Future Generation Computer Systems 173 (2025) 107882

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A novel CTGAN-ENN hybrid approach to enhance the performance and interpretability of machine learning black-box models in intrusion detection and IoT

Houssam Zouhri [a], Ali Idri [b,*]

[a] Vanguard Center, Mohammed VI Polytechnic University, UM6P, Benguerir, 43150, Morocco
[b] Software Project Management Research Team, ENSIAS, Mohammed V University, Rabat, 10000, Morocco

ARTICLE INFO

Dataset link: https://www.unb.ca/cic/datasets/index.html, https://staff.itee.uq.edu.au/marius/NIDS_datasets/

Keywords:
Intrusion detection systems
Feature selection
Generative adversarial networks
Global interpretability
SHAP
LIME
Global surrogate

ABSTRACT

Class imbalance and high-dimensional data pose significant challenges in intrusion detection systems (IDSs), impacting model performance and interpretability. This paper introduces a novel approach, CTGAN-ENN, combining explainable Conditional Tabular generative adversarial networks (CTGAN) and Edited Nearest Neighbor (ENN) with feature selection (FS) for improving IDS interpretability. The framework operates in three stages: (1) ENN undersamples majority class to reduce overlap and noise, while CTGAN generates realistic synthetic samples for minority classes; (2) two filter-based and two wrapper-based FS techniques are evaluated across three datasets (CICIDS2018, CIC-ToNIoT, NF-UNSW-NB15-v2), with optimal FS-classifier combinations identified using Scott-Knott analysis; and (3) Four interpretability techniques (SHAP, LIME, Global Surrogate (GS), and SHAP summary) are applied for local and global interpretation of four intrusion detection classifiers using two interpretability metrics. The proposed CTGAN-ENN framework is compared with state-of-the-art methods (WGAN, WGAN-GP, SMOTE, ADASYN) using Borda count ranking based on seven model performance metrics, demonstrating superior performance with accuracy rates of 99.99%, 99.64%, and 99.26% on the respective datasets. By integrating SHAP, LIME, and GS, we provide both high performance and a clear understanding of model decisions, making the CTGAN-ENN approach a powerful tool for improving IDSs. Compared to baseline approaches, CTGAN-ENN outperformed all, demonstrating the advantages of combining FS and GAN for simpler, more interpretable models.

1. Introduction

As technologies like computer applications, cloud computing, and the Internet of Things (IoT) continue to proliferate, the number of vulnerabilities and cyberattacks also increases, emphasizing the critical need to enhance network security. Unfortunately, the sheer volume and diversity of data, along with the ever-changing nature of cyber threats, provide severe obstacles for traditional intrusion detection approaches due to their inability to detect zero-day attacks and the high rate of false alarms they generate [1]. As a result, current research is looking at the utilization of artificial intelligence (AI) and machine learning (ML) approaches to create more adaptable, resilient, and robust IDS as an alternative solution capable of identifying and mitigating a wide range of security attacks. However, there are still significant limits to their practical application in real-world scenarios.

A key challenge in cybersecurity is accurately detecting network intrusions. In most cases, the majority of network flow data consists of normal traffic, while attack flows are typically rare. This discrepancy demonstrates the significant challenge presented by class imbalance, which arises when the frequency of the occurrence of one or a few classes in a dataset greatly exceeds that of others. The classes with higher prevalence are referred to as the majority classes, while those with lower occurrences are known as the minority classes [2]. In sensitive fields like cybersecurity, practitioners often focus on the minority class as it represents a particular type of attack. The imbalance rate (IR) quantifies the ratio between the majority and minority classes, calculated as shown in Eq. (1), where n_{maj} and n_{min} represent the number of instances in the majority and minority classes, respectively. A significant imbalance occurs when the number of attack data points is much smaller than the normal data, leading to class imbalance issues [3]. Resampling is a valuable technique for addressing imbalanced datasets by balancing the number of instances across different classes, utilizing two main methods: undersampling and oversampling.

* Corresponding author.
E-mail addresses: Houssam.zouhri@um6p.ma (H. Zouhri), Ali.idri@um5.ac.ma (A. Idri).

https://doi.org/10.1016/j.future.2025.107882
Received 19 November 2024; Received in revised form 6 March 2025; Accepted 1 May 2025
Available online 19 May 2025

Undersampling reduces data imbalance by removing excess simples from the majority class, while oversampling addresses this imbalance by either replicating existing instances or generating new ones for the minority classes.

$$\mathrm{IR} = \frac{n_{\mathrm{maj}}}{n_{\mathrm{min}}} \tag{1}$$

Although oversampling improves a classifier's ability to learn from minority class features and patterns, traditional methods often fail to accurately represent the true data distribution, which can degrade classifier performance. To mitigate overfitting from repetitive learning on oversampled data, oversampling techniques such as SMOTE [4] and ADASYN [5] have been developed. These methods create new artificial instances that are like rare data without simply cloning it. Despite this, these techniques can sometimes introduce noise by neglecting the contextual relationship of the data surrounding rare instances. Recently, Goodfellow and al. [6] proposed GAN as a novel approach for augmenting rare data, which holds promise for generating more realistic and useful synthetic samples. It involves two neural networks, a generator and a discriminator, that are adversarially trained to create convincing synthetic data samples. The generator and discriminator engage in a zero-sum game, where the generator aims to produce synthetic examples while the discriminator works to distinguish between the generated (fake) data and the real data it receives for evaluation [6]. On top of that, GAN has emerged as an effective oversampling approach in a variety of applications, including computer vision [7], cybersecurity [8–10], medical imaging [11], and natural language processing (NLP) [12].

Another challenge in the realm of IDSs is the issue of trust, as AI-based IDSs are often perceived as black box models. This lack of transparency reduces their interpretability and reliability, making cybersecurity organizations reluctant to trust these systems in the sensitive field of network security [13]. The field of XAI was launched in response to the growing need to interpret ML/Deep Learning models in cyberdefense applications. XAI sheds light on the black box models, explaining how they work and what they predict. According to a systematic literature review [14] of recent studies XIA in cybersecurity, the most frequently employed interpretability techniques are Local Interpretable Model-agnostic Explanations (LIME), Shapley Additive Explanations (SHAP), and rule-based explanations. Among these, LIME and SHAP are particularly prevalent. The review indicated that intrusion detection and malware classification have garnered more research focus compared to other ML applications in cybersecurity.

However, using interpretability methods to explain the behavior of black box models trained with a large number of features poses a significant challenge, as this abundance of features complicates interpretability and comprehensibility for experts. The use of a reduced feature set and interpretability methods simplify models and facilitate the implementation of IDS in real-life contexts. Numerous studies [15–18] advocate for identifying critical intrusion features as a pre-processing step to enhance the IDS performance, improve computational efficiency, and facilitate the effective deployment of IDS in real-world scenarios. Furthermore, although extensive studies [16,19–22] have made significant contributions in the field of IDS, they did not provide usable results for IDS implementation, motivating the need for our research: (1) There is currently no primary research comparing the effects of combining generative AI with FS techniques on the classification performance and interpretability of black box IDSs. (2) Many studies have been conducted on outdated datasets, potentially limiting the effectiveness and applicability of their results. (3) The majority of previous research [20–22] has focused on improving classification accuracy when using GANs to overcome class imbalance while disregarding the transparency of the predominantly used black box models. To the best of our knowledge, this is the first study to evaluate the combination of GANs with FS techniques, integrating both local and global interpretability methods in the analysis.

Therefore, the aim of this study is to overcome these challenges by introducing a hybrid sampling method-based IDS capable of addressing data imbalance issues, enhancing the performance of previous systems, and helping end-users understand the reasoning behind the detected anomalies. Our novel two-stage framework integrates ENN-based undersampling and CTGAN-based oversampling, systematically evaluates FS techniques, and incorporates SHAP, LIME, and GS for explainability using three newly introduced intrusion datasets (CICIDS2018, NF-UNSW-NB15-v2, and CIC-ToNIoT) from the Canadian Institute for Cybersecurity (CIC) [23]. The rationale behind selecting CTGAN for oversampling is its superior ability to generate realistic synthetic samples for minority classes, particularly within cybersecurity datasets, as evidenced in previous research [10,22,24]. Additionally, ENN was chosen for undersampling due to its effectiveness in cleaning boundary samples and reducing noise, which has proven beneficial in similar intrusion detection scenarios [25] and other fields [26,27]. Initially, ENN is used to effectively undersample majority classes, reducing information loss, while CTGAN generates synthetic samples for minority classes to balance the dataset. The effectiveness of CTGAN-ENN is compared against two GAN-based oversampling techniques (WGAN and WGAN-GP) and two traditional methods (SMOTE and ADASYN). These methods are commonly used to address class imbalance in IDS [21,28,29], making them a natural point of comparison for evaluating our proposed approach. Although other GAN variants or oversampling techniques exist, these were prioritized due to their proven success in enhancing IDS performance and their relevance to the current state of the field. Subsequently, four ML algorithms (RF, XGB, LGBM, and DNN) were evaluated under different synthetic data configurations (10% to 100%) to determine the optimal percentage for intrusion detection. Next, two filter-based FS techniques (Consistency-based subset selection (CON) and Pearson correlation (C)) and two wrapper-based methods (Boruta (B) and Recursive Feature Elimination (RFE)) were applied to the balanced datasets to select the optimal feature set. These FS techniques were chosen based on their proven consistency and effectiveness in previous studies [15,18]. Finally, global (GS) and local (LIME, SHAP) interpretability methods were used to evaluate the constructed black box models both quantitatively and qualitatively. These interpretability methods were selected due to their ability to provide insights into model behavior, which enhances transparency and fosters trust and usability in critical applications across diverse fields, including breast cancer classification [30], X-ray analysis [31], NLP [32], and financial distress prediction [33] and cybersecurity [19,34].

The key contributions of this study are:

- Introducing a hybrid CTGAN-ENN approach to address class imbalance, reduce feature complexity, and enhance IDS interpretability.
- Comparing wrapper-based and filter-based FS techniques to optimize feature sets for improved classification efficiency.
- Demonstrating the superior performance of the CTGAN-ENN model in enhancing intrusion detection and addressing class imbalance.
- Identifying the optimal synthetic data ratio and its impact on detection accuracy, particularly with a reduced feature set.
- Providing global and local interpretability analyses to improve model transparency, with insights into feature importance and model behavior.

To carry out this study, the research questions (RQs) listed below are addressed and discussed in the Section 6:

(1) **RQ1:** Do CTGAN and ENN-based hybrid sampling method improve the model's capacity to classify instances from minority classes in comparison to the original data and traditional oversampling methods?

(2) **RQ2:** Do wrappers outperform filters when used for cyber-attack classification?
(3) **RQ3:** What is the optimal proportion of synthetic data generated by CTGAN-ENN that maximizes intrusion detection performance when used with a reduced feature set?
(4) **RQ4:** In what ways does the selected feature set influence the global and local interpretability of the constructed black-box models?

The rest of this paper is organized as follows: Section 2 describes the different methods of preprocessing as well as the black box models and the interpretability techniques used in this study. Section 3 reviews related work dealing with the application of GANs, FS, and interpretability techniques in the field of IDS, and outlines the objectives of the proposed model. Section 4 details the datasets used, along with the performance measures and statistical tests applied to determine the best performing models, as well as the interpretability metrics. Section 5 describes the experimental design of the study, while Section 6 presents and discusses the results obtained. Finally, Section 7 provides conclusions and suggests directions for future research.

2. Background

This section offers an overview of the oversampling techniques employed to address class imbalance, the FS selection techniques used, and the interpretability techniques examined in this study.

2.1. Hybrid sampling with CTGAN and ENN

In our approach, we integrated ENN approach for selective undersampling and CTGAN for synthetic minority sample generation. ENN removes noisy majority-class samples that conflict with their k = 3 nearest neighbors, preserving informative instances and minimizing overlap with minority classes. The imbalance rate is then effectively reduced in the CTGAN phase, which generates realistic minority samples to improve class distribution.

(A) Oversampling via Conditional Tabular Generative Adversarial Network (CTGAN) model

The architecture of GANs consists of two connected neural networks: the generator and the discriminator, which are illustrated in Fig. 1. The generator, as its name indicates, is responsible for producing synthetic data from random noise, while the discriminator learns to distinguish real from fake data. Through iterative training, the generator seeks to produce data that is indistinguishable from real data, refining its results over time. CTGAN [35] is a type of GAN specifically designed to address the various challenges encountered when generating synthetic tabular data. These challenges include handling mixed data types, non-Gaussian distributions, multimodal distributions, and highly unbalanced categorical columns. To tackle these issues, CTGAN employs reversible preprocessing techniques, including mode-specific normalization for numerical variables using a Gaussian Mixture Model (GMM) to capture data clusters. For categorical variables, it uses one-hot encoding and adds noise to binary variables, enhancing variability and realism in the synthetic data. These methods ensure the generation of high-quality, diverse synthetic tabular datasets suitable for various applications. This is followed by generating the probability distribution using Softmax. The process employs a Long Short-Term Memory (LSTM) network as the generator and a Multi-Layer Perceptron (MLP) as the discriminator, notably enhancing model warm-up efficiency.

The generation model G and discrimination model D operate independently in a GAN framework, undergoing alternating iterative training. The overall network's loss function is defined in Eq. (2) as the difference between the probability distributions of real data Pr and generated data Pg. When the distributions of Pr and Pg are similar, making it challenging for D to distinguish between real and fake samples, the probability approaches 0.5, indicating that the generator can produce realistic samples effectively.

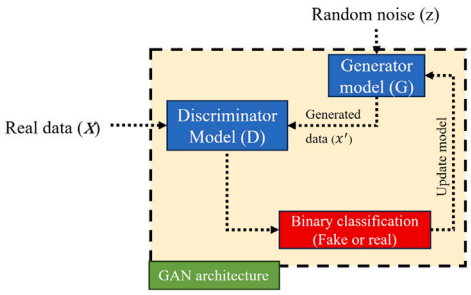

Fig. 1. The fundamental architecture of a GAN.

$$\min_G \max_D L(G, D) = \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{z\sim P_g}[\log(1 - D(G(z)))] \quad (2)$$

where: - G is the generator model that produces synthetic samples. - D is the discriminator model that classifies samples as real or generated. - x represents samples drawn from the real data distribution P_r. - z is a latent variable sampled from a prior distribution, which is mapped to $x' = G(z)$, the generated sample. - P_r is the probability distribution of real data. - P_g is the probability distribution of generated data. - $\mathbb{E}$ denotes the expectation operator, representing the statistical mean over the respective probability distributions.

The first term, $\mathbb{E}_{x\sim P_r}[\log D(x)]$, encourages the discriminator to correctly classify real samples by maximizing the probability of assigning them a high confidence score. The second term, $\mathbb{E}_{z\sim P_g}[\log(1 - D(G(z)))]$, represents the generator's objective to create samples that are indistinguishable from real ones, forcing the discriminator to misclassify them. This adversarial training process follows a min–max optimization strategy: the generator minimizes the loss to produce more realistic samples, while the discriminator maximizes it to improve classification accuracy. Over successive iterations, this competitive dynamic leads to an equilibrium where P_g closely approximates P_r, ultimately enhancing the quality and realism of generated samples.

(B) Undersampling via Edited Nearest Neighbors (ENN) model

The undersampling algorithm [36] enhances minority sample classification by selectively reducing majority samples. Utilizing the nearest neighbor principle, it calculates the distance between samples to find the closest neighbors of each majority sample. It then identifies and removes noisy samples by verifying if their labels match those of their nearest neighbors. The k-nearest neighbor (KNN) of a sample Si is defined as the set of samples in the dataset S that are closer to Si than the kth nearest neighbor of S_i. Formally, it is represented in Eq. (3) as:

$$K_{NN}(S_i, k) = \{S_j \in S \mid \text{dist}(S_i, S_j) < \text{dist}(S_i, S_i')\} \quad (3)$$

Where:

- S_i' is the kth nearest neighbor of S_i in the dataset S,
- dist denotes the distance between samples, typically using Euclidean distance.

From this definition, the primary objective of integrating ENN in our study is to reduce the imbalance rate by selectively removing instances where the label (i.e. "normal") conflicts with the majority label of their k=3 nearest neighbors (using Euclidean distance). This process minimizes overlap between the majority and minority classes, thereby lowering the imbalance ratio. As shown in Algorithm 1, we used ENN to select the instances from the majority class.

In sum, ENN is a decremental algorithm that starts with all samples and a number k. It removes a sample if the majority of its k nearest neighbors have different labels, indicating the sample is likely noise.

Algorithm 1 Edited Nearest Neighbors (ENN) Algorithm

Input: Dataset S, the majority samples S_{maj}, the number of nearest neighbors k, and the target number of deletions M.
Output: Dataset S' with reduced class imbalance.

Initialize $C_{\text{del}} = 0$.
for each sample S_i^{maj} in S_{maj} **do**
    Find its k nearest neighbors (using Euclidean distance) and store their indexes in K_{maj}.
    Select the three nearest neighbors of S_i^{maj} from K_{maj}.
    Compare the label of S_i^{maj} with the labels of its three nearest neighbors.
    if the label of S_i^{maj} differs from at least two of its three neighbors **then**
        Mark S_i^{maj} for deletion.
        Increment C_{del} by 1.
    end if
    if $C_{\text{del}} \geq M$ **then**
        break
    end if
end for
Remove all marked samples from the dataset S to obtain S' with reduced class imbalance.

2.2. Feature reduction

Building on our previous work [10], which highlighted Recursive Feature Elimination (RFE) and Boruta (B) as effective, and recent research [34], which identified CON and C as optimal for dimensionality reduction and IDS enhancement, we compare these four methods—two filters and two wrappers to select the best features.

2.3. Interpretability techniques

Interpretability techniques can be broadly categorized into global and local forms [37]. Global interpretability involves understanding the overall behavior of a model and how it generates predictions. In contrast, local interpretability focuses on explaining the reasoning behind a model's prediction for a specific input instance. Additionally, interpretability techniques can be classified based on their compatibility with ML models: agnostic and specific. Agnostic techniques, such as surrogate models, can interpret any type of model without needing to understand its internal workings. Specific interpretability techniques are tailored to particular model types, providing detailed insights into the model's internal operations. In this study, three interpretability techniques are employed.

Global Surrogate (GS): GSs are a category of ML models designed to approximate the behavior of black box models across the entire input space. By training these surrogates on the same input–output pairs used for training the black box model, valuable insights can be gained into the underlying logic of the black box. GS can come in various forms, typically transparent ML models like DTs, where the predicted labels of the black box model are used instead of the true labels in the dataset. The data used to train a GS, often referred to as Oracle data, reflects the behavior of the black box model rather than reality [38] The process of constructing surrogate models, as illustrated in Fig. 2, includes the following steps:

Generate predictions and create surrogate dataset: Replace the true labels in the original dataset with the predictions from the black box models (trainBB; Y_testBB) and create an Oracle dataset for training the Decision Tree (DT)-based surrogate model.

Build the surrogate model: Train a DT on the Oracle dataset. This DT captures the behavior of the black box model by using the generated predictions instead of the actual labels.

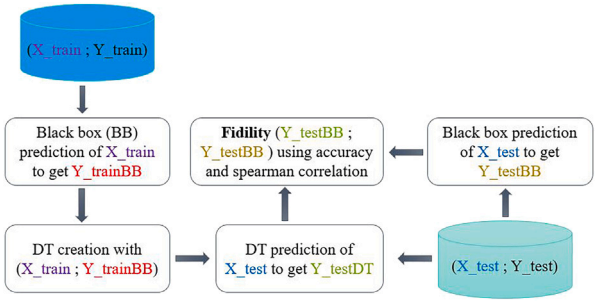

Fig. 2. Global surrogate process.

Evaluate the surrogate model: Compare the surrogate model's predictions on the test set with those of the black box model (X_testBB) using accuracy Eq. (4) and Spearman correlation Eq. (5), as these metrics were identified as the most reliable for fidelity by Schwartzenberg et al. [39]. The comprehensibility of the DT-based surrogate model is assessed by examining the depth of the DT, the number of rules, and the number of leaves.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Where: - TP is the number of true positives, - TN is the number of true negatives, - FP is the number of false positives, - FN is the number of false negatives.

$$\text{Spearman correlation}(\rho) = 1 - \frac{2\sum_{i=1}^{n} r_i}{\frac{n(n^2-1)}{2}} \tag{5}$$

Where: - r_i represents the rank difference for each pair, - n is the total number of observations.

Shapley additive explanations (SHAP): Shapley values, originating from cooperative game theory [40], offer a framework for fairly distributing the payout of a "game" among its players. In the context of ML, features act as players, collaborating and interacting to produce predictions. Lundberg et al. [41] introduced the SHAP framework, which approximates Shapley values to provide both global and local explanations. The summary plot in SHAP offers a visual depiction of global feature importance, revealing how each feature impacts model predictions overall. It displays Shapley values, representing the average contribution of each feature across various instances. Features with higher Shapley values exert greater influence on predictions, while those with lower values have less impact. This plot aids users in understanding the relative importance and directionality of features, helping identify key factors driving predictions. The Eq. (6) for calculating SHAP values is as follows:

$$\Phi_i = \frac{1}{M} \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[f_x(z')_S - f_x(z') \right] \tag{6}$$

where:

- Φ_i represents the SHAP value for feature i.
- M denotes the total number of features in the model.
- S is a subset of features excluding feature i.
- $f_x(z')_S$ and $f_x(z')$ are the model outputs with and without the features in subset S, respectively.

Local interpretable model-agnostic explanations (LIME): LIME is local model-agnostic technique introduced by Ribeiro et al. [42], operate differently from GSs by focusing on interpreting predictions at a local level. LIME constructs a surrogate interpretable model within the vicinity of a specific data point. This local surrogate is trained on a modified version of the data point's features, where the dataset is weighted based on its proximity to the data point. By employing a submodular pick algorithm, LIME presents the user

with various relevant instance explanations from the test set, offering insights into how features influence black box decisions. The LIME interpretability constraint can be expressed by Eq. (7) as follows:

$$\Phi'(x) \approx \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{7}$$

Where L represents the loss function, f is the original model, g is the interpretable surrogate model, π_x denotes the proximity measure, $\Omega(g)$ is a regularization term ensuring simplicity of the surrogate model, and G represents the class of interpretable models under consideration.

3. Related work and research objectives

In this section, we delve into recent studies that employ generative networks to address class imbalance, FS approaches, and interpretability techniques. It also highlights the main objectives guiding our research.

3.1. Related studies

Several studies have proposed diverse models and frameworks to enhance system effectiveness across various domains, such as IIoT data aggregation [43], WSN energy efficiency [44], COVID-19 and diabetes prediction [45,46], cloud efficiency [47,48], IoT drone security [49], and bioinformatics [50–52]. However, few studies have focused on enhancing interpretability while improving ML performance. In cybersecurity domain, existing works [53–55] have explored ML-based IDS improvements, yet gaps remain in FS, handling class imbalance, and interpretability. Our work addresses this gap by proposing CTGAN-ENN, a novel framework that combines generative adversarial networks, FS, and interpretability techniques to improve IDS performance and transparency.

Imbalanced intrusion datasets:

Ding et al. [20] proposed a tabular data sampling approach to address imbalanced learning. The method utilizes K-nearest neighbor (KNN) under-sampling for normal samples and a tabular auxiliary classifier GAN (TACGAN) for oversampling attack samples. Experimental results on three real intrusion detection datasets: KDDCUP99, UNSW-NB15, and CICIDS2017 demonstrate the effectiveness of the proposed approach compared to six other methods, achieving high performance with an accuracy of 93.53%. Kumar and al. [56] introduced a WCGAN-GP combined with an XGB to address the challenge of imbalanced data in IDS. The model utilizes gradient penalty to ensure stable learning. The results showed that integrating the proposed framework with the XGB classifier led to precision improvements of 96.66%, 83.36%, and 99.42% on the NSL-KDD, UNSW-NB15, and BoT-IoT datasets, respectively. Overall, the proposed approach effectively addresses data imbalance issues in IDS design by generating meaningful attack signatures. In [8], Park and al. proposed a novel WGANs-based Network IDS that addresses the data imbalance issue using a state-of-the-art generative model to produce realistic synthetic data. The authors implemented autoencoder-driven detection models based on DNN and Convolutional Neural Networks (CNN). The findings showed that the proposed models significantly outperformed existing ML and DL approaches, achieving accuracies of up to 93.2% on the NSL-KDD dataset and 87% on the UNSW-NB15 dataset. Additionally, the proposed model demonstrated efficient detection of network threats in both IoT and real-world enterprise environments, highlighting its effectiveness in resolving data imbalance and improving threat detection performance.

Moreover, Rahman and al. [57] investigated the use of 100% synthetic data generated by GANs for training ML models in IDSs to mitigate data imbalance and reduce data collection costs. Their study showed that synthetic data could effectively simulate real-world intrusion scenarios, achieving high performance. For the UNSW-NB15 dataset, they reported 90% accuracy, 91% precision, 90% recall, and 89% F1 score; for the NSL-KDD dataset, 84% accuracy, 85% precision, 84% recall, and 84% F1 score; and for the BoT-IoT dataset, perfect scores of 100% across all metrics. A recent study [21] introduced APELID, a novel IDS approach that integrates augmented-WGAN with PELID: Parallel Ensemble Learning-based Intrusion Detection to generate realistic samples for minority classes. The authors combined multiple ML models, including XGB, CBT, GBM, BME, and DNN, to improve precision and accuracy in intrusion detection. The results revealed that APELID surpasses existing methods, achieving F1-scores of 99.99% and 99.65% and False Negative Rates (FNRs) of 0.00% and 0.34% on the CSE-CIC-IDS2018 and NSL-KDD datasets, respectively. In [29], Zhao et al. proposed a novel approach to enhance Network IDS performance by integrating GANs to tackle data scarcity issues. They utilized three distinct GAN models – Vanilla GAN, WGAN, and CTGAN – to generate synthetic network traffic that closely mimics real-world behavior. Experiments with the CIC-IDS2017 dataset, enriched with GAN-generated data, demonstrated that this technique significantly improves IDS performance, particularly in detecting attacks with limited training data. The study highlights that incorporating GANs can effectively strengthen cybersecurity defenses in today's interconnected digital environment.

Interpretability in IDS:

XAI methods have been developed and implemented in various applications and fields. Prominent works focus on providing model explanations in areas such as natural language processing (NLP) [32], digital twin [58], time-series [59], and health technologies [30]. According to Arrieta et al. [60], XAI is defined as "*the capability of an AI system to provide explanations or reasoning that clarify its operations and make them comprehensible to a given audience*".

Keshk and al. [19] enhanced the performance and explainability of LSTM-based IoT network using SPIP based on four interpretability techniques (S: SHAP, P: Permutation Feature Importance (PFI), I: Individual Conditional Expectation (ICE), P: Partial Dependence Plot (PDP)). This framework offers both global and local explanations, enhancing the interpretability of cyber defense systems. By leveraging a unique set of input features extracted using SPIP and trained with LSTM, the proposed model achieved a superior detection accuracy of 87.3%, processing time, and interpretability when validated on the ToN-IoT dataset, outperforming existing techniques. Hariharan and al. [61] used PFI, SHAP, LIME, Contextual Importance, and Utility algorithms to improve the interpretability of ML models in IDS. These methods, which address both global and local scopes of explanation, were applied to RF, XGB, and LGBM models. The study compared these approaches based on accuracy, consistency, and stability, aiming to provide cybersecurity personnel with better insights into network traffic predictions. A case study focusing on DoS attack variants revealed the impact of features on prediction performance. In [62], Djenouri et al. introduced a novel framework for intrusion detection tailored to next-generation IoT environments. Their approach employs MinMax normalization for data preprocessing and the Marine Predator algorithm for FS. The selected features are then utilized to train a sophisticated recurrent neural network with an attention mechanism. To enhance model interpretability, Shapley values are computed to evaluate each feature's contribution to the predictions. Extensive simulations on the NSL-KDD dataset revealed the framework's effectiveness, achieving over 94% in both true negative and true positive rates, surpassing existing state-of-the-art methods that typically achieve below 90% on this challenging dataset.

In another study [63], the SHAP explainer was employed to examine a DL technique known as TEA-EKHO-IDS, which integrates enhanced krill herd optimization (EKHO) for breach detection in IoT-enabled cyber–physical systems. The system further enhances detection performance by combining explainable AI, bidirectional LSTM, and Bayesian optimization (BO-Bi-LSTM) for efficient classification. The findings showed that TEA-EKHO-IDS effectively identifies and classifies intrusions with a high success rate of 98.96%, offering a promising solution for improving security in industrial CPS. Nkoro et al. [64]

proposed an explainable DNN framework designed for intrusion detection within Metaverse learning environments. The authors utilized advanced FS methods and explainability techniques like SHAP and LIME to ensure the model's transparency and reliability. The findings showed that the proposed IDS model achieved a high accuracy of 99.9%, effectively distinguishing between anomalous and benign activities, thereby enhancing the security and trustworthiness of Metaverse learning platforms. Sharma and al. [65] proposed a novel DL-based IDS for IoT environments, addressing the challenges of FS and model interpretability. They developed two distinct models, a DNN and a CNN and applied these to the NSL-KDD and UNSW-NB15 datasets after employing a C filter-based FS approach. The findings showed that both models achieved higher accuracy rates compared to traditional methods, with the DNN model further enhanced by the application of explainable AI techniques, including LIME and SHAP, to provide transparency and a better understanding of the model's predictions.

4. Database description and performance criteria

This section covers the cybersecurity datasets utilized, the performance and interpretability metrics, statistical tests applied to evaluate and compare the models constructed, and the methodology proposed.

4.1. Dataset description

In this comparative study, three publicly available datasets were employed to assess the performance of the proposed methods, namely CSECIC-IDS2018 [66], CIC-ToNIoT [67], and NF-UNSW-NB15-v2 [67]. This subsection details each dataset. These datasets cover a wide range of attack scenarios and are essential resources for evaluating the performance of IDSs.

The CSECIC-IDS2018 dataset was created in collaboration between the CIC and the Communications Security Establishment (CSE). It includes various attack scenarios and consists of an infrastructure with 50 attacker machines targeting an organization with 5 departments. Each department has 420 machines and 30 servers. The dataset was captured over a 10-day period and includes network traffic captures, system logs, and features extracted using the CICFlowMeter tool. Table A.1 in the supplementary file, Appendix A shows the features corresponding to the CICFlowMeter feature set. This dataset has a total of 10,823,650 records after removing duplicate rows, of which 87.75% are benign and 13.25% are attacks as detailed in Table A.2 in the supplementary file, Appendix A.

The dataset CIC-ToN-IoT was developed by Sarhan et al. [67] derived from the ToN-IoT [68] dataset and features were extracted using the CICFlowMeter tool. The ToN-IoT dataset was created in an industrial network testbed and consisted of virtual machines running Windows, Linux, and Kali Linux systems. It was designed to capture normal and cyber-attack events in IoT networks across different scenarios.

The NF-UNSW-NB15-v2 dataset was developed by Sarhan and al. [67] derived from the UNSW-NB15 [69] dataset and features were extracted using the NetFlow tool. The NetFlow features are presented in Table A.1 in the supplementary file, Appendix A. It stands as a comprehensive resource in the realm of publicly available datasets. It has been widely adopted by numerous IDS models for both training and testing purposes due to its richness and diversity.

Table A.2 in the supplementary file, Appendix A offers an overview, including descriptions, the number of samples for both normal and attack classes, the total sample size, and the number of features extracted.

4.2. Performance criteria and statistical tests

(A) Model performance metrics:

In our study, several evaluation metrics that are appropriate for multi-label classification models with imbalanced class distributions were chosen to identify the optimal ML models. These metrics consist of Precision, Recall, F1-score, Area Under the Receiver Operating Characteristic Curve (AUC-ROC), Matthews Correlation Coefficient (MCC), G-means, and Kappa statistics. These metrics have been widely utilized in numerous research studies [21,22] to evaluate the performance of ML models in similar scenarios. Therefore, the seven metrics listed above are computed by Eqs. ((8), (9), (10), (11), (12), (13), (14)) as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

$$\text{Kappa} = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{12}$$

$$\text{AUC} = \int_0^1 TPR(FPR^{-1}(u))\, du \tag{13}$$

$$\text{G-Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{14}$$

Where, the precision, as defined in Eq.(8), measures the accuracy of positive predictions by the ratio of true positives (TP) to the sum of TP and false positives (FP). The recall (or sensitivity), as defined in Eq. (9), is the proportion of actual positives correctly identified, calculated as TP divided by the sum of TP and false negatives (FN). The F1-score, as defined in Eq. (10), combines precision and recall into a single metric, emphasizing their harmonic mean. The Matthews Correlation Coefficient (MCC), as defined in Eq. (11), assesses classification quality by considering true and false positives and negatives, providing a balanced measure even for imbalanced datasets. Kappa, as defined in Eq. (12), compares observed accuracy with expected accuracy by chance, adjusting for random agreement. AUC, as defined in Eq. (13), evaluates overall classifier performance by integrating the True Positive Rate (TPR) over the False Positive Rate (FPR), with higher values indicating better performance. G-Mean, as defined in Eq. (14), represents the geometric mean of sensitivity (recall) and specificity, ensuring a balance between these metrics. Together, these metrics offer a comprehensive framework for assessing classification models' effectiveness in distinguishing between classes.

(B) Validations, Statistical Testing, and Ranking Technique:

Validation: This study employs *K-fold cross-validation* to evaluate ML models by partitioning the data into K equal segments or folds. The model is trained using $K - 1$ of these folds and tested on the remaining fold. This process is repeated K times, with each fold serving as the test set once. The overall performance is averaged across all K iterations to give a reliable estimate of the model's effectiveness. This approach helps minimize model bias and gives an indication of how well the model generalizes to new or unseen data. A typical choice for K is 10, as it offers a favorable trade-off between bias and variance in performance estimates. Thus, with $K = 10$, the model is trained on 80% of the data and evaluated on the remaining 20%, repeating this cycle 10 times.

Statistical Test: The *Scott-Knott (SK) test* is used as a clustering technique in the context of analysis of variance (ANOVA). Developed by Scott and Knott in 1974 [70], the algorithm aims to identify distinct, non-overlapping groups based on comparisons of treatment means. Despite the development of other hierarchical clustering methods, such as those by Jolliffe (1975) [71] and Calinski and Corsten (1985) [72], the SK test remains favored for its effectiveness and simplicity. It is particularly valued for its ability to: (1) Achieve high performance compared to other statistical methods. (2) Categorize techniques into

clear, non-overlapping groups based on similar predictive performance, such as grouping methods with comparable Mean Absolute Error (MAE) values where differences are not statistically significant. In addition, unlike the Wilcoxon [73] and Friedman [74] tests, which only show if significant differences exist, the SK test goes further. It clusters models into groups based on their mean performances, allowing us to see not only the presence of significant differences but also the specific models that perform similarly or differently. This provides a clearer understanding of model relationships and performance differences.

Ranking: The *Borda Count (BC)* is a voting method used to determine election outcomes by attributing points to candidates based on their rankings from voters. Each rank is given a specific number of points, and the candidate with the highest total points wins. For this study, models are treated as candidates, and performance metrics serve as voters. The BC method was used to select the top classifier variant from the best SK clusters based on multiple metrics.

(C) Interpretability metrics:

In the field of ML, measuring interpretability is a challenge because of its variability and the absence of standardized evaluation methods. This variation in evaluation approaches makes it difficult to compare different techniques using consistent metrics, which creates a gap in interpretability assessment across research. To address this issue, our study takes a comprehensive approach, combining qualitative and quantitative analyses to assess the interpretability of the proposed NIDS model. Our main goal is to provide a nuanced understanding of model decisions in security-critical scenarios, building confidence and readiness for deployment.

In the context of interpretability, fidelity and understandability metrics have been used to assess the interpretability results:

The fidelity metric assesses how well a surrogate model (GS-DT) replicates the behavior of the original model (the trained models). It is evaluated by comparing the surrogate model's predictions with those of the original model on a separate test dataset. This comparison measures the discrepancy between the predictions of the two models, with a higher fidelity score indicating greater agreement. In this study, the fidelity was measured using accuracy and Spearman correlation, as illustrated in Eqs. (4) and (5), respectively.

Understandability was measured by taking into account factors such as the number of rules, depth, number of leaves and number of features used in GS-DT training.

5. Experimental design and proposed system

This section describes the empirical evaluation process conducted in this study, broken down into five main steps, as depicted in Fig. 3. Compared to previous studies [20,46,47] that focus on specific objectives and classifiers, this study stands out for its comprehensive approach to IDS evaluation. It examines the impact of various FS techniques on a balanced dataset using generative AI and offers a thorough post-hoc analysis of four black box models, providing extensive insights into their performance across different network contexts and attack scenarios.

The methodology of the experiment includes the following steps:

Step 1: Data Preprocessing:

This step involves three key experiments: **(1) Data preparation**: Clean the datasets by removing instances with NaN or Inf values and eliminating highly correlated features (correlation > 90%), constants, and duplicates. Normalize numerical features using Eq. (15) and reorganize categorical variables, such as "Port destination and source", into four categories (Port 21/22, Port 80, Port 8080, and others). Split each dataset into 80% training and 20% testing sets. **(2) Data augmentation**: Address class imbalance by using CTGAN-based oversampling to generate synthetic samples for minority classes and ENN-based undersampling for the majority class to reduce IR, remove

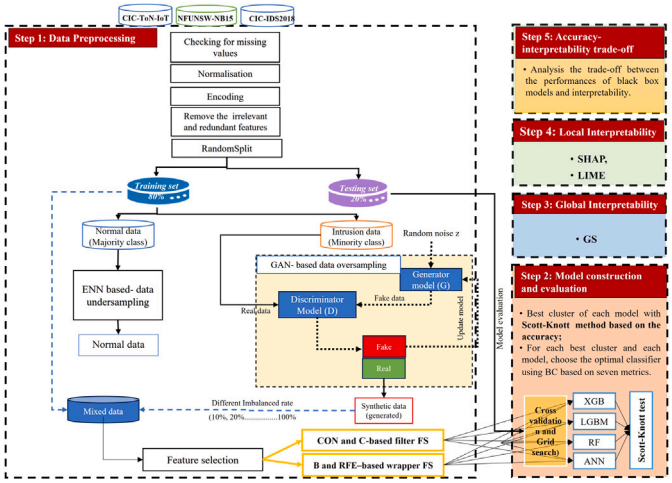

Fig. 3. Flowchart of the empirical evaluation.

noisy instances, and improve model performance. Then assess the effectiveness of our approach (CTGAN-ENN) by comparing it with two other GAN-based techniques (WGAN and WGAN-GP) and two traditional methods (SMOTE and ADASYN). **(3) Feature reduction**: Apply two wrapper methods (B and RFE) and two filter methods (CON and C) to each balanced dataset to reduce the feature space and enhance model transparency, resulting in 15 feature subsets (3 hybrid datasets ×(4 FS methods + original features)).

$$x_i = \frac{x_i - \mu}{\sigma} \tag{15}$$

where x_i is the individual data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset.

Step 2: Model construction and evaluation:

In this step, we conducted two stages of analysis. First, we employed the BC voting system, which ranks the techniques based on seven performance metrics: accuracy, precision, recall, F1-score, kappa, MCC, and AUC. Specifically, we compared the performance of our proposed CTGAN-ENN framework with four other data balancing techniques (WGAN, WGAN-GP, SMOTE, and ADASYN). This comparison was made using four black-box classifiers: XGB, RF, LGBM, and DNN, with all classifiers evaluated using their default settings and 10-fold cross-validation. We obtained a total of 24 variants for each classifier per dataset (Original dataset + five balanced datasets) ×4 models.

Next, after demonstrating the effectiveness of our proposed method (CTGAN-ENN), the impacts of filter and wrapper FS methods were compared (B, RFE, C, and CON) on the balanced datasets using a statistical clustering analysis (SK test) based on accuracy. A total of 12 additional feature subsets (4 FS techniques ×3 balanced datasets) were obtained. This comprehensive approach led to the creation of 48 classifier variants (12 balanced datasets ×4 models), allowing us to assess both the effectiveness of the data balancing methods and the impact of FS techniques. Optimal hyperparameters for each classifier were determined through grid search and 10-fold cross-validation, as presented in Table A.3 in the supplementary file, Appendix A. Once the best SK cluster for each classifier-dataset pair is determined, classifiers within this cluster are ranked using BC voting on the basis of seven metrics: precision, recall, f1-score, kappa, MCC, AUC, and Gmean, defined in Section 4, as well as the number of features on which they were trained. This strategy not only speeds up learning but also results in a simpler model for easier interpretability.

Step 3: global interpretability:

The aim of this step is to globally interpret each optimal classifier among the four black box models (Step 2) in order to understand their

behavior and the impact of each attribute on the final decision. Fig. 2 shows the GS's training process utilizing a DT for each experiment, with the black box model's predictions (Ytrain-BB and Ytest-BB) serving as class labels rather than original ground-truth labels. This improved dataset is known as the Oracle dataset. Notably, the DTs trained on the Oracle dataset mirror the behavior of the black box model rather than the ground-truth labels, since they lack access to the latter. To evaluate the surrogate model's fidelity to the black box model, criteria such as accuracy, depth, and the number of leaves were employed. Subsequently, fidelity-based SK tests and BC voting, incorporating these criteria, were performed to determine the best-performing surrogate model for each dataset. On the other hand, the SHAP summary plots are generated to provide users with insights into the relative importance and directionality of features, enabling them to understand how they collectively influence model results and to identify influential factors in predictions.

Step 4: Local interpretability:

This step applied LIME and SHAP to generate explanations for each instance in the test set. To evaluate the LIME and SHAP interpretability techniques, we focused on how each method illustrates feature importance in local explanations. Specifically, we assessed how well LIME and SHAP highlight the contribution of individual features to the model's predictions for each instance.

6. Results and discussion

This section discusses and examines the results of the empirical evaluation, aiming to address the RQs outlined in Section 1. The experiments were performed on a laptop equipped with a hexa-core Intel Core i7-8050H processor, 32 GB of RAM, and a base speed of 2.59 GHz, operating on Windows 11. The Scott-Knott statistical test was applied using R software.

6.1. Data preprocessing

We thoroughly preprocessed three intrusion datasets: CICIDS2018, CICToNIoT, and NF-UNSW-NB15-v2 to ensure their suitability for classifier construction. This preprocessing involved several steps:

Data cleaning and transformation: We began by checking for missing values and removing instances with NaN or Inf values to maintain data integrity. Highly correlated attributes (over 90%), constants, and duplicates were identified and eliminated to reduce redundancy. A critical step involved addressing duplicated rows with differing classes; we definitively removed these duplicates, which reduced the dataset size and ensured that only instances with consistent class labels remained. Numerical features were normalized using Min-Max normalization, and categorical variables such as "Port destination and source" were reorganized into specific categories (Port 21/22, Port 80, Port 8080, and others).

Data Sampling: The CTGAN model was trained through an iterative process involving both the generator (G) and the discriminator (D). Details of the hyperparameters used for both ENN and CTGAN are provided in Table A.4 in the supplementary file, Appendix A, ensuring replicability and reliability of the results. Additionally, Table 1 resumes the characteristics of each dataset after the preprocessing step, including the number of informative features, and the data distribution across different classes before and after applying the CTGAN-ENN model, highlighting the IR for each class.

Feature Selection: Once the synthetic data was generated, four FS techniques were applied to identify the most informative feature subsets. This step ensures that only relevant attributes are used for model training, improving computational efficiency and enhancing the classifier's ability to distinguish between attack and normal traffic. By reducing dimensionality, FS also helps mitigate the risk of overfitting and improves the interpretability of the model. Table A.5 in the supplementary file, Appendix A summarizes the features and their numbers resulting from applying the filters and wrappers FS techniques on the three datasets.

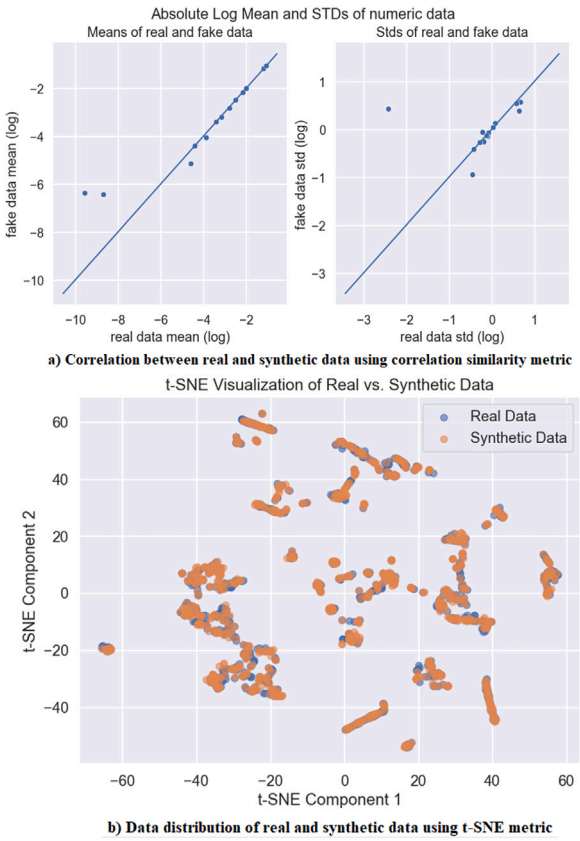

Fig. 4. Statistical and structural comparison of real and synthetic data.

Synthetic Data Validation: To validate our claims about avoiding overlap between majority and minority classes, we have incorporated both quantitative and visual analyses to assess the quality of the synthetic data generated by CTGAN-ENN. Specifically, we used TableEvaluator to compare key statistical properties (e.g., feature distributions and correlations) between the real and synthetic datasets. For instance, Fig. 4(a) compares the log-transformed means and standard deviations of real and fake data-based CTGAN-ENN for CIC-IDS2018. The left plot shows that the fake data closely approximates the real data's means, with most points aligning along the diagonal. The right plot indicates that while the standard deviations of fake data generally follow real data. According to the t-SNE, Fig. 4(b) shows a strong overlap between real and generated data, indicating that the CTGAN-ENN has effectively captured the key features of the real data in CIC-IDS2018 dataset. This quantitative analysis confirmed that the synthetic data closely mirrors the real data while preserving diversity. Furthermore, Fig. B.1 in the supplementary file shows the similarity between the distributions of the real and synthetic data for several features using cumulative sums, showing a strong overlap, suggesting that the synthetic data effectively captures the distribution of the real data. Additionally, we employed t-SNE and PCA visualizations to evaluate class separability and inter-class boundaries, as shown in Fig. B.2 in the supplementary material, which indicated that our data preprocessing techniques effectively reduced class overlap and enhancing the distinctiveness of different attack categories.

Finally, to address the challenges of computational costs, training stability, and scalability, we have proposed several mitigation strategies tailored to CTGAN-ENN. Table A.6 summarizes the key challenges and mitigation strategies for computational costs, training stability, and

Table 1
Summary of the preprocessed datasets.

Dataset	Class	Encoding	Before CTGAN-ENN		After CTGAN-ENN		Ts set	# of Informative Features
			Tr set	IR	Tr set	IR		
CSE-CIC-IDS2018	Benign	0	686,110	–	686,110	–	293,688	38
	Brute-force	1	65,613	10.46	665,613	1.03	28,431	
	DDOS	2	680,253	1.01	680,253	1.01	292,167	
	Web Attacks	3	605	113.41	600,605	1.14	250	
	Bot	4	101,552	6.76	601,552	1.14	42,968	
	Total		**1,534,133**		**3,234,133**		**657,504**	
CIC-ToN-IoT	Benign	0	696,743	–	696,743	–	300,013	42
	Scanning	1	21,736	32.05	621,736	1.12	10,737	
	Backdoor	2	18,982	36.71	618,982	1.13	8,007	
	Ransomware	3	3,314	210.24	603,314	1.15	1,495	
	MitM	4	345	2019.54	600,345	1.16	160	
	Total		**741,120**		**3,141,120**		**320,412**	
NF-UNSW-NB15-v2	Benign	0	135,549	–	135,549	–	59,695	31
	Exploits	1	10,520	12.88	120,520	1.12	6,689	
	Fuzzers	2	23,750	57.07	122,375	1.11	2,151	
	Generic	3	1,731	78.31	121,731	1.11	1,084	
	DoS	4	2,166	62.58	122,166	1.11	952	
	Total		**152,341**		**622,341**		**70,571**	

scalability in CTGAN-ENN, offering solutions such as distributed training, FS, adaptive learning rates, and incremental learning to enhance performance and scalability in real-world deployments.

6.2. Model construction and performance (RQ1)

This section presents the performance evaluation of 24 classification scenarios over each dataset, involving four classifiers (RF, XGB, LGBM, and DNN) and four distinct oversampling approaches: original (no sampling), SMOTE, ADASYN, WGAN, and WGAN-GP, along with the proposed CTGAN and ENN-based hybrid sampling methods. While traditional methods tend to replicate minority class instances, leading to overfitting, CTGAN generates realistic new samples that maintain the inherent distribution of the minority class. At the same time, ENN effectively removes noisy and borderline instances from the majority class, preventing the model from being misled by irrelevant or misclassified data. This combined approach not only improves class balance but also enhances the quality of the training set, reducing redundancy and preserving key patterns, resulting in more robust and generalizable feature representations, particularly in high-dimensional datasets. Table A.7, A.8, and A.9 in the supplementary file, Appendix A list the performance values and the ranking of each classifier applied to the CIC-IDS2018, CIC-ToN-IoT, and NF-UNSW-NB15-v2 datasets, respectively. These tables are generated without any parameter tuning and are based on seven performance metrics: accuracy, precision, recall, F1-score, kappa, MCC, and AUC.

Taking into account each oversampling method:

- From Table A.7, A.8, and A.9, it is clear that the proposed CTGAN-ENN technique exhibits exceptional performance across all datasets. CTGAN-ENN improves accuracy by approximately 0.01% (from 99.98% to 99.99%), 0.04% (from 99.60% to 99.64%), and 0.28% (from 95.97% to 96.22%) in the CIC-IDS2018, CIC-ToN-IoT, and NF-UNSW-NB15-v2 datasets, respectively, compared to the original imbalanced data and shows substantial advantages over traditional methods such as SMOTE and ADASYN. Additionally, the proposed technique surpasses both WGAN and WGAN-GP methods in BC score. These results underscore the superior overall performance of the CTGAN-ENN approach in enhancing model classification and robustness.
- The classifiers using the original imbalanced dataset generally perform well in terms of performances, but with lower BC scores compared to GAN-based methods.
- SMOTE and ADASYN traditional oversampling methods, do not significantly enhance the performance metrics. In all experiments, these methods result in lower BC scores compared to the original data, indicating limited effectiveness in handling class imbalance for IDS.
- Both WGAN and WGAN-GP show substantial improvements in the average accuracy values over the traditional methods. This indicates that GAN-based oversampling methods are also effective options for resampling the intrusion datasets. In contrast, SMOTE and ADASYN consistently yielded the lowest average accuracy values across all classifiers.
- With the data augmentation achieved through the CTGAN and ENN hybrid sampling methods, combined with ensemble learning models such as XGB and LGBM, exceptional performance in intrusion detection has been reached. The average accuracies are 99.99%, 99.64%, and 96.22% for the CSE-CIC-IDS2018, CIC-ToN-IoT, and NF-UNSW-NB15-v2 datasets, respectively.

In sum, the proposed CTGAN-ENN method outperforms all other methods across almost all metrics, achieving the highest average accuracy and BC scores. Fig. B.3 in the supplementary file, Appendix B illustrates the confusion matrix of the best accuracy achieved on the original dataset compared to the best accuracy obtained using the proposed CTGAN-ENN model across all datasets. This matrix shows the enhanced precision in correctly classifying attacks and reducing false positives and false negatives. This demonstrates the superiority of the proposed model in handling class imbalance and enhancing overall model performance. Consequently, these comparisons help us assess the effectiveness of our CTGAN-ENN method and offer further insights into addressing RQ1.

6.3. Comparative analysis of filter and wrapper-based dimensionality reduction techniques on CTGAN-ENN balanced data (RQ2)

To further address the high-dimensionality challenge and improve model interpretability, this subsection compares FS techniques after data augmentation. Specifically, we employed two filter methods (C and CON) and two wrapper methods (B and RFE) to reduce the feature space, thereby improving computational efficiency and enhancing the interpretability of the models.

In order to conduct a more thorough analysis of the CTGAN-ENN oversampling method, this subsection compares the performance of two filter methods (C and CON) and two wrapper methods (B and RFE) using four black box models (RF, XGB, LGBM, and DNN). Specifically, we investigated how the integration of FS methods influences the

Table 2
Statistics of occurrences of classifiers, filters, and wrappers in the three best SK clusters over each CTGAN-ENN-based dataset.

	CIC-IDS2018		CIC-ToN-IoT		NF-UNSW-NB15-v2	
	% of Classifiers	# of Occurrences	% of Classifiers	# of Occurrences	% of Classifiers	# of Occurrences
Single Filters (8)						
First Cluster	38%	3	0%	0	0%	0
Second Cluster	0%	0	25%	2	0%	0
TOTAL	**38%**	**3**	**25%**	**2**	**0%**	**0**
Single Wrappers (8)						
First Cluster	75%	6	25%	2	25%	2
Second Cluster	0%	0	13%	1	25%	2
TOTAL	**75%**	**6**	**38%**	**3**	**50%**	**4**

performance and efficiency of intrusion classification when combined with our proposed CTGAN-ENN oversampling technique.

In this experiment, 16 combinations (four classifiers ×four FS techniques) were systematically evaluated for each dataset to identify the best SK cluster. Hyperparameters for all combinations were chosen based on accuracy assessed through 10-fold cross-validation with grid search. The performance of each combination was then evaluated using the performance metrics on the test sets. Fig. B.4 in the supplementary file, Appendix B illustrates the SK results, consisting of three SK plots, each corresponding to a dataset. These plots feature two axes: the x-axis represents the classifier variants, with the best placements starting on the left, and the y-axis represents the accuracy values. A vertical line indicates the tenfold cross-validation results for each classifier, with a small dot at the center of each line representing the average accuracy values.

To compare the impacts of filters and wrappers on classification performance, we limit our analysis to the top two SK clusters for each dataset. Table 2 provides statistics that aid in interpreting the SK test results. It shows the frequency of occurrence of filters and wrappers in the top three clusters (column '# of occurrences') and the percentage of classifiers present in these clusters (column '%. of classifiers'). Fig. B.4 in the supplementary file, Appendix B, also helps identify which FS methods are most frequently used in the best-performing clusters, offering insights into the effectiveness of each FS technique. From Fig. B.4 in the supplementary file, Appendix B, and Table 2, it can be observed that:

- Over the CICIDS2018 dataset, we identified 5 clusters, with the first SK cluster encompassing 38% of the filter techniques (9 out of 32) and 75% of the wrapper ones (6 out of 8). The second SK cluster contains 0% of filters and 0% of wrapper ones. Therefore, in the two top SK clusters, filters account for 38% of occurrences, while wrappers make up 75% of occurrences. This indicates that wrapper techniques demonstrated a higher prevalence and effectiveness compared to filters in this dataset.
- Over the CIC-IoTToN dataset, we identified 8 clusters, with the first SK cluster encompassing 0% of the filter techniques and 25% of the wrapper ones (2 out of 8). The second cluster contains 25% of filters (2 out of 8) and 13% of wrappers (1 out of 8). Therefore, in the two best SK clusters, filters account for 25% of occurrences, while wrappers make up 38% of occurrences. This showcases the superiority of wrapper techniques compared to filters in this case.
- Over the NF-UNSW-NB15-v2 dataset, we identified 10 clusters, with the first SK cluster encompassing 0% of the filter techniques and 25% of the wrappers (2 out of 8). The second cluster contains 0% of filters and 25% of the wrappers (2 out of 8). Therefore, in the top two SK clusters, filters account for 0% of occurrences, while wrappers make up 50% of occurrences. This indicates that wrappers demonstrated a higher prevalence and effectiveness compared to filters in this dataset.

While both wrapper techniques demonstrate superior effectiveness, a direct comparison between B and RFE, based on the SK statistical test, revealed that RFE consistently appeared in the best SK cluster across all datasets (see Fig. B.3 in the supplementary file, Appendix B). Moreover, RFE achieved optimal feature selection by using fewer features without compromising model performance, making it more effective than B in enhancing classification accuracy. For example, RFE reduced the feature space by approximately 60% (from 38 to 14 features) for CIC-IDS2018 while maintaining an accuracy around 99.98%. In contrast, B required 26 features to achieve similar performance, highlighting RFE's efficiency in selecting a more compact yet effective subset of features. Additionally, this study demonstrated that the RFE and CON methods, when combined with ensemble learning techniques such as LGBM and XGB, achieved strong performance. This result is consistent with the findings of our earlier research [11,19], which evaluated various FS techniques on intrusion datasets without the CTGAN-ENN oversampling method. The key takeaway is that the introduction of the CTGAN-ENN method for data balancing did not significantly affect the performance of these FS techniques. This finding supports the conclusion that the effectiveness of FS techniques is robust, regardless of the data balancing approach used.

Moreover, reducing the feature space simplifies the model, making it less prone to overfitting and more interpretable by focusing on the most significant features. By focusing on the most relevant features, feature reduction simplifies the model's structure, making it more transparent and easier to interpret with techniques like GS models.

To determine the most effective FS technique for the three datasets, we focused on identifying classifier variants (XGB, LGBM, RF, and DNN) that not only used the smallest possible number of features but also belonged to the top-ranked SK clusters. The rationale behind this approach is that classifiers within these top clusters have demonstrated superior performance, indicating that they benefit from an optimal balance between feature reduction and model accuracy. The selected optimal classifiers were fine-tuned through a grid search process aimed at maximizing accuracy. The optimal hyperparameters identified through this process are detailed in Table 3. In addition, computational complexity metrics, including training time and testing time, are also displayed in Table 3, where up-arrows (↑) and down-arrows (↓) have been added next to each performance indicator; up-arrows indicate that a higher value is better (e.g., accuracy), while down-arrows indicate that a lower value is better (e.g., the number of features).

From Table 3, it is obvious that wrapper-based FS techniques, particularly RFE, plays a crucial role in optimizing model performance, with 14, 16, and 18 features for CIC-IDS2018, CIC-ToNIoT, and NF-UNSW-NB15-v2, respectively. For the CIC-IDS2018 dataset, RFE was the best FS technique for RF, and LGBM models, while CON was most effective for XGB and DNN models. For the CIC-ToNIoT dataset, RFE was advantageous for XGB, DNN, and LGBM, whereas CON was beneficial for RF. In the NF-UNSW-NB15-v2 dataset, the B technique was used with RF and DNN models, and RFE was employed with XGB and LGBM. Overall, wrapper-based FS techniques, especially RFE and B, demonstrated significant effectiveness in improving model accuracy and feature efficiency across all evaluated datasets.

In terms of computational complexity, XGB models demonstrated the highest efficiency across all datasets, showing the shortest training

Table 3
Optimal classifiers hyperparameters and performances using grid search over each dataset.

Model	Hyperparameters	CIC-IDS2018	CIC-ToN-IoT	NF-UNSW-NB15-v2
RF	N estimators	600	1200	800
	Max depth	26	26	24
	FS technique	RFE	CON	B
	# of features (↓)	14	12	14
	Accuracy (↑)	99.98%	99.63%	96.21%
	Training time/Testing time (↓)	55.3 s/12 s	51.9 s/9.2 s	32.3 s/5.4 s
XGB	Max depth	18	20	20
	Gamma	0.5	0.6	0.7
	FS technique	CON	RFE	RFE
	# of features (↓)	15	16	18
	Accuracy (↑)	99.99%	99.64%	96.26%
	Training time/Testing time (↓)	18.6 s/4.5 s	19.1 s/2.32 s	10.4 s/1.2 s
DNN	Number of Layers	5	5	5
	Number of Neurons per Layer	500, 250, 150, 100, 50	500, 250, 150, 100, 50	500, 250, 150, 100, 50
	Dropout rate	0.4	0.5	0.6
	FS technique	CON	RFE	B
	# of features (↓)	15	16	14
	Accuracy (↑)	99.95%	98.99%	94.95%
	Training time/Testing time (↓)	245 s/115 s	236 s/78 s	156 s/32 s
LGBM	Number of estimators	500	400	400
	Max depth	50	50	10
	Learning rate	0.01	0.01	0.01
	FS technique	RFE	RFE	RFE
	# of features (↓)	14	16	18
	Accuracy (↑)	99.99%	99.64%	96.26%
	Training time/Testing time (↓)	112 s/29 s	109 s/7.5 s	28 s/3.6 s

and testing times. For instance, XGB on the NF-UNSW-NB15-v2 dataset required only 10.4 s for training and 1.2 s for testing, reflecting its optimized performance with minimal computational demand. Similarly, XGB maintained low times on the CIC-IDS2018 and CIC-ToN-IoT datasets (18.6s/4.5s and 19.1s/2.32s, respectively).

For the other models, RF provided moderate efficiency with higher training times yet balanced computational demands. LGBM and RF models demonstrated moderate computational efficiency, with training times that were generally higher than those of XGB. Meanwhile, DNN models, although achieving high accuracy, exhibited the longest training and testing times, indicating a higher computational cost across all datasets.

6.4. Effectiveness of CTGAN-ENN under different balance rates (RQ3)

To further demonstrate the robustness of the CTGAN-ENN method in managing data with varying imbalance ratios while also reducing the execution time for training models, we mixed the CTGAN-ENN generated synthetic data with the original training data at various ratios. For classification, we employed XGB with RFE for the CIC-ToN-IoT and NF-UNSW-NB15-v2 datasets, and LGBM with RFE for CIC-IDS2018. The choice of XGB and LGBM was motivated by their high accuracy and efficient computational time as shown in Table 3, essential for testing multiple synthetic data ratios. RFE further enhanced efficiency by selecting the most informative features, reducing dimensionality, and expediting training, thus balancing accuracy with computational effectiveness across datasets. The imbalance ratios are managed manually by combining the generated synthetic data with the original data at various predefined ratios: 10%, 20%, ..., and 100%, amount of synthetic data relative to the original data.

Fig. 5 shows the performance variations of the proposed model according to different generated ratios, where 0% represents the experimental results of the original data. The y-axis represents the numerical values attributed to the four metrics (precision, recall, G-mean, and MCC). From Fig. 5, we observe that all metrics generally improve as the percentage of CTGAN-ENN generated synthetic data mixed with the original data increases. The most significant improvements in performance are observed at the 70% and 80% mixing ratios, where precision, recall, and MCC reach their highest values. This suggests that incorporating a substantial proportion of synthetic data helps in better balancing the classes, thereby enhancing the model's ability to correctly classify instances. This balance is not only key to maximizing model accuracy but also plays a significant role in optimizing computational efficiency.

6.5. Interpretability results (RQ4)

To assess the overall interpretability of each classifier, this section provides the results of global and local interpretability techniques (SHAP, LIME, and GS) applied to the optimal classifier-FS combinations (highlighted in Fig. B.4 and Table 3) across each dataset.

Global interpretability:

While the CTGAN-ENN model has demonstrated high effectiveness in detecting cyber-attacks, understanding the factors that drive its predictions is crucial for transparency. Our goal is to provide domain engineers with tools that enable swift and visual identification of anomalies in sensors or actuators.

Initially, we employed the SK test to assess whether there were significant differences between the surrogates of the four optimal classifiers, based on their fidelity values. Furthermore, we utilized the BC scores to evaluate the global interpretability of the models for each dataset, using the metrics outlined in Table 4.

Next, SHAP summary was applied to visualize the contribution of individual features. To streamline the presentation of results, we selected XGB as the representative classifier for SHAP analysis across all datasets, given its faster training times and strong performance, as demonstrated in Table 4. SHAP summary helps pinpoint the most influential features driving the model's decisions, enhancing trust in its predictions. Unlike PFI, which measures performance drops when feature values are shuffled, SHAP calculates the magnitude of feature contributions.

Table 4 presents the results of the global surrogate models (DTs) in terms of accuracy and Spearman correlation fidelity, along with the depth and number of leaves. Up-arrows (↑) and down-arrows (↓) have been added next to each performance indicator, where up-arrows signify that a higher value is better (e.g., accuracy fidelity), and down-arrows indicate that a lower value is preferred (e.g., depth).

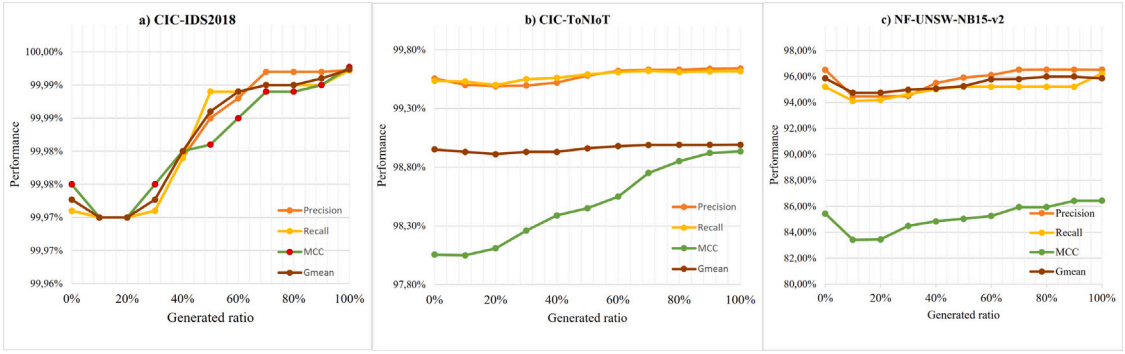

Fig. 5. Performance of CTGAN-ENN with different ratios of synthetic data, where the x-axis represents the percentages of synthetic data and the y-axis represents the values of precision, recall, G-mean, and MCC.

Table 4
Metrics comparison of surrogate models over each dataset.

Metrics	Surrogate Model	CIC-IDS2018	CIC-ToN-IoT	NF-UNSW-NB15-v2
Accuracy (↑)	XGB	99.98%	99.63%	96.18%
	RF	99.98%	99.64%	95.43%
	LGBM	99.98%	99.64%	96.18%
	DNN	99.95%	99.00%	94.95%
Fidelity (↑)	XGB	99.97%	99.42%	98.63%
	RF	99.98%	99.18%	97.70%
	LGBM	99.98%	99.41%	98.60%
	DNN	99.98%	99.67%	94.74%
# of Leaves (↓)	XGB	175	269	635
	RF	150	336	1488
	LGBM	142	269	547
	DNN	154	140	178
Depth (↓)	XGB	23	20	24
	RF	25	20	37
	LGBM	25	20	24
	DNN	24	15	19

Table 5
Global interpretability Borda winners.

Dataset	Borda Winner
CIC-IDS2018	LGBM & XGB
CIC-ToN-IoT	DNN
NF-UNSW-NB15-v2	DNN & XGB

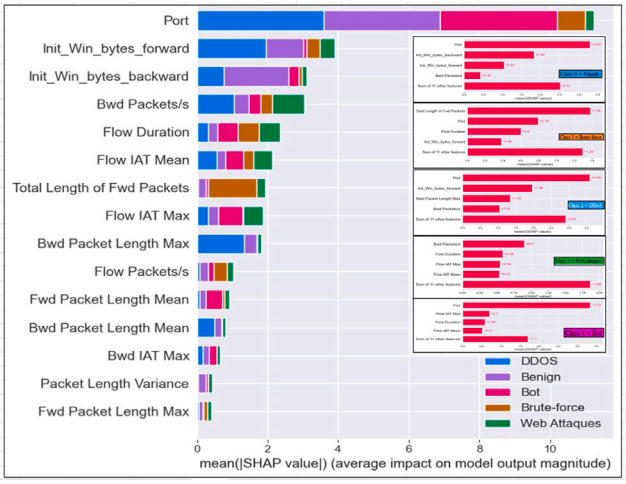

Fig. 6. Feature contributions for CIC-IDS2018 dataset using SHAP summary for XGB_CON.

According to Table 4, all surrogate models exhibit high accuracy and fidelity scores, with DNN performing best in CIC-IDS2018 and CIC-ToNIoT datasets, and XGB excelling in NF-UNSW-NB15-v2. However, when evaluating the depth of the TDs, DNN obtained the smallest depth (15 for CIC-ToNIoT and 19 for NF-UNSW-NB15-v2), except for CIC-IDS2018, where XGB outperformed with a depth of 23. The results also show that XGB, RF, and LGBM achieved higher accuracy across all datasets when comparing the surrogate models against the true labels (original dataset).

Moreover, Table 5, which ranks the models based on four global interpretability metrics (accuracy, fidelity, number of leaves, and depth), highlights that the DNN outperformed in the CIC-ToNIoT and NF-UNSW-NB15-v2 datasets, while the XGB excelled in both the CIC-IDS2018 and NF-UNSW-NB15 datasets. These findings underscore the strength of the DNN and XGB models in maintaining both interpretability and performance across different datasets.

Moreover, Fig. 6 in the main text, along with Figs. B.5 and B.6 in the supplementary file, Appendix B, provide features with their corresponding SHAP values for the testing set of each dataset using XGB model. These plots demonstrate how SHAP can quantify the impact of each feature on the model predictions. It offers a visual depiction of each feature within the SHAP summary plot. Moreover, by leveraging the information provided by Fig. 6 in the main text, along with Figs. B.5 and B.6 in the supplementary file, Appendix B, some observations can be made for each dataset separately:

For the CICIDS2018 dataset, Fig. 6 illustrates the impact of features on class predictions for the XGB_CON classifier, which relies heavily on the "Port" feature across all classes. Specifically, "Port", "Init win bytes backward", and "Init win bytes forward" are the most influential features for classifying instances as normal. For "Brute-force" attacks, key features include "Total Length of Fwd Packets". "Port", and "Flow duration". In the case of "DDoS" attacks, "Port", "Init win bytes backward", and "Bwd Packet Length Max" play crucial roles. Meanwhile, "Bwd Packets/s", "Flow duration", "Flow IAT Max", and "Flow IAT Mean" are pivotal for identifying "Web attacks". For "Bot attacks", the classifier focuses on "Port", "Flow IAT Max", and "Flow duration".

For CIC-ToNIoT dataset, Fig. B.5 reveals the contribution of various features to class predictions made by the XGB_RFE classifier, with a significant reliance on the "Idle Max" feature across all attack classes. Specifically, "Fwd Seg Size Min", "DstPort", "Init win bytes forward", and "Fwd packet length Max" are the most influential features for classifying instances as normal. For "Scanning" attacks, key features include "Idle Max", "DstPort", "Fwd packet length Min", and "Fwd Seg Size Min". In the case of "Backdoor" attacks, "Idle Max", "Bwd IAT Tot", "Init win bytes backward", and "Idle Mean" play crucial roles. Meanwhile, "Idle Max", "Fwd Seg Size Min", "Fwd packet length Max", and "Idle mean" are pivotal for identifying "Ransomware". For "MiTM" attacks, the classifier focuses on "Idle Max", "Fwd Seg Size Min", "Tot Bwd Packets", and "Init win bytes forward".

For the NF-UNSW-NB15-v2 dataset, Fig. B.6 indicates that the "MIN-TTL" feature plays a dominant role in determining predictions

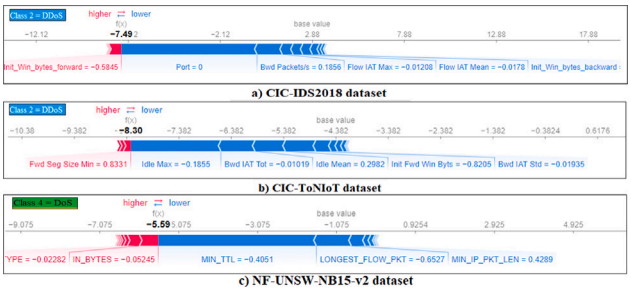

Fig. 7. SHAP Force Plot for local interpretation of XGB.

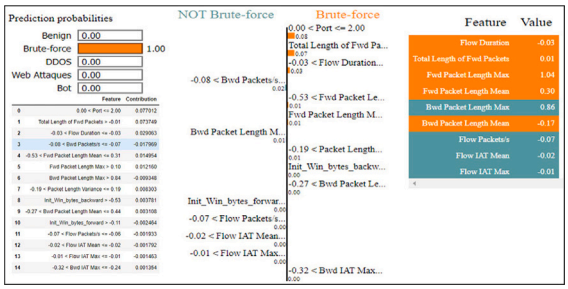

Fig. 8. LIME explanation of CIC-IDS2018 dataset instances.

across all classes, with one notable exception. In the case of the "Generic" attack class, the model relies on a combination of "MIN TTL", "IN BYTES", and "LONGEST FLOW PKT", with all three features contributing nearly equally to the classification decision. For "Normal" traffic, key features include "MIN TTL", "LONGEST FLOW PKT", "TCP WIN MAX IN", and "DstPort". In the case of "Exploits" attacks, "MIN-TTL", "SHORTEST FLOW PKT", "MIN IP PCK LEN", and "LONGEST FLOW PKT" play crucial roles. Meanwhile, "MIN TTL", "MIN IP PCK LEN", "DstPort", and "TCP FLAGS" are pivotal for identifying "Fuzzers" attacks. For "DoS" attacks, the classifier focuses on "MIN-TTL", "LONGEST FLOW PKT", "IN BYTES", and "MIN IP PCK LEN". In sum, it was observed that, over the analysis of SHAP feature contributions across the three datasets (CIC-IDS2018, CIC-ToNIoT, and NF-UNSW-NB15-v2), different features dominate predictions depending on the attack class and dataset. For CICIDS2018, the "Port" feature consistently played a pivotal role across multiple attack classes, particularly in "DDoS" and "Brute-force" attacks. Similarly, in the CIC-ToNIoT dataset, "Idle Max" emerged as the most influential feature across all attack types. In the NF-UNSW-NB15-v2 dataset, "MIN-TTL" was the primary determinant in predicting outcomes for most classes, with some variation seen in "Generic" attacks where "IN BYTES" and "LONGEST FLOW PKT" also contributed equally. These insights underscore the importance of identifying key features driving model predictions, which in turn enhances the interpretability and trustworthiness of intrusion detection models across different datasets.

Local interpretability:

SHAP provides both global and local explanations of black box's decisions, allowing us to understand the overall feature importance and how specific predictions are made. While LIME focuses solely on local interpretability by selecting a particular instance from the testing dataset. In Fig. 7, the local SHAP plot illustrates the contribution of each feature to the prediction of a DDoS instance across all datasets.

From Fig. 7(a), which represents the CIC-IDS2018 dataset using the XGB_CON model, the base value, or the average model prediction for this dataset, is 2.88. In this particular DDoS instance: "Init Win bytes Forward" is the only feature that increases the model's prediction beyond the base value. On the other hand, several features contribute to decreasing the prediction: "Port", "Bwd Packets/s", "Flow IAT Max", "Flow IAT Mean", and "Init Win bytes Backward".

In Fig. 7(b), the local SHAP plot for the scanning instance example in the CIC-ToNIoT dataset shows a similar breakdown of feature contributions for the XGB_RFE model. The base value for this dataset is the average prediction across all data points, set at −2.432. In this case, the features "DstPort", "Idl Max", and "Fwd Pkt Len Max" significantly increase the prediction for this instance, pushing the value toward the base prediction. Conversely, features such as "Init Bwd Win Byts", s"Bwd IAT Tot", "Idle Mean", "Init Win bytes Forward", and "TotLen Bwd Pkts" decrease the prediction.

From Fig. 7(c), representing the NF-UNSW-NB15-v2 dataset, the base value of the XGB_RFE model is −1.075. In this case, the features "DNS_QUERY_TYPE" and "IN_BYTES" contribute to increasing the prediction for the DoS instance, pushing the model's output above the base value. On the other hand, features such as "MIN-TTL", "LONGEST FLOW PKT", and "MIN IP PCK LEN" have a negative impact, decreasing the prediction from the base value. This balance of positive and negative feature contributions ultimately determines the final prediction made by the model for this instance.

To further analyze the local prediction effects of specific features on the model's overall decision or individual predictions, we applied LIME to explain the predictions made by the XGB model (XGB_CON for the CIC-IDS2018 dataset, and XGB_RFE for the CIC-ToNIoT and NF-UNSW-NB15-v2 datasets) for a particular instance in the testing set, as illustrated in Fig. 8 in the main text, along with Figs. B.7 and B.8 in the supplementary file, Appendix B.

In Fig. 8, focusing on the instance classified as a Brute force attack, the model predicted this class with 100% confidence. The middle section of the LIME plot highlights the top 15 features influencing this decision, where blue bars represent features that positively contributed to the prediction, while blue bars signify those with a negative impact. Notably, features such as ''Port'' ($\leq$ 2), ''Total Length of Fwd Packets'' ($>$ −0.01), and ''Flow Duration'' ($\leq$ −0.03) were key in driving the model to classify the instance as a Brute force attack, with contribution weights of 0.07, and 0.029, respectively. On the other hand, features like ''Bwd Packets/s'' ($\leq$ −0.07) and ''Bwd Packet Length Max'' ($>$ 0.10) decreased the likelihood of this classification, with weights of 0.01 and 0.009.

The table of Fig. 8 further details these feature contributions, demonstrating that the total positive contributions outweigh the negative ones, leading to the model's confident prediction of a Brute force attack.

Similarly, the predicted value for Scanning attack is obtained with 100% accuracy, as illustrated in Fig. B.7 of the supplementary file, Appendix B. For the NF-UNSW-NB15-v2 dataset, the predicted value corresponds to an Exploits attack with 97% accuracy, as depicted in Fig. B.8 of the supplementary file, Appendix B.

In sum, this analysis underscores how specific features influence the model's prediction, offering transparency into its decision-making process and providing insights into the most influential factors for detecting Brute force attacks. By understanding the balance of positive and negative feature contributions, domain engineers can gain a clearer understanding of how the model distinguishes between attack and non-attack scenarios.

6.6. The interplay between SHAP, LIME, and GS

To deepen the interpretability analysis, this study examined the interplay between local (LIME, SHAP) and global (Summary SHAP and GS) interpretability techniques. SHAP provided a unified framework for understanding feature contributions at both global and local levels, revealing key features such as "Port" in CIC-IDS2018, "Idle Max" in CIC-ToNIoT, and "MIN-TTL" in NF-UNSW-NB15-v2, which were critical for distinguishing between attack classes. LIME complemented SHAP by offering granular insights into individual predictions, highlighting how specific features influenced borderline cases, such as misclassifications between "DDoS" and normal traffic. GS models, on the other hand, provided a global perspective by approximating the decision rules of the

complex models, confirming overarching patterns such as the consistent importance of "Total Packet Length" across multiple attack types. The interplay of these methods revealed that while SHAP and LIME identified local vulnerabilities and feature-specific behaviors, GS validated these findings at a global scale, ensuring a holistic understanding of model behavior.

6.7. Comparison with state-of-the-art results

In this section, we performed a comparative analysis of the proposed approach against the most recent methods. The evaluation centers on key metrics, including accuracy performance, the number of informative features, the sampling methods employed, and the interpretability techniques utilized, along with the limitations of each method. We have specifically selected studies that used the same datasets (CIC-IDS2018, CIC-ToNIoT, NF-UNSW-NB15-v2) to ensure a fair and meaningful comparison.

The results, as presented in Table A.9 in the supplementary file, Appendix A, demonstrate that our approach outperforms most of the reference methods in both accuracy and interpretability metrics. Our method, which integrates the CTGAN-ENN technique to handle class imbalance and employs FS methods like CON and RFE, provides a significant advantage in addressing two major challenges: class imbalance and feature complexity. Additionally, the use of SHAP, LIME, and GS enables us to enhance the interpretability of black box models such as LGBM and XGB, offering deeper insights into model decisions.

While our study achieves high accuracy and integrates multiple interpretability techniques (SHAP, LIME, and GS), it shares certain limitations with previous works. Although we employed effective data balancing (CTGAN-ENN) and FS techniques (CON, RFE), our evaluation is limited to three datasets, which may impact the generalizability of our findings. These datasets—CIC-IDS2018, CIC-ToN-IoT, and NF-UNSW-NB15-v2—were carefully selected for their diversity in attack types, network environments, and data scales, ensuring a robust benchmarking process.

7. Conclusion and future work

This paper introduced the CTGAN-ENN framework, a novel solution that integrates conditional generative adversarial networks with edited nearest neighbor undersampling and advanced FS techniques. Theoretically, our approach advances the understanding of the trade-off between model accuracy and interpretability in IDSs by demonstrating that it is possible to achieve high predictive performance while still providing clear insights into model decisions through global and local interpretability methods (SHAP, LIME, and Global Surrogate). The study contributes to the field by validating that synthetic data generation combined with robust FS not only mitigates class imbalance and high-dimensionality challenges, but also enhances model transparency, a crucial aspect for trustworthy IDSs.

Practically, the CTGAN-ENN framework offers significant advantages for real-world deployment. It effectively balances imbalanced network traffic, reduces feature space to lower computational costs, and delivers faster training times while maintaining state-of-the-art accuracy across diverse datasets (CIC-IDS2018, CIC-ToNIoT and NF-UNSW-NB15-v2). Furthermore, the integration of interpretability techniques enables cybersecurity professionals to audit model decisions and quickly identify critical features that drive predictions, thus improving operational decision making in dynamic IoT environments.

Nonetheless, our study has some limitations. The evaluation was conducted on only three datasets, which may limit the generalizability of our findings across all network environments. Moreover, the performance of the CTGAN-ENN framework depends on the quality of the underlying classifiers and FS methods. While our interpretability analyses have provided valuable insights, additional techniques could be explored to further enhance model transparency.

Future work intends to focus on collaborating with cybersecurity experts to empirically validate the study's findings and enhance the framework's ability to identify security weaknesses exploited by different attack classes. Other directions include focusing on extending the evaluation to a broader range of datasets to better capture the diversity of real-world network scenarios. Further investigation into the integration of real-time detection for dynamic network monitoring, exploring advanced generative models for data balancing, and incorporating domain-specific knowledge into interpretability techniques to improve transparency in critical sectors like healthcare and finance.

CRediT authorship contribution statement

Houssam Zouhri: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Ali Idri:** Writing – review & editing, Validation, Supervision.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

A supplementary file is available that includes additional figures, tables, and extended results related to the study.

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.future.2025.107882.

Data availability

This article uses public datasets, which are publicly available online at: https://www.unb.ca/cic/datasets/index.html and https://staff.itee.uq.edu.au/marius/NIDS_datasets/. Regarding the code used in our study, it is available upon request. For those interested in obtaining the code, please contact us directly, and we will be happy to provide it.

References

[1] D.D. Bikila, J. Čapek, Machine learning-based attack detection for the internet of things, Future Gener. Comput. Syst. 166 (2025) 107630.
[2] A. Jaiswal, P. Dwivedi, R.K. Dewang, Handling imbalance dataset issue in insider threat detection using machine learning methods, Comput. Electr. Eng. 120 (2024) 109726.
[3] R. Sauber-Cole, T.M. Khoshgoftaar, The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey, J. Big Data 9 (1) (2022) 98.
[4] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[5] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322–1328.
[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[7] H. Liu, Y. Endo, J. Lee, S. Kamijo, SandGAN: Style-mix assisted noise distortion for imbalanced conditional image synthesis, Neurocomputing 559 (2023) 126762.
[8] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, D. Hong, An enhanced AI-based network intrusion detection system using generative adversarial networks, IEEE Internet Things J. 10 (3) (2022) 2330–2345, [Online].

[9] H. Ding, Y. Sun, Z. Wang, N. Huang, Z. Shen, X. Cui, RGAN-EL: A GAN and ensemble learning-based hybrid approach for imbalanced data classification, Inf. Process. Manage. 60 (2) (2023) 103235.

[10] H. Zouhri, A. Idri, Assessing the effectiveness of synthetic data generation for multi-class cyber-attacks detection using generative adversarial networks, in: 2024 World Conference on Complex Systems, WCCS, Mohammedia, Morocco, 2024, pp. 1–6.

[11] J. Zhang, X. Huang, Y. Liu, Y. Han, Z. Xiang, GAN-based medical image small region forgery detection via a two-stage cascade framework, PLoS One 19 (1) (2024) e0290303.

[12] E. Yauri-Lozano, M. Castillo-Cara, L. Orozco-Barbosa, R. García-Castro, Generative adversarial networks for text-to-face synthesis & generation: A quantitative–qualitative analysis of natural language processing encoders for Spanish, Inf. Process. Manage. 61 (3) (2024) 103667.

[13] Ruixiao Liu, Jing Shi, Xingyu Chen, Cuiying Lu, Network anomaly detection and security defense technology based on machine learning: A review, Comput. Electr. Eng. 119 (2024) 109581.

[14] C. Mendes, T. Nogueira Rios, Explainable artificial intelligence and cybersecurity: A systematic literature review, 2023, arXiv preprint arXiv:2303.01259.

[15] H. Zouhri, A. Idri, A comparative assessment of wrappers and filters for detecting cyber intrusions, in: World Conference on Information Systems and Technologies, 2024, pp. 118–127.

[16] A. Thakkar, R. Lohiya, A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions, Artif. Intell. Rev. 55 (1) (2022) 453–563.

[17] Mohammad Fatahi, Danial Sadrian Zadeh, Behzad Moshiri, Otman Basir, Entropy-based genetic feature engineering and multi-classifier fusion for anomaly detection in vehicle controller area networks, Future Gener. Comput. Syst. (2025) 107779.

[18] H. Zouhri, A. Idri, A. Ratnani, Evaluating the impact of filter-based feature selection in intrusion detection systems, Int. J. Inf. Secur. 23 (2) (2024) 759–785.

[19] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, A.Y. Zomaya, An explainable deep learning-enabled intrusion detection framework in IoT networks, Inform. Sci. 639 (2023) 119000.

[20] Hongwei Ding, Leiyang Chen, Liang Dong, Zhongwang Fu, Xiaohui Cui, Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection, Future Gener. Comput. Syst. 131 (2022) 240–254.

[21] Hoang V. Vo, Hanh P. Du, Hoa N Nguyen, Apelid: Enhancing real-time intrusion detection with augmented WGAN and parallel ensemble learning, Comput. Secur. 136 (2024) 103567.

[22] Omar Habibi, Mohammed Chemmakha, Mohamed Lazaar, Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT botnet attacks detection, Eng. Appl. Artif. Intell. 118 (2023) 105669.

[23] Datasets | Research | Canadian Institute for Cybersecurity | UNB.

[24] Abdul Majeed, Seong Oun Hwang, CTGAN-MOS: Conditional generative adversarial network based minority-class-augmented oversampling scheme for imbalanced problems, IEEE Access (2023).

[25] F. Omer Albasheer, R. Ramesh Haibatti, M. Agarwal, S. Yeob Nam, A novel IDS based on Jaya Optimizer and smote-ENN for cyberattacks detection, IEEE Access 12 (2024) 101506–101527.

[26] Chaymae Miloudi, C. Miloudi, L. Cheikhi, A. Idri, A. Abran, Bug resolution prediction for open-source software using ensembles of instance selection algorithms, in: 2023 9th International Conference on Control, Decision and Information Technologies, CoDIT, 2023, pp. 695–700.

[27] I.N.M. Adiputra, P. Wanchai, CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction, J. Big Data 11 (2024) 121.

[28] Mohammad Arafah, Iain Phillips, Asma Adnane, Wael Hadi, Mohammad Alauthman, Abedal-Kareem Al-Banna, Anomaly-based network intrusion detection using denoising autoencoder and Wasserstein GAN synthetic attacks, Appl. Soft Comput. 168 (2025).

[29] Xinxing Zhao, Kar Wai Fok, Vrizlynn L.L. Thing, Enhancing network intrusion detection performance using generative adversarial networks, 2024, arXiv preprint arXiv:2404.07464.

[30] Hajar Hakkoum, Ali Idri, Ibtissam Abnane, Global and local interpretability techniques of supervised machine learning black box models for numerical medical data, Eng. Appl. Artif. Intell. 131 (2024) 107829.

[31] Mengmeng Zhan, Xiaoshuang Shi, Fangqi Liu, Rongyao Hu, IGCNN-FC: Boosting interpretability and generalization of convolutional neural networks for few chest X-rays analysis, Inf. Process. Manage. 60 (3) (2023) 103258.

[32] Boxu Guan, Xinhua Zhu, Shangbo Yuan, A T5-based interpretable reading comprehension model with more accurate evidence training, Inf. Process. Manage. 61 (2) (2024) 103584.

[33] Zijiao Zhang, Chong Wu, Shiyou Qu, Xiaofang Chen, An explainable artificial intelligence approach for financial distress prediction, Inf. Process. Manage. 59 (4) (2022) 102988.

[34] Houssam Zouhri, Ali Idri, Hajar Hakkoum, Assessing the effectiveness of dimensionality reduction on the interpretability of opaque machine learning-based attack detection systems, Comput. Electr. Eng. 120 (2024) 109627.

[35] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, Modeling tabular data using conditional GAN, Adv. Neural Inf. Process. Syst. 32 (2019).

[36] Dennis L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybern. (3) (1972) 408–421.

[37] Christoph Molnar, Interpretable machine learning, 2020, Lulu.com.

[38] Hajar Hakkoum, Ali Idri, Ibtissam Abnane, Assessing and comparing interpretability techniques for artificial neural networks breast cancer classification, Comput. Methods Biomech. Biomed. Eng.: Imaging & Vis. 9 (6) (2021) 587–599.

[39] Carel Schwartzenberg, Tom van Engers, Yuan Li, The fidelity of global surrogates in interpretable machine learning, BNAIC/ BeneLearn 2020 (2020) 269.

[40] Kjell Hausken, Matthias Mohr, The value of a player in n-person games, Soc. Choice Welf. 18 (2001) 465–483.

[41] Scott Lundberg, A unified approach to interpreting model predictions, 2017, arXiv preprint arXiv:1705.07874.

[42] Marco Tulio Ribeiro, et al., 'Why should I trust you?' Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[43] A. Heidari, H. Shishehlou, et al., A reliable method for data aggregation on the industrial internet of things using a hybrid optimization algorithm and density correlation degree, Clust. Comput. 27 (2024) 7521–7539.

[44] Sajedeh Norozpour, Mehdi Darbandi, Proposing new method for clustering and optimizing energy consumption in WSN, Talent. Dev. Excel. 12 (2020).

[45] Karlo Abnoosian, Rahman Farnoosh, Mohammad Hassan Behzadi, A pipeline-based framework for early prediction of diabetes, J. Heal. Biomed. Inform. 10 (2) (2023) 125–140.

[46] Rahman Farnoosh, Karlo Abnoosian, A robust innovative pipeline-based machine learning framework for predicting COVID-19 in Mexican patients, Int. J. Syst. Assur. Eng. Manag. 15 (7) (2024) 3466–3484.

[47] M. Darbandi, et al., The applications of machine learning techniques in medical data processing based on distributed computing and the internet of things, Comput. Methods Programs Biomed. 241 (2023).

[48] M. Darbandi, S. Haghgoo, M. Hajiali, A. Khabir, Prediction and estimation of next demands of cloud users based on their comments in CRM and previous usages, in: 2018 International Conference on Communication, Computing and Internet of Things, IC3IoT, Chennai, India, 2018, pp. 81–86.

[49] Arash Heidari, Nima Jafari Navimipour, Mehmet Unal, A secure intrusion detection platform using blockchain and radial basis function neural networks for internet of drones, IEEE Internet Things J. 10 (10) (2023) 8445–8454.

[50] S. Noor, A. Naseem, H.H. Awan, et al., Deep-m5U: a deep learning-based approach for RNA 5-methyluridine modification prediction using optimized feature integration, BMC Bioinformatics 25 (2024) 360.

[51] S. Khan, S.A. AlQahtani, S. Noor, et al., PSSM-Sumo: deep learning based intelligent model for prediction of sumoylation sites using discriminative features, BMC Bioinformatics 25 (2024) 284.

[52] Salman Khan, Mukhtaj Khan, Nadeem Iqbal, Mohd Amiruddin Abd Rahman, M. Khalis Abdul Karim, Deep-PiRNA: Bi-layered prediction model for PIWI-interacting RNA using discriminative features, Comput. Mater. Contin 72 (2) (2022) 2243–2258.

[53] Adnan Helmi Azizan, et al., A machine learning approach for improving the performance of network intrusion detection systems, Ann. Emerg. Technol. Comput. (2021).

[54] Ali Saeed Alfoudi, Mohammad R. Aziz, Zaid Abdi Alkareem Alyasseri, Ali Hakem Alsaeedi, Riyadh Rahef Nuiaa, Mazin Abed Mohammed, Karrar Hameed Abdulkareem, Mustafa Musa Jaber, Hyper clustering model for dynamic network intrusion detection, IET Commun. (2022).

[55] Osamah Ahmed, Enhancing intrusion detection in wireless sensor networks through machine learning techniques and context awareness integration, Int. J. Math. Stat. Comput. Sci. 2 (2024) 244–258.

[56] Vikash Kumar, Ditipriya Sinha, Synthetic attack data generation model applying generative adversarial network for intrusion detection, Comput. Secur. 125 (2023) 103054.

[57] Saifur Rahman, Shantanu Pal, Shubh Mittal, Tisha Chawla, Chandan Karmakar, SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security, Internet Things 26 (2024) 101212.

[58] Kazuma Kobayashi, Syed Bahauddin Alam, Explainable, interpretable, and trustworthy AI for an intelligent digital twin: A case study on remaining useful life, Eng. Appl. Artif. Intell. 129 (2024) 107620.

[59] Toshitaka Hayashi, Dalibor Cimr, Hamido Fujita, Richard Cimler, Interpretable synthetic signals for explainable one-class time-series classification, Eng. Appl. Artif. Intell. 131 (2024) 107716.

[60] Alejandro Barredo Arrieta, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[61] S. Hariharan, R.R. Rejimol Robinson, R.R. Prasad, et al., XAI for intrusion detection system: comparing explanations based on global and local scope, J. Comput. Virol. Hacking Tech. 19 (2) (2023) 217–239.

[62] Youcef Djenouri, et al., Interpretable intrusion detection for next generation of internet of things, Comput. Commun. 203 (2023) 192–198.

[63] Sivanandam Sivamohan, S.S. Sridhar, Sivamohan Krishnaveni, TEA-EKHO-IDS: An intrusion detection system for industrial CPS with trustworthy explainable AI and enhanced krill herd optimization, Peer- To- Peer Netw. Appl. 16 (4) (2023) 1993–2021.
[64] Ebuka Chinaechetam Nkoro, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, Dong-Seong Kim, Detecting cyberthreats in metaverse learning platforms using an explainable DNN, Internet Things 25 (2024) 101046.
[65] Bhawana Sharma, Lokesh Sharma, Chhagan Lal, Satyabrata Roy, Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach, Expert Syst. Appl. 238 (2024) 121751.
[66] IDS 2018 | datasets | research | Canadian institute for cybersecurity | UNB, 2018.
[67] Mohanad Sarhan, et al., Netflow datasets for machine learning-based network intrusion detection systems, in: Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10, 2021, Springer International Publishing.
[68] Nour Moustafa, Ton_IoT datasets. IEEE dataport, 2019.
[69] Nour Moustafa, Jill Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, 2015, pp. 1–6.
[70] Andrew Jhon Scott, Martin Knott, A cluster analysis method for grouping means in the analysis of variance, Biometrics (1974) 507–512.
[71] I.T. Jolliffe, Cluster analysis as a multiple comparison method, in: Applied Statistics: Proceedings of Conference at Dalhousie University, 1975, pp. 159–168.
[72] T. Caliński, L.C.A. Corsten, Clustering means in ANOVA by simultaneous testing, Biometrics (1985) 39–48.
[73] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[74] Jie Liu, Yubo Xu, T-Friedman test: a new statistical test for multiple comparison with an adjustable conservativeness measure, Int. J. Comput. Intell. Syst. 15 (1) (2022) 29.