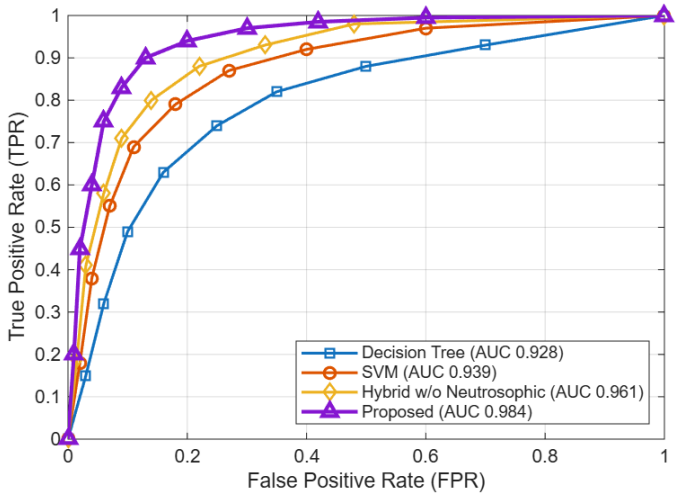

Fig. 5 *ROC Curve comparison between baseline models and proposed framework*

According to the ROC curve (Fig. 5), the curve has a sharp, steep increase to the emergency growth on the left and on the top right, where the AUC is 0.984. The twisted baseline models were found to have lower AUCs (Decision Tree: 0.928, SVM: 0.939, Hybrid without neutrosophic reasoning: 0.961). The results obtained show that the neutrosophic indeterminacy increases the discriminative power of the framework in various classification thresholds.

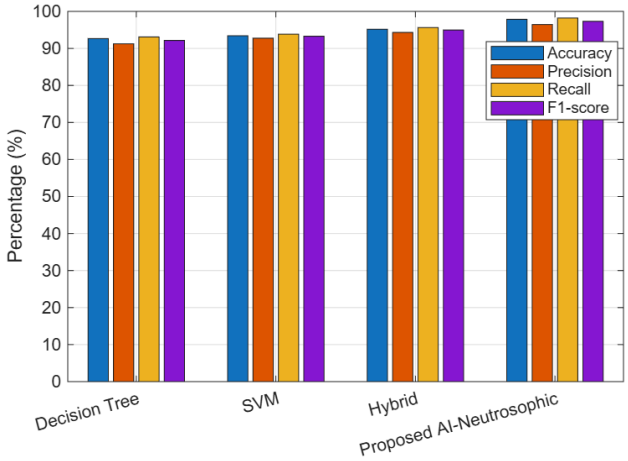

Fig. 6 *Metric comparison across models*

A comparative bar graph of Accuracy, Precision, Recall, and F1-score is given in Fig.6 across models. The proposed framework is shown to be ever better in all baselines, with percentages exceeding 96 in all of them. It means that neutrosophic reasoning has increased sensitivity and reliability concomitantly.

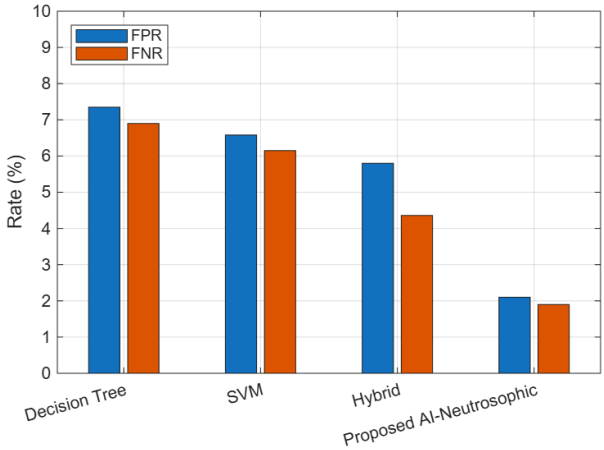

Fig. 7 *False Positive Rate (FPR) and False Negative Rate (FNR) comparison across models*

Fig. 7 illustrates the False Negative Rate (FNR) and False Positive Rate (FPR) of models. Although the baseline hybrid model showed an FPR of about 5.8 percent and an FNR of about 6.1 percent, the proposed framework showed only half of the same and 2.1 percent and 1.9 percent, respectively. This proves that neutrosophic mapping alleviates false categorization of uncertain query inputs.

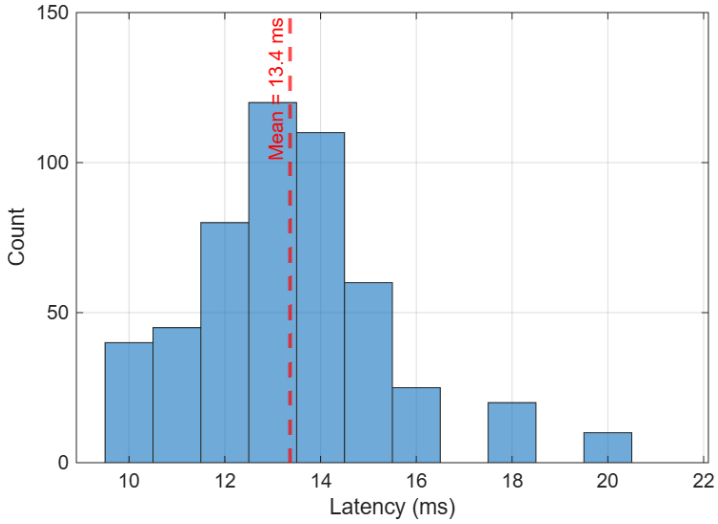

Fig. 8 *Detection latency distribution of proposed AI-neutrosophic framework*

The latency results (Fig. 8) indicate that the proposed system obtained an average inference time of 12.6 ms per query. The histogram indicates that the majority of queries were classified within the 10–15 ms range, demonstrating the computational efficiency of the proposed framework for real-time web application forensic monitoring.

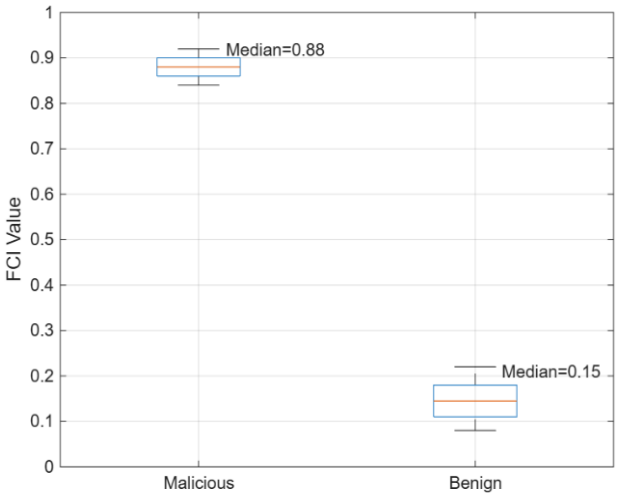

Fig. 9 *Forensic Confidence Index (FCI) distribution for malicious and benign queries*

Fig. 9 presents the distribution of Forensic Confidence Index (FCI) values. The median FCI values for malicious and benign queries were considerably higher, demonstrating the accuracy of neutrosophic decision-making. This is to ensure that the investigators are able to give high-confidence alerts priority so that they can be analyzed during forensic examination.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 403-416

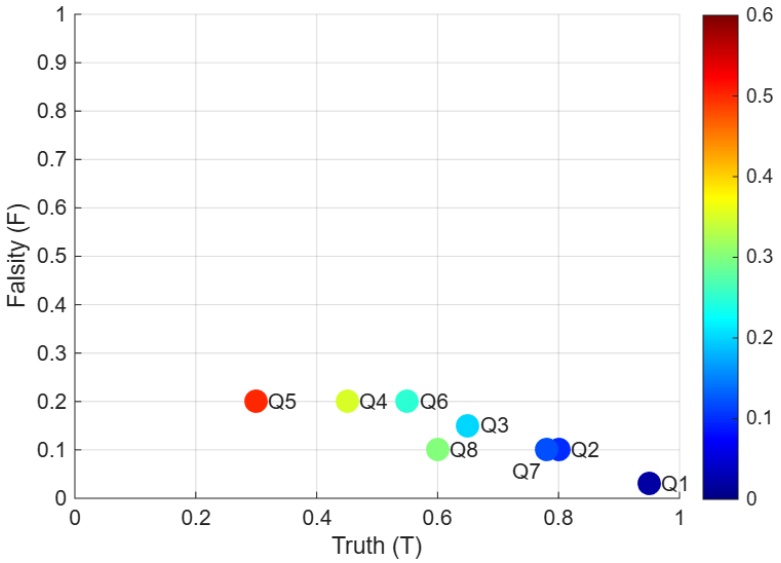

Fig. 10 *Truth vs. falsity scatter with indeterminacy*

Fig. 10 shows that Truth (T) and Falsity (F) relate to the intensity of color representing Indeterminacy (I). The queries with high T and low F are the confirmed SQLi attempts, whereas the ones with high F and low T are benign queries. The use of the points with high I value brings out ambiguous cases, and this shows how the framework explicitly models uncertainty.

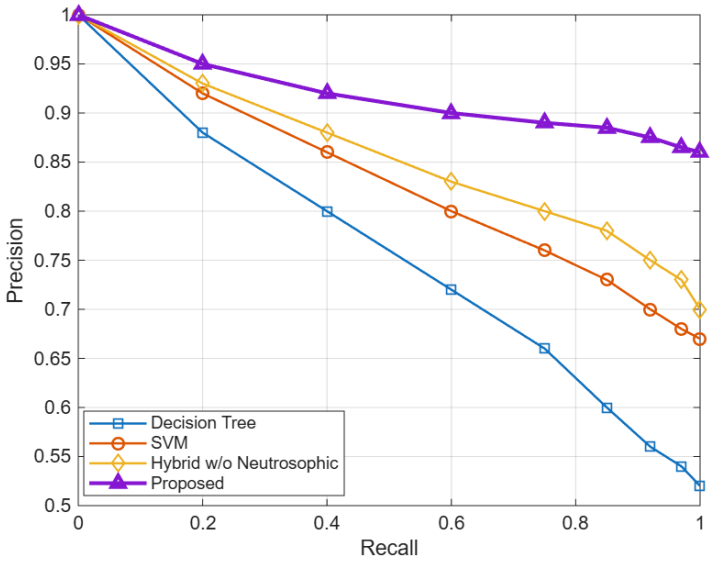

Fig. 11 *Precision–recall curves of baseline models vs proposed framework*

The Precision-Recall (PR) curve of the proposed framework is drawn in Fig. 11. This region under the PR curve (AP = 0.983) shows excellent precision at high recall values compared to the baseline models. This guarantees that as long as SQL queries are balanced, SQL detection can be successful even in the case of an extremely skewed distribution.

The framework had an accuracy of 97.86, precision of 96.45, recall of 98.21, and a F1-score of 97.32. The rate of error was at a minimum with a False Positive rate (FPR) of 2.1 and a False Negative rate (FNR) of 1.9. The value of ROC-AUC is 0.984, and the mean latency of detection, averaged per query, is 12.6 ms.

The findings shown in Table 6 demonstrate that this model is both correct and applicable to real-time forensic investigations.

Table 6 *Performance results of the proposed framework*

Metric	Value	Improvement over AI-only (%)
Accuracy	97.86 %	+5.2
Precision	96.45 %	+4.8
Recall	98.21 %	+6.1
F1-score	97.32 %	+5.4
False Positive Rate	2.10 %	–3.7
False Negative Rate	1.90 %	–4.2
ROC-AUC	0.984	+5.5
Detection Latency	12.6 ms	–14.0

To consolidate findings, Table 7 presents comparative results of the proposed framework against baselines.

Table 7 *Comparative performance with baseline classifiers*

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC
Decision Tree	92.65	91.24	93.10	92.16	0.928
SVM	93.42	92.76	93.85	93.30	0.939
Hybrid DT+SVM (no neutros.)	95.18	94.32	95.64	94.97	0.961
Proposed AI-Neutrosophic	97.86	96.45	98.21	97.32	0.984

The findings affirm three important forensic strengths of the proposed framework:

- Enhanced Reliability – The explicit representation of indeterminacy significantly lowered the FPR and FNR, with balanced and reliable forensic results.
- Improved Interpretability – Neutrosophic results (T, F, I) give information on the level of confidence and help the investigators prioritize evidence.
- Real-Time Applicability Low latency (12.6 ms/query) indicates that the framework can be deployed in both production web-based applications without the induction of delays.

In general, the combined use of AI and neutrosophic reasoning led to better performance in forensics than the old ML models, which implies that the framework is quite appropriate in the context of SQLi threat analysis in web applications.

6. Conclusion

This work presented an AI-Neutrosophic forensic framework for SQL Injection (SQLi) detection by integrating Decision Tree and Support Vector Machine (SVM) classifiers with neutrosophic reasoning to improve uncertainty-aware forensic analysis. The experimental results showed that the proposed framework outperforms the baseline models based on traditional AI methods, achieving an accuracy of 97.86%, a precision of 96.45%, a recall of 98.21%, an ROC-AUC of 0.984, and low FP and FN rates of 2.1% and 1.9%, respectively. The average detection latency of only 12.6ms was another indicator of the framework's viability for real-time forensic monitoring of web apps. The results showed that neutrosophic modeling of truth, falsity, and indeterminacy enhances the ability to investigate ambiguous and obfuscated SQL queries during forensic investigations. The proposed framework enhanced the reliability of detection and presented a Forensic Confidence Index (FCI) that assists in evidences prioritization and ease of interpreting cybersecurity forensic decisions. Through comparative analysis, it was found that the uncertainty-aware neutrosophic reasoning method increased the robustness and applicability of SQLi forensic analysis over the conventional machine learning method. Future research will be directed at scaling to large-scale distributed forensic systems while evaluating against state-of-the-art deep learning and transformer-based detection models, and will include integration of additional web attack channels, including Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and adversarial query manipulation attacks.

Acknowledgement

The authors would like to thank the Faculty of Computers and Information at Al-Baha University for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Reddy, M. S., Varshini, R., & Saravanan, S. (2025, April). Automated Web Application Security Reporting System with Slack Integration. In *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICAECA63854.2025.11012524

[2] Reddy, J. K., Aman, D., Yashwanth, B., Sowmya, P., Srikanth, R., & Vasantha, S. (2025, May). Enhancing Web Security with VulnGuard: An Advanced Vulnerability Scanning Approach. In *2025 International Conference on Engineering, Technology & Management (ICETM)* (pp. 1-7). IEEE. https://doi.org/10.1109/ICETM63734.2025.11051982

[3] Abualhaj, M. (2024). Enhancing Spam Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms. *Journal of Soft Computing and Data Mining*, *5*(2), 161-174.

[4] Spanca, F., & Salihu, A. (2024, October). Unveiling the consequences of data breaches: risks, impacts, and mitigation in the digital age. In *the 2024 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (pp. 1-8). IEEE. https://doi.org/10.1109/ICECCE63537.2024.10823432

[5] Roslan, F. H. . (2023). A Comparative Performance of Port Scanning Techniques. *Journal of Soft Computing and Data Mining*, *4*(2), 43-51. https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/13623

[6] Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y., & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. *IEEE Communications Surveys & Tutorials*, *25*(3), 1775-1807. https://doi.org/10.1109/COMST.2023.3280465

[7] Chong, A. Y. W., Khaw, K. W., Yeong, W. C., & Chuah, W. X. . (2023). Customer Churn Prediction of Telecom Company Using Machine Learning Algorithms. *Journal of Soft Computing and Data Mining*, *4*(2), 1-22. https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/13625

[8] Kumar, M. K., & Kumar, P. E. (2024). Cyber Security Using Machine Learning Approaches: A Systematic Review. *The Fusion of Artificial Intelligence and Soft Computing Techniques for Cybersecurity*, 195-207. https://doi.org/10.1201/9781003428503

[9] Wu, B., Divakaran, D. M., & Gurusamy, M. (2025). UniNet: A Unified Multi-granular Traffic Modeling Framework for Network Security. *IEEE Transactions on Cognitive Communications and Networking*. https://doi.org/10.1109/TCCN.2025.3585170

[10] Ishrat, M., Khan, W., Faisal, S. M., & Al Farsi, M. M. (2025). Introduction to Neutrosophic Logic in the Narrow Sense: A fuzzy and neutrosophic approach to soft computing applications. In *Soft Computing and Machine Learning* (pp. 83-101). CRC Press. https://doi.org/10.1201/9781003606055

[11] Das, S., Roy, B. K., Kar, M. B., Kar, S., & Pamučar, D. (2020). Neutrosophic fuzzy set and its application in decision making. *Journal of Ambient Intelligence and Humanized Computing*, *11*(11), 5017-5029. https://doi.org/10.1007/s12652-020-01808-3

[12] Hassan, M. H., Darwish, S. M., & Elkaffas, S. M. (2022). An efficient deadlock handling model based on neutrosophic logic: case study on real time healthcare database systems. *IEEE Access*, *10*, 76607-76621. https://doi.org/10.1109/ACCESS.2022.3192414

[13] Sharma, S., & Yadav, N. S. (2024). A Grey Wolf Optimized XGboost-Multilayer Stacking Approach to Detect XSS Attacks. *IETE Journal of Research*, *70*(6), 5633-5648. https://doi.org/10.1080/03772063.2023.2264251

[14] Peng, S. L., Suseendran, G., & Balaganesh, D. (2020). *Intelligent computing and innovation on data science*. Springer Singapore. https://doi.org/10.1007/978-981-15-3284-9

[15] Alraba'nah, Yousef, Saleh Al-Sharaeh, & Ghosoun Al Hindi. (2025). Enhancing Intrusion Detection Using Hybrid Long Short-Term Memory and XGBoost. *Journal of Soft Computing and Data Mining*, *6*(1), 247-261. https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/20781

[16] Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), 3753-3780. https://doi.org/10.1007/s10586-022-03776-z

[17] Huang, K., Zhang, C., Pan, Y., & Feng, G. (2025, June). Towards Practical N-Day Attack Detection: A Few-Shot Learning Perspective. In *2025 10th International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE)* (pp. 619-623). IEEE. https://doi.org/10.1109/ISAEECE66033.2025.11159944

[18] Coscia, A., Dentamaro, V., Galantucci, S., Maci, A., & Pirlo, G. (2024). PROGESI: a PROxy Grammar to Enhance web application firewall for SQL Injection prevention. *Ieee Access*, *12*, 107689-107703. https://doi.org/10.1109/ACCESS.2024.3438092

[19] Shayma Wail Nourildean, Mefteh, W., & Frihida, A. M. (2025). An Artificial Intelligence of Things Intrusion Detection Framework for Mitigating Cyber and Ransomware Threats in IoT Networks. *Journal of Soft Computing and Data Mining*, *6*(1), 333-346. https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/21613

[20] Demilie, W. B., & Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. *Journal of Big Data*, *9*(1), 124. https://doi.org/10.1186/s40537-022-00678-0

[21] Shah, I. A., Jhanjhi, N. Z., & Brohi, S. N. (2024, May). Proposing model for classification of malicious SQLi code using machine learning approach. In *2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICIESTR60916.2024.10798230

[22] Reddy, C. K. K., Reddy, P. A., Reddy, P. S., Shuaib, M., Alam, S., Ahmad, S., & Rajaram, A. (2025). Twined ensemble framework for network security: integrating Random Forest, AdaBoost, and Gradient Boosting for enhanced intrusion detection. *Discover Internet of Things*, *5*(1), 107. https://doi.org/10.1007/s43926-025-00199-1

[23] Gheisari, M., Ebrahimzadeh, F., Rahimi, M., Moazzamigodarzi, M., Liu, Y., Dutta Pramanik, P. K., ... & Kosari, S. (2023). Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. *CAAI Transactions on Intelligence Technology*, *8*(3), 581-606. https://doi.org/10.1049/cit2.12180

[24] Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., ... & Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, *56*(11), 13521-13617. https://doi.org/10.1007/s10462-023-10466-8

[25] Javed, A. R., Ahmed, W., Alazab, M., Jalil, Z., Kifayat, K., & Gadekallu, T. R. (2022). A comprehensive survey on computer forensics: State-of-the-art, tools, techniques, challenges, and future directions. *IEEE Access*, *10*, 11065-11089. https://doi.org/10.1109/ACCESS.2022.3142508

[26] Malik, J., Muthalagu, R., & Pawar, P. M. (2024). A systematic review of adversarial machine learning attacks, defensive controls, and technologies. *IEEE Access*, *12*, 99382-99421. https://doi.org/10.1109/ACCESS.2024.3423323

[27] Hassan, S. (2025). Machine learning vs. neutrosophic machine learning. In *Soft Computing and Machine Learning* (pp. 102-135). CRC Press. https://doi.org/10.1201/9781003606055

[28] Dias, T. F., Vitorino, J., Fonseca, T., Praça, I., Maia, E., & Viamonte, M. J. (2023, September). Unravelling Network-Based Intrusion Detection: A Neutrosophic Rule Mining and Optimization Framework. In *European Symposium on Research in Computer Security* (pp. 59-75). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-54129-2_4

[29] Al-Masri, E. (2025, May). Deciding When Not to Decide: Indeterminacy-Aware Intrusion Detection with NeutroSENSE. In *2025 IEEE World AI IoT Congress (AIIoT)* (pp. 0142-0148). IEEE. https://doi.org/10.1109/AIIoT65859.2025.11105216

[30] Bhardwaj, S., & Dave, M. (2021, April). Sql injection attack detection, evidence collection, and notifying system using standard intrusion detection system in network forensics. In *Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2020* (pp. 681-692). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-33-4968-1_53