JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

A Self-Healing DDoS Detection Framework Using Generative AI and Transfer Learning

ASMA SHAHEEN[1], OSAMA REHMAN[2+], IBRAHIM M. H. RAHMAN[3] AND SOHAIB AHMED[2]
[1]*Department of Software Engineering, Bahria University, Karachi,Pakistan*
[2]*School of Computer Science, Taylor's University, Subang Jaya, Malaysia*
[3]*Information Technology, The Open Polytechnic of New Zealand, Wellington, New Zealand*
E-mail: asmashaheen.bukc@bahria.edu.pk, osama.rehman@taylors.edu.my, ibrahim.rahman@openpolytechnic.ac.nz, sohaib.ahmed@taylors.edu.my

Distributed Denial of Service (DDoS) attacks represent a major threat to network resilience, while also being considered among the most persistent and damaging attack vectors capable of crippling major systems, including critical infrastructure. The rapid expansion of DDoS attacks driven by increase in IoT devices, cloud computing resources, and 5G technologies, has outpaced traditional network defenses. Such conventional defense systems are further deemed ineffective with the emerging AI-generated DDoS attack patterns. Static mechanisms such as rule-based Intrusion Detection Systems (IDS) and conventional machine learning models fail to counter evolving DDoS attacks that exploit adaptive evasion techniques. This study introduces a self-healing DDoS detection framework that integrates Generative AI technique, namely Gaussian Copula Synthesizer, and transfer learning techniques for dynamic adaptability to emerging DDoS attacks. A Gaussian Copula Synthesizer generates high-fidelity synthetic traffic preserving real attack correlations, enabling continuous retraining of a Recurrent Neural Network (RNN) based model through iterative transfer learning and without catastrophic forgetting. Evaluated on the CIC-DDoS2019 dataset, the framework boosts detection accuracy from 64% to 96.8% across retraining cycles. Analysis highlights the model's adaptive learning behavior and performance stability. Despite added computational cost, the trade-off is justified by significant security gains.

Keywords: DDoS detection, generative AI, recurrent neural networks, synthetic data generation, intrusion detection system

1. INTRODUCTION

The digital transformation of society has created an increasingly interconnected world where business operations, personal communications and critical infrastructures rely heavily on network connectivity. This reliance has created unprecedented vulnerabilities to cyber-attacks, with DDoS attacks representing one of the most persistent and damaging threats to online services. The sophistication and scale of DDoS attacks have increased and evolved dramatically, with recent attack volumes exceeding 3 terabits per second, capable of overwhelming even robust network infrastructures [1].
The DDoS attack volume has expanded considerably through the proliferation of Internet of Things (IoT) devices, cloud computing platforms, and 5G networks [2]. Modern DDoS

Received ?, ?; revised ?, ?; accepted ?, ?.
+Corresponding author

campaigns employ increasingly sophisticated techniques, including multi-vector attacks, IoT botnets, and low-rate attacks that mimic legitimate traffic patterns [3]. These developments have exposed the limitations of traditional security solutions, which struggle to adapt to the dynamic nature of contemporary cyber threats.

Conventional DDoS detection mechanisms primarily fall into two categories: signature-based detection and anomaly-based detection. Signature-based systems maintain databases of known attack patterns but fail to recognize novel attack variants [4]. Anomaly-based systems establish baselines of normal behavior but generate excessive false positives in complex network environments [5]. Both approaches share a critical limitation, i.e. being fundamentally static implementations that cannot autonomously adapt to evolving attack patterns.

Machine learning-based solutions have shown promise in DDoS attacks detection but typically suffer from model decay as attack patterns evolve over time [6]. Retraining models on new data presents practical challenges, including the scarcity of labeled attack data, computational costs, and the risk of catastrophic forgetting [7]. This creates a gap in cybersecurity defenses, hence the need for systems that can continuously adapt to emerging threats without human intervention or to address performance degradation on previously learned DDoS attack patterns.

Existing methods often struggle to adapt to evolving attack patterns, leaving networks vulnerable to emerging DDoS attack strategies. Static approaches may lead to high false-positive and false-negative rates, reducing the reliability and efficiency of detection [8]. Consequently, there is a critical need for dynamic, intelligent defense mechanisms capable of adapting to evolving DDoS attack patterns.

This research addresses these challenges by proposing a self-healing DDoS detection framework that leverages Gaussian Copula Synthesizer and transfer learning to create continuously adaptive defense capabilities. Our work integrates a Gaussian Copula Synthesizer with transfer learning to enable continuous adaptation, allowing models not only to retain performance but also to evolve with emerging threat patterns advancing toward fully autonomous and resilient cybersecurity systems. Our work extends these developments by employing a Gaussian Copula Synthesizer optimized for network traffic and embedding it within a continuous learning pipeline. The gaussian copula-based approach has the potential to preserve the relationships between features while introducing greater diversity in the data distribution, producing statistically valid synthetic samples that improve knowledge retention during incremental transfer learning. Such capabilities prove vital in dealing with catastrophic forgetting issues often faced in incremental learning of the classification models.

The specific objectives of this work are given below:

1. To develop a synthetic data generation mechanism using Gaussian Copula models that can produce realistic network traffic patterns representing novel attack strategies.

2. To design an iterative transfer learning protocol that enables continuous model adaptation without catastrophic forgetting of previously learned attack patterns.

3. To implement and evaluate a recurrent neural network architecture optimized for temporal pattern recognition in network traffic flows.

4. To comprehensively analyze the performance improvements, learning dynamics, and computational trade-offs associated with the proposed approach.

2. RELATED WORK

AI-enabled Intrusion Detection Systems (IDS) have gained significant attention in recent years for having the potential to detect sophisticated DDoS attacks [9]. Several IDS tools have emerged with the potential to host an intelligent DDoS classification model built using AI algorithms [10]. AI-enabled DDoS classification models have been increasingly applied to network security across various domains, including deployment within intrusion detection and adversarial defense systems. Moreover, several studies have explored AI-driven mechanisms against poisoning attacks against DDoS classifiers, evasion, and API-based threats, showing its potential to strengthen future network security [11]. Among prominent cyber threats, DDoS attacks remain particularly disruptive, exploiting network vulnerabilities through botnets and malicious traffic to exhaust system resources [12, 13]. The evolving nature of emerging DDoS attacks has motivated continuous research on detection and mitigation methods.

Early DDoS detection relied on signature-based IDS [4], which compared incoming traffic with known DDoS attack signatures. Although effective for known DDoS threats patterns, such detection mechanisms often failed to detect novel or obfuscated attack patterns, such as those DDoS attacks that mimic behavior of flash events [14]. Anomaly-based IDS later emerged, while learning statistical baselines of normal traffic and identifying deviations as potential attacks [5]. However, such models often suffered from high false positives in dynamic networks. On the other hand, hybrid approaches combining both detection modes that improved coverage but still struggled against adaptive and evolving DDoS attacks [15].

The integration of machine learning based DDoS detection models within IDS enabled automated traffic classification as benign or DDoS attack traffic. Classical ML algorithms, such as Decision Trees, Support Vector Machines, and K-Nearest Neighbors, provided improved generalization but required manual feature engineering and frequent retraining of the classification models [6]. Deep learning methods, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated superior capability in modeling complex spatial–temporal dependencies in DDoS and benign network traffic data [16, 17]. Yet, these models demand extensive labeled datasets and computational resources, limiting real-time adaptability.

Generative AI techniques including Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), transformers, and large language models (LLMs) have gained traction in the realm of DDoS attacks for predictive analysis, anomaly detection, and adaptive defense [18]. GANs have been increasingly used for synthetic network traffic generation, data augmentation, and model evaluation [19, 20], while VAEs have shown effectiveness in identifying anomalies through compressed latent representations of normal traffic [21].

More recently, copula-based approaches have been proposed for generating statistically consistent synthetic normal traffic and cyber-attacks data, preserving inter-feature correlations crucial for realistic network traffic simulation [22, 23]. Such probabilistic mod-

els assume class-dependent data distributions and can accurately reproduce the statistical behavior of real network traffic flows [24]. In [25], an unsupervised intrusion detection framework combined GANs, Temporal Convolutional Networks (TCNs), and self-attention mechanisms to detect unknown cyber-attacks without labeled data, achieving competitive accuracy on benchmark datasets.

Recent research has emphasized self-healing and adaptive cybersecurity systems capable of detecting and correcting model degradation [9]. Frameworks in [7] addressed automatic identification of performance drift, while [26] presented lightweight adaptive models for IoT security. These studies primarily focus on degradation recovery rather than proactive model evolution. A summarized comparative analysis of the surveyed literature for AI-enabled DDoS attack detections is provided in Table 1 depicting the key problem solved, techniques adopted, results obtained and findings discovered.

3. THE PROPOSED METHOD

The overarching architecture of our self-healing DDoS detection framework implements a continuous cycle of adaptation, as illustrated in Figure 1. The system operates through four interconnected phases: (1) initial model training on real network data, (2) synthetic data generation representing novel attack patterns, (3) iterative model retraining through transfer learning, and (4) performance evaluation and feedback. This cyclic process enables the system to autonomously expand its detection capabilities while maintaining performance on previously learned patterns.

A detailed flow mechanisms of the proposed framework depicting the retraining and self-healing processes of the base model using synthetically generated datasets is presented by Figure 2.

3.1 Data Preprocessing and Balancing

The CIC-DDoS2019 dataset forms the basis of this study, providing extensive DDoS and benign network traffic samples. Raw PCAP files were processed using CICFlowMeter-V3 to extract 88 flow-based features describing temporal, spatial, and statistical properties such as duration, protocol, packet and byte counts, rates, and TCP flags. The dataset exhibits severe class imbalance (Table 2), with attack traffic exceeding benign flows by roughly 150:1, which could bias learning models toward the majority class.

The preprocessing workflow (Table 3) included data cleaning, integration, transformation, and balancing. Cleaning involved removing MSSQL traffic and erroneous rows; integration combined benign, SYN, and UDP subsets. Label encoding and timestamp decomposition prepared features for model training. To address imbalance, representative undersampling was applied to SYN and UDP attacks, while SMOTE oversampling generated additional benign instances.

Finally, features were scaled using min–max normalization and refined through mutual information–based selection. The resulting balanced dataset (Table 4) comprised 103,912 records equally distributed between attack and benign classes, represented by 93 most informative features.

Table 1. Comparative analysis of AI-enabled DDoS attack detection techniques.

Author (Year)	Dataset	Problem Resolved	Technique / Algorithm	Results / Key Findings
Mahawish et al. (2026)[9]	ToN_IoT, BoT_IoT	Building a self-healing mechanisms catered towards IoT based resource constrained devices	Self-healing IDS, based on machine learning algorithms that enhances IoT network security.	Random Forest (RF) achieves 59.6% accuracy before self-healing and 99.9% after self-healing, with the high true positive and true negative rates in detecting emerging DoS/DDoS attacks.
Rauba et al. (2025)[7]	CICIDS 2017, UNSW-NB15	Self-healing ML models for continuous adaptation to threats	Self-healing ML frameworks (H-LLM)	H-LLM achieved 0.76 accuracy vs. baseline of 0.52 (no self-healing) on corrupted data, a 46% relative improvement. Ablation study confirmed that all four components (monitoring, diagnosis, adaptation, testing) are essential.
Zhao et al. (2024)[19]	NSL-KDD, UNSW-NB15, CI-CIDS2017	generated synthetic data Improving NIDS with GAN-generated synthetic data	GANs (Vanilla, WGAN, CTGAN)	Baseline RF model on CIC-IDS2017 achieved accuracy 0.9972, but Botnet class had poor baseline (Precision: 0.87, Recall: 0.46, F1: 0.60). After augmenting with 99× WGAN-generated Botnet samples, Botnet detection improved to Precision: 1.00, Recall: 0.82, F1: 0.90 — a 30% improvement in F1-score over baseline. Increasing synthetic samples showed diminishing returns beyond 4× augmentation.
Lent et al. (2024)[20]	CICDDoS2019, SDN-based datasets	GAN-based anomaly detection for DDoS	GANs (Vanilla, WGAN, CTGAN) + GRU	Achieved F1-score of 0.998 (MCC: 0.998) on first dataset and F1-score of 0.990 (MCC: 0.986) on second dataset. Mitigation algorithm dropped 99% of malicious flows. GRU neurons demonstrated the best balance of performance and computational efficiency among GRU, LSTM, TCN, CNN, and DNN architectures.
Tariq et al. (2024)[16]	IoT-23, UNSW-NB15, N-BaIoT	Deep learning techniques for IoT security	CNNs, RNNs, GANs, Transformers	Transformer Models and GNNs show promising results in contextualizing multi-stage cyber-attacks. DL techniques (CNNs, RNNs) demonstrated efficacy in anomaly detection and predictive maintenance with reduced false positives.
Kushal et al. (2024)[15]	UNSW-NB15, CI-CIDS2017	Hybrid IDS combining signature-based and anomaly-based detection	C5 Decision Trees, LSTM	SIDS (C5). TP rate = 97.3%, FP rate = 8%. Normal class F1 = 0.902, Attack class F1 = 0.897. AIDS (LSTM-200). Detection rate = 90%, FAR = 17%, AUC = 0.936. System autonomously updates signature database from detected anomalies.
Houssou et al. (2022)[22]	Synthetic dataset	Synthetic dataset generation for cybersecurity and finance	Copula-based data generation	Copulas method outperformed SMOTE and autoencoders in preserving both dependency structure and marginal distributions. Spearman correlation μdiff = 0.005 (copulas) vs. 0.008 (SMOTE). Copulas was the only method where K-S test could not reject the hypothesis of same distribution (p ¿ 5%). Autoencoders produced the least accurate results.
Kumar & Venugopalan (2017)[5]	KDD99, NSL-KDD, DARPA	General review of IDS techniques	Multiple IDS approaches	Comprehensive morphological framework and taxonomy of IDS have been done by surveying architectures, deployment modes, detection methods, and evaluation metrics.
Chaudhari & Patil (2017)[4]	KDD99, NSL-KDD, UNSW-NB15	Classification of IDS and available datasets	SVM, Decision Tree, KNN, K-Means, Fuzzy C-Means	SVM has least execution time and high accuracy with smaller datasets. In addition, Decision Tree has high detection rate for large datasets while K-Means is faster for small data, K-Medoids better as data points increase.
This Paper	**CIC-DDoS 2019**	**Improving DDoS detection through adaptive self-healing mechanisms**	**Gaussian Copula Synthesizer and RNN**	**Gaussian Copula increased performance from 64 to 96.85 using RNN algorithm. Performance of proposed framework compared to CTGAN based models where the performance of CTGAN based model drops on original CIC-DDoS 2019 dataset from 96.55% to 49.31%.**

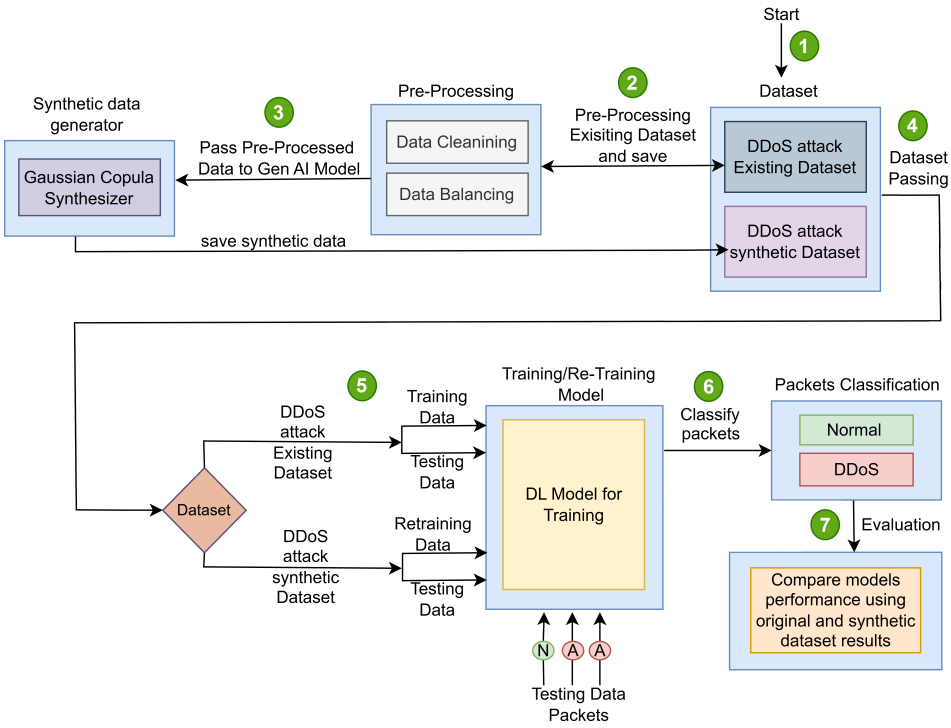


Fig. 1. Overall architecture of the self-healing DDoS detection framework, showing the continuous cycle of model training, synthetic data generation, transfer learning, and evaluation.

Table 2. Initial CIC-DDoS2019 dataset composition showing class imbalance.

Dataset	Features	Name	Type	Count	Class Total	Dataset Total
CIC-DDoS2019	88	DDoS	UDP	1,022,170	4,966,244	4,999,667
			SYN	3,944,074		
		Benign	Benign	33,423	33,423	

Table 3. Data preprocessing workflow for the CIC-DDoS2019 dataset.

Step	Description
Data Cleaning	Removed MSSQL data and invalid rows from UDP traffic
Data Integration	Merged benign, SYN, and UDP subsets
Data Transformation	Decomposed timestamps and label-encoded categorical fields
Data Reduction	Undersampled SYN/UDP attacks to balance benign data
Data Discretization	Not required

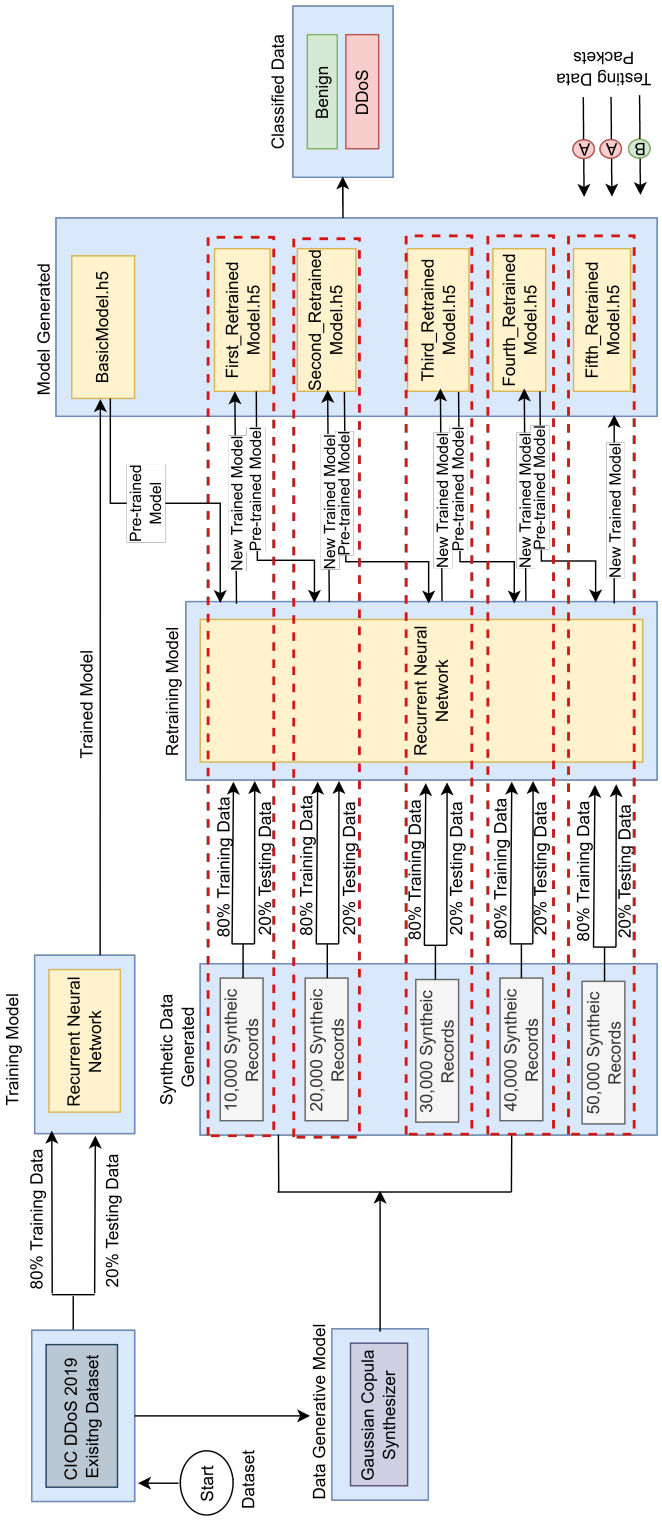


Fig. 2. Detailed architecture of the self-healing DDoS detection framework, showing the continuous cycle of model training, synthetic data generation, transfer learning, and evaluation.

Table 4. Dataset composition after preprocessing and balancing.

Dataset	Features	Name	Type	Count	Class Total	Dataset Total
CIC-DDoS2019	93	DDoS	UDP	26,783	51,956	103,912
			SYN	25,173		
		Benign	Benign	51,956	51,956	

3.2 Base Model Architecture

We designed an RNN architecture specifically optimized for temporal pattern recognition in network traffic flows. The choice of RNN was motivated by its ability to capture sequential dependencies and temporal patterns in traffic behavior, which are critical for distinguishing between legitimate bursts of activity and coordinated attack patterns [17]. Table 5 summarizes the components of the architecture depicted in Figure 3. The model contains approximately 45,000 trainable parameters and was compiled with the Adam optimizer (learning rate = 0.001), binary cross-entropy loss function, and accuracy as the primary evaluation metric. The use of multiple dropout layers and moderate layer sizes was designed to prevent overfitting while maintaining sufficient capacity to learn complex traffic patterns.

Model: "sequential"

Layer (type)	Output Shape	Param #
simple_rnn_2 (SimpleRNN)	(None, 1, 128)	28,288
dropout_3 (Dropout)	(None, 1, 128)	0
simple_rnn_3 (SimpleRNN)	(None, 64)	12,352
dropout_4 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2,080
dropout_5 (Dropout)	(None, 32)	0
dense (Dense)	(None, 64)	2,112
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 44,897 (175.38 KB)
Trainable params: 44,897 (175.38 KB)
Non-trainable params: 0 (0.00 B)

Fig. 3. Detailed architecture of the RNN base model showing layer types, output shapes, and parameter counts.

3.3 Transfer Learning Protocol

To enable continuous adaptation without catastrophic forgetting, we implemented a sophisticated transfer learning protocol illustrated in Figure. 4. The approach leverages the concept of progressive neural networks, where knowledge from previous learning stages

Table 5. RNN Model Architecture.

Layer	Description
Input	Sequences of network traffic features with shape $(10, 93)$, representing 10 time steps and 93 features.
1st RNN	SimpleRNN with 128 units and ReLU activation; captures short-term temporal patterns.
1st Dropout	Dropout rate $= 0.3$ to prevent overfitting by randomly deactivating neurons during training.
2nd RNN	SimpleRNN with 64 units and ReLU activation; captures higher-level temporal dependencies.
2nd Dropout	Dropout rate $= 0.3$ for additional regularization.
1st Dense	Fully connected layer with 32 units and ReLU activation for feature transformation and abstraction.
3rd Dropout	Dropout rate $= 0.2$ applied after the dense layer for regularization.
2nd Dense	Fully connected layer with 64 units and ReLU activation for further feature refinement.
Output	Dense layer with 1 unit and sigmoid activation for binary classification (attack vs. benign).

is preserved through network expansion and controlled information flow. The transfer learning process operates as follows:

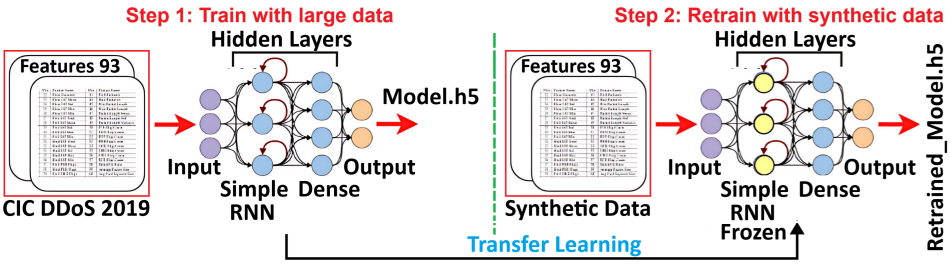


Fig. 4. Transfer learning protocol showing the iterative process of model retraining with frozen lower layers and trainable higher layers.

1. **Base Model Training**: The initial model (M_0) is trained on the original balanced dataset using early stopping with patience=10 epochs to prevent overfitting.
2. **Layer Freezing Strategy**: For each retraining iteration, the lower layers of the network (first RNN layer and associated dropout) are frozen to preserve previously learned feature representations. The higher layers (second RNN layer and dense layers) remain trainable to adapt to new patterns.
3. **Learning Rate Reduction**: The learning rate is reduced by a factor of 0.2 during retraining to enable fine-tuning rather than radical weight changes.

4. **Iterative Retraining**: Each new model (M_i) is initialized with the weights of the previous model (M_{i-1}) and fine-tuned on the new synthetic data batch for a reduced number of epochs (20 vs. 60 for initial training).

5. **Knowledge Consolidation**: After each retraining phase, model performance is evaluated on both the new synthetic data and a held-out validation set from original data to ensure no catastrophic forgetting occurs.

This approach ensures that the model continuously expands its detection capabilities while maintaining performance on previously learned patterns, effectively addressing the stability-plasticity dilemma in continual learning.

3.4 Synthetic Data Generation with Gaussian Copula

The synthetic data generation process employs a Gaussian Copula Synthesizer [27] specifically optimized for network traffic data. Copulas are powerful statistical tools that model multivariate distributions by separating the marginal distributions from the dependency structure, making them particularly suitable for network data with complex feature correlations. The synthetic data generation process illustrated in Figure 5 is summarized in Algorithm 1. This process ensures that the generated synthetic data preserves both the marginal distributions of individual features and the complex correlation structure between features, resulting in realistic network traffic patterns that represent novel attack strategies.

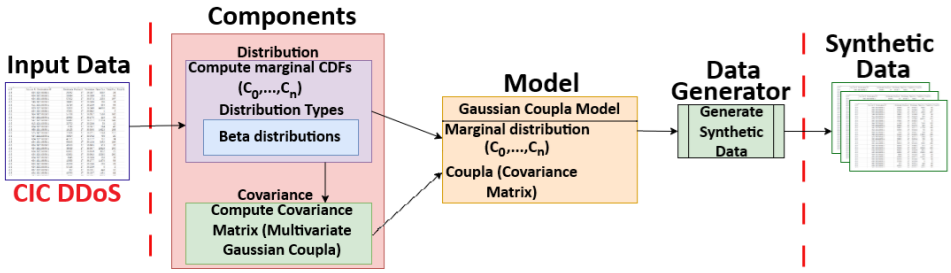


Fig. 5. Gaussian Copula architecture for synthetic data generation, showing the transformation process from original data to synthetic samples.

3.5 Catastrophic Forgetting

The catastrophic forgetting issue is investigated over Conditional Tabular Generative Adversarial Network (CTGAN) as a popular synthetic data generating technique. The data generation mechanism is set up in which the preprocessed CIC-2019 dataset is inputted into the CTGAN to generate the synthetic data of different batches. The initial model is trained using the original CIC-2019 dataset and the generated model is saved as base_model.h5 as the base model for being re-trained by the next batch of synthetic data. While having a total of five iterations in synthetic datasets generation and model re-training, a total of five new and retrained models are generated with the aim to improve detection performance by each new model.

Algorithm 1 Gaussian Copula-Based Synthetic Data Generation

Require: Dataset $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$
Ensure: Synthetic dataset $\mathbf{X}^*$

for $i = 1$ to n **do**
    Estimate empirical CDF: $F_{X_i}(x) = P(X_i \leq x)$
    Identify theoretical distribution via KS-test
end for
Transform features to uniform variables: $U_i = F_{X_i}(x_i), \quad U_i \sim U(0,1)$
Transform to standard normal space: $Z_i = \Phi^{-1}(U_i), \quad Z_i \sim \mathcal{N}(0,1)$
Estimate correlation structure: $R = \text{corr}(Z_1, Z_2, \ldots, Z_n)$
Generate new samples: $Z^* \sim \mathcal{N}(0, R)$
Inverse transformations: $U^* = \Phi(Z^*), \quad X^* = F^{-1}(U^*)$
return $\mathbf{X}^*$

The first trained model and last trained model are tested with the original CIC-2019 dataset to verify if the last model still retains the learning of base model, and the generated accuracy results are depicted in Figure 6. Results show the accuracy drop from 96.55% (for the first trained model) to 49.31% (the last trained model) indicating catastrophic forgetting by the last model trained using synthetic datasets. The underlying phenomenon occurred due to CTGAN generated synthetic data closely resembled the original distribution, causing weight overwriting during repeated fine-tuning. This limitation motivates the adoption of a copula-based synthetic data generation as it preserves the relationships between features while introducing greater diversity in the data distribution, producing statistically valid synthetic samples that improve knowledge retention during incremental transfer learning.

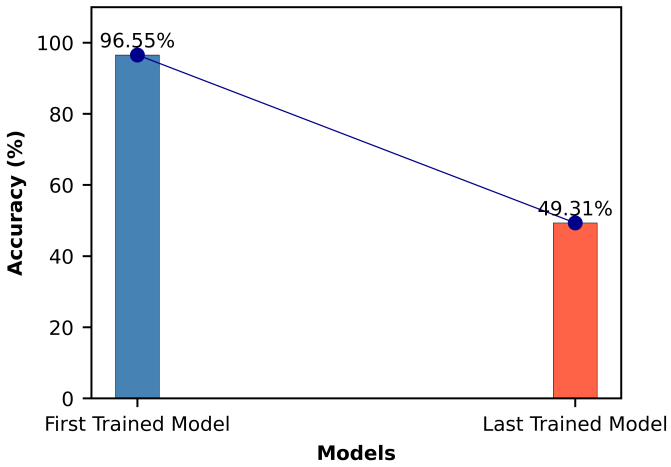


Fig. 6. Accuracy performance of the first trained model (using original CIC-2019 dataset) vs. last re-trained model (using synthetic datasets generated through CTGAN technique).

4. EXPERIMENTAL SETUP

The preprocessed and balanced CIC-DDoS2019 dataset containing 103,912 instances was partitioned using stratified sampling to maintain class distribution. The split allocation was as follows: Training Set (70%): 72,738 instances for initial model training, Validation Set (15%): 15,587 instances for hyperparameter tuning and early stopping and Test Set (15%): 15,587 instances for final evaluation. Additionally, a separate holdout set of 10,000 instances was reserved from the original data for evaluating model performance on unseen real traffic patterns after retraining.

Five batches of synthetic data were generated using the Gaussian Copula synthesizer, with progressively increasing number of instances from 10,000 for batch 1 to 50,000 for batch 5. Each batch was evaluated using multiple quality metrics, as summarized in Table 6, including Accuracy (predictive accuracy of a model trained on synthetic data and tested on real data), Data Validity Score (percentage of valid records), Data Structure Score (preservation of covariance structure), Column Shapes Score (similarity of marginal distributions), and Column Pair Trends Score (preservation of feature correlations).

All models were implemented in Python using TensorFlow 2.8 and Keras 2.8. Experiments were conducted on a dedicated research server with the following configuration: Intel Xeon Gold 6248R (3.0GHz, 24 cores), NVIDIA RTX A6000 (48GB VRAM), 256GB DDR4 RAM and 2TB NVMe SSD for storage.

Training parameters were consistent across all experiments: batch size=32, initial learning rate=0.001 (reduced by factor 0.2 during retraining), early stopping patience=10 epochs, and maximum epochs equal to 60 for initial training and 20 for retraining.

Model performance was evaluated using a set of metrics. These metrics were computed for each model iteration on both the synthetic test data and the holdout real data to evaluate generalization capabilities.

5. RESULTS AND DISCUSSION

This section presents the results concerning synthetic data quality, classification performance, and model efficiency.

5.1 Synthetic Data Quality Assessment

Table 6 summarizes the quality evaluation of synthetic network traffic generated using the Gaussian Copula model. The Data Validity Score (DVS=100%) confirms that all generated records adhere to the original feature constraints and data types, while the Data Structure Score (DSS=100%) indicates perfect preservation of multivariate relationships. The Column Pair Trends Score (CPTS) remains above 90.6% across all batches, reflecting strong retention of feature correlations critical for DDoS detection, where interactions (e.g., between packet count and flow duration) are often more discriminative than individual features. The Column Shapes Score (CSS), averaging around 70.7%, shows good but not perfect preservation of marginal distributions, a reasonable trade-off given the maintained covariance structure. Each synthetic batch (10K, 20K, 30K, etc.) was generated independently from the complete CIC-DDoS2019 dataset. Consequently, larger batches do not contain the synthetic records generated in previous batches instead, each

Table 6. Quality metrics for synthetic data batches across different sizes.

DDoS	Benign	Total	Accuracy	DS	DSS	CSS	CPTS
4994	5006	10000	80.68%	100%	100%	70.72%	90.64%
10000	10000	20000	80.77%	100%	100%	70.64%	90.8%
15020	14980	30000	80.76%	100%	100%	70.70%	90.81%
20000	20000	40000	80.78%	100%	100%	70.72%	90.84%
24995	25005	50000	80.79%	100%	100%	70.72%	90.86%

synthesis process produces a new and distinct set of synthetic samples while preserving the statistical properties and inter-feature dependencies learned from the original dataset. Figure 7 visualizes feature-wise similarity, showing high consistency across most features. Categorical features (e.g., Source IP, Destination IP, SimilarHTTP, Label) achieve perfect similarity via Total Variation Distance, while continuous features vary slightly based on their distributions. Figure 8 confirms that the synthetic data maintains the desired 50/50 class balance.

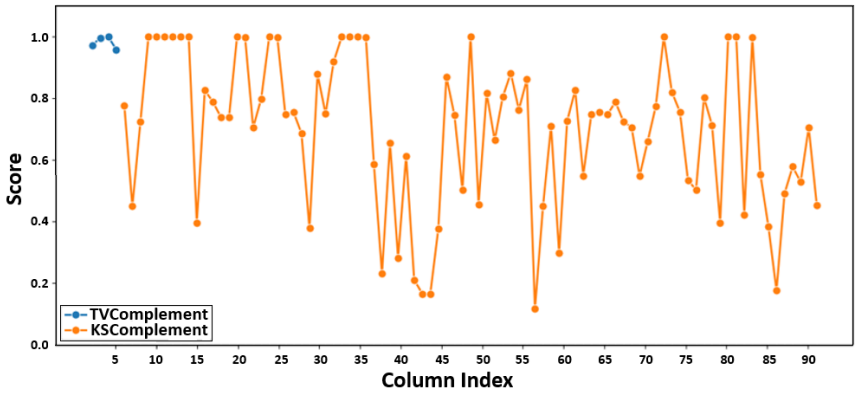


Fig. 7. Feature-wise similarity scores between real and synthetic data showing strong preservation of statistical properties.

5.2 Detection Performance and Adaptation Analysis

Figure 9 summarizes the framework's performance across iterative retraining. The baseline model (M_0) achieved only 64.0% accuracy on unseen synthetic data, underscoring the limitation of static models against novel attacks. After the first retraining (M_1), accuracy rose sharply to 95.95%—a gain of 31.95 percentage points—demonstrating rapid adaptation. Further iterations (M_2–M_5) refined the model, stabilizing around 96.8%, indicating convergence and diminishing returns as the synthetic data remained within the learned feature space.

Based on the results obtained, it is observed that accuracy improves significantly after the 1st re-training cycle. However, several further experiments were executed to obtain experimental proof for behavior of the system as the re-training cycle is repeated further

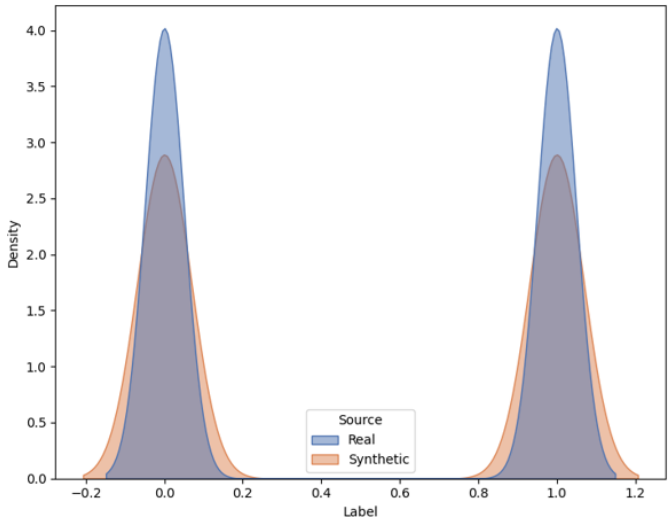


Fig. 8. Class distribution comparison between real and synthetic datasets showing maintained balance.

with new synthetic datasets. The accuracy results exhibited by the 2nd to 5th models show a marginal difference in performance reflecting that there is no need to perform re-training after the 1st training cycle. However, as the attack patterns can vary significantly in case of new zero-day DDoS attacks, hence the need to re-train with new synthetic datasets can trigger even after the 1st cycle.

Figure 10 provides a reflection of the F1-score exhibited by the five devised classification models. Similar to that portrayed by Figure 9, a major increase in F1-score is shown by the second model after re-training the original model with the first batch of synthetic datasets. In contrast, almost similar F1-score is exhibited by the subsequent models where no major increase or decline in performance is noticed.

The confusion matrix metrics in Figure 11 highlight key learning dynamics. The True Positive Rate (recall) rose from 30.26% to 97.80%, while the True Negative Rate (specificity) initially dropped from 96.27% to 94.17%, then recovered to 95.87% by M_4, reflecting an adaptation-refinement tradeoff. The False Negative Rate showed the largest gain, falling from 69.74% to 2.20%, minimizing missed detections.

Overall, the iterative transfer learning process enables effective adaptation to new attack patterns while maintaining strong performance on previously learned behaviors, following a predictable cycle of rapid adaptation and gradual stabilization.

5.3 Learning Dynamics and Pattern Analysis

The learning process exhibited three distinct phases reflecting the model's adaptation behavior. The initial phase (M_0–M_1) showed rapid improvement in recall and accuracy, indicating efficient acquisition of new detection capabilities from representative patterns. The refinement phase (M_1–M_3) involved reconciling new and prior knowledge, marked by improved specificity and reduced false positives. The final stabilization phase (M_4–M_5)

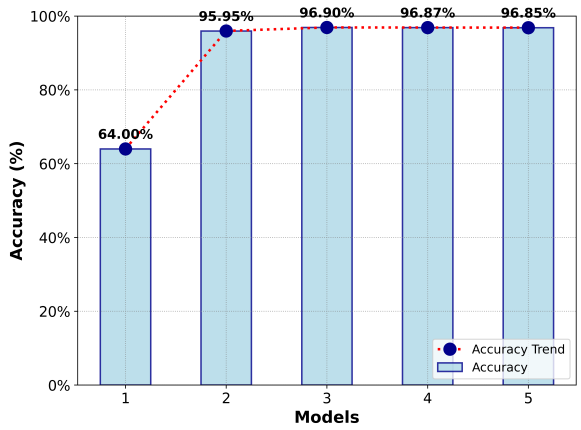


Fig. 9. Classification accuracy across model iterations showing improvement from 64% to 96.85%.

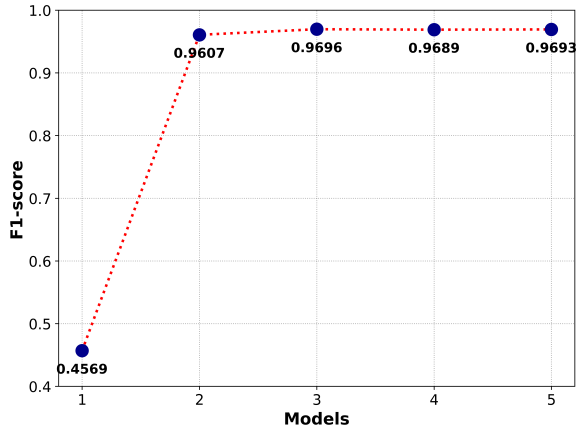
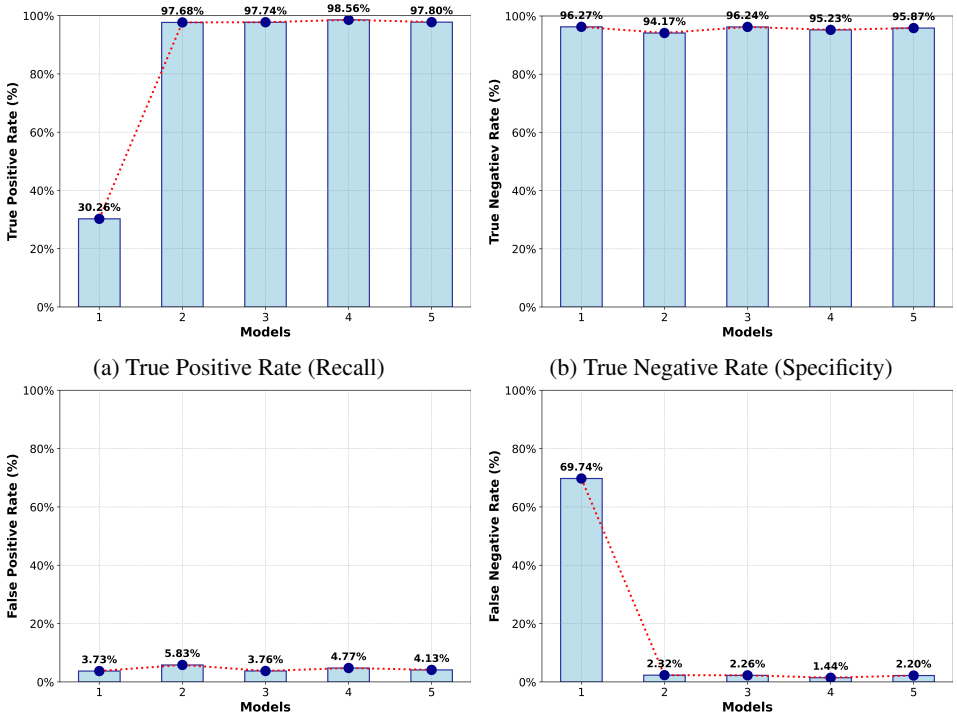


Fig. 10. F1-score across model iterations showing improvement from 0.4569 to 0.9693.

showed marginal gains, suggesting effective integration of synthetic data.

Per-class analysis revealed slightly higher accuracy for UDP flood detection than for SYN floods, likely due to more distinct UDP features. However, this gap narrowed over iterations, indicating enhanced recognition of subtle SYN flood characteristics. Rapid early improvement followed by convergence showed no signs of overfitting, supported by effective regularization. Transfer learning proved efficient, requiring only 35% of the time needed for training from scratch while achieving superior performance. These dynamics suggest practical retraining strategies: frequent, smaller updates are more efficient than infrequent large ones, and targeted retraining on weaker attack classes can improve robustness.



(a) True Positive Rate (Recall)

(b) True Negative Rate (Specificity)

(c) False Positive Rate

(d) False Negative Rate

Fig. 11. Confusion matrix metrics across model iterations showing adaptation and learning dynamics.

5.4 Computational Efficiency and Trade-off Analysis

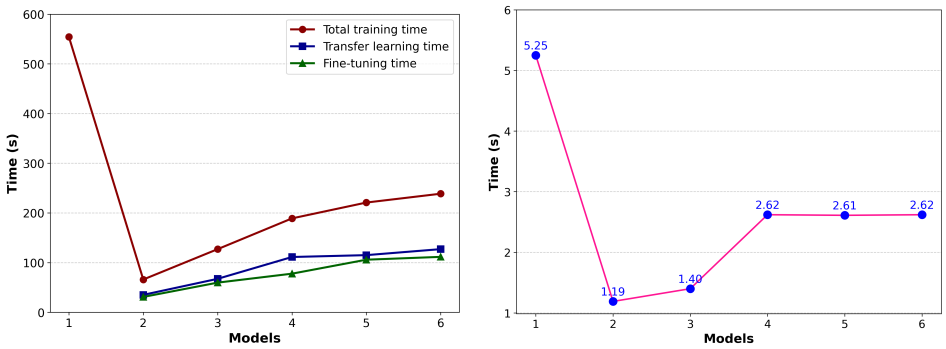
Figure 12 summarizes computational performance. Training time scaled linearly with synthetic batch size, increasing from 324 s (initial) to 1,487 s (largest retraining), far below the 2,800 s needed for full training.

Inference time rose modestly from 4.92 s to 6.84 s for 10,000 instances (39%), while memory usage increased from 48 MB to 67 MB due to added parameters both within acceptable deployment limits.

Overall, the moderate computational overhead is outweighed by a 32.8-point accuracy gain, yielding a favorable performance–security trade-off. Retraining can also be scheduled during off-peak hours or executed in scalable cloud environments for practical deployment.

6. Conclusions and future work

The proposed self-healing DDoS detection framework advances autonomous cybersecurity by enabling continuous adaptation to evolving threats. Combining Gaussian Copula



(a) Training Time (b) Testing Time

Fig. 12. Computational performance metrics showing training and testing times across model iterations.

Synthesizer for synthetic data creation with transfer learning for model refinement, the framework achieves notable improvements in detection accuracy (from 64% to 96.8%) while maintaining computational efficiency. This marks a shift from reactive to proactive cybersecurity, fostering intelligent systems that evolve with the threat landscape.

The study contributes a continuous adaptation framework integrating Gaussian Copula Synthesizer and transfer learning for resilient detection, validates the Gaussian Copula synthesizer for realistic traffic generation, and introduces an iterative transfer learning protocol that mitigates catastrophic forgetting. Empirical results confirm significant accuracy gains supported by detailed analysis of learning dynamics and deployment trade-offs.

Future work would extend the framework to broader threat vectors, consideration of other datasets for model building along with in-domain and cross-domain analysis, integration of real-time synthetic data generation, and incorporate explainability for transparency. Additionally, the impact of variations in other system parameters, including those pertaining to transfer learning and layer freezing strategies can prove a valuable research direction. Finally, performance metrics, including ROC-AUC, MCC and statistical significance can be investigated for the proposed framework to further deepen the understanding of system behavior.

ACKNOWLEDGMENT

The authors would like to clarify that an AI based tool (specifically ChatGPT) was used solely for grammar checking, rephrasing, and improving the structure of certain parts of the manuscript. The tool was not used to generate any part of the content or results. All analyses and findings presented in this paper are original and reflect the authors' own work.

REFERENCES

1. J. Wang, L. Yu, J. C. Lui, and X. Luo, "Modern ddos threats and countermeasures: Insights into emerging attacks and detection strategies," *arXiv preprint arXiv:2502.19996*, 2025, accessed: 2025-08-21. [Online]. Available: https://arxiv.org/abs/2502.19996
2. W. Dimitrov, "The impact of the advanced technologies over the cyber attacks surface," in *Artificial Intelligence and Bioinspired Computational Methods*, R. Silhavy, (ed.). Cham: Springer International Publishing, 2020, pp. 509–518.
3. H. Beitollahi and G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Computer Communications*, Vol. 35, no. 11, 2012, pp. 1312–1332.
4. R. R. Chaudhari and S. P. Patil, "Intrusion detection system: classification, techniques and datasets to implement," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 4, no. 2, 2017, pp. 1860–1866.
5. D. A. Kumar and S. Venugopalan, "Intrusion detection systems: a review," *International Journal of Advanced Research in Computer Science*, Vol. 8, no. 8, 2017, pp. 356–370.
6. A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting ddos attacks in software-defined networking," *Sensors*, Vol. 23, no. 9, 2023, p. 4441.
7. P. Rauba, N. Seedat, K. Kacprzyk, and M. van der Schaar, "Self-healing machine learning: A framework for autonomous adaptation in real-world environments," *Advances in Neural Information Processing Systems*, Vol. 37, 2025, pp. 42 225–42 267.
8. A. Spyros, A. Kougioumtzidou, A. Papoutsis, E. Darra, D. Kavallieros, A. Tziouvaras, and I. Kompatsiaris, "A comprehensive survey of manual and dynamic approaches for cybersecurity taxonomy generation," *Knowledge and Information Systems*, 2025, pp. 1–27.
9. M. Fatima, O. Rehman, N. Z. Jhanjhi, and S. Ali, "Sh-ids: a resilient self-healing intrusion detection framework against dos and ddos attacks in iot systems," *Scientific Reports*, Vol. 16, 2026, p. 19728. [Online]. Available: https://doi.org/10.1038/s41598-026-48932-2
10. S. Sutradhar, K. Chowdhury, S. Deb *et al.*, "A comprehensive review of ddos attack prevention, detection, and mitigation in iot and sdn-iot networks," *Discover Internet of Things*, Vol. 6, 2026, p. 67. [Online]. Available: https://doi.org/10.1007/s43926-026-00341-7
11. C. Benzaĭd and T. Taleb, "Ai for beyond 5g networks: A cyber-security defense or offense enabler?" *IEEE Network*, Vol. 34, no. 6, 2020, pp. 140–147.
12. J. M. Biju, N. Gopal, and A. J. Prakash, "Cyber attacks and its different types," *International Research Journal of Engineering and Technology*, Vol. 6, no. 3, 2019, pp. 4849–4852.
13. J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, Vol. 80, no. 5, 2014, pp. 973–993.

14. S. Agha, O. Rehman, and I. M. H. Rahman, "Improving discriminating accuracy rate of ddos attacks and flash events," *International Journal of Cyber Warfare and Terrorism*, Vol. 11, no. 4, 2021, pp. 21–42.
15. S. Kushal, B. Shanmugam, J. Sundaram, and S. Thennadil, "Self-healing hybrid intrusion detection system: an ensemble machine learning approach," *Discover Artificial Intelligence*, Vol. 4, no. 1, 2024, p. 28.
16. U. Tariq, I. Ahmed, A. K. Bashir, and M. A. Khan, "Securing the evolving iot with deep learning: a comprehensive review," *Kurdish Studies*, Vol. 12, no. 1, 2024, pp. 3426–3454.
17. I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, Vol. 15, no. 9, 2024, p. 517.
18. M. Andreoni, W. T. Lunardi, G. Lawton, and S. Thakkar, "Enhancing autonomous system security and resilience with generative ai: A comprehensive survey," *IEEE Access*, Vol. 12, 2024, pp. 109 470–109 493.
19. X. Zhao, K. W. Fok, and V. L. Thing, "Enhancing network intrusion detection performance using generative adversarial networks," *Computers Security*, Vol. 145, 2024, p. 104005.
20. D. M. B. Lent, V. G. d. S. Ruffo, L. F. Carvalho, J. Lloret, J. J. P. C. Rodrigues, and M. L. Proença, "An unsupervised generative adversarial network system to detect ddos attacks in sdn," *IEEE Access*, Vol. 12, 2024, pp. 70 690–70 706.
21. N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative deep learning to detect cyberattacks for the iot-23 dataset," *IEEE Access*, Vol. 10, 2022, pp. 6430–6441.
22. R. Houssou, M.-C. Augustin, E. Rappos, V. Bonvin, and S. Robert-Nicoud, "Generation and simulation of synthetic datasets with copulas," *arXiv preprint arXiv:2203.17250*, 2022. [Online]. Available: https://arxiv.org/abs/2203.17250
23. Y. Yigit, M. A. Ferrag, M. C. Ghanem, I. H. Sarker, L. A. Maglaras, C. Chrysoulas, N. Moradpoor, N. Tihanyi, and H. Janicke, "Generative ai and llms for critical infrastructure protection: Evaluation benchmarks, agentic ai, challenges, and opportunities," *Sensors (Basel)*, Vol. 25, no. 6, Mar 2025, p. 1666.
24. S. Kamthe, S. Assefa, and M. Deisenroth, "Copula flows for synthetic data generation," *arXiv preprint arXiv:2101.00598*, 2021.
25. P. F. de Araujo-Filho, M. Naili, G. Kaddoum, E. T. Fapi, and Z. Zhu, "Unsupervised gan-based intrusion detection system using temporal convolutional networks and self-attention," *IEEE Transactions on Network and Service Management*, Vol. 20, no. 4, 2023, pp. 4951–4963.
26. V. S. M. Bonam, C. Sasidhar Ravi, S. M. Yellepeddi, T. Ahmad, and A. K. Reddy, "Adaptive machine learning frameworks for iot cybersecurity: Real-time anomaly detection in low-power networks," *Nanotechnology Perceptions*, Vol. 15, no. 3, 2019, pp. 303–325.
27. H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, and I. Molloy, "Using probabilistic generative models for ranking risks of android apps," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS 2012)*. ACM, 2012, pp. 241–252.



Asma Shaheen received her BS and MS degrees in Software Engineering from Bahria University Karachi Campus, Pakistan. She is currently working as a Lab Engineer in the Department of Software Engineering at Bahria University Karachi Campus. Her research interests include cybersecurity, machine learning, generative AI, and software engineering.



Dr. Osama Rehman is a Senior Lecturer at School of Computer Science, Taylor's University, Malaysia. He received his B.S. in Computer Engineering and M.S. in Network Engineering, both from Sir Syed University of Engineering and Technology, Pakistan, followed by his Ph.D. in Electrical and Computer Engineering from Sultan Qaboos University, Oman. He served as a faculty member at the Department of Software Engineering, Bahria University, Pakistan. Prior to that, he was a Postdoctoral Researcher, a Research Assistant, both at Sultan Qaboos University – Oman, and a Lecturer at Iqra University - Pakistan. Dr. Rehman has 40+ research publications, and his current research interests are mainly in Cybersecurity, Internet of Things and Connected Vehicles.



Dr. Ibrahim M. H. Rahman is a Senior Academic Staff Member and Degree Lead in the Department of Information Technology at the Open Polytechnic of New Zealand. He holds a BE in Computer Systems Engineering from NED University of Engineering and Technology, an MS in Computer Networks from Sir Syed University of Engineering and Technology (Pakistan), and a PhD in Engineering from Victoria University of Wellington (New Zealand). He completed his postdoctoral research at the Computational Media Innovation Centre (CMIC), Victoria University of Wellington. Dr. Rahman has published extensively in reputable journals and conferences. His research interests include but not limited to computer vision, machine learning, cybersecurity and signal processing.



Dr. Sohaib Ahmad is working as a Senior Lecturer in the School of Computer Science, Faculty of Innovation and Technology, Taylor's University, Malaysia. He received PhD degree from Massey University, Auckland, New Zealand. In addition, he also received his PGD degree from Massey University, Auckland, New Zealand with distinction. Earlier, he graduated as BS in Computer Engineering from Sir Syed University of Engineering and Technology, Karachi, Pakistan. He has around 16 years of teaching and research experience. He has published more than 40 peer-reviewed articles in high-quality journals and conferences. He also voluntarily provides editorial services to various high-impact factor journals. His research interests include educational technology, AI in education, AI in software engineering and agile software development.