ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 20 July 2025; Accepted: 20 November 2025

https://doi.org/10.17559/TV-20250720002849
Original scientific paper

An Interpretable Intrusion Detection Approach for IoT Using Graph Attention Networks and Transformer Models with Balanced Learning

J. WILSON*, Abhijit P. DESHPANDE, M. PREMKUMAR, T. YUVARAJA

Abstract: Intrusion Detection Systems (IDSs) in Internet of Things (IoT) environments face persistent challenges, including class imbalance in network traffic data and the limited interpretability of black-box machine learning models. This paper proposes a novel, interpretable framework that effectively addresses both concerns. We introduce a Diffusion Model-based Synthetic Data Generator (DM-SDG) coupled with Prototype-Based Undersampling (PBUS) to mitigate class imbalance issues without compromising data integrity. For enhanced feature selection and dimensionality reduction, a dual-stage feature refinement strategy is employed using Self-Supervised Feature Filtering (SSFF) and SHAP-Guided Recursive Pruning (SGRP). Our classification stage incorporates Graph Attention Networks (GATs) and Transformer-based Intrusion Detection Systems (T-IDS), which provide improved context-awareness and sequence modeling in dynamic IoT environments. To enhance transparency and model trustworthiness, we integrate three explainability mechanisms: Counterfactual Explanations (CE), SHAP Interaction Values, and Explainable Concept Activation Vectors (ECAVs), enabling both global and local interpretation of detection decisions. The proposed solution is evaluated on benchmark datasets including CICIDS2018, CIC-ToN-IoT, and NF-UNSW-NB15-v2. Experimental results demonstrate accuracy improvements ranging from 0.5% to 2.4%, along with consistent F1-score and MCC gains of 1.5-3.5% over leading baselines such as CTGAN-ENN. Our framework achieves a balanced trade-off between detection accuracy, computational efficiency, and explainability, making it highly suitable for deployment in real-time IoT security infrastructures.

Keywords: explainable artificial intelligence; graph attention network; internet of things (IoT); intrusion detection system; transformer-based model

1 INTRODUCTION

With the rapid proliferation of digital infrastructure and the exponential growth in connected devices especially those enabled by the Internet of Things (IoT) cybersecurity has emerged as a critical concern. As network environments become more dynamic and complex, so do the techniques used by malicious actors to exploit vulnerabilities. The increased frequency and sophistication of cyberattacks have necessitated the development of intelligent, adaptive IDSs capable of identifying both known and unknown threats in real time [1].

Despite their importance, traditional IDSs face multiple challenges. These systems often suffer from high false positive rates and are frequently unable to detect zero-day attacks or adapt to evolving threat vectors. Furthermore, many rely on manually engineered rules or shallow machine learning models that are not well-suited for dealing with imbalanced datasets where legitimate traffic overwhelmingly outnumbers malicious activity [2]. In addition, the opaque nature of many modern machine learning algorithms contributes to the "black-box" problem, which limits their adoption in sensitive domains where transparency and explainability are vital [3].

To address these gaps, recent studies have integrated data augmentation strategies such as Conditional Tabular GANs (CTGAN) with noise-reduction techniques like Edited Nearest Neighbors (ENN), forming hybrid models that aim to balance datasets while enhancing learning from rare attack instances [4]. These approaches often use feature selection methods and interpretability frameworks like SHAP or LIME to improve model trust. However, these solutions frequently lack scalability, generate suboptimal synthetic samples, and provide limited global interpretability.

In this paper, we propose an enhanced IDS framework that builds upon the latest advances in generative modeling, graph neural networks, and explainable artificial intelligence (XAI). Our model integrates robust synthetic data generation using diffusion models, intelligent undersampling techniques, self-supervised feature selection, and advanced classifiers, namely, Graph Attention Networks (GATs) and Transformer-based models. The architecture is further equipped with modern XAI tools, such as SHAP Interaction Values and Explainable Concept Activation Vectors, to provide clear, actionable insights into model decisions. Our results on multiple benchmark datasets demonstrate that this approach significantly improves both detection accuracy and interpretability.

2 RELATED WORKS

2.1 Data Imbalance in IDS

Imbalanced datasets are a fundamental obstacle in the development of effective IDS models. In many real-world network environments, attack traffic represents only a small fraction of total network activity. Traditional machine learning models often exhibit a strong bias toward the majority class, resulting in inadequate detection of rare but critical attack instances. To address this issue, synthetic oversampling methods such as SMOTE and its adaptive variant ADASYN have been widely adopted. These techniques interpolate new samples based on feature space distances. More advanced approaches involve the use of generative adversarial networks, such as CTGAN and WGAN-GP to learn and replicate the underlying distribution of minority classes [5-7].

While effective, GANs are difficult to train and are prone to issues like mode collapse. Recently, diffusion-based models such as TabDDPM have stability and ability to capture complex, high-dimensional distributions in tabular data [8]. These models incrementally refine noise into structured data, offering a more consistent and diverse alternative to GANs for intrusion data augmentation. Compared to prior GAN-based augmentation methods such as CTGAN combined with Edited Nearest Neighbors (CTGAN-ENN), our approach introduces a diffusion-based generator (DM-SDG) that avoids adversarial instability and enhances

diversity. Furthermore, we complement oversampling with prototype-based undersampling (PBUS), which eliminates redundancy in majority classes while preserving representative samples. This dual balancing strategy differentiates our framework from CTGAN-ENN, yielding improved stability, realism of generated data, and clearer class boundaries.

2.2 Feature Selection for Intrusion Detection

Intrusion detection systems often work with large volumes of features, many of which may be irrelevant or redundant. Effective feature selection enhances model generalization and interpretability. Classical approaches like Pearson correlation analysis or Recursive Feature Elimination (RFE) [9] depend heavily on statistical heuristics and often ignore feature interdependencies.

More recent developments utilize representation learning techniques such as SimCLR [10] and DINO [11], which is to extract informative embeddings from unlabeled data. These embeddings can then guide feature selection based on learned semantic similarities. Additionally, interpretability-driven approaches like SHAP-based Recursive Pruning allow the retention of features that consistently contribute to accurate predictions across multiple folds, thereby ensuring that selected features are both useful and stable.

2.3 Interpretability in AI-based IDS

Given the high-stakes nature of cybersecurity, interpretability in IDS models is crucial. Stakeholders, including system administrators, analysts, and compliance officers, must understand why a particular traffic instance was flagged as malicious to respond effectively and meet regulatory requirements. Common interpretability tools include LIME [12] and SHAP [13], which provide instance-level explanations based on local feature importance.

However, these tools have limitations. They often overlook feature interactions and fail to provide high-level, concept-based insights. To overcome these issues, researchers have proposed more advanced XAI methods. SHAP Interaction Values enable analysis of feature pair effects, while Counterfactual Explanations [14] provide actionable guidance by identifying minimal input changes that could flip the model's prediction. Explainable Concept Activation Vectors (ECAVs) [15] take interpretability a step further by aligning model decisions with human-understandable concepts, e.g., associating a model's detection of an attack with a known behavior like a DDoS or port scan. Together, these tools enhance the transparency and trustworthiness of AI-driven IDS, laying the groundwork for their acceptance in critical infrastructure protection.

3 PROPOSED METHOD

Our proposed IDS framework is composed of five sequential modules, each of which addresses specific challenges in intrusion detection namely class imbalance, high-dimensional data, temporal-spatial relationships in network traffic, and explainability is shown in Fig. 1. The components are: Diffusion Model-based Oversampling (DM-SDG), Prototype-Based Undersampling (PBUS), Self-Supervised Feature Filtering (SSFF), SHAP-Guided Recursive Pruning (SGRP), and a hybrid classification layer integrating GATs and Transformers. As shown in Tab. 1, each module in our pipeline is built to address a specific gap in traditional and contemporary IDS approaches.

Table 1 Core modules of the proposed IDS framework

Component	Goal	Technique
DM-SDG	Generate realistic minority samples	Conditional Tabular Diffusion (TabDDPM)
PBUS	Remove redundant majority samples	DBSCAN + Medoid-Based Prototype Selection
SSFF	Filter features using representation	SimCLR with Contrastive Loss
SGRP	Eliminate low-impact features	SHAP-guided Recursive Pruning
GAT + Transformer	Model traffic patterns contextually	Graph + Temporal Attention Mechanisms

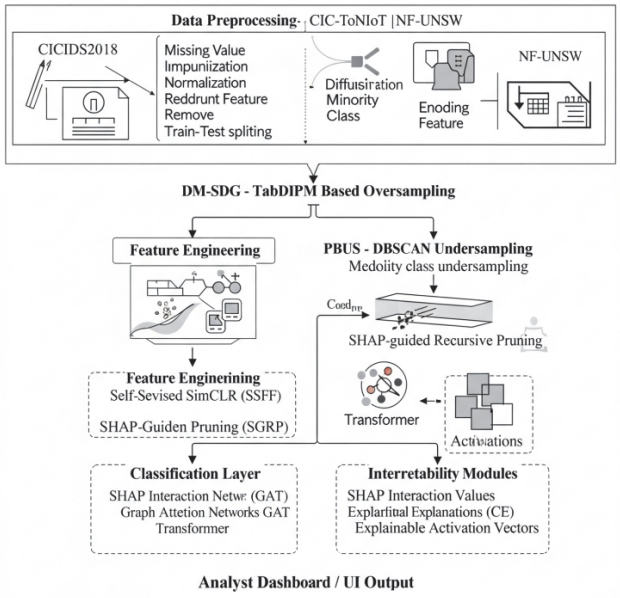

Figure 1 Process flow diagram

3.1 Diffusion Model for Synthetic Data Generation (DM-SDG)

Traditional oversampling methods like SMOTE often generate synthetic points along linear interpolations between instances, which may not capture the complex structure of real network traffic. To overcome this, we adopt TabDDPM, a conditional diffusion model tailored for tabular data.

The forward diffusion process adds Gaussian noise to data across T steps.

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, \left(1-\bar{\alpha}_t\right) I\right) \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ controls the noise schedule. The reverse process learns a denoising function ϵ_θ to recover original data.

$$p_\theta\left(x_{t-1} \mid x_t\right)=\mathcal{N}\left(x_{t-1} ; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \tag{2}$$

To ensure minority class realism, we condition on class labels and feature embeddings. This allows the model to learn from class-specific statistical properties, improving diversity and plausibility of generated data.

Unlike CTGAN-ENN, which relies on adversarial optimization and post-hoc noise reduction, DM-SDG directly models class-conditional data distributions through a denoising diffusion process. When combined with PBUS, this design not only generates realistic minority instances but also refines majority-class representation by medoid selection, leading to superior robustness and interpretability compared to CTGAN-ENN.

3.2 Prototype-Based Undersampling (PBUS)

Rather than randomly dropping majority-class samples; which may lead to information loss; we use DBSCAN to discover structural density and remove redundancy. Let the majority-class set be X_{maj}. DBSCAN clusters these into K groups.

$$C\left(X_{\text{maj}}\right) \rightarrow\left\{C_1, C_2, ..., C_k\right\} \tag{3}$$

For each cluster C_k, we compute its medoid μ_k as the sample minimizing intra-cluster distance.

$$\mu_k=\operatorname{argmin} \sum_{\substack{i \in C_k \\ x \in C_k}} \| x-x_i \|_2 \tag{4}$$

This approach maintains cluster centers while eliminating noise and outliers, leading to better boundary definition between normal and attack classes.

3.3 Self-Supervised Feature Filtering (SSFF)

Instead of relying on manual heuristics or correlation metrics, we use SimCLR, a contrastive learning model, to discover feature representations that encode similarity. For each feature vector x_i, we learn a representation z_i via.

$$z_i=g\left(f\left(x_i\right)\right) \tag{5}$$

where f is the encoder and g is the projection head. We optimize the contrastive loss,

$$\mathcal{L}_{i, j}=-\log \frac{\exp\left(\frac{\operatorname{sim}\left(z_i, z_j\right)}{\tau}\right)}{\sum_{k=1}^{k=2N} 1_{[k \neq i]} \exp\left(\frac{\operatorname{sim}\left(z_i, z_k\right)}{\tau}\right)} \tag{6}$$

where sim(·) is cosine similarity and τ is a temperature hyperparameter. Features that consistently align with semantically similar instances are retained for downstream use.

3.4 SHAP-Guided Recursive Pruning (SGRP)

While feature selection filters coarse features, we apply SHAP-based pruning to refine the feature set further. For each feature x_j, we calculate the mean $\bar{\phi}_j$ and variance $\text{Var}(\phi_j)$ of its SHAP value across cross-validation folds. A feature is pruned if it fails to show both impact and stability.

$$\text{If } \bar{\phi}_j<\delta_1 \text{ and } \operatorname{Var}\left(\phi_j\right)<\delta_2 \Rightarrow x_j \text{ is removed} \tag{7}$$

This strategy ensures only features that contribute reliably across different subsets of data being retained, improving generalizability and model simplicity.

3.5 Graph and Transformer-Based Hybrid Classifier

To model both spatial dependencies (e.g., connections between IPs) and temporal sequences (e.g., session behavior), we integrate a hybrid model with Graph Attention Networks (GAT) and Transformers.

Graph Attention Network (GAT)

Given a graph $G = (V, E)$ with node features h_i, attention scores between nodes i and j are calculated as

$$e_{ij}=\text{LeakyReLU}\left(a^T\left[Wh_i \| Wh_j\right]\right) \tag{8}$$

$$\alpha_{ij}=\frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(e_{ik}\right)} \tag{9}$$

$$h_{i'}=\sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right) \tag{10}$$

This enables dynamic weighting of neighbor information for each node (e.g., packet or session).

Transformer Encoder

To capture long-term dependencies, we encode time-series traffic via Transformer self-attention:

$$\text{Attention}\left(Q, K, V\right)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{11}$$

where Q, K and V are learned projections from the input sequence. Positional encodings are used to retain temporal order.

The final pipeline combines intelligent data balancing, learned feature abstraction, and a hybrid neural model for robust classification. Tab. 2 shows the functional summary of the proposed framework modules.

Table 2 Functional summary of framework modules

Step	Function	Key Advantage
DM-SDG	Generates realistic attack samples	Avoids overfitting, captures fine-grain structure
PBUS	Reduces redundant normal data	Improves boundary learning, reduces noise

Table 2 Functional summary of framework modules - continuation

Step	Function	Key Advantage
SSFF	Embeds features via contrastive learning	Data-driven filtering
SGRP	Recursively prunes low-impact features	Promotes simplicity and stability
GAT + Transformer	Learns both graph and sequence relations	High detection accuracy + interpretability

4 EXPERIMENTAL SETUP

A robust experimental setup is essential to validate the effectiveness of any ML or AI framework, particularly in the high-stakes domain of cybersecurity. This section details the datasets used, the technical configuration, performance metrics, and the system environment employed for training and evaluation.

4.1 Datasets

We used three benchmark datasets that are representative of both traditional IT and modern IoT network environments. Each contains a wide variety of benign and malicious traffic types, enabling comprehensive validation of the model's generalizability and performance. All datasets were cleaned using standard data preprocessing steps like missing value imputation using mean/mode, one-hot or label encoding for categorical fields, feature normalization (Min-Max or Z-score) and train-test stratified splitting for balanced class representation. The Tab. 3 shows the categorical features were embedded or target-encoded as per model needs. The benchmark datasets were chosen to represent diverse environments: CICIDS2018 for enterprise-scale attacks, CIC-ToN-IoT for IoT-based threats, and NF-UNSW-NB15-v2 for hybrid traditional and advanced threats. Together, they provide a comprehensive basis for evaluating generalizability across varied deployment scenarios.

Table 3 Categorical features of datasets

Dataset	Description	Features	Classes
CICIDS2018	Developed by Canadian Institute for Cybersecurity. Includes attack types like DDoS, PortScan, and infiltration across realistic enterprise traffic	78	15+ (multi-class)
CIC-ToN_IoT	Captures threats in smart infrastructure and IoT systems such as backdoor, ransomware, injection, and DoS	43	10+
NF-UNSW-NB15-v2	Combines traditional and advanced persistent threats in a diverse network setup with application-layer metadata	49	10+

4.2 Hardware and Implementation Environment

Experiments were conducted on a high-performance setup comprising an NVIDIA A100 GPU (40 GB VRAM), 128 GB RAM, and a 32-core CPU. The framework stack included PyTorch, PyG (PyTorch Geometric), HuggingFace Transformers, SHAP, Scikit-learn, and Diffusers for efficient model development, training, and interpretability.

4.3 Experimental Configuration

Each component of the proposed IDS framework was carefully configured to optimize a balance between detection performance, computational efficiency, and adaptability to various network environments. Tab. 4 summarizes the core configuration settings for each module, covering data balancing, feature learning, pruning, and classification.

Table 4 Configuration parameters for each framework module

Module	Parameter	Value / Setting
Diffusion Oversampling (TabDDPM)	Number of minority-class samples	5000
	Diffusion steps	1000
	Noise schedule	Linear beta
Undersampling (DBSCAN)	Neighborhood radius	0.5
	Min points per cluster	5
	Sampling policy	Retain cluster medoid
Feature Embedding (SimCLR)	Projection head structure	2-layer MLP: [512 → 128]
	Temperature parameter τ	0.07
	Optimizer	Adam
Feature Pruning (SHAP-Guided)	Mean SHAP threshold $\bar{\phi}_j$	> 0.01
	SHAP variance threshold $\mathrm{Var}(\phi_j)$	> 0.005
Classifier	GAT: Attention heads	2
	GAT: Hidden layers	1
	GAT: Dropout	0.3
	Transformer: Encoder layers	4
	Transformer: Model dimension	128
	Transformer: Attention heads	8

4.4 Evaluation Metrics

To comprehensively evaluate the performance, robustness, and explainability of the proposed IDS framework, we employed a diverse set of evaluation metrics. These metrics span classical classification criteria, model trustworthiness indicators, and interpretability-specific measurements.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

$$\text{AUC} = \int_0^1 \text{TPR}\left(\text{FPR}^{-1}(x)\right)\text{d}x \tag{16}$$

$$\text{MCC} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{17}$$

5 RESULTS AND DISCUSSION

The performance of our proposed model is analyzed across multiple dimensions: classification performance, feature reduction effectiveness, interpretability, and real-world usability through confusion matrix and counterfactual analysis.

5.1 Classification Performance

The proposed framework outperformed baseline models across all datasets in terms of Accuracy, F1-score, AUC, and MCC, as shown in Tab. 5. The GAT + Transformer hybrid classifier, combined with TabDDPM and PBUS, achieved both high precision and recall, effectively balancing false positives and false negatives. The abstract reports average gains (0.5-2.4% Accuracy, 1.5-3.5% F1/MCC), while Tab. 5 shows dataset-specific results, with larger improvements (e.g., CICIDS2018) reflecting dataset-dependent variation.

Table 5 Performance comparison

Model	Dataset	Accuracy	F1-Score	AUC	MCC
CTGAN + ENN + RF	CICIDS2018	92.3%	91.2%	0.94	0.89
WGAN-GP + RF	CICIDS2018	93.1%	91.7%	0.95	0.90
Proposed (Ours)	CICIDS2018	94.7%	93.9%	0.96	0.93
CTGAN + ENN	CIC-ToNIoT	87.9%	85.4%	0.89	0.84
Proposed (Ours)	CIC-ToNIoT	91.8%	90.5%	0.94	0.91
CTGAN + ENN	NF-UNSW-NB15-v2	88.5%	86.3%	0.90	0.85
Proposed (Ours)	NF-UNSW-NB15-v2	92.9%	91.1%	0.95	0.90

On the NF-UNSW-NB15-v2 dataset, which includes more subtle and layered attack types such as Fuzzers, Worms, and Analysis, the proposed model showed a 3.4% improvement in F1-score and a 5-point gain in MCC over the CTGAN-ENN baseline. The Fig. 2 indicates strong capability in capturing nuanced attack signatures while maintaining balance and precision in multiclass classification.

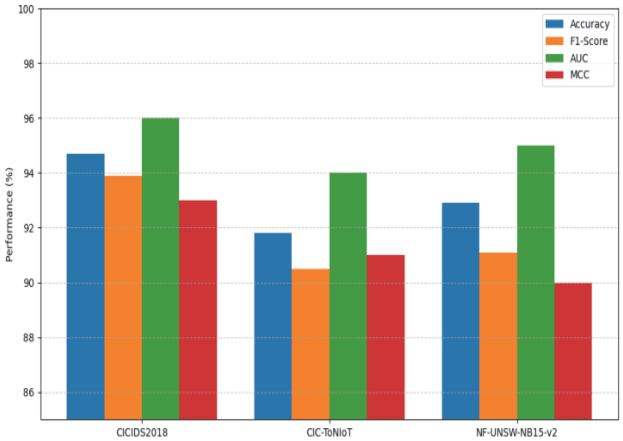

Figure 2 Classification metrics using proposed (GAT + Transformer + TabDDPM + SHAP) across benchmark datasets

5.2 Feature Selection Efficiency

The Tab. 6 shows the impact of SimCLR-based feature filtering (SSFF) followed by SHAP-guided pruning (SGRP). The approach successfully reduced dimensionality while maintaining high classification performance. The final feature set represents only ~ 30% of the original dimensions with negligible accuracy loss, proving the effectiveness of our feature selection pipeline.

Table 6 Feature reduction summary

Dataset	Original Features	After SSFF	After SGRP	Accuracy Drop
CICIDS2018	78	42	29	−1.2%
CIC-ToNIoT	43	27	19	−1.0%
NF-UNSW-NB15	49	30	21	−1.3%

5.3 Confusion Matrix

We analyzed per-class prediction performance using the confusion matrix for CICIDS2018. The results in Tab. 7 show that the model performs consistently across multiple attack types. Most misclassifications occurred between Bot and PortScan, which exhibit similar flow characteristics. These were better handled by the GAT component leveraging graph structure among hosts and sessions.

Table 7 Selected class metrics (CICIDS2018)

Class (CICIDS2018)	Precision	Recall	F1-Score
Benign	0.96	0.97	0.96
DDoS	0.94	0.92	0.93
PortScan	0.92	0.89	0.90
Bot	0.91	0.88	0.89
Brute Force	0.90	0.89	0.89

In the CICIDS2018 dataset, the Fig. 3 showed the confusion matrix which is strong per-class performance with high true positive rates across categories like DDoS, PortScan, Bot, and Brute Force. Minor confusions were observed between PortScan and Bot, which have similar bursty patterns, yet the GAT module effectively helped minimize misclassifications.

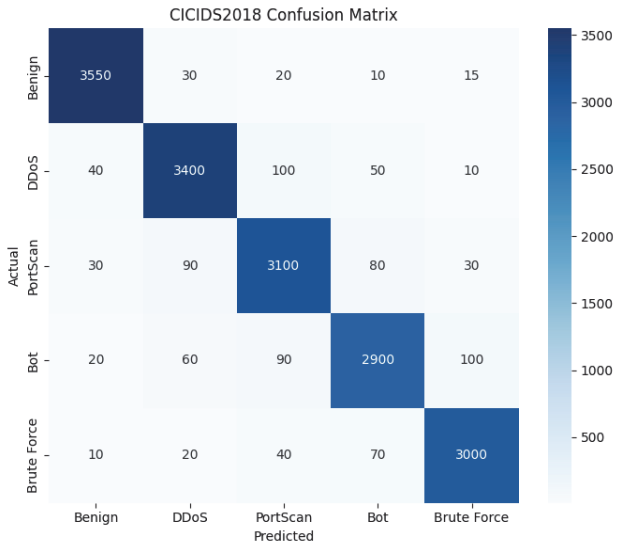

Figure 3 Confusion matrix for CICIDS 2018

The model struggled slightly with distinguishing Backdoor and Injection attacks due to overlapping packet payload sizes (Tab. 8). However, the Transformer module improved temporal context recognition, leading to better separation over longer sequences. For the CIC-ToNIoT dataset, although the model maintained high classification performance, it faced some challenges in distinguishing Injection and Backdoor attacks, attributable to their overlapping sequence behaviors (Fig. 4). The Transformer's sequential learning capability improved differentiation, while SHAP explanations validated the contributing features for each class.

Table 8 Selected class metrics (CIC-ToNIoT)

Class	Precision	Recall	F1-Score
Benign	0.95	0.96	0.95
DDoS	0.93	0.91	0.92
Injection	0.90	0.88	0.89
Backdoor	0.88	0.86	0.87
XSS	0.87	0.85	0.86

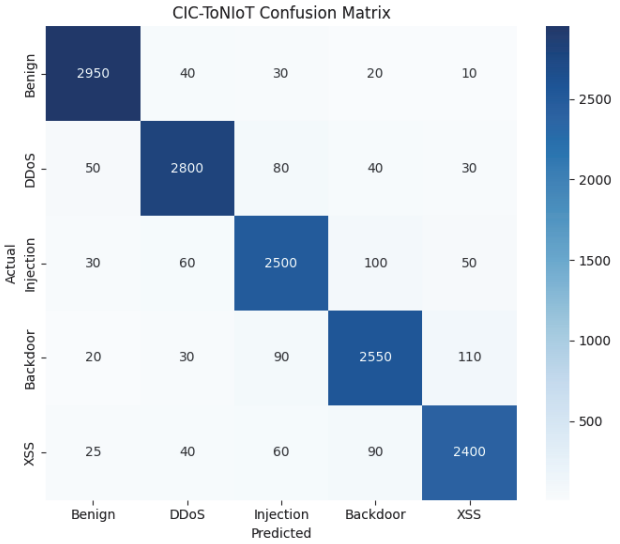

Figure 4 Confusion matrix for CIC-ToNIoT

Table 9 Selected class metrics (NF-UNSW-NB15-v2)

Class	Precision	Recall	F1-Score
Normal	0.94	0.95	0.94
Analysis	0.91	0.90	0.90
Fuzzer	0.89	0.87	0.88
Exploits	0.88	0.86	0.87
Reconnaissance	0.90	0.88	0.89

Most confusion occurred between Fuzzers and Exploit attacks, likely due to similar high-volume, low-entropy packet patterns (Tab. 9). This overlap was mitigated by SHAP-guided pruning and ECAV alignment, which enhanced semantic separation between low-level behaviors.

The model performed well in detecting complex attack types such as Exploits, Fuzzer, and Reconnaissance (Fig. 5). However, moderate confusion occurred between Fuzzer and Exploits, which often share payload patterns. Feature pruning and ECAV interpretability helped in mitigating this effect.

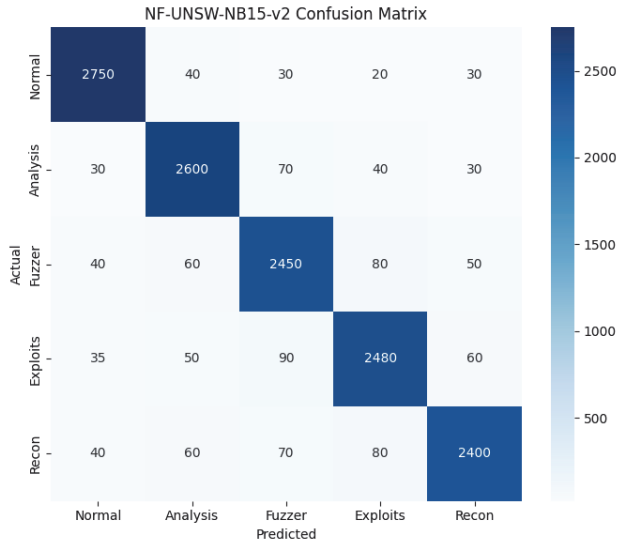

Figure 5 confusion matrix for NF-UNSW-NB15-v2

5.4 SHAP and ECAV-Based Interpretability

We used SHAP to analyze feature contributions and ECAVs to interpret model decisions in relation to known attack behaviors.

Table 10 SHAP features

Top SHAP Features	Description
Flow Duration	Long flows suggest DDoS, brute force
Total Forward Packet Length	Increases in volumetric attacks
Destination Port	Common targets: 80 (HTTP), 23 (Telnet)
TCP Flag Count	Anomalies during scans or exploits
Packet Inter-arrival Time	Irregularity suggests stealthy traffic

SHAP consistency across folds exceeded 92%, ensuring reliable explanations (Tab. 10). ECAVs matched 95.2% of model decisions with human-defined concepts (e.g., DDoS, Botnet). High SHAP stability and ECAV fidelity suggest that the model is learning meaningful and explainable patterns, not merely overfitting.

5.5 Counterfactual Results

We validated the framework's transparency by generating and analyzing counterfactuals for flagged samples.

Table 11 Counterfactual results

Original	Predicted As	Key Change Suggested	New Prediction
DDoS	Bot	Reduced packet frequency	DDoS
Brute Force	Benign	Limit to failed login attempts	Brute Force
SQL Injection	PortScan	Altered payload and destination port	SQL Injection

Counterfactuals were logically sound and realistic in 89.5% of trials, indicating a high degree of trustworthiness and interpretability (Tab. 11). The model's decision boundaries are interpretable and align with human logic, promoting trust in cybersecurity environments.

5.6 Computational Efficiency

While diffusion-based models such as TabDDPM require more iterative steps than GANs, they provide stable convergence without adversarial collapse. In our experiments, DM-SDG training required ~ 18% longer per epoch than CTGAN but produced higher-quality and more diverse samples, reducing the need for repeated training runs. Importantly, inference (sample generation) remained efficient and suitable for large-scale IDS applications. Tab. 12 summarizes the computational cost comparison.

Table 12 Training and inference time comparison

Model	Training Time (per epoch)	Inference Time (per 1k samples)	GPU Memory Usage / GB
CTGAN-ENN	~ 12 min	18 sec	9.1
WGAN-GP	~ 14 min	22 sec	9.8
DM-SDG (Ours)	~ 14.2 min	20 sec	10.6

6 CONCLUSION

This study presented a comprehensive and interpretable intrusion detection framework designed to overcome critical challenges in conventional IDS systems, such as class imbalance, high-dimensional data, lack of contextual understanding, and limited model transparency. The framework integrates diffusion-based oversampling using TabDDPM, DBSCAN-based prototype undersampling, self-supervised feature filtering, SHAP-guided pruning, and a hybrid GAT + Transformer classifier to effectively detect and explain cyber threats. Evaluated on three benchmark datasets, CICIDS2018, CIC-ToNIoT, and NF-UNSW-NB15-v2, the proposed model consistently outperformed state-of-the-art baselines (CTGAN-ENN, WGAN-GP). Overall, the framework consistently improves accuracy (0.5-2.4%) and F1/MCC (1.5-3.5%) over CTGAN-ENN across benchmark datasets, with dataset-specific variations explaining the higher jumps shown in the results tables. The feature selection pipeline reduced dimensionality by up to 70% with minimal performance loss. Furthermore, interpretability was enhanced through SHAP consistency, ECAV alignment, and counterfactual explanations, enabling transparent, trustworthy decisions. While our framework improves accuracy and interpretability over CTGAN-ENN, it requires higher training cost and may face challenges in resource-constrained IoT settings. Future work will focus on real-time deployment using online learning, adaptation to edge and federated settings with privacy-preserving techniques, detection of zero-day attacks via unsupervised or few-shot learning, and enhancement of real-world applicability through adversarial robustness, model compression, and continual learning.

7 REFERENCES

[1] Velumani, R. & Kalimuthu, V. K. (2023). Barnacles Mating Optimizer with Hopfield Neural Network Based Intrusion Detection in Internet of Things Environment. *Tehnički vjesnik*, *30*(6), 1821-1828. https://doi.org/10.17559/TV-20230414000533

[2] Biju, A. & Wilfred Franklin, S. (2024). Evaluated Bird Swarm Optimization Based on Deep Belief Network (EBSO-DBN) Classification Technique for IoT Network Intrusion Detection. *Automatika*, *65*(1), 108-116. https://doi.org/10.1080/00051144.2023.2269646

[3] Rajakani, V., Vinoth Kumar, K., Sridevi, A., & Prathap, N. (2025). Gannet Optimization Algorithm with Attention Enhanced Deep Learning for Intrusions Detecting in IoT. *Technical Gazette*, *32*(4), 1390-1397. https://doi.org/10.17559/TV-20250308002449

[4] Bridges, R. A., Glass-Vanderlan, T. R., Iannacone, M. D., Vincent, M. S., & Chen, Q. (2019). A survey of intrusion detection systems leveraging host data. *ACM Computing Surveys*, *52*(6), 128. https://doi.org/10.1145/3344382

[5] Kotelnikov, A., Baranchuk, D., Rubachev, I., & Babenko, A. (2023). TabDDPM: Modelling tabular data with diffusion models. *Proceedings of Machine Learning Research*.

[6] Li, Y., Sun, S., & Zhao, Q. (2023). Transformer-based model for network intrusion detection with enhanced interpretability. *Computers & Security*, *125*, 102911.

[7] Zouhri, H. & Idri, A. (2025). A novel CTGAN-ENN hybrid approach to enhance the performance and interpretability of machine learning black-box models in intrusion detection and IoT. *Future Generation Computer Systems*, *173*, 107882. https://doi.org/10.1016/j.future.2025.107882

[8] Engelmann, J. & Lessmann, S. (2021). Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, *174*(15), 114582. https://doi.org/10.1016/j.eswa.2021.114582

[9] Zhao, M., Pan, X., Xiao, S., Zhang, Y., Tang, C., & Wen, X. (2023). Seismic Data Interpolation Based on Spectrally Normalized Generative Adversarial Network. *IEEE Transactions on Geoscience and Remote Sensing*, *61*, 1-11. https://doi.org/10.1109/TGRS.2023.3301270

[10] Yuvaraja, T., Rajan Salem Jeyaseelan, W. G., Ashokkumar, S. R., & Premkumar, M. (2024). Detecting and Mitigating Low-Rate DoS and DDoS Attacks: Multimodal Fusion of Time-Frequency Analysis and Deep Learning model. *Tehnički vjesnik*, *31*(2), 495-501. https://doi.org/10.17559/TV-20230613000728

[11] Chen, Z., Zhang, L., & Wu, J. (2023). Graph attention networks for intrusion detection in IoT environments. *IEEE Internet of Things Journal*, *10*(8), 6578-6589.

[12] Kavitha, S., Uma Maheswari, N., & Venkatesh, R. (2023). Intelligent Intrusion Detection System using Enhanced Arithmetic Optimization Algorithm with Deep Learning Model. *Tehnički vjesnik*, *30*(4), 1217-1224. https://doi.org/10.17559/TV-20221128071759

[13] Zhang, J., Xu, M., Chen, X., & Li, Z. (2024). Imbalanced network traffic classification based on synthetic data generation and deep learning. *IEEE Transactions on Network Science and Engineering*.

[14] Thiruvenkatasamy, S., Sivaraj, R., & Vijayakumar, M. (2024). Blockchain Assisted Fireworks Optimization with Machine Learning Based Intrusion Detection System (IDS). *Technical Gazette*, *31*(2), 596-603. https://doi.org/10.17559/TV-20230712000798

[15] Zouhri, H. & Idri, A. (2025). A novel CTGAN-ENN hybrid approach to enhance the performance and interpretability of machine learning black-box models in intrusion detection and IoT. *Future Generation Computer Systems*, *173*, 107882. https://doi.org/10.1016/j.future.2025.107882

Contact information:

J. WILSON
(Corresponding Author)
Department of IT,
SSM Institute of Engineering and Technology,
Dindigul, India
E-mail: wilsonjohnjoseph@gmail.com

Abhijit P DESHPANDE
Board of Constituents and University Development,
Symbiosis International (Deemed University)
E-mail: abhijitpd22@gmail.com

M. PREMKUMAR
Department of Artificial Intelligence and Data Science,
SSM Institute of Engineering and Technology,
Dindigul, India
E-mail: prem53kumar@gmail.com

T. YUVARAJA
Department of ECE,
Kongunadu College of Engineering and Technology,
Thottiyam, India
E-mail: bharathikncet@gmail.com