Unsupervised DDoS Detection in High-Speed Networks: An Evaluation Using Real Transit Provider Data

Murilo A. Chianfa[1], **Rodrigo S. Miani**[2], **Bruno B. Zarpelão**[1]

[1] Computer Science Department – State University of Londrina (UEL)
P.O. Box 10.011 – ZIP Code 86057-970 – Londrina – PR – Brazil

[2]School of Computer Science (FACOM) – Federal University of Uberlândia (UFU)
P.O. Box 593 – ZIP Code 38400-902 – Uberlândia – MG – Brazil

murilo.chianfa@uel.br, miani@ufu.br, brunozarpelao@uel.br

Abstract. *Distributed denial-of-service (DDoS) attack detection has been widely studied in the past decade by academia. Despite progress having been made, recent surveys show that detection in environments such as Internet Transit Providers (ITP) remains challenging due to high-speed constraints. This study evaluates four anomaly detection algorithms, namely Autoencoder, Isolation Forest, Local Outlier Factor, and One Class Support Vector Machine, using three datasets collected from operational ITPs during confirmed DDoS attacks. The evaluation considers four temporal aggregation windows and three feature selection configurations, with the objective of analyzing the predictive capacity of the algorithms under different temporal and feature selection settings. The results show that the Autoencoder detection achieved the best results when using the most aggressive feature selection configuration and the shortest temporal aggregation windows.*

1. Introduction

Distributed denial-of-service (DDoS) attacks remain a major threat to large-scale networks. Beyond known well-established campaigns, attackers have increasingly attempted to congest high-capacity links between critical peering points [Cheng et al. 2024]. In this setting, Internet Service Providers (ISPs), especially in this work, Internet Transit Providers (ITPs), are both potential victims and key control points, since they operate in the long-haul of main links and routing policies that steer traffic across the Internet. Several detection mechanisms have recently been proposed, but fewer have focused on high-speed network environments[1].

Addressing this, [Latif et al. 2023] and more recently [Koumar et al. 2025c] proposed network outlier detectors for these environments, but none specifically focused on DDoS attacks based on Link Flood Attacks (LFA), which has a different defense mechanism than other network attacks. A natural line of defense in this scenario is to exploit the flexibility of the inter-domain routing. Rerouted-based DDoS defenses leverage Border Gateway Protocol (BGP) mechanisms such as traffic engineering, remote triggered blackholing (RTBH), and detours through scrubbing centers to alleviate congestion along the targeted links [Tran et al. 2019].

[1]Defined as networks exceeding 100 Gbps of link capacity according to [Haseeb ur Rehman et al. 2023].

According to the National Institute of Standards and Technology (NIST) guidelines on resilient inter-domain traffic exchange [Sriram and Montgomery 2019], the mitigation of DDoS for ISPs depends on cooperative mitigation between the upstream transit provider and downstream networks, implicitly assuming the availability of reliable DDoS detection components at or near ITP boundaries.

A growing body of work investigates DDoS detection using statistical and deep learning models [Koumar et al. 2025b, Da Silva Ruffo et al. 2024]. However, such models can be complex to deploy on the ITP scale, typically focusing on detection outcomes without considering the feasibility of automated mitigation actions required to handle large-scale and transit-link DDoS attacks in high-speed networks, such as traffic redirection to scrubbing centers or cloud-based mitigation [Smith and Schuchard 2018].

Moreover, many DDoS detection systems assume fully supervised settings with abundant labeled attack samples, whereas operational networks rarely possess labeled traffic during model development [Meyer et al. 2025]. Therefore, it is important to evaluate DDoS detection models using real operational traffic from high-speed networks, ensuring that predictive performance is measured under realistic operational constraints. In this context, closer cooperation between academia and industry is encouraged to allow realistic evaluations [Hiesgen et al. 2024].

This work focuses on unsupervised flow-based anomaly detection for DDoS detection on the ITP side. Three real-world datasets from operational ITPs were collected during DDoS incidents with confirmed downstream outages, resulting in highly imbalanced series in which DDoS accounts for only a small fraction of the traffic. Based on this setting, the objective of this study is to evaluate and compare one-class anomaly detection models for DDoS detection under realistic operational constraints, where labeled attack traffic is typically unavailable during model development. Four unsupervised models from distinct methodological families were evaluated, namely Autoencoder (AE), Isolation Forest (IF), One-Class Support Vector Machine (OCSVM), and Local Outlier Factor (LOF), with the goal of comparing their predictive performance and practical suitability for operational deployment.

Detection is performed over time-windowed flow aggregates, allowing us to analyze the relationship between window length and detection effectiveness in high-speed networks, where delayed responses may lead to rapid service degradation. In addition, the effect of feature space reduction is investigated by applying a Pearson Correlation Coefficient (PCC) filter to an initial set of 149 features, in order to quantify the trade-off between detection precision and efficiency.

The remainder of this paper is organized as follows. § 2 reviews related work on DDoS detection, with an emphasis on unsupervised and one-class approaches in high-speed networks. § 3 describe the datasets, the feature extraction process, and the methodological framework adopted in this study. § 4 presents and discusses the experimental results. Finally, § 5 concludes the paper, discusses the limitations of the proposed approach, and outlines the directions for future work.

2. Related Work

Focusing on high-speed networks, Koumar et al. [Koumar et al. 2025a] created real-world datasets from an operational ISP located in the Czech Republic and later

presented a method to detect network outliers using statistical time series algorithms, the Mean Outlier Detector and Seasonal Auto Regressive Integrated Moving Average [Koumar et al. 2025b]. Although working based on recently collected real traces, the study highlights an average of 375 outliers detected hourly over a period of one month, which can be a problem in the ITP context according to [Smith and Schuchard 2018], where in general these attacks can be considered a routing problem.

Another branch aiming to enhance the detection is the use of machine learning and deep learning techniques. Venceslau et al. [Venceslau et al. 2025] proposed a hybrid correlation-based stacking approach for one-class detection, where three models, LOC, IF, and OCSVM, were used as base detectors. Their anomaly scores were combined and passed to a Random Forest, which is trained to distinguish between normal and anomalous behavior based on the input data from the input received. The method is evaluated on the public URG'16 and CIC-IDS2017 datasets using ROC curves, AUC and F1-score, and shows gains in terms of false-positive reduction when compared with individual detectors. However, the stacking architecture introduces a supervised stage that depends on labeled attack traffic for training the final classifier and is tuned for evaluation on benchmark datasets only without considering temporal aggregation or feature redundancy control.

Paolini et al. [Paolini et al. 2025] conducted a comprehensive survey and an experimental study of one-class anomaly detection methods among AE, LOC, IF and OCSVM algorithms in several benchmark datasets. In addition to detection accuracy, runtime and resource consumption in constrained environments was measured, highlighting AE as an attractive candidate. These findings support the inclusion of AE alongside one-class models in the present study.

Salahuddin et al. [Salahuddin et al. 2022] and more recently Goldschmidt and Kučera in [Goldschmidt and Kučera 2024] studied the impact of several windowing aggregation techniques for feature extraction, as well as using variations of AE in their pipelines, concluding the benefits over a flow-by-flow or packet-per-packet feature extraction approach, especially when working with high-speed networks where the need for real-time detection is a must.

Table 1 contrasts the proposed approach with related work across key technical dimensions. Although prior studies cover individual aspects of these dimensions, none of them jointly address all of them within a single work.

Table 1. Comparison between related works and the proposed approach

Work	real-world traffic	high-speed networks	one-class detection	temporal aggregation	feature selection
[Koumar et al. 2025b]	✓	✓	–	✓	–
[Venceslau et al. 2025]	–	–	✓	–	–
[Paolini et al. 2025]	–	–	✓	–	✓
[Salahuddin et al. 2022]	–	✓	✓	✓	–
[Goldschmidt and Kučera 2024]	–	✓	✓	✓	–
This work	✓	✓	✓	✓	✓

3. Materials and Methods

This section describes the overall architecture adopted in this work, as shown in Figure 1. In total, 144 experimental validation scenarios were carried out, covering all combinations over (i) three datasets collected using sampled flow records from ITP border routers (DS1, DS2, and DS3) presented in § 3.1, (ii) four temporal aggregation windows (1 s, 10 s, 60 s, and 300 s), described in § 3.2, (iii) three PCC values for feature selection (0.50, 0.70, and 0.90) denoted as θ in § 3.2.1, and (iv) four anomaly detection models (AE, IF, OCSVM and LOF), described in § 3.3, then thresholding in § 3.4, and evaluation in § 3.5.

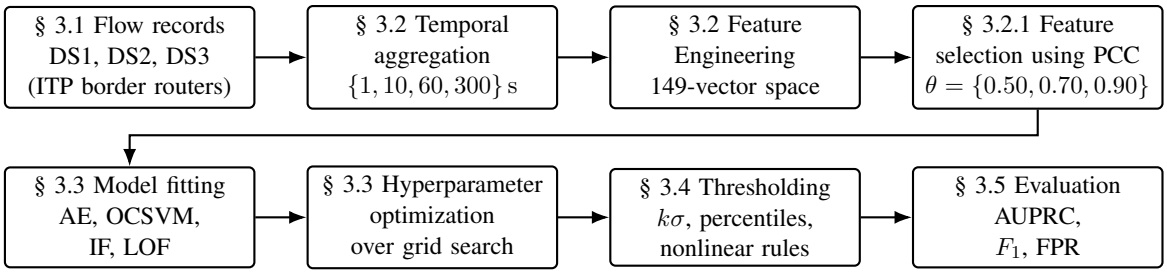


Figure 1. Overall experimental design adopted in this work.

3.1. Data Collection

Three real-world datasets were built from operational ITPs during confirmed DDoS attack incidents. The first dataset, denoted `itp-downstream-http-flood` (DS1), contains traffic corresponding to an application-layer HTTP flood attack targeting downstream customer networks, hitting the speed limit of 10 Gbps of the circuit. The second dataset, `itp-multivector-udp-100gbps-peak` (DS2), captures a large-scale multi-vector UDP flood attack that reached a peak throughput of about 100 Gbps coming from a carpet bombing attack[2]. The third dataset, `itp-synack-customer-outage` (DS3), captures a SYN–ACK reflection attack that caused service outage after 2 hours of continuous attack.

Together, these datasets cover three different types of DDoS attacks, each impacting different dimensions of network flows (e.g. volumetric behavior and protocol dynamics), which motivates their joint use to increase the robustness and generality of the proposed analysis.

All traffic was collected using the NetFlow version 9 protocol [Claise 2004], configured to export from production edge routers within the ITP environment, at the border between customer aggregation and upstream transit peers. Exporters were configured with fixed active and inactive timeouts T and a packet sampling ratio of $1{:}k$, being T and k set evenly on all exporters according to the operator's policies. The statistical summary of each dataset can be seen in Table 2.

The flows were filtered using a destination-specific criteria to isolate flows relevant to each attack scenario. AS-based filters such as `dst as 65550` were employed to extract traffic directed to the specific AS network that were known to be under attack. The `nfdump` [Haag 2025] utility suite was used to collect the data.

[2]According to Bhardwaj et al. [Bhardwaj et al. 2021], a carpet bombing attack targets entire subnets by distributing traffic across a large number of destination IP addresses.

Table 2. Statistics of the extracted datasets.

Dataset	Duration	#Flows	Size	Sampling k	$T_{\text{active}}/T_{\text{inactive}}$
DS1	3d 14h 45m	9.43×10^6	1024MB	1000	1800s/15s
DS2	8d 2h 0m	1.95×10^8	20.76GB	1000	60s/15s
DS3	7d 18h 20m	2.25×10^6	230MB	4000	60s/15s

After filtering, the datasets were chronologically partitioned into three disjoint subsets. The training partition consisted of multiple days of traffic that exhibited normal behavior prior to the onset of the attack. The validation partition comprised one to two days of recent baseline traffic immediately preceding the attack periods. The test partition encompassed two to five days of traffic that included both normal conditions and confirmed attack intervals as recommended by [Ring et al. 2019].

The attack windows were manually annotated using precise start and end timestamps obtained from operator incident reports, providing ground-truth labels for the evaluation of detection performance. The share between attack and non-attack can be seen in Table 3.

Table 3. Share of attack periods on test set (1s windows).

	DS1	DS2	DS3
Total time in seconds	139,800	353,100	239,100
Number of DDoS seconds	319	2,734	6,591
Total of the anomalous share	0.23%	0.77%	2.76%

3.2. Feature Engineering

All flows were partitioned into fixed-length time tumbling windows of duration Δt of 1, 10, 60, and 300 seconds [Goldschmidt and Kučera 2024]. For each window, let N denote the number of flows observed, and for each flow $i \in \{1, \ldots, N\}$ let p_i, b_i and d_i denote, respectively, the number of packets, the number of bytes and the flow duration. For any generic quantity $x_i \in \{p_i, b_i, d_i\}$, the total volume in the window was defined as

$$\forall x \in \{p, b, d\}, \quad T_x = \sum_{i=1}^{N} x_i. \tag{1}$$

Given the per-flow sample $\{x_i\}_{i=1}^{N}$ for a generic variable $x \in \{p, b, d\}$, standard first- and second-order empirical moments were then defined in order to characterize both the central tendency and dispersion of the flows in each window, namely the mean, variance, standard deviation, and coefficient of variation:

$$\mu_X = \frac{1}{N} \sum_{i=1}^{N} x_i, \qquad \sigma_X^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_X)^2, \tag{2}$$

$$\sigma_X = \sqrt{\sigma_X^2}, \qquad \mathrm{CV}_X = \frac{\sigma_X}{\mu_X} \quad (\mu_X > 0), \tag{3}$$

yielding, for each window, features such as *packets_mean*, *bytes_variance*, *duration_std* and *duration_cv*. To capture higher-order shape information, the third and fourth standardised central moments were also computed, namely skewness and kurtosis:

$$\gamma_{1,X} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_X)^3}{\sigma_X^3}, \qquad \gamma_{2,X} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_X)^4}{\sigma_X^4}, \tag{4}$$

where $\gamma_{1,X}$ measures the distribution asymmetry and $\gamma_{2,X}$ quantifies its tail heaviness, resulting in features such as *bytes_skewness* and *packets_kurtosis*.

Information-theoretic features were computed using Shannon entropy to characterize the randomness and diversity of traffic distributions. For a discrete random variable X with support $\mathcal{X}$ and probabilities $\{p_x\}_{x\in\mathcal{X}}$, the entropy is defined as

$$H(X) = -\sum_{x\in\mathcal{X}} p_x \, \log_2 p_x. \tag{5}$$

Entropy features were derived for categorical traffic attributes, such as source and destination ports, IP addresses, AS numbers, and geographic country codes, providing a compact summary of the diversity of endpoints within each aggregation window.

In addition to entropy, the feature set includes measures of diversity, inequality, concentration, temporal dynamics, and protocol behavior, such as Simpson diversity, Gini coefficient, traffic concentration metrics with top-K ratios and Herfindahl–Hirschman Index, and heavy-tail indicators based on Hill estimators.

Further feature groups include indicators of amplification attacks, port-usage distributions across well-known, registered, and ephemeral ranges, asymmetry measures comparing source and destination entropies, inter-arrival time statistics, and IP behavioral patterns. Together, these categories yield a feature space of 149 dimensions. [3]

3.2.1. Feature Selection

Given the large number of derived features, a correlation-based filter strategy was adopted to reduce redundancy in the feature space before model training. Since the anomaly detection setting considered in this work is pure unsupervised (i.e. labels are only available just for evaluation), a model-agnostic, filter-type method is preferable to wrapper or embedded approaches that require labeled data or repeated model fitting.

Inspired by [Kamaldeep et al. 2023], the selection of features was based on the Pearson's correlation coefficient (PCC), calculated between all pairs of features using only the training partition. For two features X and Y with paired observations $\{(x_i, y_i)\}_{i=1}^{N}$, the PCC is defined as

$$\rho_{X,Y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{6}$$

where $\bar{x}$ and $\bar{y}$ denote the sample means of X and Y, respectively. We operate on the absolute value $|\rho_{X,Y}|$ to capture both positive and negative linear dependencies, as both forms of strong correlation indicate redundancy.

[3] All features are available in the github repository for further analysis and replication of results `https://github.com/MuriloChianfa/isp-ddos-auto-detector`.

For each dataset and temporal aggregation window, a complete correlation matrix was computed on the training data. Then a simple but reproducible pruning procedure was applied. First, a fixed ordering of features is established (e.g., by their original index). Next, the upper triangular part of the absolute correlation matrix was scanned, and whenever a pair of features (f_i, f_j) satisfies

$$|\rho_{f_i,f_j}| > \theta, \tag{7}$$

the latter feature in the pair (according to the fixed ordering) is marked for removal. By iterating through all pairs in this manner, each feature is removed at most once, and the resulting subset contains no pair of features with absolute correlation that exceeds the chosen cutoff value θ. This procedure is purely deterministic given the ordering of features and the value of θ, ensuring that the experiments are exactly reproducible.

Once the feature subset has been determined on the training partition, the same selection mask is applied to the validation and test partitions, so that all models are trained and evaluated on exactly the same reduced feature space for a given configuration. To study the impact of different levels of redundancy control, three correlation cutoff values were systematically evaluated: $\theta \in \{0.50, 0.70, 0.90\}$, which generated different vectors of features for each dataset.

3.3. Model Architectures

Four anomaly-detection algorithms were implemented and evaluated, covering both classical machine learning and deep learning approaches to unsupervised outlier detection. All models were trained exclusively on traffic labeled as normal from the training partition, reflecting realistic operational constraints in ITP environments where labeled attack data are absent during model development.

The autoencoder model is a deep neural network designed to learn compact representations of normal traffic and to detect anomalies based on reconstruction error. An architecture of mirrored encoder–decoder was chosen, where the encoder consists of $L \in \{2, 3, 4\}$ hidden layers, and the number of neurons in each hidden layer $\mathbf{h} = (h_1, \ldots, h_L)$ was defined as fractions of the input dimension d, as shown in Table 4. All input features were scaled to $[0, 1]$ by Min-Max normalization. The training was carried out using the Adam optimizer with initial learning rate η, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and numerical stability parameter $\varepsilon = 10^{-7}$. Training continues for up to 100 epochs with mini-batches of size B. Early stopping with patience of $P = 20$ and a dropout rate of $p_{\text{drop}} = 0.1$ were applied to reduce overfitting.

For IF, an ensemble of isolation trees T was trained using subsamples of size m (or `auto`) and an optional subsample fraction of features $f \in (0, 1)$. The contamination parameter c was tuned to determine the proportion of samples labeled as anomalies for high-imbalanced datasets. For OCSVM, a ν-OCSVM was adopted with $\nu \in (0, 1)$ controlling the boundary tightness and kernel choices in $\{\texttt{rbf}, \texttt{sigmoid}, \texttt{poly}\}$ with parameters γ and degree q (poly only). For LOF, density deviations were calculated using the closest neighbors k, using different distance metrics (including Minkowski order p), neighbor-search algorithm A, a leaf size ℓ treated as tunable parameters and a contamination c. The input features for these were scaled using the Standard Scaler.

Hyperparameters were selected via grid search in the validation partition, using the best negative mean for their respective anomaly scores generated on the validation data after training, as the primary selection criterion. The search spaces explored are summarized in Table 4.

Table 4. Hyperparameters selection based on grid search.

Model	Search space	Model	Search space
OCSVM	$\nu \in [0.01, 0.30]$ kernel $\in \{\texttt{rbf}, \texttt{sigmoid}, \texttt{poly}\}$ $\gamma \in \{\texttt{scale}, \texttt{auto}\} \cup [10^{-4}, 10^{-1}]$ $q \in \{2, 3, 4, 5\}$ (poly only)	IF	$T \in [50, 500]$ $m \in [64, 512] \cup \{\texttt{auto}\}$ $c \in [0.01, 0.20]$ $f \in [0.5, 1.0]$
AE	$L \in \{2, 3, 4\}$ $h_1 \in [0.60d,\ 0.95d]$ $h_{L+1} \in [0.50h_L,\ 0.80h_L]$ $B \in \{32, 64, 128\}$ $\eta \in \{0.001, 0.005, 0.01, 0.05\}$ $p_{\text{drop}} = 0.1$ $P = 20$	LOF	$k \in [20, 200]$ $c \in [0.01, 0.30]$ $A \in \{\texttt{ball\_tree}, \texttt{kd\_tree}\}$ $\ell \in [10, 200]$ metric $\in \{\texttt{minkowski}, \texttt{euclidean}, \texttt{manhattan}, \texttt{chebyshev}\}$ $p \in \{1, 2, 3\}$ (minkowski only)

3.4. Thresholds for Anomaly Detection

After training, each model produces a raw anomaly score $s(x)$ for every temporal window x on the test partition. A unified anomaly score $a(x)$ is defined as a transformation of the raw score $s(x)$, i.e., $a(x) = f(s(x))$, where (f) is chosen such that higher values consistently indicate a higher likelihood of anomalies. Whenever an algorithm produces a score for which lower values indicate stronger anomalies, the score is inverted to match this convention [Venceslau et al. 2025]. Thus, a window is flagged as anomalous when

$$a(x) > \tau. \tag{8}$$

For AE, $a(x)$ is computed as the reconstruction error (MSE). For IF, $a(x)$ is obtained from the model's isolation/normality score and is inverted when needed so that larger values correspond to easier isolation (more anomalous). For OCSVM, $a(x)$ is obtained from the decision function (signed distance to the separating boundary) and is inverted when necessary so that higher values indicate stronger outliers. For LOF, $a(x)$ is obtained from the local density deviation score, and the same inversion is applied whenever the implementation returns the opposite convention.

The thresholds τ were computed from the anomaly scores on the validation set. The set of validation anomaly scores is indicated by S, and its mean and standard deviation are indicated by μ and σ, respectively. The following strategies were evaluated:

$$\tau = \mu + k\sigma, \quad k \in \{1, \ldots, 80\}, \tag{9}$$

$$\tau_p = \text{perc}(p, S), \quad p \in \{0.95, 0.99, 0.995, 0.999\}, \tag{10}$$

$$\tau_{\text{exp}} = \exp(\mu + 10\sigma), \tag{11}$$

$$\tau_{\text{sig}} = \frac{1}{1 + \exp\big(-(\mu + 3\sigma)\big)}. \tag{12}$$

Each model and dataset specific thresholds were empirically selected and encoded in configuration files for reproducible evaluation, available on the repository.

3.5. Evaluation Metrics

Each model was evaluated using standard classification metrics computed from confusion matrices obtained by comparing the prediction of the model with the ground-truth labels. For each temporal window, the predictions are categorized as true positive (TP, correctly detected DDoS), true negative (TN, correctly classified normal traffic), false positive (FP, misclassified normal traffic as DDoS), or false negative (FN, not detected DDoS). From these values, the following metrics were derived:

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

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

The F_1-score, which is the harmonic mean of precision and recall, is defined as

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{15}$$

The true-positive rate (TPR) is the same as Recall, and the false-positive rate (FPR) is given by

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{16}$$

Finally, the Precision Recall curve (AUPRC) was also used, defined as

$$\text{AUPRC} = \int_0^1 P(R)\, dR. \tag{17}$$

The quantity is computed numerically from the empirical curves. In this work, the F_1-score was adopted as the primary metric, with AUPRC used as a complementary measure. AUPRC was preferred over AUROC due to its more informative behavior under label imbalance, as discussed by [McDermott et al. 2025, Komadina et al. 2024].

4. Experimental Results

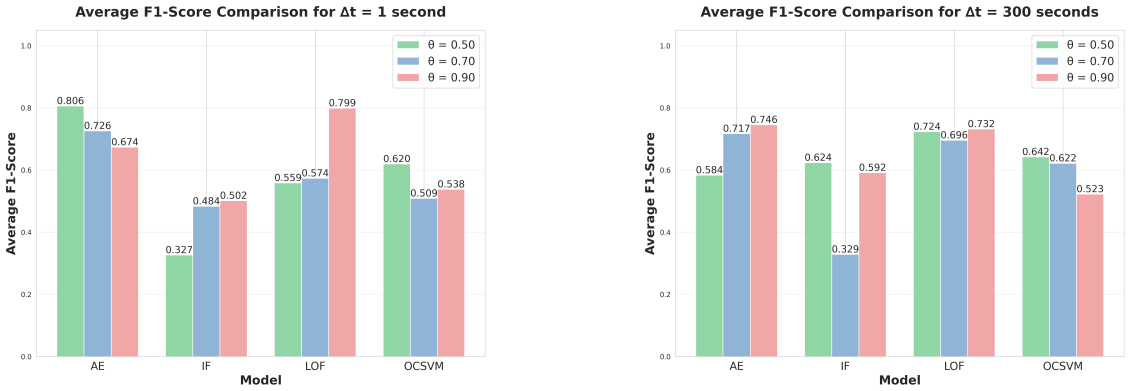
All experiments were carried out on a dual Intel Xeon E5-2683 v4 CPU running at 2.10 GHz, 128 GB of RAM and a NVIDIA GeForce GTX 1050 Ti GPU. The operating system was Debian 12, with all models and scripts implemented in Python 3.12.2. The AE model was implemented using TensorFlow 2.16.1, and all other models were implemented using Scikit-learn 1.7.1. For feature extraction and selection, NumPy 1.26.4, Pandas 2.3.1, and SciPy 1.16.0 were used.

Firstly, the average F_1-score was computed across the datasets for each algorithm and each aggregation window Δt, considering the cutoff values $\theta = 0.50$ and $\theta = 0.90$. The cutoff $\theta = 0.70$ was omitted to emphasize the comparison between the most aggressive and conservative approach, as was the case for $\Delta t = 60\,\text{s}$ and $\Delta t = 10\,\text{s}$.

Starting with $\Delta t = 300\,\text{s}$, the results show that higher cutoff values tend to benefit most algorithms: AE achieved an average F_1-score of 0.746 with $\theta = 0.90$, compared to

0.584 with $\theta = 0.50$, while LOF presented stable performance with average F_1-scores of 0.724 and 0.732, respectively.

For the shortest aggregation window, $\Delta t = 1\,\text{s}$, a different behavior can be observed. Under $\theta = 0.50$, AE achieved the highest average F_1-score among all configurations (0.806), followed by OCSVM (0.620) and LOF (0.559). When increasing the cutoff to $\theta = 0.90$, LOF benefited significantly, improving its average F_1-score to 0.799. Figure 2 summarizes the mean F_1-score obtained when comparing the results.



(a) F_1-score for $\Delta t = 1\,\text{s}$. (b) F_1-score for $\Delta t = 300\,\text{s}$.

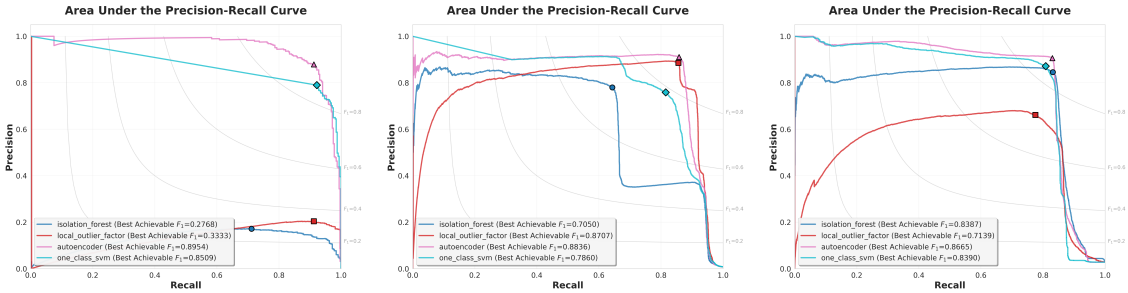
Figure 2. F_1-score comparison for $\Delta t = 1\,\text{s}$ and $\Delta t = 300\,\text{s}$ at $\theta \in \{0.50, 0.70, 0.90\}$

Based on the results discussed previously, the AE was identified as the best-performing model for the shortest aggregation window when using $\Delta t = 1\,\text{s}$ and $\theta = 0.50$, achieving the highest average F_1-score among all algorithms evaluated. Table 5 presents a subset of the numerical results for this configuration, reporting Precision, Recall, F_1-score, and FPR for all models in the considered datasets.

Table 5. Detection performance for $\Delta t = 1\,\text{s}$ and $\theta = 0.50$.

Dataset	Model	Precision	Recall	F_1	FPR
DS1	AE	**0.7772**	0.9404	**0.8511**	**0.0006**
	IF	0.1440	0.9060	0.2485	0.0123
	LOF	0.1702	**0.9906**	0.2904	0.0111
	OCSVM	0.4738	0.9342	0.6287	0.0024
DS2	AE	**0.9138**	0.6628	**0.7683**	**0.0005**
	IF	0.4012	0.6672	0.5011	0.0078
	LOF	0.8718	0.5768	0.6943	0.0007
	OCSVM	0.2679	**0.9444**	0.4174	0.0202
DS3	AE	**0.9177**	0.7063	0.7983	0.0018
	IF	0.8029	0.1348	0.2309	**0.0009**
	LOF	0.5888	0.8360	0.6910	0.0167
	OCSVM	0.7897	**0.8377**	**0.8130**	0.0064

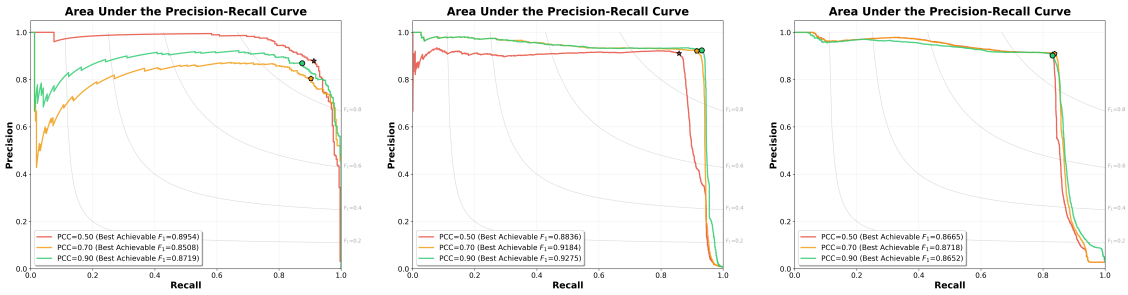
To provide a more detailed view of the comparative behavior between the models, Figure 3 shows the AUPRC for all models throughout the three datasets under $\Delta t = 1\,\text{s}$, evaluated using the most restricted PCC cutoff value of $\theta = 0.50$, suggesting that the AE was the most stable among all models and datasets in this setting.



(a) Comparing AUPRC for DS1. (b) Comparing AUPRC for DS2. (c) Comparing AUPRC for DS3.

Figure 3. Comparing AUPRC for the three datasets using $\theta = 0.50$ and $\Delta t = 1$ s. IF, LOF, AE, and OCSVM are shown in blue, orange, purple, and green, respectively. The highlighted markers denotes the highest achievable F_1 along the curve.

In that context, a comparison was generated using AUPRC between the different PCC cutoff values, see Figure 4. It is observed that the selection of features for the AE evaluation had little impact on its performance. In particular, the use of more aggressive redundancy control (lower θ values) does not lead to a systematic degradation in precision–recall performance. In several scenarios, the differences between $\theta = 0.50$, $\theta = 0.70$, and $\theta = 0.90$ remain small and within a narrow band. This suggests that a substantial fraction of the original 149 features is redundant and that the AE was able to exploit a more compact feature set with a negligible loss in its average precision, different from what the Figure 3 shows for the other models.

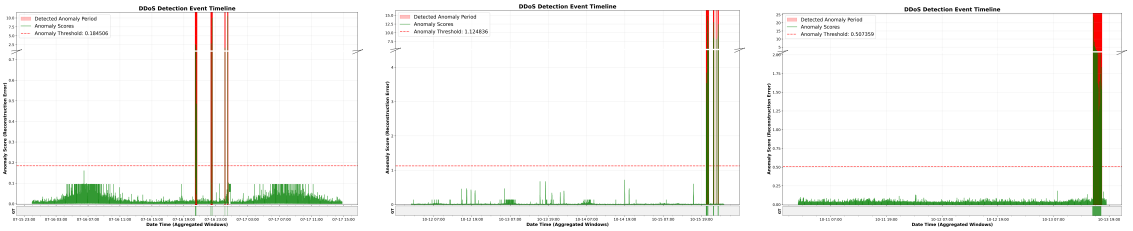


(a) Comparing AUPRC for DS1. (b) Comparing AUPRC for DS2. (c) Comparing AUPRC for DS3.

Figure 4. Comparing AUPRC for the AE under $\theta \in \{0.50, 0.70, 0.90\}$ PCC cutoff values for all datasets using $\Delta t = 1$ s. Curves correspond to $\theta = 0.50$ in red, $\theta = 0.70$ in orange, and $\theta = 0.90$ in green. The highlighted markers denotes the highest achievable F_1 along the curve.

Finally, to complement the analysis, Figure 5 depicts the temporal behavior of AE on the datasets under $\Delta t = 1$ s and $\theta = 0.50$. The detection timelines show that the AE produces anomaly score peaks aligned with the confirmed attack periods, while remaining mostly below the detection threshold during benign intervals.

In contrast with the results shown in Figure 3 and Figure 4, these timelines highlight the AE behavior, in which consistent performance between datasets representing different network environments and DDoS types can be seen. This suggests that while the other models (IF, LOF, and OCSVM) may perform well under specific scenarios, the AE provided a robust result across heterogeneous scenarios.



(a) Detection Timeline for DS1. (b) Detection Timeline for DS2. (c) Detection Timeline for DS3.

Figure 5. Detection Timeline for AE under $\theta = 0.50$ for all datasets using $\Delta t = 1$ s. The anomaly score (reconstruction error) over time, detection threshold, and flagged periods are shown, highlighting the DDoS against the ground truth (GT).

4.1. Discussion

When the shortest aggregation window ($\Delta t = 1$ s) was adopted together with the most restrictive PCC cutoff ($\theta = 0.50$), the AE achieved the most consistently favorable AUPRC across all evaluated datasets. Importantly, this performance was well balanced among the datasets representing different network environments and attack scenarios, rather than being driven by strong results in a single scenario.

This stability is operationally relevant for ITP deployments, where mitigation actions are typically integrated with routing or policy-driven responses (e.g., triggering flowspec rules or rerouting via traffic steering mechanisms). Under these conditions, a detector that maintains a robust precision-recall trade-off across different operational settings is better aligned with a predictable behavior.

Operating in a 1-second aggregation window enables near real-time detection, as decisions can be made after observing only one second of traffic, allowing faster mitigation compared to longer intervals. Moreover, AE preserved strong performance when using a reduced feature set, which reduces the computational cost for feature extraction and model training, an important requirement in high-speed traffic scenarios.

Nonetheless, there are still limitations. This analysis focused on offline detection over fixed configurations. In practice, traffic is non-stationary, and a concept drift method may be required to periodic retraining. Finally, although precision-recall metrics were emphasized, an operational assessment (e.g., time-to-detection or computational cost) was not quantified, where these factors may influence deployment choices.

5. Conclusion and Future Work

In this work, unsupervised flow-based DDoS detection for ITP environments was investigated under realistic operational constraints, using three real-world datasets collected during confirmed incidents. Four one-class detectors (AE, IF, OCSVM, and LOF) were evaluated over multiple aggregation windows and PCC cutoff configurations, with the objective of analyzing the predictive capacity of the algorithms under these settings. Overall, AE achieved the best results with the lowest variability across all datasets in short aggregation windows and compact feature sets.

Future work should include validation under broader traffic dynamics, assessment of the robustness to concept drift, and exploration of online learning strategies while also quantifying computational costs to further improve operational reliability.

6. Acknowledgments

The authors thank the ITPs for granting access to operational telemetry and for their support in building the datasets used in this study. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001.

This manuscript also benefited from AI-assisted tools in accordance with the SBC ethical guidelines. Overleaf and Writefull supported language polishing and consistency checks, Mathematica and Wolfram Assistant were used to generate selected mathematical expressions, ChatGTP 5.1 Pro Extended Thinking helped with tables and text organization, and GitHub Copilot (Student Edition), providing access to the Claude Sonnet 4 model, which supported code development for the experiments.

References

Bhardwaj, A., Mangat, V., Vig, R., Halder, S., and Conti, M. (2021). Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions. *Computer Science Review*, 39:100332.

Cheng, S., Jin, D., Ma, Y., Chen, S., He, H., and Yang, J. (2024). LinkGuard: Link Flooding Attack Detection and Mitigation via Spatio-Temporal Graph Convolutional Network. *IEEE Transactions on Network Science and Engineering*, 11(5):4059–4075. Conference Name: IEEE Transactions on Network Science and Engineering.

Claise, B. (2004). Cisco Systems NetFlow Services Export Version 9. RFC 3954.

Da Silva Ruffo, V. G., Carvalho, L. F., Lloret, J., and Proenca, M. L. (2024). Unsupervised DDoS Detection Using Entropy Features and f-AnoGAN in Software-Defined Networks. In *2024 11th International Conference on Software Defined Systems (SDS)*, pages 35–41, Gran Canaria, Spain. IEEE.

Goldschmidt, P. and Kučera, J. (2024). Windower: Feature Extraction for Real-Time DDoS Detection Using Machine Learning. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2024, NOMS 2024*. Institute of Electrical and Electronics Engineers Inc.

Haag, P. (2025). nfdump: Network flow data dump. Accessed: 2025-10-11.

Haseeb ur Rehman, R. M. A., Aman, A. H. M., Hasan, M. K., Ariffin, K. A. Z., Namoun, A., Tufail, A., and Kim, K.-H. (2023). High-speed network ddos attack detection: A survey. *Sensors*, 23(15):6850.

Hiesgen, R., Nawrocki, M., Barcellos, M., Kopp, D., Hohlfeld, O., Chan, E., Dobbins, R., Doerr, C., Rossow, C., Thomas, D. R., Jonker, M., Mok, R., Luo, X., Kristoff, J., Schmidt, T. C., Wählisch, M., and Claffy, K. (2024). The Age of DDoScovery: An Empirical Comparison of Industry and Academic DDoS Assessments. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, pages 259–279. Association for Computing Machinery.

Kamaldeep, Malik, M., and Dutta, M. (2023). Feature Engineering and Machine Learning Framework for DDoS Attack Detection in the Standardized Internet of Things. *IEEE Internet of Things Journal*, 10(10):8658–8669.

Komadina, A., Martinić, M., Groš, S., and Mihajlović, Ž. (2024). Comparing Threshold Selection Methods for Network Anomaly Detection. *IEEE Access*, 12:124943–124973.

Koumar, J., Hynek, K., Čejka, T., and Šiška, P. (2025a). CESNET-TimeSeries24: Time Series Dataset for Network Traffic Anomaly Detection and Forecasting. *Scientific Data*, 12(1):338. Publisher: Nature Publishing Group.

Koumar, J., Pesek, J., Jerabek, K., and Cejka, T. (2025b). Towards Building Network Outlier Detection System for Network Traffic Monitoring. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2025, NOMS 2025*. Institute of Electrical and Electronics Engineers Inc.

Koumar, J., Pesek, J., Jerabek, K., and Čejka, T. (2025c). Towards Building Network Outlier Detection System for Network Traffic Monitoring. In *NOMS 2025-2025 IEEE Network Operations and Management Symposium*, pages 1–6, Honolulu, HI, USA. IEEE.

Latif, H., Suárez-Varela, J., Cabellos-Aparicio, A., and Barlet-Ros, P. (2023). Detecting Contextual Network Anomalies with Graph Neural Networks. In *Proceedings of the 2nd on Graph Neural Networking Workshop 2023*, GNNet '23, pages 25–30, New York, NY, USA. Association for Computing Machinery.

McDermott, M. B. A., Zhang, H., Hansen, L. H., Angelotti, G., and Gallifant, J. (2025). A Closer Look at AUROC and AUPRC under Class Imbalance. arXiv:2401.06091 [cs].

Meyer, B. H., Pozo, A. T. R., Nogueira, M., and Zola, W. M. N. (2025). Enhancing Intrusion Detection Systems with representation methods: A comparative study. In *2025 IEEE Symposium on Computational Intelligence in Security, Defence and Biometrics (CISDB)*, pages 1–7.

Paolini, D., Dini, P., Soldaini, E., and Saponara, S. (2025). One-Class Anomaly Detection for Industrial Applications: A Comparative Survey and Experimental Study. *Computers*, 14(7).

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. (2019). A survey of network-based intrusion detection data sets.

Salahuddin, M. A., Pourahmadi, V., Alameddine, H. A., Bari, M. F., and Boutaba, R. (2022). Chronos: DDoS Attack Detection Using Time-Based Autoencoder. *IEEE Transactions on Network and Service Management*, 19(1):627–641.

Smith, J. M. and Schuchard, M. (2018). Routing around congestion: Defeating ddos attacks and adverse network conditions via reactive bgp routing. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 599–617.

Sriram, K. and Montgomery, D. (2019). Resilient interdomain traffic exchange: BGP security and DDos mitigation. Technical Report NIST SP 800-189, National Institute of Standards and Technology, Gaithersburg, MD.

Tran, M., Kang, M. S., Hsiao, H.-C., Chiang, W.-H., Tung, S.-P., and Wang, Y.-S. (2019). On the Feasibility of Rerouting-Based DDoS Defenses. In *On the Feasibility of Rerouting-Based DDoS Defenses*, pages 1169–1184. IEEE Computer Society.

Venceslau, F., Souza, R., Silva, F., and Monteiro, J. (2025). Correlação híbrida baseada em stacking para detecção de anomalias em redes de computadores. In *Anais do XXV Simpósio Brasileiro de Cibersegurança*, pages 1003–1010, Porto Alegre, RS, Brasil.