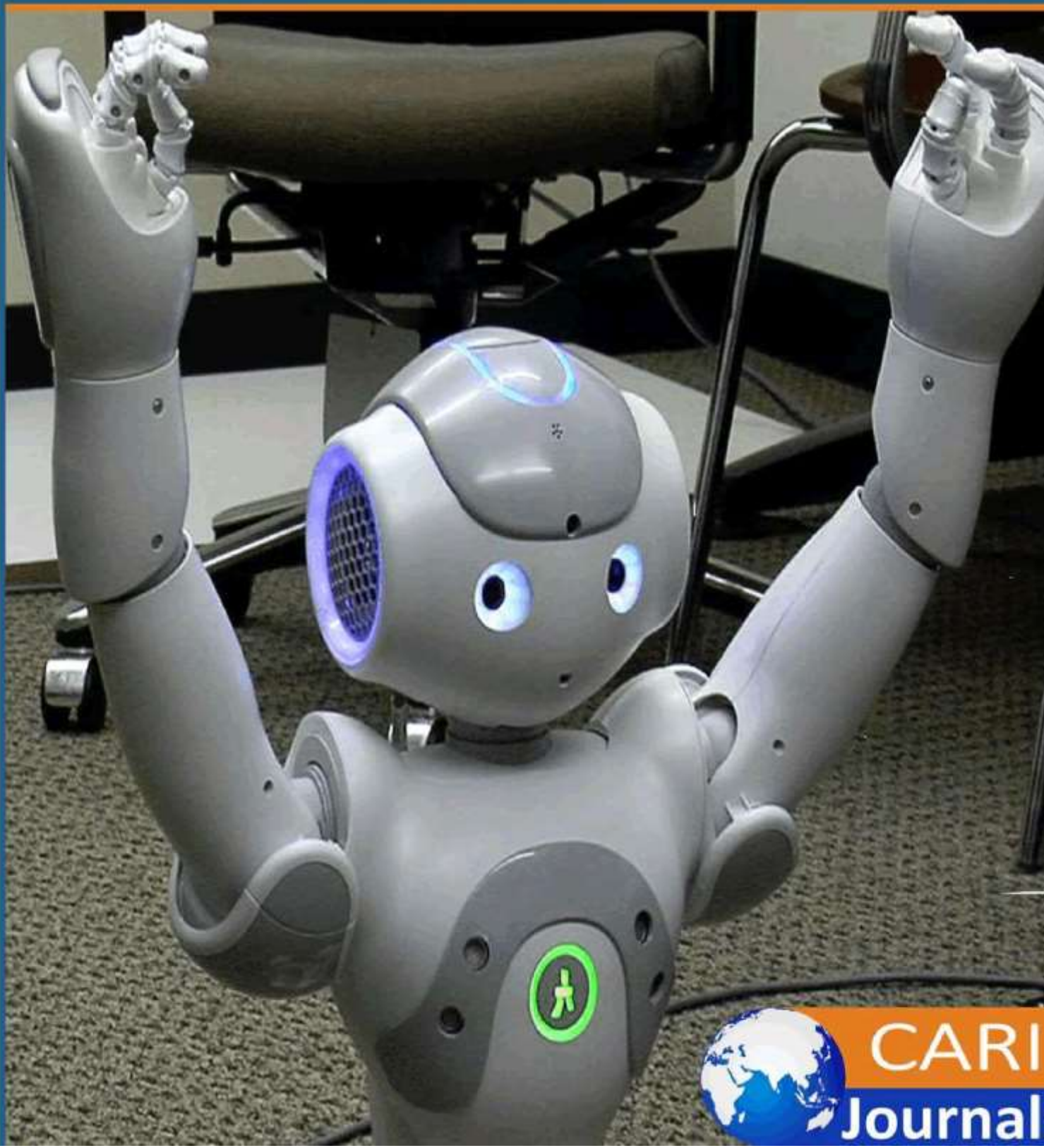
International Journal of
Computing and
Engineering
(IJCE)

AI-Driven Synthetic Data Engineering: A Framework for Generating, Validating and Managing High-Fidelity Data for Intelligent Information Systems



CARI Journal

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

AI-Driven Synthetic Data Engineering: A Framework for Generating, Validating and Managing High-Fidelity Data for Intelligent Information Systems

[1*]Dr. Abdinasir Ismael Hashi, [2]Mr.Abdirizak Mohamed Hashi,
[3]Mr. Abdirizak Hussein Mohamed
[1,3]Somali National University
[2]Jazeera University

Accepted: 23rd Oct, 2026, Received in Revised Form: 9th Nov, 2026, Published: 23rd Nov, 2026

Abstract

Cyber threats are continuously evolving and it is more than ever that integrated solutions for intrusion detection, intelligent decision making and secure access control are required. Current solutions tend to tackle threat classification, collaborative intelligence, and cryptographic security, and are seldom integrated in coherent cyber defense systems. This study introduces an AI-native autonomous cyber defense framework that combines multi-agent orchestration of the LLM with post quantum-based zero-trust architecture. The framework consists of data acquisition, data pre-processing, multi-agent intelligence, threat analysis, zero-trust enforcement and autonomous response. To classify the network traffic into 15 different categories, it is evaluated by using CICIDS2017 database using a Random Forest classifer. After preprocessing and removing duplicate data, 106,126 records were retained and split into 70% training and 30% testing data. On 31838 test records, the classifier had an accuracy of 99.17%, weighted F1 of 99.14% and macro F1 of 91.90%. Web Attack XSS showed a recall rate of ~37.76%, showing some issues with detecting XSS for the class. The report of experiments establishes classification performance on the data tested, rather than effectiveness of the entire framework, and an integrated architectural design connecting intelligent threat assessment with zero-trust enforcement and post-quantum cryptographic mechanisms is added to the study.

Keywords: *Synthetic Data Engineering; Artificial Intelligence; Data Quality Validation; Privacy-Preserving Data Generation; Intelligent Information Systems.*

1. Introduction

Artificial intelligence (AI) and data-driven technologies are becoming essential components of intelligent information systems in the age of digital transformation. [1]. Artificial intelligence technologies are widely used in medical, financial, educational, business analytics, cybersecurity, and decision-making applications [2]. These systems depend heavily on the quality, diversity, and representativeness of datasets. But, there are some challenges associated with collecting and managing real-world data such as privacy issues, restricted data availability, expensive data acquisition, missing data, class imbalance and data sharing constraints [3-6]. The restrictions may limit the generalizability and reliability of AI models. To overcome these challenges, the new idea of synthetic data engineering has been proposed to produce synthetic datasets that exhibit the key statistical features and relationships of the original data without directly simulating any individual data records [7].

AI-based synthetic data engineering involves systematically using AI and machine learning to create, test, verify and maintain synthetic datasets [8]. AI-based methods incorporate existing data to learn complex patterns, dependencies and distributions, unlike traditional data augmentation techniques that are based on simple transformations or predefined rules [9]. There are techniques like Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), or probabilistic generative models that could generate synthetic samples to serve various analytical purposes that are able to replicate the appearance of real samples [10-13]. These methods offer ways to increase the availability of data, aid in model building, allow for controlled sharing of data and minimize reliance on particular real-world data [14]. However, the value of synthetic data is contingent upon its statistical accuracy, analytical value, variation and privacy safeguards.

Although recent progress has been made in generating high fidelity synthetic data is still a difficult task [15]. Synthetic datasets might lack important relationships between attributes; might not capture minority class characteristics; and might not capture unusual observations [16]. Moreover, if the synthetic records turn out to be extremely realistic, this may lead to privacy issues when synthetic records bear a striking resemblance to the original training examples. [17]. However, data quality problems, varying validation processes and lack of lifecycle management may also restrict the use of synthetic data in intelligent information systems [18]. Thus, an integrated approach is needed to link the generation of synthetic data with systematic validation, quality assurance, privacy assessment and good data management to enable proper data use [19]. Such a framework should assess the realism of generated records as well as the suitability of generated records for downstream machine learning tasks.

To meet these requirements, the proposed framework for generating, validating and managing high fidelity data for intelligent information systems, “AI-Driven Synthetic Data Engineering,” provides an integrated framework [20-22]. The framework has three main phases: synthetic data generation, validation of synthetic data and management of synthetic data. In the generation stage, the original dataset is analysed, cleaned, transformed and fed to a suitable generative model based on AI [23]. At the validation stage, the data produced is

assessed by statistical similarity, attribute relationships, distributional consistency, data completeness and downstream predictive performance. Before sharing or deployment, it is also essential to assess privacy and disclosure risks with the synthetic data [24]. Finally, at the management stage, data documentation, version control, storage, reproducibility and continuous data quality monitoring are key. These steps build a robust pathway to the creation of dependable and valuable synthetic datasets.

The Adult Census Income Dataset could be used as an experimental dataset to illustrate the applicability of the proposed framework [25]. This data set encompasses socio-economic and demographic information, including age, education, occupation, working hours, among others, with the goal variable being the income category [26]. Its property gives it a proper test setting to conduct experiments on the generation of synthetic data with mixed numerical and categorical features. Once the data is pre-processed, an appropriate generative model could be used to learn the underlying patterns of the data and to create synthetic records [27]. The resulting data set could then be analysed by comparing the statistical characteristics of the resulting dataset with the original one and by assessing the effectiveness of the resulting data set for income classification tasks [28]. There are reported measures of predictive utility (e.g., accuracy, precision, recall and F1-score) and other measures of data quality (e.g., statistical assessments and privacy assessments). The study aims the following study objectives:

- To develop an AI-driven synthetic data generation framework that produces realistic and representative synthetic datasets using appropriate generative models.
- To pre-process and analyse the Adult Census Income Dataset by handling missing or inconsistent values, encoding categorical attributes and preparing numerical features for synthetic data generation.
- To evaluate the quality of synthetic data using statistical similarity measures, distributional comparisons and feature-correlation analysis.
- To assess the predictive utility of synthetic data by training and evaluating machine learning classifiers using metrics such as Accuracy, Precision, Recall and F1-score.
- To compare the proposed framework with existing approaches using clearly defined evaluation criteria to identify its strengths, limitations and potential improvements.

2. Related Work

Intelligent reasoning, collaboration capabilities and tool-based decision-making were the major advantages of autonomous LLM agents, but autonomous LLM agents also face significant challenges related to security, trust, authorization, communication, and verification. Yao et al., 2022 [29] proposed their reasoning-and-action generation framework ReAct, where reasoning and action were generated to interact with external environments, however, the ReAct framework did not address key security aspects, including secure communication between agents, cryptographic identity, quantum resistance, and formal security guarantees. While AutoGen Wu et al., (2023) [30] was an important step in

customizable LLM agents that could interact and exchange information with humans and external tools, it doesn't solve the problems of post-quantum authentication, dynamic trust evaluation and risk-based privilege adaptation. Greshake et al., (2023) [31] highlighted the dangers posed by IPI, presenting a method for exploiting external data to cause the LLM to behave in ways that were unintended. Wallace et al., (2024) [32] focused on privilege differentiation using instruction hierarchy and Huang et al., (2024) [33] highlighted the importance of verifying, validating, monitoring at runtime and formalizing trustworthy LLM systems. Overall, these studies confirm that it was important to secure the execution of agents, to control their privileges and to formally verify their trustworthiness, but most of these works consider only a single agent or a single type of security violation rather than provide full coverage for distributed multi-agent workflows.

Recent studies have also reinforced the security aspect by analyzing the capabilities of agents, assessing risk and exploring post-quantum cryptography. AgentDojo, introduced by Debenedetti et al., in (2024) [34], comprised 97 authentic tasks and 629 security test cases designed to evaluate prompt-injection attacks and defences in agent-utilizing tools, whereas Agent Security Bench (ASB), developed by Zhang et al., in (2025) [35], featured 10 scenarios, 10 agents, more than 400 tools, 23 attack/defense strategies, and 8 assessment metrics. Alquwayfili et al., (2025) [36] came up with Quantigence, which integrated multi-agent orchestration, cryptographic analysis, threat modelling, standards compliance and risk assessment, with tool-supported processing bringing the accuracy up to 98% and the complex-query rubric coverage to 89%. Drammeh et al., (2025) [37] showed that multi-agent orchestration was effective for incident response, with a high actionable recommendation rate (100% vs. 1.7% for single-agent systems). Specifically, Campbell et al., (2026) [38] focused on migration of PQC for agentic AI and pointed out the issues of confidentiality and integrity that may arise with future quantum attacks, highlighting the migration of critical trust anchors. Although these advances have been made, there was significant scatter and overlap in the literature on orchestration, attack detection, benchmarking, risk assessment, formal verification and PQC migration. A clear gap therefore exists in integrating post-quantum-resilient identity and communication, dynamic risk-aware trust and authorization, continuous security monitoring and formal verification within a single multi-agent LLM orchestration framework

3. Research Methodology

This study suggests an AI-Native Autonomous Cyber Defense Framework that combines Multi-Agent Large Language Models (LLMs) with a Post-Quantum Zero Trust Architecture (PQ-ZTA) to deliver intelligent, adaptive and resilient cyber defence against emerging cyber threats. The approach brings together independent threat detection, decision-making processes, ongoing trust verification and quantum secure cryptographic assurance. There are six major phases of the overall framework: Data Acquisition, PreProcessing, Multi-Agent Intelligence, Threat Analysis, PostQuantum Zero Trust Enforcement and Autonomous Response. The study method is shown in figure 2.

3.1 Cybersecurity Data Collection

The CICIDS2017 dataset [39], developed by the Canadian Institute for Cybersecurity, serves as a well recognised benchmark in cybersecurity research. The dataset was established to rectify the deficiencies of older datasets by integrating realistic network traffic and current attack scenarios. The dataset produced in a regulated setting contained both innocuous and harmful traffic, resembling real-world conditions. The dataset encompasses a substantial array of attacks, including brute force (FTP and SSH), DoS and DDoS, botnet operations, online assaults (SQL injection and XSS), infiltration, and port scanning. In addition to PCAP files, CICIDS2017 provides flow-based characteristics derived using CIC-Flow-Meter, which encompass over 84 statistical parameters for each flow (as illustrated in Table 2). Owing to its magnitude, diversity, and realistic attributes, it serves as an exceptional dataset for evaluating deep learning-driven cloud-edge cybersecurity threat detection systems. The division of the proposed dataset into training and testing sets is presented in Table 1.

Table 1: Division of Training and Testing for the CICIDS2017 Dataset

Category	Class	Flow Count	Percentage (%)	Training (70%)	Testing (30%)
Benign	Benign	2,273,097	76.75	1,591,167	681,929
DDoS	DDoS	231,073	7.802	161,751	69,321
DoS	Heartbleed	11	0.0003	7	3
DoS	DoS Slowloris	5,796	0.1957	4,057	1,738
DoS	DoS GoldenEye	10,293	0.3475	3,087	7,205
DoS	DoS SlowHTTPTest	5,499	0.1856	3,849	1,649
DoS	DoS Hulk	231,073	0.0392	161,751	69,321
Web Attack	SQL Injection	5,796	0.2121	4,057	1,738
Web Attack	Brute Force	7,938	0.2906	5,556	2,381
Web Attack	XSS	5,897	0.2158	4,127	1,769
Infiltration	Infiltration	10,293	0.3768	7,205	3,087
Port Scan	Port Scan	158,930	5.8184	111,251	47,679
Brute Force	FTP-Patator	1,769	0.2906	1,238	530
Brute Force	SSH-Patator	5,897	0.2158	4,127	1,769
Bot	Bot	1,966	0.0719	1,376	589

3.2 Data Pre-processing

The Data Pre-processing Layer is crucial to the deep learning framework as it guarantees the model sufficiently readies unrefined input data for training, evaluation, and real-time inference [30]. The calibre and arrangement of input data profoundly influence the efficacy of deep learning models. An effective preprocessing technique enhances model accuracy and reduces the risk of overfitting, facilitating improved generalisation to novel data. This stage encompasses normalisation, feature extraction, sequence formatting, and encoding of categorical variables. Figure 3 illustrates the structure of the Data pre-processing layer.

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

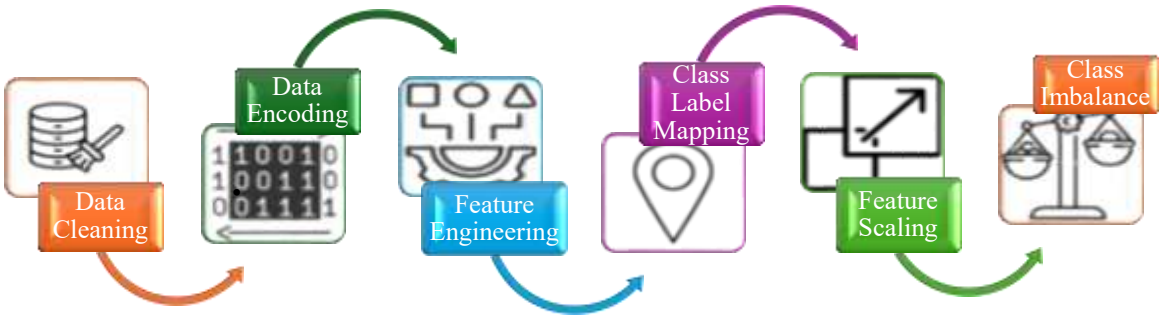


Figure 1: Data-preprocessing layer

- **Data Cleaning**

Data cleansing procedures eliminate extraneous information, redundancies, and absent values to enhance data integrity. Incomplete entries are rectified by statistical imputation, so guaranteeing that deep learning models could be trained efficiently in a uniform and dependable fashion.

$$x_i = \begin{cases} \bar{x}, & x_i \; missing \\ x_i, & otherwise \end{cases}$$

- **Data Encoding**

Data encoding transforms categorical variables into numerical formats using one-hot or label encoding, enabling neural networks to handle protocol types, services, and flags without introducing ordinal bias.

$$c_j \rightarrow [0,0,\dots,1,\dots,0]$$

- **Feature Engineering**

Feature engineering involves the selection and transformation of pertinent attributes such as packet rate and flow length to reduce dimensionality while enhancing the discriminative capacity for precise cyber-security threat identification.

$$MI(f,y) = \sum p(f,y) log \frac{p(f,y)}{p(f)p(y)}$$

Table 2: 84 Feature attributes

Feature Name	Feature Name	Feature Name	Feature Name
Flow ID	Flow Packets/s	Fwd Packets/s	Fwd Avg Packets/Bulk
Source IP	Flow IAT Mean	Bwd Packets/s	Fwd Avg Bulk Rate
Source Port	Flow IAT Std	Min Packet Length	Bwd Avg Bytes/Bulk
Destination IP	Flow IAT Max	Max Packet Length	Bwd Avg Packets/Bulk
Destination Port	Flow IAT Min	Packet Length Mean	Bwd Avg Bulk Rate
Protocol	Fwd IAT Total	Packet Length Std	Subflow Fwd Packets
Timestamp	Fwd IAT Mean	Packet Length Variance	Subflow Fwd Bytes
Flow Duration	Fwd IAT Std	FIN Flag Count	Subflow Bwd Packets
Total Fwd Packets	Fwd IAT Max	SYN Flag Count	Subflow Bwd Bytes
Total Backward Packets	Fwd IAT Min	RST Flag Count	Init_Win_bytes_forward
Total Length of Fwd Packets	Bwd IAT Total	PSH Flag Count	Init_Win_bytes_backward
Total Length of Bwd Packets	Bwd IAT Mean	ACK Flag Count	act_data_pkt_fwd
Fwd Packet Length Max	Bwd IAT Std	URG Flag Count	min_seg_size_forward
Fwd Packet Length Min	Bwd IAT Max	CWE Flag Count	Active Mean
Fwd Packet Length Mean	Bwd IAT Min	ECE Flag Count	Active Std
Fwd Packet Length Std	Fwd PSH Flags	Down/Up Ratio	Active Max
Bwd Packet Length Max	Bwd PSH Flags	Average Packet Size	Active Min
Bwd Packet Length Min	Fwd URG Flags	Avg Fwd Segment Size	Idle Mean
Bwd Packet Length Mean	Bwd URG Flags	Avg Bwd Segment Size	Idle Std
Bwd Packet Length Std	Fwd Header Length	Fwd Header Length	Idle Max
Flow Bytes/s	Bwd Header Length	Fwd Avg Bytes/Bulk	Idle Min

- **Class Label Mapping**

Class label mapping organises detailed attack labels into a uniform category to streamline the classification process and improve model generalisation across various attack types and typical network traffic patterns.

$$y' = g(y), \quad y' \in \{1, \dots, C\}$$

- **Feature Scaling**

Feature scaling standardises numerical variables to a uniform range by standardisation or min-max normalisation, hence preventing high-magnitude features from overshadowing others and facilitating quicker convergence of the neural network during training.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- **Class Imbalance Handling**

Addressing class imbalance employs resampling or cost-sensitive techniques to equalise the minority and majority classes, thus enhancing the detection of infrequent cyber-attacks and reducing the bias toward typical traffic samples.

$$x_{new} = x_i + \lambda(x_j - x_i), \lambda \in (0,1)$$

3.3 AI-Native Multi-Agent LLM Layer

AI-Native Multi-Agent LLM Layer is the intelligent decision-making component of the proposed autonomous cyber defense system, which integrates multiple specialized Large Language Model (LLM) agents to process security events, reason with contextual

information and produce collective responses in real-time [33]. The proposed architecture is different from traditional single-model architectures by spreading cybersecurity responsibilities across domain-specific agents such as network monitoring, threat intelligence, vulnerability assessment, incident response and Zero Trust policy management [34]. Intermediate results from each agent in a multi-agent system are exchanged via an orchestration layer in order to allow agents to collectively reason and reduce false-positive results [35]. Let the incoming context of cybersecurity concerns be represented by $X = \{x_1, x_2, \ldots, x_n\}$ which refers to individual network events, authentication logs, endpoint activity or threat intelligence records. The contextual embedding created by an LLM agent is represented as:

$$H_i = f_{\text{LLM}}(X_i),$$

where H_i is the semantic representation of the input context X_i and $f_{\text{LLM}}(\cdot)$ denotes the transformer-based language model responsible for contextual reasoning [36]. Each autonomous agent evaluates the processed context and predicts the probability of different cyber threat categories using a softmax classifier:

The semantic representation of the input context X_i is represented as H_i and $f_{\text{LLM}}(\cdot)$ represents the transformer-based language model used to reason about the context [36]. Every autonomous agent makes a prediction about each of the possible categories of cyber threats according to the processed context, based on a softmax classifier.

$$P(y = k \mid H_i) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}},$$

Where z_k is the output logit value of threat class k and K is the number of threat classes [37]. Each agent's confidence weighted decision, computed as $R_i = P(y \mid H_i)$. The outputs of all N collaborating agents are combined in the orchestration layer, which provides a unified security decision:

$$R_{\text{final}} = \sum_{i=1}^{N} w_i R_i,$$

subject to

$$\sum_{i=1}^{N} w_i = 1,$$

The confidence weight w_i is assigned to the i^{th}h agent based on its past accuracy and context. Final autonomous decision is made using the threat class having the most confident aggregated threat,

$$\hat{y} = \arg\max(R_{\text{final}}).$$

To continuously improve the intelligence of the framework, the model parameters are updated using gradient descent after each validated incident:

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t),$$

Here, θ represents the trainable parameters of the LLM agents, η is the learning rate and L(θ) is the optimization loss. The AI-Native Multi-Agent LLM Layer facilitates accurate threat interpretation, adaptive decision-making, autonomous cyber defense and quick adaptation to new attack strategies while ensuring scalability and resilience in complex enterprise environments through collaborative reasoning, contextual understanding and adaptive learning [38].

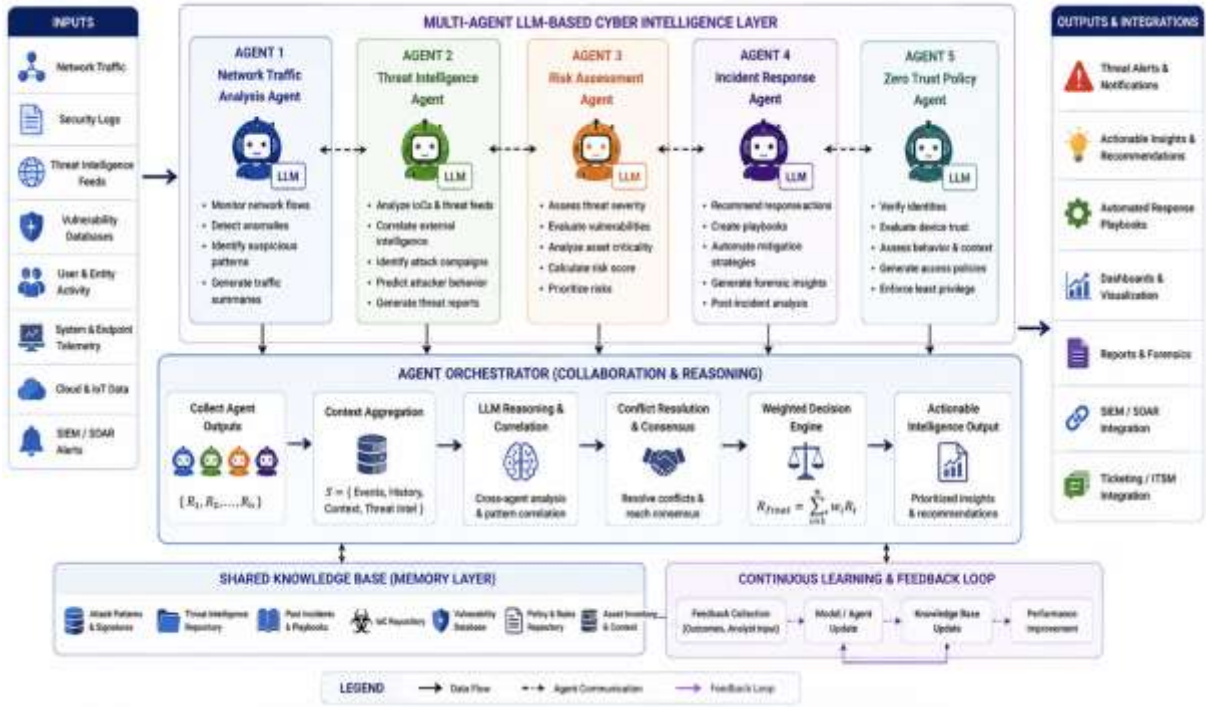


Figure 2: Multi-agent LLM-based cyber intelligence layer

3.4 Post-Quantum Zero Trust Architecture

Once an attack is detected, all communication requests are continuously verified for Zero Trust. There is no default trust for user, device, or application [39]. The framework verifies identities, validates endpoint security, analyzes behavioral context and re-evaluates trust before and maintains access.

The framework includes NIST-standardized post-quantum cryptographic algorithms that would guard against future threats from quantum computers:

- **ML-KEM (CRYSTALS-Kyber)** for quantum-resistant key establishment [40].
- **ML-DSA (CRYSTALS-Dilithium)** for digital signatures, authentication and integrity verification [41].

Encrypted communication is expressed as:

$$C = Enc_{ML-KEM}(M, K)$$

where:

- M= Original message
- K= Quantum-resistant session key
- C= Encrypted ciphertext

Digital signatures are generated as:

$$S = Sign_{ML-DSA}(M, SK)$$

Where SK is the sender's private signing key. Signature verification is performed using the corresponding public key to ensure authenticity and integrity.

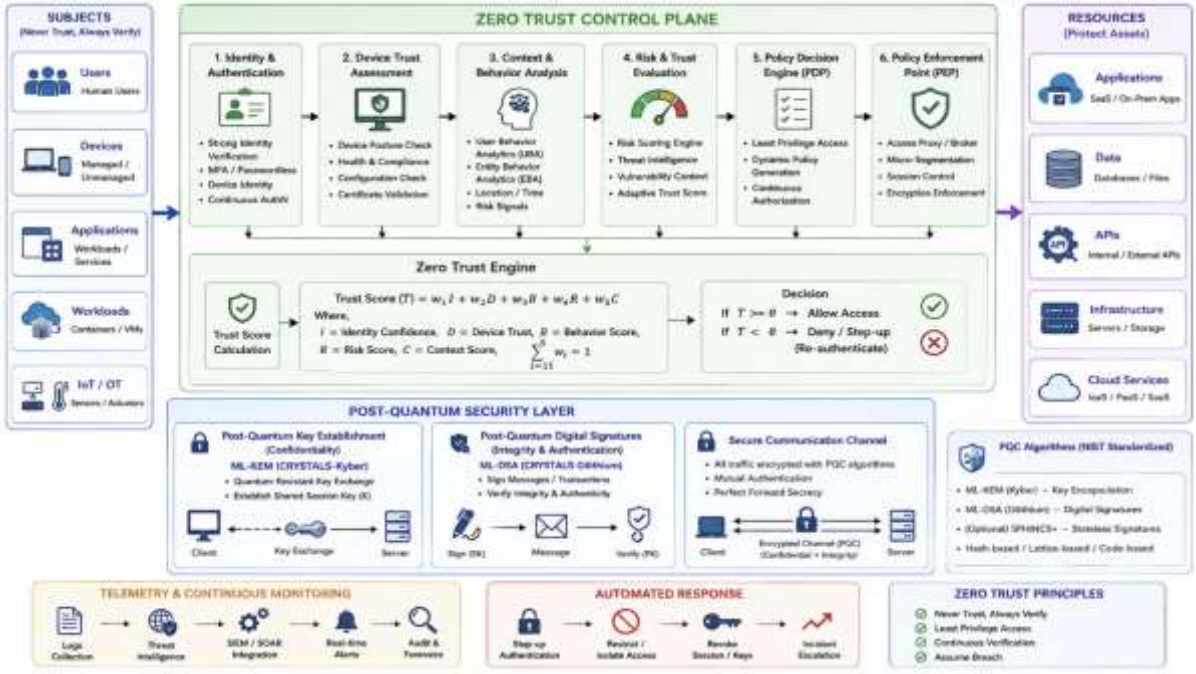


Figure 3: Post-Quantum Zero Trust Architecture

3.5 Evaluation Metrics

The proposed framework is evaluated using the following metrics:

Threat Detection

The model's performance was evaluated using Equations (1)–(5), where $A_{accuracy}$, $F1_{score}$, $P_{precision}$ and R_{recall} are the relevant variables .

- "TP (true positive): If the model predicts Norm, it is the accurate response.
- FP (false positive): If the model predicts Norm, the accurate response is Attack.
- TN (true negative): If the model predicts Attack and this is the right response.
- FN (false negative): If the model predicts Attack, the accurate response is Norm".

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TP+FN+FP+TN}} \quad (1)$$

$$\text{Precision} = \frac{NA}{NA+FP} \quad (2)$$

$$\text{Recall} = \frac{NA}{FN+NA} \quad (3)$$

$$\text{F1 score} = \frac{2\times\text{precision}\times\text{recall}}{\text{recall}+\text{precision}} \quad (4)$$

$$MCC = \frac{TP\times TN - FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

Response Performance

$$MTTD = \frac{\sum_{i=1}^{N}(t_{detect,i} - t_{start,i})}{N} \quad (6)$$

$$MTTR = \frac{\sum_{i=1}^{N}(t_{response,i} - t_{detect,i})}{N} \quad (7)$$

System Performance

$$\text{Throughput} = \frac{\text{Number of Processed Flows}}{\text{Processing Time}} \quad (8)$$

$$\text{Latency} = t_{output} - t_{input} \quad (9)$$

$$\text{Memory}(\%) = \frac{\text{Used Memory}}{\text{Total Memory}} \times 100 \quad (10)$$

$$\text{CPU}(\%) = \frac{\text{CPU Time Used}}{\text{Total CPU Time}} \times 100 \quad (11)$$

Zero Trust Performance Metrics

$$\text{Trust} = w_1 I + w_2 D + w_3 B + w_4 R \quad (12)$$

$$\text{Access Accuracy} = \frac{\text{Correct Access Decisions}}{\text{Total Access Requests}} \quad (13)$$

4. Results & Analysis

The performance of the proposed intrusion detection model is assessed with respect to the other existing CICIDS2017 based models. The class distribution of the proposed cyber-threat classification experiment for development sample CICIDS2017 is shown in Figure 4. DDoS, BENIGN and PortS could each have ~20,000 records, with DoS Hulk having ~14,600 records and DoS GoldenEye having ~10,300 records. The other categories have smaller volumes, especially for the categories: Infiltration, SQL Injection and Heartbleed. The development sample had been limited to around 20,000 records per class, so these are not necessarily the total amount of records contained in the CICIDS2017 archive. The distribution is used for the evaluation of the classifier over common and less-frequently-observed network traffic categories.

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

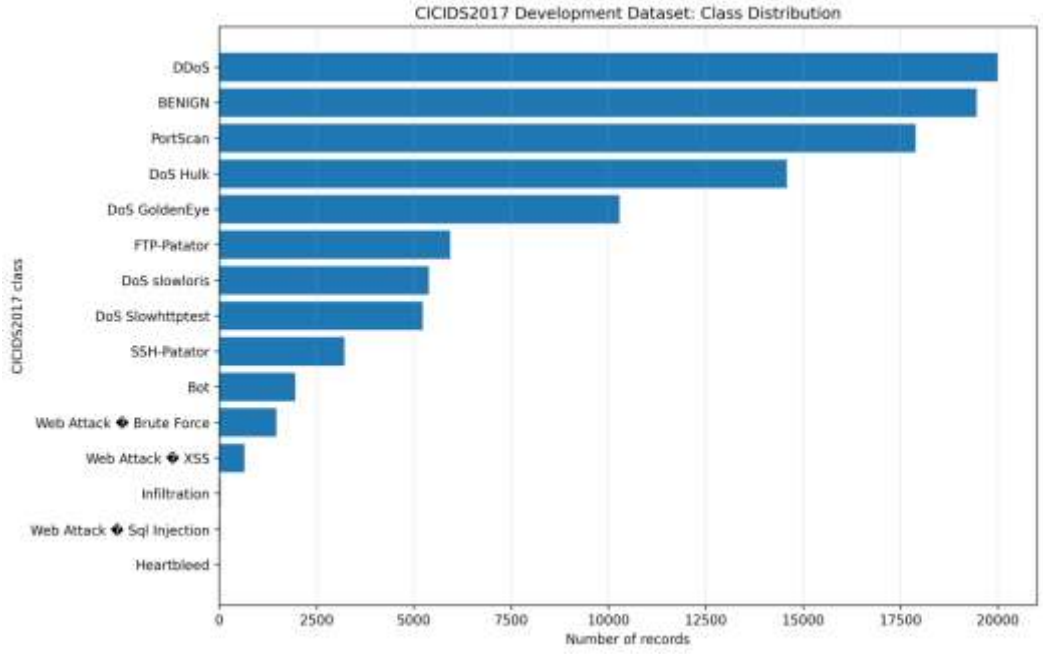


Figure 4: CICIDS2017 Development-Sample Class Distribution

The sample in Figure 5 demonstrates the impact of duplicate removal on a sample of its development. There were 119616 data points in the sample, but 13490 data points were found to be duplicates and removed from the sample. This is about an 11.28% cut. When comparing, the picture that emerges is that the number of duplicate records in each category affected the representation of the individual categories differently. The resulting data set is used to build the model and test the trained model. This preprocess stage improves in this way to remove repetitive observations from the experimental sample and for a more reliable estimation of classification performance on the preprocessed data.

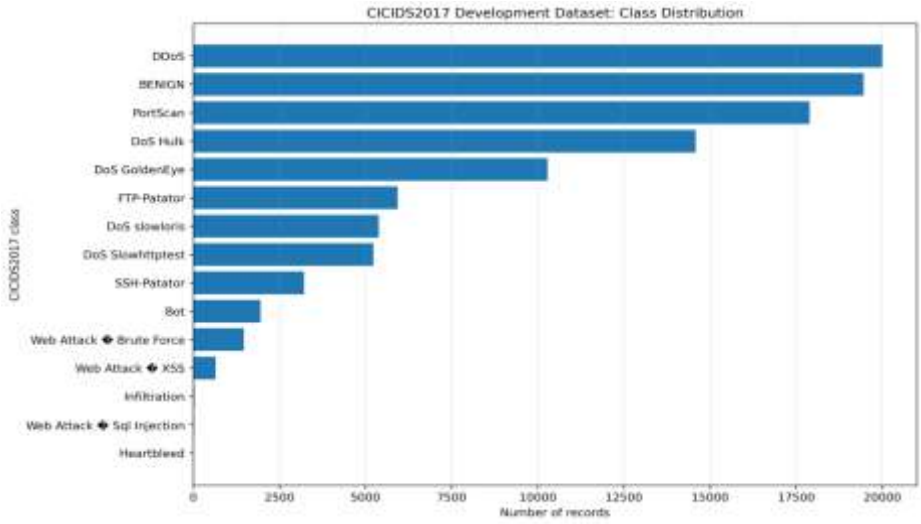


Figure 5: Class Counts Before and After Duplicate Removal

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

Figure 6 shows the distribution of the 15 traffic classes in the training and test partitions. For the cleaned dataset, 106,126 records are available of which 74,288 are training records and 31,838 are test records, with the proportion being 70:30. The amount of records contributed to the test partition has increased from 5,000 to 6,000 by DDoS, 5,837 by BENIGN and 5,367 by PortScan. In comparison, Infiltration has 11 records, SQL Injection has six and Heartbleed has three. The feature-label integrity checks ensured that the matching feature columns across the partitions matched the corresponding records. Class-support values serve as a context to be used when understanding the evaluation metrics of the model at both the aggregate level and at the class level.

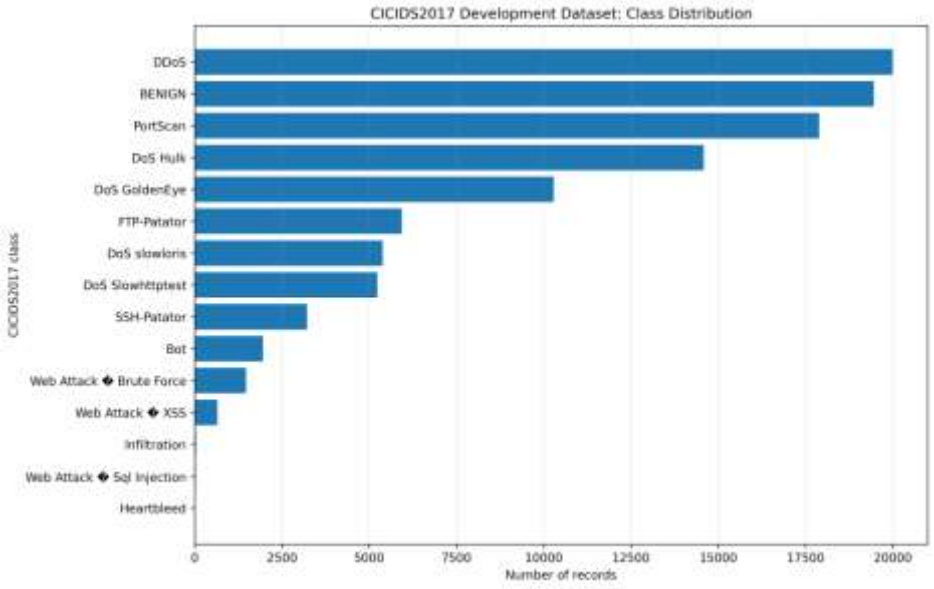


Figure 6: Training and Test Partition Class Distribution

The confusion matrix from the Random Forest (RF) classifier on 31,838 test records is shown in Figure 7. Overall, the model's accuracy was 99.17% and the number of mispredictions was 265. The correct classification of the records were 5,993 were classified as DDoS, 5,824 were classified as BENIGN, 5,363 were classified as PortScanning. Web Attack XSS was the most common misclassification type, as were Web Attack Brute Force. In other words, there were 114 XSS records classified as Brute Force and 78 Brute Force records classified as XSS. These results show that the classification of the overall performance is very good and a specific distinction between the categories of web-attack that is worthy of exploration in future model refinement.

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

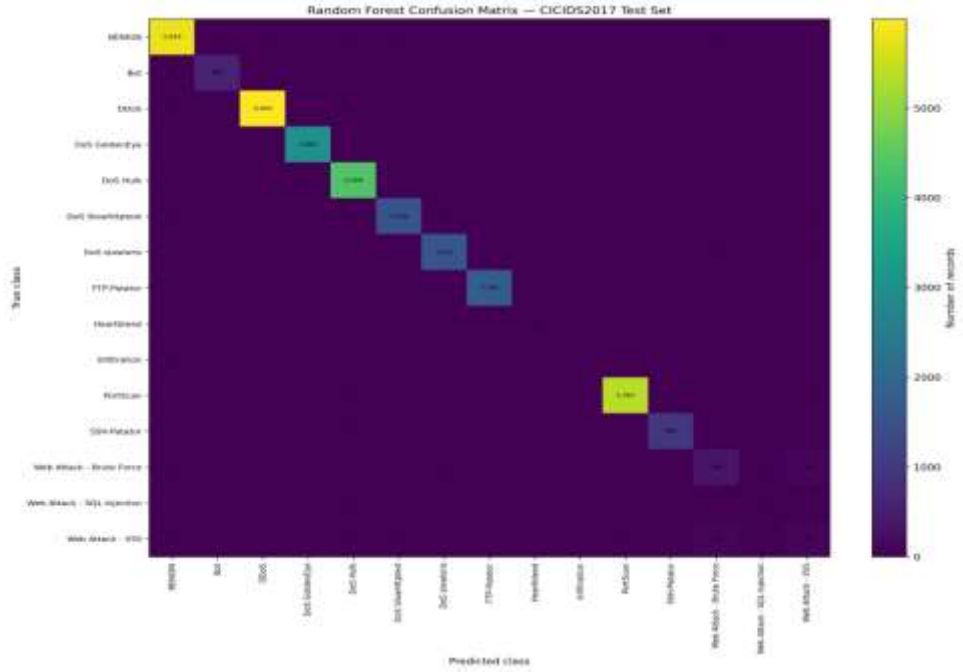


Figure 7: Random Forest Confusion Matrix

Precisions, recalls and F1-scores are compared over the 15 traffic classes in Figure 8. The macro F1 score for this model was 91.90% and the weighted F1 score was 99.14%. The scores for most classes having a relatively high number of tests were close to 1.00. Web Attack XSS was the more complex of classification patterns, with a recall of around 37.76% and F1 score of around 42.1%. Web Attack Brute Force had a recall of around 80.73% and F1 score of around 77.7%. A total of 11 test records were used to calculate infiltration recall, which came to 63.64%. These class-specific results supplement the overall accuracy and offer a more granular understanding of accuracy for attacks in certain classes.

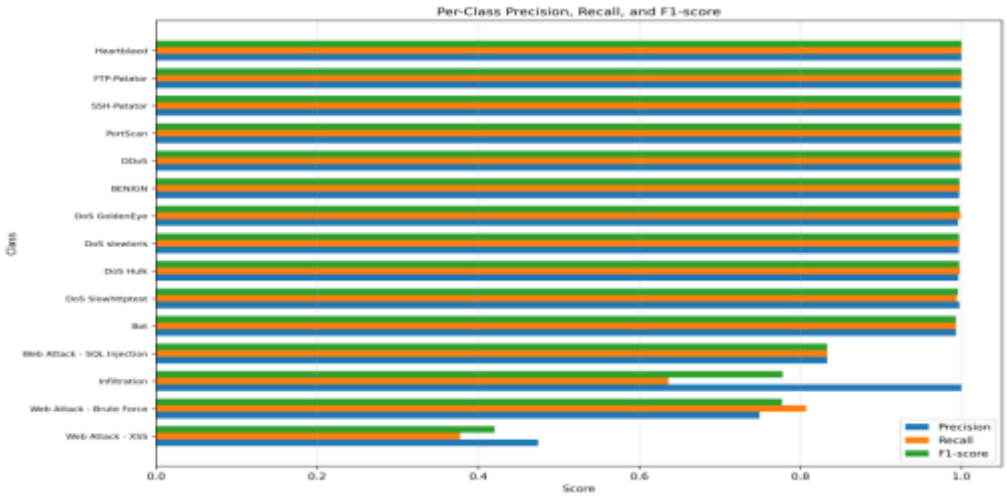


Figure 8: Per-Class Precision, Recall and F1-Score

Figure 9 displays the twenty most important features based on the fitted Random Forest model's impurity-based relevance. The most important feature is Destination Port (diameter),

with an importance score of about 0.065, followed by Init_Win_bytes_backward (diameter), with an importance score of about 0.055 and Init_Win_bytes_forward (diameter), with an importance score of about 0.037. Fwd Packet Length Max has an approximate 0.030 importance and Bwd Header Length has an approximate 0.027 importance. Other significant characteristics are packet rates, flow duration, inter-arrival time and packet-length statistics. The ranking shows that the information about the connection being connected, the TCP-window attributes and traffic-flow characteristics was important to the decisions made by the fitted classifier. These values are the relative importance of the features for a model, not causal relationships between features and cyberattacks.

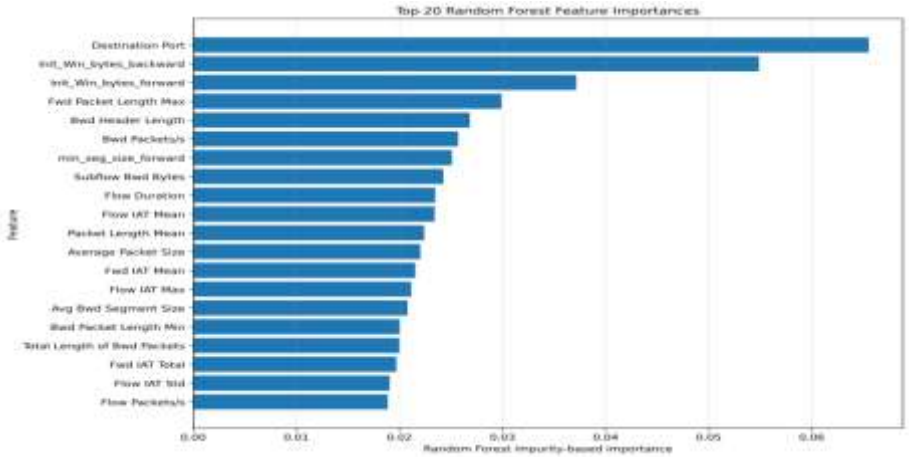


Figure 9: Top 20 Random Forest Feature Importance's

The maximum probabilities predicted for the correct and incorrect predictions are compared in figure 10. The average maximum predicted probability over the test set was about 0.9900. The mean confidence for correct predictions was 0.9927 and the mean for incorrect predictions was 0.6783. The distributions show that in general incorrect predictions have lower and more variable, levels of confidence than correct predictions. This separation indicates that prediction confidence may help prioritize records for additional assessment. However, the maximum predicted probability should not necessarily be interpreted as a calibrated estimate of the probability that a prediction is correct.

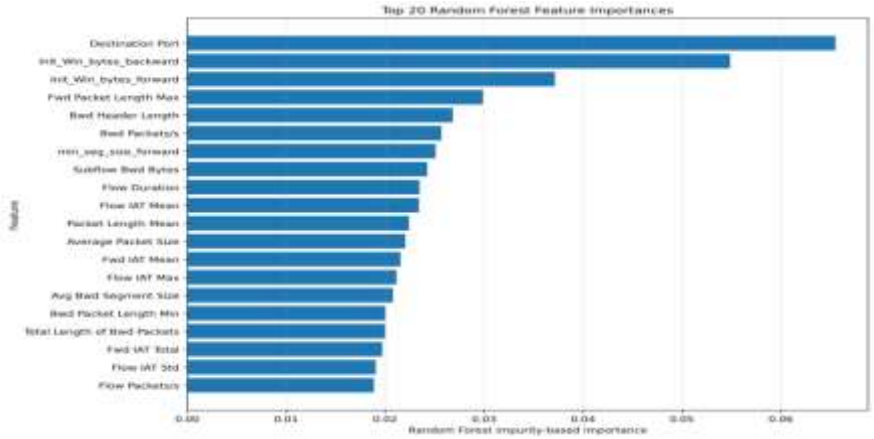


Figure 10: Prediction Confidence for Correct and Incorrect Classifications

The one-versus-rest receiver operating characteristic (ROC) curves for the 15 traffic classes are shown in Figure 11. The curves show the true positive rate vs. false positive rate for different thresholds in the classification process. The corresponding ROC-AUC results are around 1.000 for most classes, 0.998 for Web Attack Brute Force, 0.987 for Web Attack XSS and 0.916 for Web Attack SQL Injection. The results show high class-discrimination for the test set used. The results must be interpreted with class support as there are six test records for the SQL Injection attack and three for the Heartbleed attack. The ROC scores thus reflect performance on this experimental partition and not the true generalizability to other network settings.

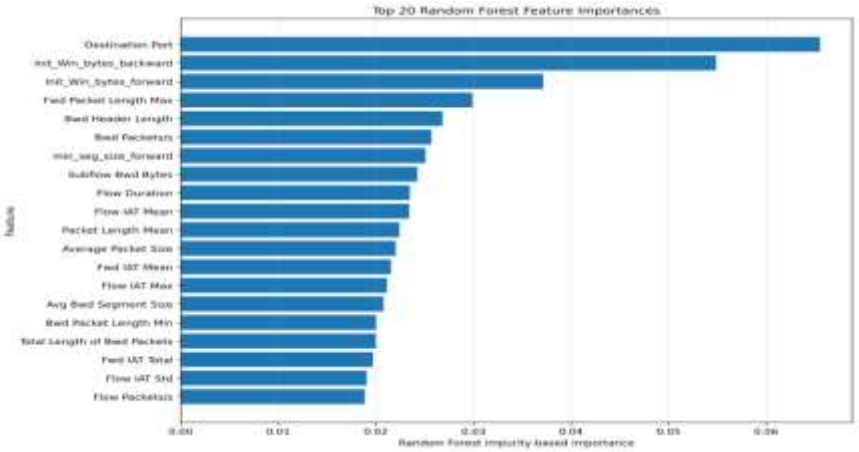


Figure 11: One-versus-Rest ROC Curves

Figure 12 illustrates the precision–recall performance of the individual traffic classes. The trade-off between precision and recall as the classification thresholds vary is represented by these curves. Many other classes reach a precision of 1.000 and the average precision for Web Attack XSS, Web Attack SQL Injection and Infiltration is about 0.449, 0.772 and 0.811, respectively. As seen on the XSS curve, the accuracy deteriorates more rapidly as the number of recalled XSS instances grows, suggesting that there are more false positives when trying to retrieve more XSS instances. This analysis complements the evaluation performed on the ROC curve which is on negative-class retrieval. The results are very helpful for determining performance for less common attack categories but are subject to healthy cautions in very few class sizes.

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 8, Issue No. 4, pp. 72 - 93, 2026
www.carijournals.org

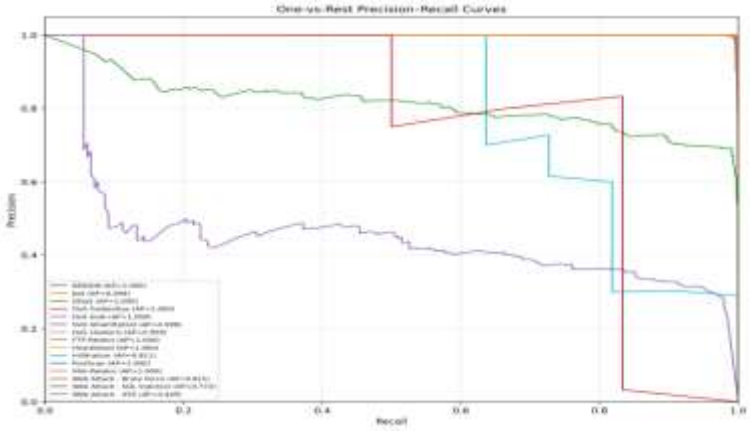


Figure 12: One-versus-Rest Precision–Recall Curves

The results in Table 13 show the ROC-AUC obtained for each individual traffic class. The graph shows an AUC of 1.000 for the majority of classes and an AUC of 0.998 for Web Attack Brute Force, 0.987 for Web Attack XSS and 0.916 for Web Attack SQL Injection. These results are reflective of good discrimination of the classes evaluated using ranking criteria. The lesser SQL Injection score indicates that it is a class to be examined more closely. The high ROC-AUC with the low recall value also indicates that the two metrics reflect different aspects of the system's performance. These values should be read in conjunction with the class supports and with care in relation to the overall detection capability of the model.

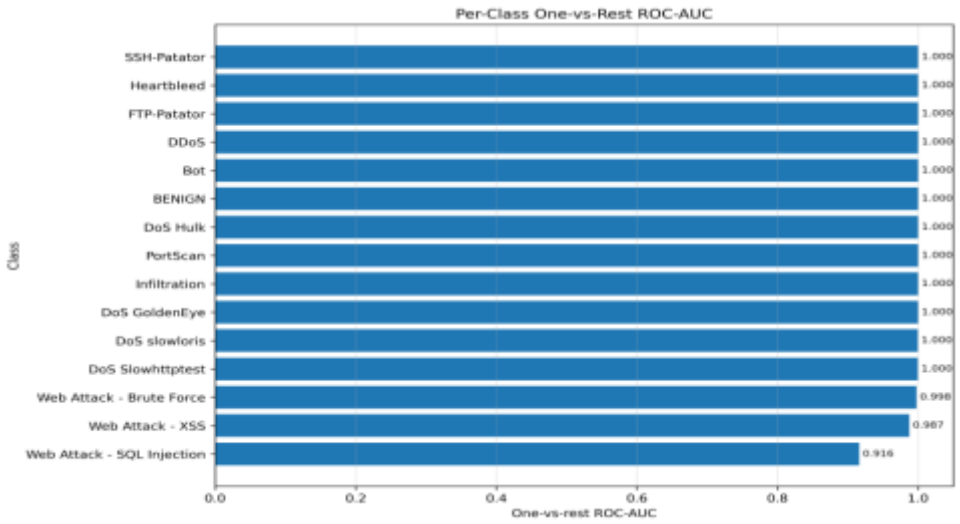


Figure 13: Per-Class ROC-AUC

Figure 14 shows the most common real-to-predicted class errors by the baseline classifier. The highest amount of error is for Web Attack XSS, which falls under the Web Attack Brute Force category with 114 records. The other 78 records are classified as reverse confusion: Brute Force classified as XSS. The two types of errors together constitute 192 out of the 265 total errors, or about 72.45%. DoS Hulk is the next most common pair at eight records, DoS GoldenEye. This number of errors is a concentration of errors, suggesting that the overall

confusion pattern for the experiment is between the two classes of web attacks. The analysis identifies a direction to pay attention to in order to better discriminate between the related attack signatures.

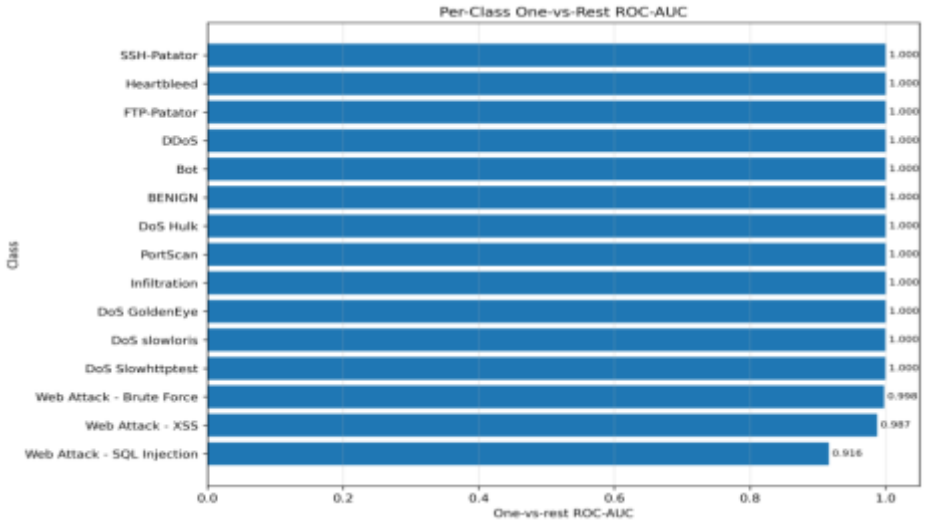


Figure 14: Major Baseline Misclassification Pairs

Figure 15 depicts the correlation among the confidence threshold, the percentage of baseline mistakes identified for additional evaluation, and the corresponding workload.Sixty-hundred-six of these records were selected (202 out of 265 baseline errors, 75.23%). The number of selected records was 888 and the number of captured errors was 231, which is 87.17% of the selected records. At 0.99, the selected set increased to 2,155 records, capturing 257 errors, or 96.98%. The results show that as the threshold increases, the number of records needing to be reviewed increases and so do errors that are captured. The results demonstrate the feasibility of confidence-guided review, where an operational threshold needs to be determined with data other than the validation data and the capacity for reviews needs to be factored in.

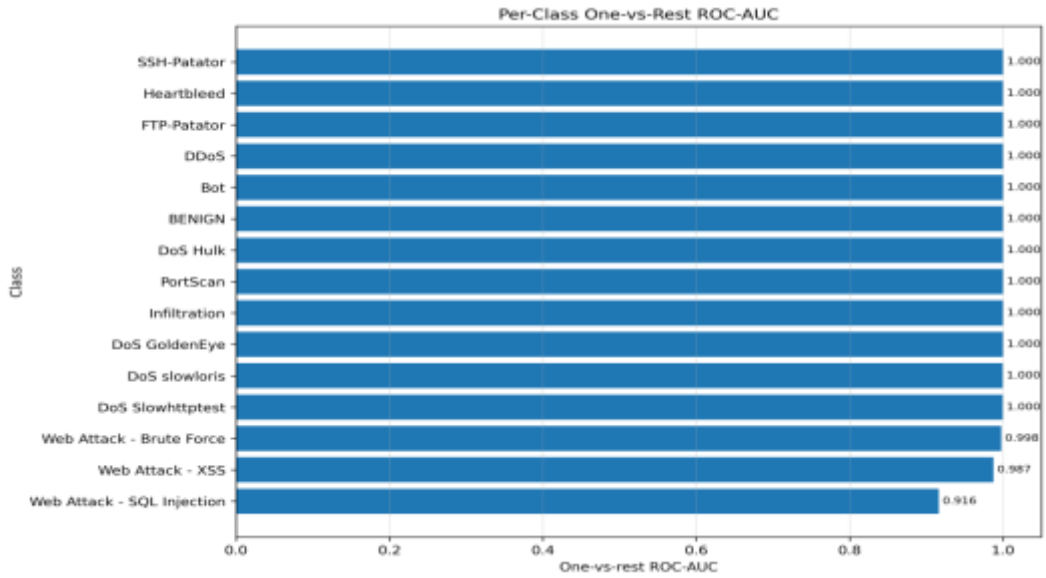


Figure 15: Review-Policy Trade-Off: Error Capture versus Workload

Table 3 displays the results of a comparison to other models found in the literature, which allowed us to delve deeper into the algorithm's correctness, where it could be seen that the proposed Random Forest model had an accuracy of 99.17%, which is the highest among the models presented. It achieved a higher accuracy than the original CICIDS2017 random forest model (96.12%), anomaly-based intrusion detection approach (98.30%) and logistic regression (98.78%) by 0.87, 3.05 and 0.39 percentage points, respectively. These results show a good performance in classification on the CICIDS2017 development sample. However, the overall superiority should be taken with a pinch of salt, as there are pre-processing differences, class distribution and differences in their evaluation protocols.

Table 3: Comparative Performance of Existing Intrusion Detection Methods on CICIDS2017

Author(s) & Year	Method / Architecture	Accuracy
Anomaly-Based Intrusion Detection (2022)	Random Forest feature selection with an ensemble of Naïve Bayes, QDA and ID3	98.30%
Pinto et al. (2025)	Random Forest using original CICIDS2017 flows	96.12%
Recent deterministic comparison (2025)	Logistic Regression on CICIDS2017, without SMOTE	98.78%
Proposed Work [2026]	Random Forest (300 trees), 15-class classification on CICIDS2017 development sample	99.17%

5. Conclusion

This study introduces a framework for autonomous artificial intelligence-enabled cyber defense features potentially aimed at incorporating intrusion detection, collaborative threat analysis, zero-trust access enforcement, and post-quantum cryptographic protection. The architecture integrates multi-agent large language model orchestration seamlessly into a permanent identity verification and security-driven response process, offering a cohesive and coordinated design for intelligent cybersecurity operations. The experimental validation with CICIDS2017 validated the accuracy of the Random Forest classifier as 99.17%, a weighted F1-score of 99.14% and a macro F1-score of 91.90% in 15 categories of network traffic. The results suggest good classification accuracy for the set used in the evaluation. However, the gap between the macro and the weighted ones F1-scores shows some variation in performance within the attack categories. Especially the Web Attack XSS hit in particular that had a score around 37.76% showed how tricky it can be to spot some types of malicious traffic. However, interpretation of aggregate performance metrics is important, as a good example of this confusion is the situation with Web Attack Brute Force and XSS. The main contribution of the present study is an integrated approach to a multi-agent cybersecurity intelligence and the enforcement of the zero-trust domain and the use of post-quantum security mechanisms. This planning design would give a framework for coordinating threat assessment, access control and security responses. The reported experiments demonstrate the classification accuracy of the Random Forest component but these are not individual tests of the overall system's effectiveness.

References

1. Soundarapandiyan, Praveen Sivathapandi Rajalakshmi. "AI-driven synthetic data generation for financial product development: Accelerating innovation in banking and fintech through realistic data simulation." (2022).
2. Maxwell, Brian. "Next Generation AI Systems for Data Intelligence Security and Scalable Cloud Architectures." *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)* 9, no. 2 (2026): 753-783.
3. Koussouris, Sotiris and Dimitrios Panopoulos. "The Convergence of Data-Centric Engineering and Adaptive AI Operations: A Comprehensive Framework for Trustworthy Hybrid Systems." *Technical Annals* 2, no. 3 (2026).
4. Nieminen, Walter, Haben Gebreweld, Arto Liuha, Mikko Nissinen, Mario Verdugo, Aki Mikkola and Antero Kutvonen. "Synthetic data for predictive maintenance: A systematic review and framework for Industry 4.0 applications." *Journal of Intelligent Manufacturing* (2026): 1-23.
5. Sims-Handcock, Chad. "Design and Evaluation of an AI-Driven Pipeline for Synthetic Tabular Data Generation." PhD diss., Stellenbosch: Stellenbosch University, 2026.
6. Polczynski, Stephanie, John D. Hastings, Varghese Vaidyan and Kyle Korman. "Bridging the Smart City Cybersecurity Data Gap Through AI-Driven Synthetic Dataset Generation." In *2026 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pp. 1-10. IEEE, 2026.
7. Waseem, Hafiz Muhammad, Saif Ul Islam, Nikolaos Matragkas, Gregory Epiphaniou, Theodoros N. Arvanitis and Carsten Maple. "Review of generative AI for synthetic data generation: a healthcare perspective." *Artificial Intelligence Review* 59, no. 2 (2025): 55.
8. Strauss, Shawn, Robert Corby and Steven Brown. "Diffusion-Generated Synthetic Medical Data for Robust Anomaly Detection and Drug Manufacturing Optimization in AI-Driven Hospitals."
9. Pasculli, Giuseppe, Marco Virgolin, Puja Myles, Anna Vidovszky, Charles Fisher, Elisabetta Biasin, Miranda Mourby et al. "Synthetic data in healthcare and drug development: definitions, regulatory frameworks, issues." *CPT: Pharmacometrics & Systems Pharmacology* 14, no. 5 (2025): 840-852.
10. Fabuyi, Jumai. "Leveraging synthetic data as a tool to combat bias in artificial intelligence (AI) model training." *Available at SSRN 5082577* (2024).
11. Rahman, Mohammad Ishtiaque, Md Razuan Hossain, S. M. Sayem and Forhan Bin Emdad. "Exploring the role of synthetic data in the future of ai in healthcare: A scoping review of frameworks, challenges and implications." *Intelligence-Based Medicine* (2025): 100342.
12. Mendes, Jorge M., Aziz Barbar and Marwa Refaie. "Synthetic data generation." (2025).
13. Goldberg, Benjamin, Randall Spain, Kevin Owens, Jeremy Lanman, C. P. Kwon, Kevin Gupton and P. A. Butler. "A data strategy for data-driven training management: Artificial intelligence and the Army's synthetic training environment." In *Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*. 2023.

14. Velocia, Clarice and K. Kumar. "Synthetic data in Ai healthcare research: A review of use cases, benefits and risks." (2025).
15. Lazzaretti, André E., Farnoosh Rahmatian, Rodrigo J. Riella and Ashish Pandharipande. "High-Fidelity Sensing and AI for Power Grid Monitoring and Intelligence: A Review." *IEEE Sensors Journal* (2026).
16. Danish, Tanvir and Anil Kapure. "Advancing AI/ML Healthcare with Cloud-Enabled MLOps Generative AI and Scalable AI Workflows in Data Engineering." (2024).
17. Löhr, Andreas and Conor Haines. "Mastering Training Data Generation for AI-Integrating High-Fidelity Component Models with Standard Flight Simulator Software." In *PHM Society European Conference*, vol. 8, no. 1, pp. 7-7. 2024.
18. Yousif, Ibrahim. "AI-Driven Autonomous Manufacturing: A Novel Taxonomy, Vision-Guided Planning and Diversity-Aware Active Learning Frameworks." (2025).
19. Kavala, Yuvaraj. "Smart ERP: Scalable data engineering frameworks using artificial intelligence." *International Journal of Computational Mathematical Ideas (IJCMI)* 15, no. 3 (2023): 1248-1262.
20. Nguyen, Phat, Minjung Kim, Elaina Nichols and Hwan-Sik Yoon. "AI-driven digital twins for manufacturing: A review across hierarchical manufacturing system levels." *Sensors (Basel, Switzerland)* 26, no. 1 (2025): 124.
21. Goyal, Mandeep and Qusay H. Mahmoud. "A systematic review of synthetic data generation techniques using generative AI." *Electronics* 13, no. 17 (2024): 3509.
22. Mei, Ruo-Syuan, Guangze Li, Sixian Jia, Lu Huang, Li-Wei Shih, Jorge Arinez, Jeffery Abell and Chenhui Shao. "Synthetic data generation in smart manufacturing applications: A systematic review." *Available at SSRN 5726804* (2025).
23. Udechukwu, Lisa Mmesoma, Gbenro Charles Opeke, Emonena Patrick Obrik-Uloho, Olufunke Cynthia Metibemu and Rukayat Oluwabukola Olasege. "Information governance framework for AI-generated synthetic patient data in healthcare research: Balancing utility, privacy and algorithmic bias mitigation." *Asian Journal of Advanced Research and Reports* 19, no. 11 (2025): 30-45.
24. Gangadharaiah, Dhanush. "Synthetic data generation for enhanced computer vision applications: a CAD model and Blender approach." (2023).
25. Nadi, Faheem and Anil Kapure. *Cloud-Native AI/ML Data Engineering with Generative AI MLOps and Scalable AI Workflows for Healthcare Innovation*. 2024.
26. Thapa, Padam Jung. "Toward Robust Semantic Segmentation in Levee Infrastructure Monitoring: Enhancing Accuracy with High-Fidelity Synthetic Data and Ensemble Learning." (2025).
27. Richard, Heston, Jade Adrian and John Nicole. "GENERATIVE AI-DRIVEN HETEROGENEOUS DATABASE MIGRATION WITH INTEGRATED SECURITY VALIDATION AND SCHEMA HARMONIZATION."
28. Tiwari, Vaibhavi, Venkata Sai Ram Dasari and Krishanu Agrawal. "Synthetic data as a catalyst: Applications in healthcare, environment and beyond." In *2025 IEEE 16th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0417-0426. IEEE, 2025.

29. Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan and Yuan Cao. "React: Synergizing reasoning and acting in language models." arXiv preprint arXiv:2210.03629 (2022).
30. Wu, Qingyun, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang et al. "Autogen: Enabling next-gen llm applications via multi-agent conversation." arXiv preprint arXiv:2308.08155 (2023).
31. Greshake, Kai, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz and Mario Fritz. "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection." In Proceedings of the 16th ACM workshop on artificial intelligence and security, pp. 79-90. 2023.
32. Wallace, Eric, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke and Alex Beutel. "The instruction hierarchy: Training llms to prioritize privileged instructions." arXiv preprint arXiv:2404.13208 (2024).
33. Huang, Xiaowei, Wenjie Ruan, Wei Huang, Gaojie Jin, Yi Dong, Changshun Wu, Saddek Bensalem et al. "A survey of safety and trustworthiness of large language models through the lens of verification and validation." Artificial Intelligence Review 57, no. 7 (2024): 175.
34. Debenedetti, Edoardo, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer and Florian Tramèr. "Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents." Advances in neural information processing systems 37 (2024): 82895-82920.
35. Zhang, Hanrong, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang and Yongfeng Zhang. "Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents." In International Conference on Learning Representations, vol. 2025, pp. 35331-35366. 2025.
36. Alquwayfili, Abdulmalik. "Quantigence: A Multi-Agent AI Framework for Quantum Security Research." arXiv preprint arXiv:2512.12989 (2025).
37. Drammeh, Philip. "Multi-agent LLM orchestration achieves deterministic, high-quality decision support for incident response." arXiv preprint arXiv:2511.15755 (2025).
38. Campbell, Robert. "Post-Quantum Cryptography Migration for Agentic AI Systems." Computers 15, no. 7 (2026): 434.
39. https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

2026 by the Authors. This Article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/)