

# Explainable Anomaly Detection in Network Traffic Using Normalizing Flows

Lior Shafir<sup>ID</sup>, Raja Giryes<sup>ID</sup>, and Avishai Wool<sup>ID</sup>

**Abstract**—Anomaly detection in network traffic is critical for identifying deviations from normal behavior—including sophisticated cyber threats and previously unseen attacks—especially when anomalous examples are absent from the training data. The escalating complexity of cyber-attacks necessitates developing methods that not only identify low-likelihood traffic but also provide insights into its anomalous nature and deviations from normal behavior, enabling effective response and troubleshooting. In this work, we leverage the unique capabilities of normalizing flows (NF), a state-of-the-art reversible generative model for exact density estimation, to detect anomalies using only normal traffic. Our approach fundamentally differs from previous methods by utilizing NF’s exact likelihood computation for unsupervised detection and combining it with Shapley values to introduce a novel feature selection framework for guiding the selection of discriminative features in anomaly detection, while also providing statistically grounded enhanced explanations for detected anomalies, pinpointing potential root causes. Through experiments on CICIOT-2023, ISCXTor2016, and CICIDS2017, we demonstrate that our NF-based approach outperforms existing state-of-the-art methods for unsupervised anomaly detection. Notably, on the CICIOT-2023 dataset, we achieve an accuracy of 0.9951, comparable or higher than supervised methods, despite being trained solely on normal data.

**Index Terms**—DDoS attacks, intrusion detection system, flow-based network intrusion detection, network flow.

## I. INTRODUCTION

WITH the exponential growth of connected devices, particularly in the Internet of Things (IoT) domain, the cyber threat landscape has evolved significantly. According to Cisco [1], the proliferation of IoT devices is expected to reach hundreds of billions by 2030. This surge in connectivity presents a vast attack surface, increasing the complexity and frequency of potential cyber threats. Consequently, the number of types of anomalous network traffic that threaten the internet is larger than ever before. Hence, with such a broad spectrum of low-likelihood network traffic, anomaly detection remains a critical challenge.

Anomalous traffic may refer to malicious attacks but it can also include benign deviations—such as infrequent legitimate variations in traffic or undesired behaviors stemming from regulatory compliance, privacy constraints, or capacity limitations.

Received 16 January 2025; revised 10 August 2025; accepted 8 September 2025; approved by IEEE TRANSACTIONS ON NETWORKING Editor E. Bertino. Date of publication 13 October 2025; date of current version 5 January 2026. (Corresponding author: Lior Shafir.)

The authors are with the School of Electrical Engineering, Tel Aviv University, Tel Aviv 6997801, Israel (e-mail: lior.shafir@gmail.com).

Digital Object Identifier 10.1109/TON.2025.3617580

Network Intrusion Detection Systems (NIDSs) detect attacks via signature- or anomaly-based inspection of network traffic [2], [3]. Recent progress in machine learning (ML) has opened new opportunities for improving the accuracy of traffic classification tasks, including the detection of anomalous or malicious activity. However, ML-based NIDS still face substantial challenges due to the complexity of modern ML models and the need to process high-dimensional or time-series network data [4].

Supervised ML approaches, in particular, face several limitations in this context: Firstly, they require large amounts of labeled anomalous data for training, which is often unavailable. Secondly, they encounter challenges with extreme class imbalance, which further degrades detection accuracy, skewing model learning towards majority classes and hindering the identification of minority traffic classes. Lastly, collecting anomalous traffic data is a costly and time-consuming task due to its elusive and sporadic nature.

To address the lack of anomalous data for training, some approaches have explored the use of deep generative models, such as Generative Adversarial Networks [5] (GANs), to synthetically create anomaly samples [6]. However, while generative neural methods like GANs and Variational Autoencoders [7] (VAEs) have demonstrated prominent performance results on tasks like learning the distribution of natural images, neither of them allows for exact computation of the probability density of new data points.

In this work, we propose a novel approach leveraging Normalizing Flows (NF) [8] for anomaly detection. NF is a reversible generative model framework that produces tractable distributions, where both sampling and density-estimation directions (i.e., the generative and normalizing directions, respectively) can be efficient and exact.

NF learn an invertible mapping between a simple base distribution and a complex data distribution, enabling exact likelihood computation. By training the flow solely on normal network traffic data, anomalies can be detected as low-likelihood samples under the learned distribution, eliminating the need for labeled anomaly examples during training.

While NF have generative abilities, we leverage its density estimation capability rather than its generative direction. Our approach fundamentally differs from other anomaly detection works that employ generative methods such as GANs, AEs, and even NF in that we do not generate synthetic anomalies followed by training a supervised classifier.

Such models are inherently sensitive to the proportion and authenticity of anomalies present in the training set. Firstly, the generated synthetic anomalies do not overlap well with real anomaly samples [9], which may lead to suboptimal training, and reduce the model's detection effectiveness in real-world scenarios. Secondly, according to [9], the model's performance decreases when the anomaly-to-normal sample ratio is modified. This indicates that the model's efficacy is closely tied to specific sample ratios, which may not be feasible to predict in real-world scenarios where the rate of anomalies can be unpredictable.

Feature selection is a critical challenge in anomaly detection, particularly in unsupervised settings, where including irrelevant or redundant features can obscure the essential characteristics of normal traffic and degrade the model's performance. To address this, we propose a novel Shapley-based [10] feature selection framework that leverages NF's probability-based evaluation function. This approach enables us to identify and rank discriminative features effectively, overcoming the limitations of traditional feature selection methods that often struggle in unsupervised contexts. Additionally, our use of Shapley values enhances the interpretability of detection results, providing statistically grounded explanations for why a particular traffic sample is anomalous and how it deviates from normal patterns.

We evaluate our method on three benchmarks: CICIOT-2023 [11], ISCXTor2016 [12], and CICIDS2017 [13]. While two of these datasets contain comprehensive and diverse sets of attack scenarios, we also extend our evaluation to the ISCXTor2016 dataset, which focuses on encrypted network traffic and includes both Non-Tor and Tor network traffic across seven different application categories. Notably, the recent CICIOT-2023 dataset encompasses 33 distinct attacks across seven attack categories. To the best of our knowledge, our work is the first study to evaluate this dataset using only normal traffic.

Our contributions may be summarized as follows:

- We introduce a novel anomaly detection method that leverages NF's exact density estimation capabilities to model normal network traffic and identify anomalies as low-likelihood samples.
- We integrate the NF probability-based evaluation function with Shapley values [10] to propose a novel wrapper feature selection method tailored for unsupervised learning, demonstrating its efficacy through extensive experiments. Additionally, through simulations and real attack examples, we demonstrate the explainability benefits of using Shapley values with NF to provide statistically grounded explanations for detected anomalies.
- We evaluate our method on three publicly available datasets: two from the DDoS and IDS cyber-attacks domain, and one from the encrypted traffic classification domain. We show that our flow-based approach outperforms existing state-of-the-art anomaly detection methods for binary classification tasks. Perhaps most strikingly, on the recent CICIOT-2023 dataset, which contains 33 attacks across 7 different categories, we demonstrate that our method achieves a detection accuracy comparable

or higher than supervised methods, despite being trained solely on normal data.

## II. RELATED WORK

**Anomaly-Based Intrusion Detection.** Anomaly detection in network traffic has been extensively studied over the past few decades, cf. [14], [15]. In recent years, deep learning-based methods have gained significant traction, offering improved accuracy and robustness. Proposed methods include Support Vector Machines (SVM), Convolutional or Recurrent Neural Networks (CNN/RNN), and more, cf. [16].

More recently, the application of generative models such as VAEs and GANs for anomaly-based detection has been thoroughly investigated. Zavrak and Iskefiyeli [17] employ both Autoencoders (AEs) and VAEs in a semi-supervised learning framework to identify unknown attacks based on flow features extracted from network traffic. Another work by Min et al. [18] explores the use of a memory-augmented deep autoencoder for network anomaly detection. It focuses on leveraging deep learning techniques, specifically AE, enhanced with memory components to improve anomaly detection accuracy in network traffic data. Azmin and Islam [19] combine a variational Laplace AE (VLAЕ) with a deep neural network (DNN) to enhance intrusion detection accuracy.

GANs [5] have also been used for adversarial traffic generation and anomaly-based intrusion detection. E.g., NetShare [20] is a GAN-based framework for generating synthetic IP header traces, Gadot [21] is a GAN-based framework aimed at enhancing the detection of DDoS attacks, and NIDSGAN [22] is a GAN-based framework for generating realistic adversarial network traffic flows.

Wang et al. [23] present Def-IDS, an ensemble-based defense mechanism designed to protect deep learning-based network intrusion detection systems from adversarial attacks.

Peng et al. [24] also propose a framework where a GAN is used to generate adversarial network traffic, which is then used to train the NIDS. The GAN-generated adversarial examples help the NIDS to better identify and mitigate potential threats.

While various studies employed VAEs and GANs for anomaly detection, they face challenges like intractable marginal likelihoods and mode collapse. Additionally, GANs require a large amount of training data to generate network traffic samples. Thus, the acquisition of malicious samples is still a challenging task.

Another category of unsupervised learning approaches includes classical algorithms such as Isolation Forest (IF) and more recent self-supervised methods based on contrastive learning. These techniques are trained without access to labels and aim to identify structural irregularities in the data. For example, Zhang et al. [25] propose an online IF-based model for malicious traffic detection in SD-WAN environments, where the model is trained on a mixture of benign and attack traffic without using labels. Similarly, Li et al. [26] present a self-supervised contrastive learning framework that learns representations for intrusion detection using data augmentations and contrastive loss, without label supervision during the pretraining stage. However, unlike our approach, these methods are typically trained on a mixture of normal and

anomalous samples, relying on their presence in the training data to identify structural irregularities. In contrast, our model is trained exclusively on clean benign traffic.

**Normalizing Flows for Anomaly Detection.** NF have received attention in the field of anomaly detection, particularly due to their capability to model complex probability distributions. Various studies in fields other than network traffic detection have utilized NF for anomaly detection, producing promising outcomes. Gudovskiy et al. [27] proposed CFLOW-AD, which employs conditional NF for anomaly detection with localization. CFLOW-AD characterizes the distribution of normal network-based features and calculates precise data likelihoods of the examined features, demonstrating faster and more compact performance compared to earlier models. Yet, it requires a specialized design that is distinct from standard conditional NF models.

Yu et al. [28] introduced FastFlow, a detection technique that extends conventional NF to two-dimensional space. Initially, a feature extractor gathers visual features from normal samples, which are then input into a 2D flow model to estimate the probability distribution. FastFlow shows enhanced accuracy and efficiency over previous methods. Dias et al. [29] developed an unsupervised density estimation approach for trajectory anomaly detection using NF. They applied RealNVP and Masked Autoregressive Flow (MAF) to model trajectory data and identify anomalies, with their experimental results showing that NF outperform traditional density-based methods such as Local Outlier Factor (LOF) and Gaussian Mixture Models (GMM). Ryzhikov et al. [30] proposed a model-agnostic anomaly detection process based on NF, addressing class-imbalanced classification by integrating existing anomalous samples during training and reconfiguring one-class classification as two-class classification. The experimental findings indicated that NFAD surpassed existing anomaly detection methods.

Concurrently to us, Dang et al. [9] proposed a three-stage framework that uses only normal traffic data to generate pseudo-anomaly samples. Their approach also involves the use of NF to generate synthetic anomalous samples, and training a supervised classifier to distinguish between normal and pseudo-anomaly samples. They show state-of-the-art performance on several datasets.

Our work differs from [9] in several key aspects. Firstly, while their research addresses encrypted traffic binary classification, our work addresses network intrusion detection of more than 35 different attacks across two IDS and DDoS datasets. Secondly, their method uses full packet information, including payload data, whereas we rely solely on flow header statistics. Thirdly, their approach incorporates a supervised classifier for detecting anomalies, whereas we utilize the log probability of NF to identify anomalous traffic. For a fair comparison we also employed the Tor dataset used by [9] and we provide a comparative analysis of the results.

### III. PRELIMINARIES

**Normalizing flows**, which gained prominence through the work of Rezende and Mohamed [8] in variational inference, and Dinh et al. [31] for their application in density estimation,

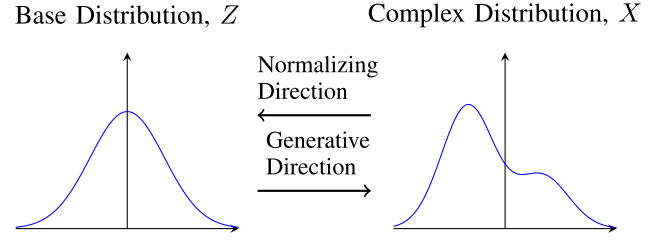


Fig. 1. NF Key Idea Illustration. The left figure is the density function of the source  $p_Z(z)$ . The right figure is the density function of the target distribution  $p_X(x)$ .

are a powerful framework in ML for constructing flexible probability distributions by transforming a simple base distribution through a series of invertible mappings.

While in the past, flow-based generative models have gained less attention in the research community compared to GANs and VAEs, they have become popular recently [32], and have been used in various applications such as speech processing (e.g., NVIDIA WaveGlow [33]), image generation (e.g., OpenAI Glow [34]), and reinforcement learning [35].

The key idea is to represent a complex target distribution  $p_X(x)$  as the result of applying a series of invertible transformations to a simple base distribution  $p_Z(z)$ , such as a standard normal distribution (see Figure 1).

Consider a random variable  $x \in \mathbb{R}^D$  with a complex distribution  $p_X(x)$  that we aim to model. NF construct a mapping  $f$  such that  $x = f(z)$ , where  $z$  is a latent variable with a known and simple distribution  $p_Z(z)$ . The mapping  $f$  is chosen to be invertible, with an inverse  $f^{-1}$ , allowing us to express  $z$  as  $z = f^{-1}(x)$ . To compute the density  $p_X(x)$  under the transformation  $f$ , we use the change of variables formula. Thus, we get

$$p_X(x) = p_Z(z) \left| \det \left( \frac{\partial f^{-1}(x)}{\partial x} \right) \right|. \quad (1)$$

Since  $z = f^{-1}(x)$ , we can rewrite this as:

$$p_X(x) = p_Z(f^{-1}(x)) \left| \det \left( \frac{\partial f(z)}{\partial z} \right) \right|^{-1}. \quad (2)$$

However, calculating the Jacobian determinant of  $f^{-1}$  directly can be challenging. Thus, we use the fact that:

$$\det \left( \frac{\partial f^{-1}(x)}{\partial x} \right) = \left( \det \left( \frac{\partial f(z)}{\partial z} \right) \right)^{-1}. \quad (3)$$

To facilitate computation, especially in a ML context, it is common to work with the natural logarithm of the densities, which transforms the products into sums, simplifying the gradients computation. Thus, the log-density is given by:

$$\log p_X(x) = \log p_Z(f^{-1}(x)) - \log \left| \det \left( \frac{\partial f(z)}{\partial z} \right) \right|. \quad (4)$$

NF are typically constructed by composing multiple simple invertible transformations, each contributing to the overall flexibility of the model. Let

$$f = f_K \circ f_{K-1} \circ \dots \circ f_1 \quad (5)$$



be a composition of  $K$  invertible transformations  $f_k$ . The log-density transformation through a sequence of flows is then given by:

$$\log p_X(x) = \log p_Z(z_0) - \sum_{k=1}^K \log \left| \det \left( \frac{\partial f_k(z_{k-1})}{\partial z_{k-1}} \right) \right|, \quad (6)$$

where  $z_k = f_k(z_{k-1})$  for  $k = 1, \dots, K$  and  $z_0 = f^{-1}(x)$ .

In practice, the choice of transformations  $f_k$  is critical for ensuring that the flow is both expressive and computationally efficient. Common choices include affine coupling layers, autoregressive flows, and more complex neural network-based transformations. Each transformation must be designed to allow efficient computation of the determinant of the Jacobian and its inverse.

By leveraging these transformations, NF can model highly complex data distributions while maintaining tractable density estimation and sampling capabilities. This makes them particularly well-suited to tasks such as anomaly detection, where exact likelihood evaluation is crucial.

**Shapley values.** Based on cooperative game theory, Shapley values [10] provide a way to fairly distribute the “payout” (in our context, feature importance) among the features of a model based on their contribution to the model’s predictions. They are used to interpret complex ML models by assigning an importance value to each feature reflecting its contribution to the prediction.

The Shapley value for a feature is calculated as the average marginal contribution of that feature across all possible subsets of features. This ensures a fair distribution of importance, adhering to properties such as efficiency (the total importance is distributed among features), symmetry (features that contribute equally receive equal values) and additivity (the sum of the individual contributions equals the total contribution of the features) [36].

Formally, the Shapley value for a feature  $i$  in a model with  $n$  features is defined as the weighted average of the marginal contributions of  $i$  across all possible subsets of features. Let  $S$  be a subset of features not including  $i$ , and  $v(S)$  represent the value function (e.g., model output) for the subset  $S$ . The Shapley value  $\phi_i$  for feature  $i$  is given by:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)],$$

where  $N$  is the set of all features,  $n = |N|$ , and  $v(S \cup \{i\}) - v(S)$  quantifies the contribution of  $i$  when added to subset  $S$ . This formulation ensures fairness by adhering to the properties mentioned above. Due to the exponential complexity of calculating Shapley values for all subsets, efficient approximations are often employed in practice.

Shapley values can provide both global and local explanations. Global explanations offer an overall importance ranking of features for the entire dataset, helping to understand which features are generally most influential. Local explanations, on the other hand, provide feature importance for individual predictions, offering insights into why a specific prediction was made by highlighting which features had the most influence on that particular outcome. We are particularly interested in

local explanations to understand why certain anomalies deviate from normal traffic patterns. By focusing on local Shapley values, we can pinpoint the specific features that contribute most significantly to an anomaly, enhancing our ability to diagnose and respond to unusual network behaviors.

#### IV. PROPOSED METHOD

We propose a novel approach using bidirectional NF to learn the distribution of normal network traffic, and then use its normalizing direction, which is essentially a probability density function, to distinguish between traffic anomalies and normal behavior.

##### A. Data Processing and Formulation

For all input network traffic data, we consider only the header information at the flow level. The features we use are aggregated flow statistics such as duration, packet size statistics, inter-arrival times (IAT), and network and transport layers characteristics (e.g., flag counts and window size) for each direction of the flow traffic. The data is then represented as a two-dimensional matrix  $\mathbf{X}$  with  $N$  rows and  $D$  columns,  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^\top$ , where each row  $\mathbf{x}_i \in \mathbb{R}^D$  is a vector of  $D$  features representing a flow sample. Each sample has a label  $y_i$  where  $y_i \in \{0, \dots, C\}$ , with 0 representing a normal traffic label, and  $y_i$  representing an anomaly class (e.g., attack vector or traffic category depending on the classification task). Since our model relies only on normal traffic to estimate the likelihood function, our classifier is trained only with samples where  $y_i = 0$ . Thus, class imbalance issues, which exist in all datasets that we evaluate, do not impact our proposed model’s operation. We later show how we handle the class imbalance in the testing phase. We also remove any features related to ports, IP addresses, or absolute timestamps, as they might be dependent on the setup on which the dataset was created or might have artifacts with the flow label.

**Feature Normalization.** NF require feature normalization. We use z-score normalization, where the features are normalized to have a mean of zero and a standard deviation of one. It is given by  $z_i = (x_i - \mu)/\sigma$  where  $\mu$  and  $\sigma$  are the mean and standard deviation of the features, respectively. This normalization method is chosen over alternatives such as min-max scaling or max-abs scaling due to its effectiveness in handling outliers and its common use in statistical modeling.

##### B. Using NF for Anomaly Detection

**Training Phase.** During the training phase, the input data is the matrix  $\mathbf{X}$  that contains a set of normal network traffic samples only. The NF framework optimizes and learns the parameters of the transformation that suit the normal traffic data distribution we are trying to model, with the objective of maximizing the log-likelihood of the input  $\mathbf{X}$ , as given by Equation (6).

**Loss Function.** The equivalent minimization problem to maximizing the log-likelihood can be formulated as minimizing the negative log-likelihood. Therefore, the loss function for training our NF model is defined as:

$$\mathcal{L} = - \sum_{i=1}^N \log p_X(\mathbf{x}_i), \quad (7)$$

**Algorithm 1** Anomaly Detection Using NF

---

```

1 Function TrainFlow( $X = \{x_1, \dots, x_N\}, E_{NF}$ ):
2   Normalize features using z-score normalization;
3   for  $epoch = 1$  to  $E_{NF}$  do
4     Sample minibatch  $X^+ \sim X$ ;
5     Transform to latent space:  $Z^+ = f^{-1}(X^+; \theta)$ ;
6     Compute log-likelihood:  $\log p_X(X^+) =$ 
        $\log p_Z(Z^+) - \log |\det(\partial f / \partial Z)|$ ;
7     Compute loss:  $\mathcal{L} = -\sum \log p_X(X^+)$ ;
8     Update parameters  $\theta$  via gradient descent on  $\mathcal{L}$ ;
9   return trained flow model  $f_\theta$ ;
10 Function ClassifySamples( $f_\theta, X_{in}, \tau$ ):
11   for each input sample  $x \in X_{in}$  do
12     Compute log-likelihood  $\log p_X(x)$  using  $f_\theta$ ;
13     if  $\log p_X(x) < \tau$  then
14       Label  $x$  as anomaly;
15     else
16       Label  $x$  as normal;
17   return predicted labels;
18 Function ComputeThreshold( $f_\theta, X_{val}^+, X_{val}^-$ ):
19   Compute log-likelihoods
      $\ell^+ = \{\log p_X(x) \mid x \in X_{val}^+\}$ ;
20   Compute log-likelihoods
      $\ell^- = \{\log p_X(x) \mid x \in X_{val}^-\}$ ;
21   for candidate threshold  $\tau$  in range of  $\ell^+ \cup \ell^-$  do
22     Compute TPR and FPR based on  $\ell^+$  and  $\ell^-$ 
       using  $\tau$ ;
23     Compute score:  $s_\tau = \text{TPR}(\tau) - \text{FPR}(\tau)$ ;
24   return threshold  $\tau^*$  that maximizes  $s_\tau$ ;

```

---

where  $N$  is the number of samples in  $\mathbf{X}$ , and  $\mathbf{x}_i$  is the feature vector of the  $i$ -th sample. This loss function encourages the model to increase the probability of observing the given samples under the modeled distribution, effectively fitting the normal traffic profile, as outlined in the TRAINFLOW function of Algorithm 1.

### C. Feature Selection

Feature selection techniques can be categorized into two main groups:

- **Wrapper Methods:** These evaluate subsets of features by training and validating models on different combinations, optimizing performance metrics.
- **Filter Methods:** These rely on statistical properties, such as correlation or mutual information, to rank and select features.

We have developed and examined several novel wrapper feature selection techniques that integrate Shapley values with the log-likelihood function, which serves as the evaluation metric for our NF model. While Shapley values have become a cornerstone in the field of explainable artificial intelligence (XAI), their typical application is to attribute feature importance for a given model or to explain individual predictions, rather than being explicitly used as a tool for feature selection. Furthermore, as discussed in some studies [36], the use of Shapley-based attribution methods for feature selection has

been found to be unsuitable and, in certain cases, even counterproductive.

Our aim is not to provide an exhaustive theoretical or empirical study on the use of Shapley values in feature selection. Instead, we propose an approach that leverages Shapley values to guide feature selection while avoiding the computational burden of an exhaustive search through the entire model space of  $2^F$  subsets.

**Feature Selection Approaches.** Our technique focuses on *local* feature Shapley values, computed using small subsets of samples, rather than relying on global Shapley values across the entire model. We explore two distinct feature selection approaches tailored to different settings:

**Few-Shots:** This semi-supervised method for feature selection leverages a limited set of labeled anomalies. Using this set, we compute the Shapley values for each sample prediction on a trained NF model. The marginal contribution of a feature is evaluated based on its impact on the detection score (log probability) for anomaly samples compared to normal samples. Intuitively, an effective discriminative feature should exhibit a negative marginal contribution to anomaly detection scores and a positive marginal contribution to normal detection scores.

We employ the following forward feature selection procedure to evaluate features in the dataset:

- 1) Initialize a base set of features  $S$ .
- 2) For each feature  $f \in F \setminus S$ :
  - Train a Normalizing Flow detection model  $NF_f = NF(S \cup \{f\})$  using the training set.
  - Compute  $C_f = \text{contribution}(f, NF_f, a_{set}, n_{set})$ .
- 3) Select the top  $k$  features based on their  $C_f$  scores.

Here,  $a_{set}$  refers to the limited set of anomalies, and  $n_{set}$  is a similarly sized set of normal samples. We experimented with various initial sets  $S$  and contribution metrics to evaluate the distribution of Shapley values across  $a_{set}$  and  $n_{set}$ . To determine the most appropriate  $S$  and contribution metric, we analyzed the correlation between Shapley-based contribution scores and the NF model's detection performance on a validation set, measured using the AUROC metric.

As detailed in Section V, selecting  $S$  with one or more discriminative features hindered comparisons of marginal contributions for tested features. The primary issue is that the Shapley values of tested features are obscured by interactions with existing features in  $S$ . In such cases, each trained NF model effectively represents a different “game” with unique characteristics, in which the Shapley values are not necessarily comparable. Consequently, Shapley values vary significantly across normal and anomaly samples and across trained models.

To address this, we tested  $S$  containing a single “dummy” feature with controlled distributions. As demonstrated in our results, using a random noise feature in  $S$  and training NF models with the noise feature alongside the tested feature provides a method that clearly distinguishes good and poor discriminative features.

**Transductive:** In this approach, we take an untagged sampling of both normal and anomaly samples, without their labels. By combining the log-likelihood function with Shapley values, we identify features that exhibit high variance in Shapley value distributions. A discriminative feature will

TABLE I  
LIST OF ATTACKS AND NUMBER OF SAMPLES IN CICIOt-2023 DATASETS

Attack	Samples	Attack	Samples	Attack	Samples
<b>DDoS</b>		<b>Dos</b>		<b>Mirai</b>	
ACK fragmentation	285,104	UDP flood	3,318,595	GREIP flood	751,682
UDP flood	5,412,287	SYN flood	2,028,834	Greeth flood	991,866
SlowLoris	23,426	HTTP flood	71,864	UDPPlain	890,576
ICMP flood	7,200,504	TCP flood	2,671,445	<b>Web-based</b>	
RSTFIN flood	4,045,285	<b>Recon</b>		Sql injection	5245
PSHACK flood	4,094,755	Ping sweep	2,262	Command injection	5,409
HTTP flood	28,790	OS scan	98,259	Backdoor malware	3,218
UDP fragmentation	286,925	Vulnerability scan	37,382	Uploading attack	1,252
TCP flood	4,497,667	Port scan	82,284	XSS	3,846
SYN flood	4,059,190	Host discovery	134,378	Browser hijacking	5,859
SynonymousIP	3,598,138	<b>Spoofing</b>		<b>Brute Force</b>	
ICMP Fragmentation	452,489	DNS	178,911	Dictionary brute force	13,064
Benign	1,098,195	Arp	307,593		

typically show both relatively high and low Shapley values, indicating significant impacts on the likelihood computations for anomalies compared to normal traffic. For each feature, we compute the mean absolute Shapley value for all samples in the set. This overall score, termed the *Transductive Shapley Value*, is expected to be relatively high for features that can effectively separate normal samples from anomalies:

$$\text{Transductive Shapley Value}_i = \frac{1}{N} \sum_{j=1}^N |\phi_{i,j}|,$$

where  $\phi_{i,j}$  is the Shapley value of the  $i$ -th feature for the  $j$ -th sample and  $N$  is the total number of samples. Feature selection experiments results for both Few-Shots and Transductive approaches are provided in Section V.

#### D. Likelihood-Based Detection

Once the NF model is trained using normal network traffic, we utilize the corresponding likelihood function to distinguish between unseen anomalies and normal samples. The key idea is that normal samples should have high likelihoods under the learned distribution, while anomalies should have low likelihoods.

During the detection phase, we calculate the log-likelihood of each incoming network traffic sample using the trained model. The decision to classify a sample as normal or anomalous is based on a predefined threshold. The detection function  $f$  is defined as:

$$f(x) = \begin{cases} \text{Normal} & \log p_X(x) > \text{threshold} \\ \text{Anomalous} & \log p_X(x) \leq \text{threshold} \end{cases}$$

The threshold value is an important parameter that determines the sensitivity of the detection system. It can be selected based on the desired trade-off between false positive and false negative rates, often determined through empirical analysis or validation on a labeled dataset.

## V. EVALUATION

### A. Datasets

We incorporate two datasets focused on intrusion detection, along with an additional dataset for encrypted traffic clas-

TABLE II  
LIST OF ATTACKS AND NUMBER OF FLOW INSTANCES IN THE CICIDS2017 DATASET

Attack	Samples	Attack	Samples
Benign	2,273,097	DoS - Hulk	231,073
PortScan	158,930	DoS - GoldenEye	10,293
DDoS	128,027	DoS - Slowloris	5,796
Botnet	1,966	DoS - Slowhttptest	5,499
FTP-Patator	7,938	SSH-Patator	5,897
Web BruteForce	1,507	Web XSS	652
SQL Injection	21	Heartbleed	11
Infiltration	36		

TABLE III  
SUMMARY OF TOR VS NON-TOR IN ISCXTor2016

Type	Number of Flows
Tor	14,508
Non-Tor	18,758
<b>Total</b>	<b>33,266</b>

sification.<sup>1</sup> This diverse selection allows us to compare our work with a broader set of state-of-the-art anomaly detection methods including various generative models.

**CICIOt-2023:** The CICIOt-2023 [11] dataset is designed to evaluate intrusion detection systems (IDS) within IoT environments. The dataset provides more than 45 million attack flows and more than one million normal traffic flows, capturing a wide variety of attacks categorized into seven classes: Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance, Web-based, Brute Force, Spoofing, and Mirai. More details about the specific attacks included in the dataset can be found in Table I. All state-of-the-art classification methods evaluated on this dataset in the literature are supervised.

<sup>1</sup>Throughout this paper, we refer to anomalous samples that are labeled as malicious in the dataset as “attacks.” Our anomaly detector is trained exclusively on traffic deemed “normal” — typically the benign or majority class — and flags deviations from this distribution. In the CICIDS2017 and CICIOt-2023 datasets, the training set consists of benign traffic, and anomalous samples correspond to labeled attacks. In the ISCXTor2016 dataset, the model is trained on Non-Tor traffic, and Tor traffic is treated as anomalous. While some anomalies may be benign in practice (e.g., rare but legitimate activity), our terminology reflects dataset-specific labeling conventions.

**ISCXTor2016:** The ISCXTor2016 [12] dataset focuses on the classification of encrypted traffic, particularly Tor network traffic. This dataset includes both Tor and non-Tor traffic (see Table III), capturing various internet activities such as browsing, streaming, and file transfers. It contains detailed labels indicating the type of application, allowing for the evaluation of traffic classification methods in an encrypted context. The dataset emphasizes the challenge of identifying and classifying traffic when encryption is used to obfuscate the data.

Using time-based features such as flow duration and inter-arrival times, ISCXTor2016 aids in distinguishing between different types of traffic. This focus on temporal characteristics is crucial for the effective analysis and classification of encrypted traffic, where traditional packet inspection methods are not feasible.

**CICIDS2017:** The CICIDS2017 [13] dataset is a comprehensive benchmark for evaluating intrusion detection systems. This dataset captures a wide range of attack scenarios, including DoS, DDoS, brute force, XSS, SQL injection, infiltration, port scans, and botnet attacks as detailed in Table II. Collected over five days, the dataset includes detailed labels and meta-data for each instance, facilitating a thorough evaluation of IDS techniques.

The dataset contains both raw network traffic and extracted features, facilitating different levels of analysis. Flow-based features include metrics such as duration, packet sizes, and various counts of flags and protocol-specific characteristics. This extensive set of features, combined with the variety of attack types, makes CICIDS2017 a valuable resource for developing and testing advanced intrusion detection systems. While the CICIDS2017 dataset has been extensively evaluated and studied, there are limited results evaluating this dataset using a binary classifier trained solely on normal traffic. To compare our method with state-of-the-art techniques, we evaluate DoS/DDoS attacks (Wednesday traffic, i.e., DoS Hulk, DoS Goldeneye, Slowloris, and Slowhttptest) similar to the evaluation in [18] which utilized a semi-supervised, memory-augmented auto-encoder approach for anomaly detection on CICIDS2017, and we compare our results to their approach.

## B. Testing Methodology

We consider a binary classification model that classifies each input flow as Normal or Anomalous. The datasets we use are highly imbalanced. In CICIOT-2023, there are significantly more attack flow instances than benign flows, while in ISCXTor2016 and CICIDS2017, there are significantly more Non-Tor and benign flow instances than Tor and attack flows, respectively.

While class imbalance does not impact the training phase, it does affect testing. For each dataset, we combine all normal traffic together and all malicious traffic together (in the case of attack datasets; for the Tor dataset, we combine all Tor traffic applications), then shuffle them. For each dataset, we randomly sampled 20,000 normal flows and 10,000 anomalous flows. We then split the normal flows into training (10,000 samples) and testing (10,000 samples) sets.

In these considerations, we follow similar balancing and sampling schemes used for testing by other works that evaluate their binary classification models using CICIOT-2023 [11], [37], CICIDS2017 [18], and ISCXTor2016 [9]. This approach ensures proper baselines for comparing the performance and accuracy results of our method to these state-of-the-art classification models.

We conducted experiments five times using different random seeds and reported the mean Area Under the Receiver Operating Characteristic Curve (AUROC). Additionally, for the CICIOT-2023 dataset, we compared our method to supervised methods. Therefore, we also reported the F1 score, Recall, Precision, and Accuracy, using a threshold that maximizes the True Positive Rate (TPR) minus the False Positive Rate (FPR).

For AUROC-based evaluations, no threshold tuning is required. However, when reporting threshold-dependent metrics such as Precision, Recall, F1-score, and Accuracy, we select the decision threshold using a separate validation set. This validation set is completely disjoint from both the training and test sets and contains 1,000 normal and 1,000 anomaly samples—amounting to 10% of the test set size. The final metrics are then computed on the fully held-out test set comprising 10,000 normal and 10,000 anomaly flows.

We also validated that performance metrics remained consistent when evaluated on the full datasets. However, for consistency and comparability with prior works, we report performance on downsampled test sets as described.

**Implementation Details.** We train our NF model using PZFlow [38]. The models are trained for 100 epochs with a batch size of 1024. XGBoost [39] classifiers which we use as benchmarks (See Section V-D), are trained with learning rate of 0.1, using 10 trees, each has a 10 depth limit. To estimate Shapley values for any model, we make use of the open-source code provided by the authors of [10].

We release an implementation of our proposed method. The code is available at: <https://github.com/lshafir/NF-anomaly-detection>

## C. Feature Selection Experiments Results

Here we present experimental results for the two Shapley based feature selection approaches introduced in Section IV-C.

**Few Shots (Semi-Supervised) Approach.** Given a dataset and an input set of features, the *Few Shots* feature selection technique leverages a limited number of anomaly samples and Shapley value distributions to identify features with strong discriminative power.

We experimented with the wrapper forward feature selection procedure using different combinations of base features  $S$  and various Shapley distribution contribution metrics.

To evaluate this procedure, we analyzed the correlation between Shapley value contributions on anomaly and normal samples with the detection performance of the trained models. For each experiment, we followed these steps:



- 1) Select a base set of features  $S$ .
- 2) For each feature not in  $S$ , train a NF model  $NF_f$  using  $S \cup \{f\}$ .
- 3) Compute Shapley value distributions on small sample sets of  $k$  anomaly samples and  $k$  normal samples.
- 4) Evaluate the detection performance of  $NF_f$  on the full testing set to obtain an AUROC score.
- 5) Compare the observed Shapley value distributions with the detection AUROC scores of the models.

We set  $k = 20$  in the following experiments. This choice reflects the few-shot learning approach, which typically involves learning or making inferences from a very limited number of examples (commonly 1 to 100 samples per class). The goal is to evaluate whether a model or interpretability method can extract meaningful information from scarce labeled data. This value is supported by an ablation study we conducted, evaluating the impact of different few-shot sample sizes, as described later in §V-E.3.

Note that the NF models are trained solely on normal samples and use a probability (likelihood) score during detection. A feature's Shapley value for an individual prediction represents the impact of that feature on the output detection score. Thus, an effective discriminative feature should exhibit a negative marginal contribution to anomaly sample detections and a positive marginal contribution to normal sample detection scores.

While we explored various initial subsets of features, our results revealed that including both strong and weak discriminative features in the base set  $S$  caused varying interactions with tested features, obscuring their individual contributions. We experimented with several absolute and relative metrics for evaluating Shapley distributions but observed that interactions among features in  $S$  significantly affected the interpretability of results.

As an example, Figure 3 shows results from an experiment where we selected a single feature, *Packet Length Variance*, as the base set  $S$ . We tested over 50 other features from the CICIDS2017 dataset, each trained alongside the base feature. The box plot illustrates the 10 features with the highest AUROC scores (rightmost) and the 10 features with the lowest AUROC scores (leftmost). For each feature, the AUROC score of the corresponding model is presented above the feature name. The Shapley value distributions for the base feature and the tested feature are displayed in the top and bottom areas of the plot, respectively.

From the results, we observed two key findings: (i) the Shapley value distribution of the base feature varied significantly across trained models, even when evaluated on the same set of anomaly (attack) samples, and (ii) no clear correlation was observed between the Shapley value distributions of the tested features and the model detection performance.

Given these results, we sought a “neutral” feature to serve as a baseline for evaluating the Shapley value contributions of other features. We experimented with a single “dummy” feature with controlled distributions. While a random feature with a normal distribution exhibited similar behavior to the earlier experiments, using a random noise feature produced

TABLE IV  
SPEARMAN CORRELATION BETWEEN THE AUROC SCORE AND THE 90TH PERCENTILE OF SHAP VALUE DISTRIBUTIONS ACROSS TESTED FEATURES IN CICIDS2017. THE RESULTS COMPARE TWO BASE FEATURE CONFIGURATIONS: A SYNTHETIC RANDOM NOISE FEATURE VERSUS A DISCRIMINATIVE FEATURE (*Packet Length Variance*) USED AS THE BASE SET  $S$

SHAP Correlation with AUROC		
Feature	Spearman $\rho$	P-Value
<b>Anomaly samples (20 samples)</b>		
Random Noise	0.838	$2.36 \times 10^{-16}$
Packet Len Var	-0.211	0.116
<b>Normal samples (20 samples)</b>		
Random Noise	0.509	$4.44 \times 10^{-5}$
Packet Len Var	-0.070	0.603

notably different results. For this experiment, we selected  $S$  as a single random noise feature sampled uniformly between  $-1$  and  $1$ , adding it to the training/testing sets and the limited sets of 20 anomaly (attack) and normal (benign) samples.

Figures 2 and 4 present the results of this experiment using the 20 anomaly and 20 normal samples, respectively. Similar to the earlier setup, we tested over 50 features and provide results for the 10 features with the highest AUROC detection scores (rightmost) and the 10 features with the lowest AUROC detection scores (leftmost). As shown in Figure 2, features with relatively high AUROC results exhibit significantly negative Shapley value distributions (on the 20 anomaly samples), with negative median values. In contrast, features associated with relatively low AUROC results either displayed positive Shapley value distributions for anomaly samples (indicating insufficient discrimination) or exhibited no significant contribution compared to the random noise feature. This suggests that these features have minimal negative or positive impact on detection.

Importantly, the random noise feature maintained an insignificant Shapley value distribution (centered around zero with low standard deviation), enabling meaningful comparisons between the Shapley value distributions of the tested features.

To quantify this observation, we computed the *Spearman correlation* between the AUROC score of models trained with each feature and the 90th percentile of its SHAP value distribution, across sets of  $k = 20$  samples. We performed this analysis on both normal and anomaly samples from the CICIDS2017 dataset.

Table IV summarizes the correlation results for using the random noise as the base feature versus using a discriminative feature such as *Packet Length Variance* as the base feature.

As shown, there is a significant positive correlation between SHAP distributions and classifier performance when using the random noise baseline.

We observed similar results when performing the same experiment on the ISCXTor2016 dataset, with the random noise feature as the base feature (see Figure 5).



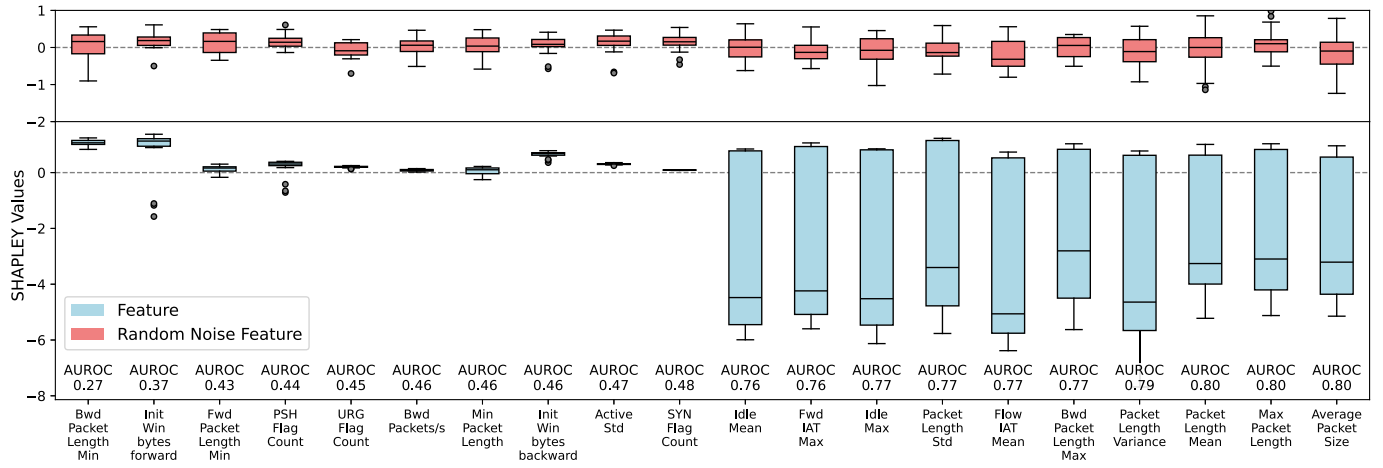


Fig. 2. The box plot visualizes the distribution of marginal Shapley values computed on 20 anomaly samples for various features from the CICIDS2017 dataset, each trained alongside a random noise feature. The plot demonstrates a clear correlation between features with predominantly negative Shapley value distributions and detection performance on the full testing set. The rightmost ten features have the highest AUROC scores, while the leftmost ten features have the lowest AUROC scores. In total, over 50 different features were evaluated.

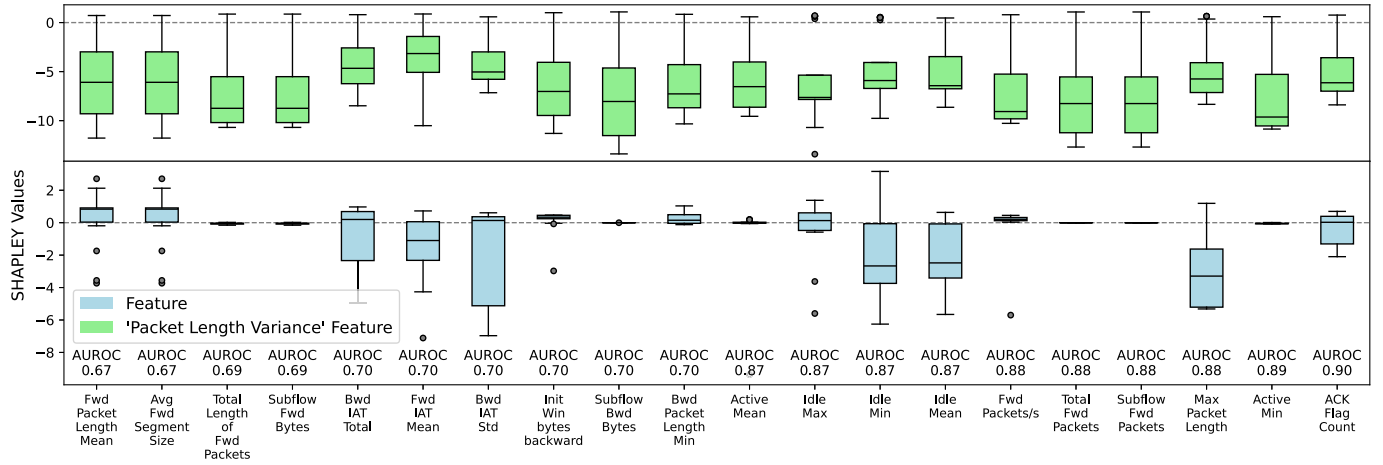


Fig. 3. The box plot visualizes the distribution of marginal Shapley values computed on 20 anomaly samples for various features from the CICIDS2017 dataset, each trained alongside the 'Packet Length Variance' feature. The plot illustrates how the marginal Shapley values of 'Packet Length Variance' interact differently with the tested features across various trained models.

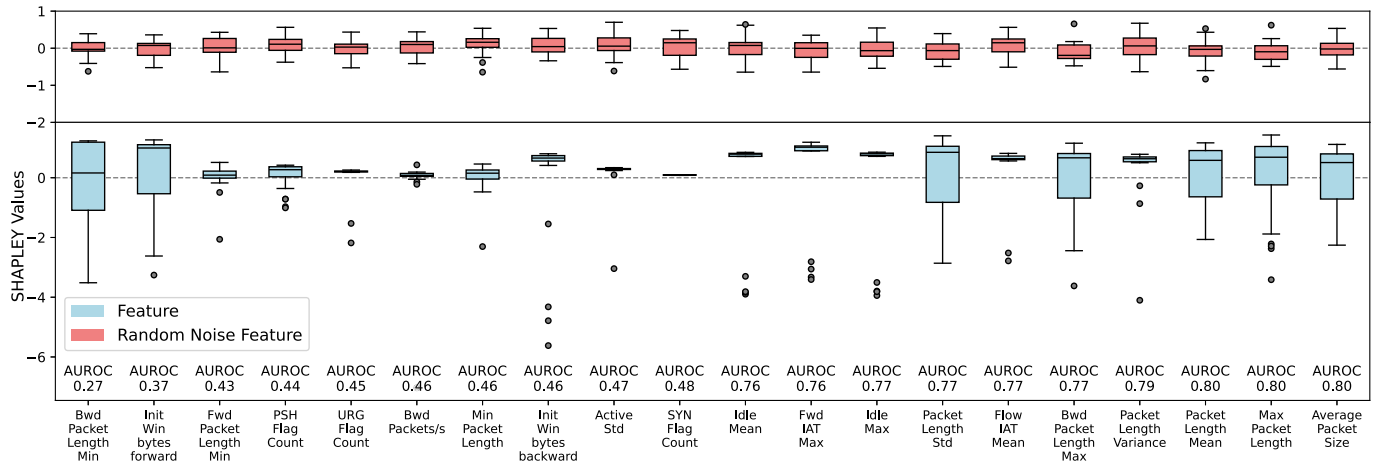


Fig. 4. The box plot visualizes the distribution of marginal Shapley values computed on 20 benign samples for various features from the CICIDS2017 dataset, each trained alongside a random noise feature. The plot illustrates how well each feature aligns with and represents the benign traffic characteristics in the trained models.

These results support the use of summary statistics of anomaly and normal samples—as a practical criterion for SHAP value distributions—computed on a small number of ranking features. For example, we found that the 90th quantile

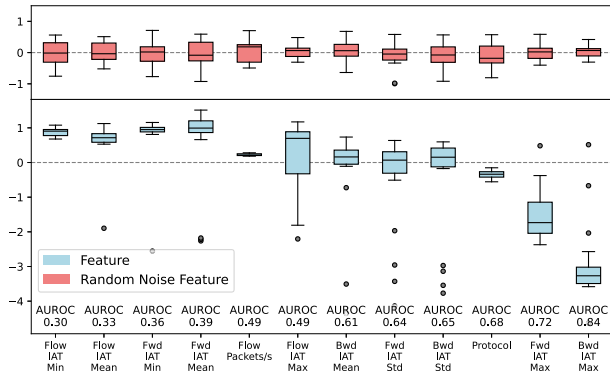


Fig. 5. The box plot visualizes the distribution of marginal Shapley values computed on 20 anomaly samples for various features from the ISCXTor2016 dataset, each trained alongside a random noise feature. The plot demonstrates a clear correlation between features with predominantly negative Shapley value distributions and AUROC score on the full testing set.

of SHAP values for anomaly and benign samples correlates strongly with the AUROC scores of corresponding models. This enables effective identification of discriminative features for anomaly detection, and facilitates the exclusion of features that exhibit minimal or misleading contributions to the detection task.

**Transductive Approach.** In the second approach, we have a mixed set of random samples without their labels. Since we do not know their labels, we cannot compare Shapley values of anomalies and normal samples across this set. For each feature, we compute the mean absolute Shapley value for all samples in the set. This overall score, termed the *Transductive Shapley Value*, is expected to be relatively high for features that can effectively separate normal samples from anomalies:

$$\text{Transductive Shapley Value}_i = \frac{1}{N} \sum_{j=1}^N |\phi_{i,j}|$$

where:

- $\phi_{i,j}$  is the Shapley value of the  $i$ -th feature for the  $j$ -th sample.
- $N$  is the total number of samples.

Here, we demonstrate the performance of our feature selection approach on the ISCXTor2016 dataset. We selected a set of five features: ‘Bwd IAT Std’, ‘Active Max’, ‘Flow Packets/s’, ‘Idle Min’, ‘Flow IAT Max’ which are not highly correlated, and tested the AUROC results of training a normalizing flow model using combinations of two features out of this set. The results are shown in Figure 6.

Additionally, we trained a NF model using these five features, and calculated both Transductive Shapley scores, with a random sampling of 100 Tor samples and 100 non-Tor samples out of the testing sets.

The Transductive Shapley Scores are presented in Table V. The feature ‘Bwd IAT Std’ stands out with the highest score of 1.44, followed by ‘Flow IAT Max’ with a score of 1.04. This approach confirms the importance of these features in

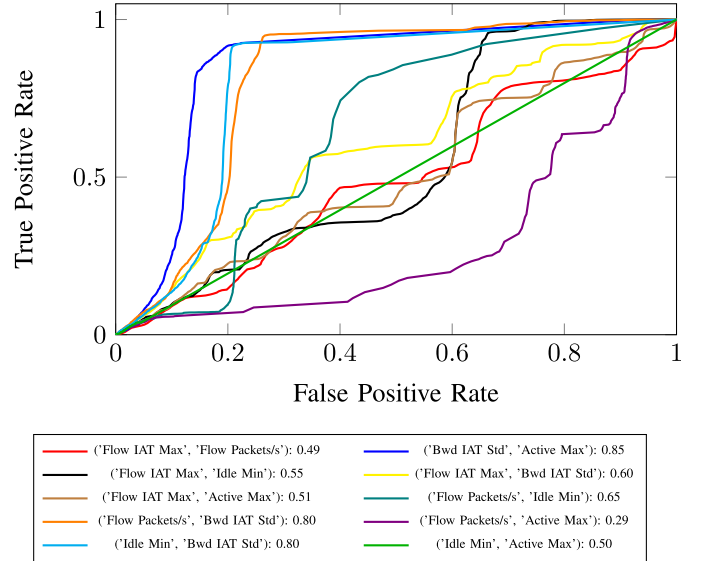


Fig. 6. AUROC value of the ROC curve of each NF model that was trained on a pair of features using the ISCXTor2016. The ROC curve of the ‘Bwd IAT Std’ and ‘Active Max’ features achieves the best AUROC score of 0.85.

TABLE V  
OUR SHAPLEY SCORES FOR DIFFERENT FEATURES

Feature	Transductive Shapley Score
Bwd IAT Std	1.44
Active Max	0.42
Flow Packets/s	0.19
Idle Min	0.42
Flow IAT Max	1.04

capturing the underlying characteristics of the data, even without labeled anomalies.

Figure 6 illustrates the ROC curves for various pairs of features. The AUROC values for each pair are annotated in the legend. Notably, the pair (‘Bwd IAT Std’, ‘Active Max’) achieves the highest AUROC score of 0.85. This aligns with our Shapley score analysis, further validating the effectiveness of these features in distinguishing normal traffic from Tor traffic. Other pairs, such as (‘Flow Packets/s’, ‘Bwd IAT Std’) and (‘Idle Min’, ‘Bwd IAT Std’), also show strong performance with AUROC scores of 0.80.

Our analysis using the Transductive approach suggests that some features (e.g., ‘Bwd IAT Std’ and ‘Active Max’) have the potential to contribute significantly to effective anomaly detection in the ISCXTor2016 dataset. The ROC curves strengthen these findings, demonstrating high AUROC scores for pairs of features that include ‘Bwd IAT Std’.

#### D. Performance Evaluation Results

**ISCXTor2016:** In Table VI, we compare the AUROC of our method with various state-of-the-art methods on the ISCXTor2016 dataset. We compare our results against state-of-the-art anomaly detection methods, including distribution learning-based methods [43], [44], [45], reconstruction-based methods [40], [44], knowledge distillation-based methods [41],

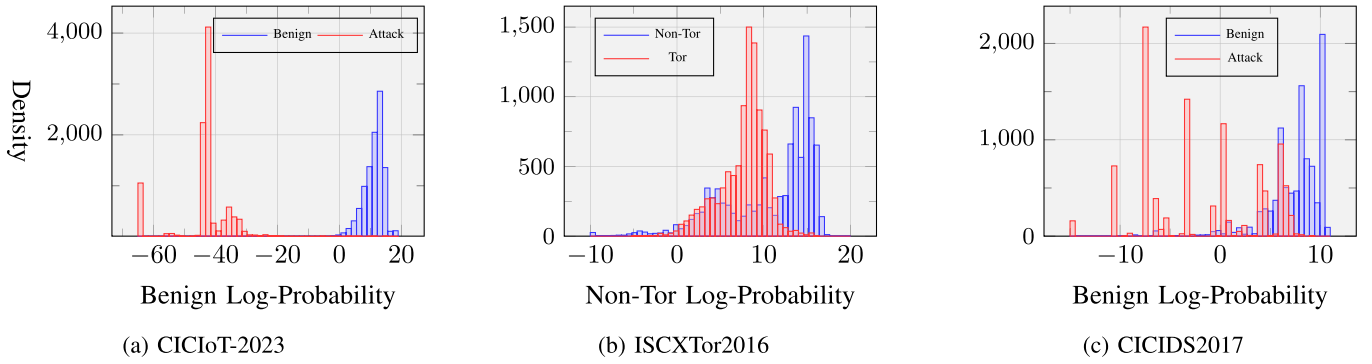


Fig. 7. Our NF detection method results in histograms showing the distribution of normal and anomaly samples in the testing sets of the three datasets we evaluated.

TABLE VI

AUROC COMPARISON AGAINST STATE-OF-THE-ART ANOMALY DETECTION METHODS [9] ON THE ISCXTOR AND NON-TOR DATASETS

Dataset: ISCXTor-2016			
Method	Training		AUROC
	Non-Tor	Tor	
GNomaly [40]	10,000	0	0.7823
CFlow[27]	10,000	0	0.7025
STFPM [41]	10,000	0	0.7371
PatchCore[42]	10,000	0	0.7434
PADIM [43]	10,000	0	0.7516
FastFlow[28]	10,000	0	0.6689
DRAEM [44]	10,000	0	0.7028
DFM[45]	10,000	0	0.7514
DFKDE[46]	10,000	0	0.7356
Reverse Distillation[47]	10,000	0	0.7450
Kitsune[48]	10,000	0	0.7800
Dang et al. [9]	10,000	0	0.8458
<b>Ours</b>	10,000	0	<b>0.8731</b>

[47], normalization flow-based methods [27], [28], and memory matching-based methods [42] as reported by Dang et al. [9]. In addition, we compare our results with Kitsune [48] by replicating their method using the publicly available implementation provided by the authors.

As shown in Table VI, our method achieves the highest AUROC of 0.8731, significantly outperforming the state-of-the-art methods. Specifically, our method outperforms the recent work by Dang et al. [9], which also utilizes NF for the generation of synthetic anomalies. Note that both flow-based methods of [9] and our method exceed the highest result among the other state-of-the-art methods by 8.12% and 11.6%, respectively. A visual representation of the anomaly score distribution of our method on ISCXTor2016 appears in Figure 7b. Here, the non-Tor and Tor classes are less separated than the normal and attack traffic on CICIOT-2023 (see Figure 7a). However, these results are still better than those achieved by other methods.

Our method's best results are even higher, with an AUROC of 0.932, when using only two features, one of which is the 'Protocol Type' field of the flow. However, since we found this feature as highly correlated with the Tor class, we focused on experiments excluding the 'Protocol Type' feature to ensure unbiased evaluation. The best reported results are achieved by

TABLE VII

AUROC PERFORMANCE OF OUR METHOD AGAINST STATE-OF-THE-ART METHODS ON CICIDS2017 DATASET

Dataset: CIC-IDS-2017			
Method	Training		AUROC
	Benign	Attack	
OCSVM[18]	1,590,881	0	0.7684
Kitsune[48]	10,000	0	0.8500
AE[18]	1,590,881	0	0.8758
MemAE[18]	1,590,881	265,817	0.9101
SparseMemAE [18]	1,590,881	265,817	0.8961
<b>Ours</b>	10,000	0	<b>0.9303</b>

an ensemble of two NF models. The first model is trained using the 'Flow IAT Std' and 'Flow Bytes/s' features, while the second model is trained using the 'Flow Packets/s' and 'Bwd IAT Max' features. When using an ensemble of more than one NF-based classifier, a flow is classified as an anomaly only if all classifiers in the ensemble detect it as an anomaly. To calculate the joint ROC and the corresponding AUROC, we normalize the detection thresholds and select the minimum score for each sample. It is noteworthy that the best AUROC result for a single NF classifier is 0.851, achieved using only the 'Bwd IAT Std' and 'Active Mean' features.

**CICIDS2017:** To evaluate the effectiveness of our approach on the CICIDS2017 dataset, we compare our results against several competitive methods, including OCSVM (one-class SVM), AE (autoencoder), MemAE [18] (memory-augmented autoencoder), SparseMemAE [18], and Kitsune [48] (ensemble of AEs). It is important to note that OCSVM, AE and Kitsune are trained using only normal samples, while MemAE and SparseMemAE utilize a limited number of attack samples, employing a semi-supervised approach. The comparison results are presented in Table VII. As shown, the OCSVM model exhibits low AUROC performance against DDoS attacks, while the AE-based models demonstrate overall high performance. Our proposed method outperforms the AE-based methods, achieving an AUROC of 0.93 for DDoS attack detection on the CICIDS2017 dataset.

Our best results were obtained using a NF model trained on a set of five features: 'Bwd Packet Length Mean', 'Fwd Packets/s', 'ACK Flag Count', 'Total Length of Bwd Packets',

TABLE VIII

PERFORMANCE OF OUR METHOD AGAINST STATE-OF-THE-ART METHODS ON CICIOT-2023 DATASET. OUR METHOD IS UNSUPERVISED, TRAINED USING NORMAL TRAFFIC ONLY, WHILE THE COMPARED METHODS ARE SUPERVISED METHODS

Dataset: CICIOT-2023						
Method	Training		Performance Metrics			
	Benign	Attack	Recall	Pr.	F1-Score	Acc.
Ours	10K	0	0.9903	1.0000	0.9951	0.9951
RF[11]	800K	36M	0.9651	0.9654	0.9653	0.9968
Adaboost[11]	800K	36M	0.9473	0.9656	0.9563	0.9959
DNN[11]	800K	36M	0.9333	0.9476	0.9403	0.9944
LR[37]	8.45K	8.45K	0.9834	0.9837	0.9834	0.9843
XGBoost	10K	10K	0.9865	0.9984	0.9932	0.9925

and ‘Flow Duration’. These features encompass TCP flag counts, packet length distribution in the downstream direction, and attack packets’ rate in the upstream direction.

**CICIOT-2023:** In Table VIII, we present the performance of our method compared to various state-of-the-art methods on the CICIOT-2023 dataset. To the best of our knowledge, our method is the first to evaluate the CICIOT-2023 dataset using only normal traffic data for training. Thus, all compared baseline methods are supervised.

Ghorbani et al. [11] applied several ML methods to evaluate the CICIOT-2023 dataset. Their results indicated that supervised methods like Adaboost, Random Forest (RF), and Deep Neural Network (DNN) achieved high performance, with accuracy metrics exceeding 98%. Additionally, Khan et al. [37] reported high detection performance using Logistic Regression (LR). However, it is important to note that our NF model, despite being unsupervised, achieves comparable or higher results than these supervised methods. Furthermore, we implemented an XGBoost [39] classifier and trained it on two random sets of 10,000 benign samples (the same training set as our method) and 10,000 attack samples (supervised method). We then evaluated its performance using the same testing sets used in our experiments.

In addition to the quantitative results, we also provide a visual representation of our method’s performance. As shown in Figure 7a, the histogram of the detection results illustrates the excellent capability of our NF model in distinguishing between normal and anomaly traffic on the CICIOT-2023 dataset.

Our NF model stands out as a highly effective unsupervised anomaly detection method. The results demonstrate that it can achieve or exceed the performance of leading supervised methods, making it a valuable tool for intrusion detection in IoT environments.

### E. Ablation Study

1) *Generalizing Across Datasets:* In realistic network deployments, anomaly detection models are likely to encounter benign traffic patterns that were not present during training. To assess the robustness of our approach in such scenarios, we evaluate its ability to distinguish malicious anomalies from previously unseen or rare benign traffic.

TABLE IX

DETECTION PERFORMANCE WHEN TRAINING AND TESTING ON DIFFERENT COMBINATIONS OF CICIDS2017 (BENIGN AND CYBERATTACKS TRAFFIC DATASET) AND CICDDoS2019 (BENIGN AND DDoS ATTACKS) DATASETS

Train		Test		Detection Result AUROC
DDoS2019	IDS2017	DDoS2019	IDS2017	
	✓		✓	0.93
	✓	✓		0.89
✓		✓	✓	0.93
✓			✓	0.88

While the CICIOT-2023 and CICIDS2017 datasets both include benign samples, their underlying feature sets differ significantly. Therefore, we conduct cross-dataset generalization experiments using two datasets with comparable feature sets: CICIDS2017 — one of the three datasets evaluated in this work — and CICDDoS2019, an additional dataset with a comparable feature set. Specifically, we train the model on the benign traffic of one dataset and evaluate it on the benign traffic of the other, ensuring that the testing benign samples are entirely unseen during training.

Table IX presents the results across different combinations of training and testing datasets. As shown, performance slightly degrades when generalizing across datasets compared to within-dataset evaluation. However, the detection results remain consistently high, demonstrating that our method is robust in distinguishing between attacks and previously unseen benign traffic.

2) *Using Additional XAI Methods:* To evaluate the generality of our proposed XAI-based feature selection framework, we investigate whether the few-shot approach—originally formulated with SHAPLEY-based explainer—can be applied with other explanation methods.

In this experiment, we replace Shapley values with LIME [49] (Local Interpretable Model-agnostic Explanations)—another widely used XAI method—and repeat the few-shot feature selection procedure. Specifically, we compute the importance scores of features for each anomaly sample and analyze their relationship with classifier performance. As in our original SHAPLEY-based method, each feature is tested by training NF model on it alongside a neutral baseline feature (random noise), and the 90th percentile of its marginal importance values over a small set of 20 attack or normal samples is recorded. We then measure the correlation between these values and the AUROC of the classifier trained using each feature.

As shown in Table X, both SHAP and LIME yield statistically significant positive correlations between importance values and detection performance when the baseline is a random noise feature. Conversely, when using a discriminative feature such as Packet Length Variance as the baseline, the correlations become insignificant, indicating the impact of feature interactions on XAI-based comparisons. These results are consistent with the SHAPLEY analysis in §V-C.

In addition, Figure 8 visually demonstrates that LIME yields similar explanation patterns to SHAP: features associated with high AUROC scores tend to have negative importance values



TABLE X

COMPARISON BETWEEN FEW-SHOTS FEATURE SELECTION APPROACH USING THE SHAP AND LIME XAI IMPORTANCE ATTRIBUTION METHODS. WE CALCULATE THE SPEARMAN CORRELATION BETWEEN THE 90TH QUANTILE DISTRIBUTIONS OF MARGINAL SHAP/LIME VALUES COMPUTED ON 20 ATTACK SAMPLES FOR EACH FEATURE AND THE AUROC SCORE OF THE CLASSIFIER USING THAT FEATURE

Correlation between feature importance values and classifier AUROC				
Feature	SHAP Spearman Correlation	SHAP P-Value	LIME Spearman Correlation	LIME P-Value
Anomaly samples (20 samples)				
Random Noise	0.838	2.36e-16	0.594	8.65e-07
Packet Len Var	-0.211	0.116	-0.040	0.752
Normal samples (20 samples)				
Random Noise	0.509	4.44e-05	0.533	1.65e-05
Packet Len Var	-0.070	0.603	-0.060	0.656

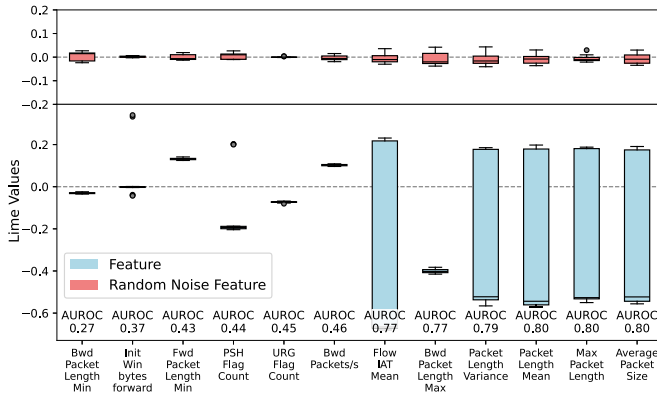


Fig. 8. The box plot shows the distribution of marginal LIME values for 20 anomaly samples from CICIDS2017, each trained alongside a random noise feature. As in Fig. 2 (SHAP-based), features with predominantly negative LIME value distributions correlate with high AUROC scores.

across the anomaly samples, while weak features exhibit dispersed or positive distributions. This behavior supports our hypothesis that discriminative features produce more consistent and interpretable attribution patterns when evaluated in isolation with a controlled baseline.

3) *Ablating Few-Shots Sample Size*: The Few-Shots feature selection technique leverages a *limited* number of anomaly samples and their corresponding SHAPLEY value distributions to identify features with strong discriminative power. In this subsection, we ablate the size of the anomaly and normal sample sets used in this procedure to evaluate the impact of different few-shot sample sizes on the stability and reliability of the feature selection results.

To this end, we repeat the procedure described in §V-C using varying sample sizes  $k \in \{10, 20, 50, 100, 200\}$  for both anomaly and normal samples, drawn from the CICIDS2017 dataset. For each value of  $k$ , we compute the Spearman and Pearson correlations (along with their associated  $p$ -values) between the AUROC score of each feature and the 90th percentile of the marginal XAI values computed using SHAP. In addition, we replicate the same experiment using the LIME method as described in §V-E.2.

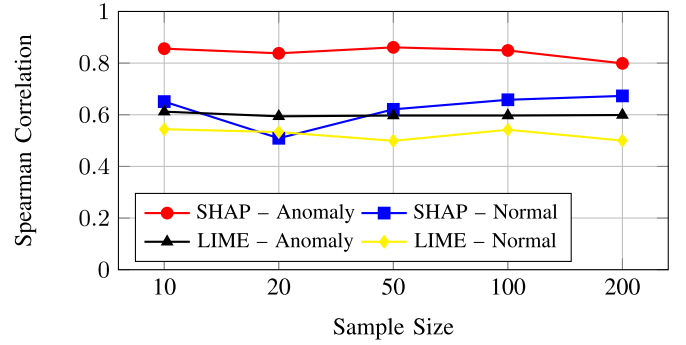


Fig. 9. Stability of few-shots feature selection results across varying sample sizes.

Table XI and Figure 9 present the correlation results across the different sample sizes. As shown, all evaluated values of  $k$ , including the smallest sizes (e.g.,  $k = 10$ ), yield consistently high and statistically significant correlations when the base feature is a random noise feature. Conversely, when using a strong discriminative feature such as *Packet Length Variance* as the baseline, the correlations become negligible or insignificant, regardless of sample size.

These findings confirm that our few-shots feature selection framework remains effective even when applied to small sets of labeled examples.

## VI. EXPLAINABILITY

In this work, we are interested not only in enhancing detection performance, but also in understanding what makes a low likelihood sample an anomaly and how it deviates from normal behavior. Thus, we combine the NF classifier with Shapley values, to provide statistically grounded explanations for detected anomalies, potentially improving the root-cause analysis of anomalous traffic.

While Shapley values can be used with a variety of methods, such as Random Forests, Gradient Boosting Machines, and Neural Networks, treating them as models with unknown internal logic, it attributes to each feature the change in the expected model prediction when conditioning on that feature. This means that Shapley values essentially explain the *model* prediction but not necessarily why a sample deviates from a set of samples statistically. Furthermore, the Shapley values are influenced by the internal mechanisms and nature of the model, meaning that the explanations may reflect the model's complexity rather than the true cause of the deviation.

Here, we leverage the statistical nature of the NF likelihood predictions with Shapley values. This approach provides a statistical importance measure to the Shapley values, producing explanations that elucidate why a particular traffic sample is anomalous and how it deviates from normal patterns.

**Simulation.** We start with a comparison of using Shapley values with NF and two other methods: one supervised (XGBoost) and another unsupervised (OCSVM). We create a simple simulation of normal and anomaly samples with two features:  $F_1$  and  $F_2$ .

Figure 10 illustrates a set of normal samples (blue points) and anomaly samples (red points). For a comparison, we

TABLE XI

CORRELATION VALUES AND P-VALUES BETWEEN THE 90TH PERCENTILE OF THE IMPORTANCE VALUE DISTRIBUTIONS (FROM SHAP AND LIME) AND AUROC CLASSIFICATION SCORE, FOR TWO NEUTRAL FEATURES (RANDOM NOISE AND PACKET LENGTH VARIANCE) ACROSS VARYING SAMPLE SIZES

Feature	Correlation Metric		SHAP					LIME				
			Sample size					Sample size				
			10	20	50	100	200	10	20	50	100	200
Attack	Random Noise	Pearson	0.873	0.871	0.868	0.860	0.840	0.608	0.541	0.555	0.556	0.555
		Spearman	0.856	0.838	0.861	0.849	0.799	0.612	0.594	0.597	0.597	0.599
		P-Value	1.07E-17	2.36E-16	4.47E-18	3.76E-17	5.68E-14	3.32E-07	8.65E-07	7.51E-07	7.64E-07	6.83E-07
	Packet Length Variance	Pearson	0.083	-0.159	0.000	-0.046	-0.058	-0.062	-0.097	-0.044	-0.049	-0.030
		Spearman	0.122	-0.211	-0.029	0.029	-0.073	-0.041	-0.040	-0.018	-0.019	-0.002
		P-Value	0.365	0.116	0.833	0.830	0.590	0.762	0.752	0.895	0.891	0.989
Benign	Random Noise	Pearson	0.746	0.499	0.667	0.724	0.697	0.505	0.499	0.478	0.476	0.484
		Spearman	0.651	0.509	0.621	0.658	0.673	0.544	0.533	0.499	0.542	0.500
		P-Value	3.17E-08	4.44E-05	1.98E-07	1.33E-10	7.23E-09	1.03E-05	1.65E-05	6.78E-05	1.13E-05	6.39E-05
	Packet Length Variance	Pearson	-0.181	-0.201	-0.062	-0.073	-0.122	-0.239	-0.147	-0.207	-0.139	-0.132
		Spearman	-0.155	-0.070	-0.099	-0.092	-0.139	-0.068	-0.060	-0.123	-0.050	-0.059
		P-Value	0.250	0.603	0.465	0.496	0.302	0.614	0.656	0.361	0.711	0.665

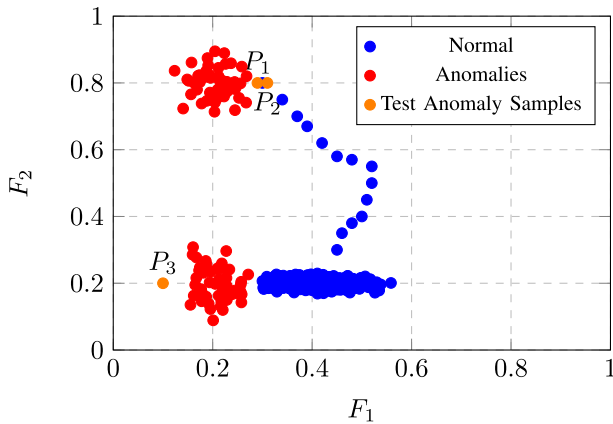


Fig. 10. A simulation of normal and anomaly samples with two features.

train two unsupervised classifiers (NF and OCSVM) on the blue samples only, and an additional XGBoost classifier is trained on both blue and red samples (supervised). Then, we select three points:  $P_1$ ,  $P_2$ , and  $P_3$  and compute their corresponding Shapley values using the three methods. Note that  $P_1$ ,  $P_2$  deviate from the majority of normal samples more pronouncedly in  $F_2$  than in  $F_1$ . Additionally,  $P_1$  has an  $F_1$  value lower than the minimum  $F_1$  value of all normal samples.  $P_3$  deviates mostly in  $F_1$ .

For a binary classifier, Shapley values represent the contribution of each feature to the model's output. The sign of the Shapley value indicates whether the feature's presence increases or decreases the likelihood of the sample being classified as an anomaly.

Table XII presents the Shapley values for points  $P_1$ ,  $P_2$ , and  $P_3$  as computed by the three different models: NF, OCSVM, and XGBoost. It is evident from the table that while OCSVM was trained on the same data as NF, its corresponding Shapley values do not provide any useful explanation for  $P_1$  and  $P_2$ . Specifically, both features  $F_1$  and  $F_2$  have the same Shapley value of  $-0.89$  for these points. While OCSVM is designed to find a boundary that encloses the majority of the data points of the learned class, a few points within the training data might affect the placement of the boundary, which in

TABLE XII

COMPARISON OF SHAPLEY VALUES ACROSS THREE DIFFERENT METHODS: XGBOOST, OCSVM, AND NF. THE SHAPLEY VALUES INDICATE THE CONTRIBUTION OF EACH FEATURE TO THE PREDICTION OF THREE SAMPLES:  $P_1$ ,  $P_2$ , AND  $P_3$  (SEE FIGURE 10), EACH WITH TWO FEATURES,  $F_1$  AND  $F_2$ . POSITIVE SHAPLEY VALUES INDICATE THAT THE FEATURE CONTRIBUTES POSITIVELY TO THE PREDICTION (NORMAL), WHILE NEGATIVE VALUES INDICATE A NEGATIVE CONTRIBUTION (ANOMALY)

Sample	Shapley Values					
	XGBoost		OCSVM		NF	
	$F_1$	$F_2$	$F_1$	$F_2$	$F_1$	$F_2$
$P_1$	0.323	0	-0.89	-0.89	-7.787	-9.382
$P_2$	-1.916	0	-0.89	-0.89	-3.313	-9.878
$P_3$	-0.951	0	-1.83	0.05	-3.4e38	0

this particular example, also affects the Shapley values of anomalous samples.

On the other hand, the Shapley values computed by the XGBoost model do not attribute any contribution to  $F_2$  for any of the points, failing to recognize the significance of  $F_2$  in differentiating between normal and anomalous samples (indicating that XGBoost operates solely on  $F_1$ ). This highlights the limitations of using XGBoost for this particular anomaly detection task. NF, however, offers a more nuanced explanation by assigning different Shapley values to  $F_1$  and  $F_2$  for each point, which reflects the true nature of their deviations.

**Using Real-World Attack Samples.** Next, we analyze Shapley values with NF on attack samples from CICIOT-2023. Two classifiers were trained using five features: 'HTTPS', 'Protocol Type', 'Magnitude', 'Variance', and 'fin\_count'. Our NF model was trained on 10,000 normal samples, while the XGBoost baseline used the same 10,000 normal samples plus 10,000 attack samples (supervised).

To obtain test anomalies with expected explanations serving as ground truth, we constructed subsets of attack samples termed *X-Feature Deviant Attack Samples*. In each subset, four features follow the distribution of normal samples, while one feature ('X') takes values unlikely under the normal distribution. To quantify this deviation, we compute the mean likelihood of each feature using the normal training set (10,000

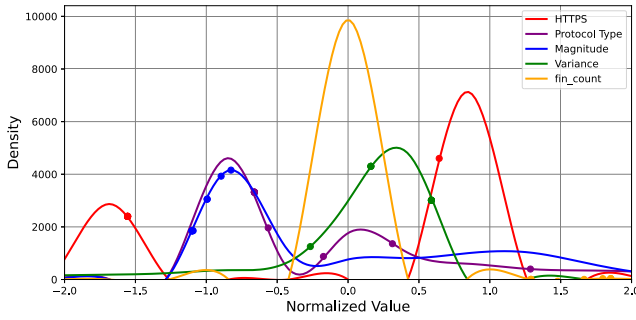


Fig. 11. CICIOT-2023 ‘fin\_count’ feature deviant attack samples illustration. We plot the normalized distributions of five features over 10,000 normal samples. The points represent the feature values of the attack samples.

TABLE XIII

COMPARISON BETWEEN SHAPLEY VALUES USING OUR NF MODEL VS. XGBOOST ON REAL-WORLD CICIOT-2023 ATTACK SAMPLES. AS A GROUND-TRUTH WE USE SUBSETS OF ATTACK SAMPLES, IN WHICH ONLY ONE FEATURE EXHIBITS VALUES THAT ARE UNLIKELY TO BE PRESENT IN THE NORMAL DISTRIBUTION. NOTE THAT LOW NF SHAPLEY VALUES CORRELATE WITH THE DEVIANT FEATURES

Shapley Values on CICIOT-2023 Attack Samples			
Feature	Ground Truth Mean Likelihood	XGboost Mean SHAP	NF Mean SHAP
<b>‘Variance’ feature deviant attacks (91 samples)</b>			
HTTPS	0.761	-0.073	-2.764
Protocol Type	0.397	-0.099	-7.994
Magnitude	0.461	<b>-0.550</b>	-0.086
<b>Variance</b>	<b>0.024</b>	-0.132	<b>-11.751</b>
fin_count	4.488	-0.001	-0.707
<b>‘fin_count’ feature deviant attacks (10 samples)</b>			
HTTPS	0.834	-0.041	-1.554
Protocol Type	0.840	-0.300	0.159
Magnitude	0.659	<b>-0.407</b>	-0.771
Variance	0.871	-0.073	0.131
<b>fin_count</b>	<b>0.002</b>	0.014	<b>-10.213</b>

samples). Likelihoods are estimated with Gaussian Kernel Density Estimation (Gaussian KDE), a non-parametric method for probability density estimation.

As an example, Figure 11 shows the distribution of each feature in the normal training set (10,000 samples). For each attack sample in the ‘fin\_count’ deviant subset, the normalized feature value is placed on the corresponding distribution curve. The ‘fin\_count’ values exhibit very low likelihood, whereas the other four features remain relatively likely.

We then examine the Shapley values obtained from these two classifiers. For each such ground-truth sample, we expect that the Shapley value will attribute a relatively high value to the feature that deviates from the normal distribution, with a negative sign indicating that this feature decreases the likelihood of the sample being detected as normal. Table XIII shows the results on two subsets of feature deviant attacks (i.e., ‘Variance’ with 91 samples and ‘fin\_count’ with 10 samples), together with the computed likelihood which is used as a ground-truth measure.

As shown in Table XIII, the XGBoost Shapley values always attribute the ‘Magnitude’ feature as the most domi-

nant feature for all ground-truth samples. In the ‘fin\_count’ ground-truth subset, the XGBoost mean Shapley value even has a positive sign, indicating that this feature increases the likelihood of being detected as normal despite the very low likelihood of this feature across the normal samples (0.002). On the other hand, the NF Shapley values clearly highlight the “deviant” feature as the one causing the test sample to be detected as an anomaly, underscoring the significant improvement in explainability when using our method.

Our analysis demonstrates the potential of combining NF with Shapley values to enhance network traffic anomaly detection explainability. While our findings include anecdotal evidence and specific case studies, the promising results indicate a significant opportunity for future work to further validate and expand upon this approach in broader contexts.

## VII. CONCLUSION

In this study, we demonstrated the efficacy of NF for anomaly detection in network traffic. By leveraging the density estimation capabilities of NF, our approach effectively identifies low-likelihood samples, marking them as potential anomalies. This method avoids the need for labeled anomaly data during training, which is a significant advantage in real-world scenarios where such data is often unavailable.

Our experiments on the CICIOT-2023, ISCXTor2016, and CICIDS2017 datasets show that our model achieves high accuracy, outperforms existing state-of-the-art methods for unsupervised anomaly detection, and, when combined with Shapley values, supports feature selection and provides valuable explanations for detected anomalies. While our method effectively models the distribution of normal traffic within each dataset, we note that performance slightly degrades when training on one dataset and evaluating on another, reflecting distribution shifts between environments. Addressing such generalization challenges is an important direction for future work, for example through self-supervised pretraining techniques. Overall, our results indicate that NF offer a powerful and interpretable solution for anomaly detection in network traffic.

## REFERENCES

- [1] E. Bout, V. Loscri, and A. Gallais, “How machine learning changes the nature of cyberattacks on IoT networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 248–279, 1st Quart., 2021.
- [2] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, “An overview of IP flow-based intrusion detection,” *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 343–356, 3rd Quart., 2010.
- [3] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. 4150, Jan. 2021.
- [4] S. Gopalakrishnan, N. Tuptuk, and S. Hailes, “Machine learning-based intrusion detection systems: Deployment guidelines for industry,” PETRAS National Centre of Excellence for IoT Systems Cybersecurity, London, U.K., Tech. Rep., 2023, doi: [10.14324/0000rp.10190465](https://doi.org/10.14324/0000rp.10190465). [Online]. Available: <https://discovery.ucl.ac.uk/id/eprint/10190465/>
- [5] I. Goodfellow et al., “Generative adversarial nets,” in *Proc. NIPS*, 2014, pp. 2672–2680.
- [6] A. S. Dina, A. B. Siddique, and D. Manivannan, “Effect of balancing data using synthetic data on the performance of machine learning classifiers for intrusion detection in computer networks,” *IEEE Access*, vol. 10, pp. 96731–96747, 2022.

- [7] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
- [8] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1530–1538.
- [9] Z. Dang, Y. Zheng, X. Lin, C. Peng, Q. Chen, and X. Gao, "Semi-supervised learning for anomaly traffic detection via bidirectional normalizing flows," 2024, *arXiv:2403.10550*.
- [10] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4765–4774.
- [11] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.
- [12] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 253–262.
- [13] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.
- [14] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proc. DARPA Inf. Survivability Conf. Expo.*, Apr. 2003, pp. 303–314.
- [15] P. D. Bojović, I. Bašičević, S. Ocovaj, and M. Popović, "A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method," *Comput. Electr. Eng.*, vol. 73, pp. 84–96, Jan. 2019.
- [16] S. Naseer et al., "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
- [17] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.
- [18] B. Min, J. Yoo, S. Kim, D. Shin, and D. Shin, "Network anomaly detection using memory-augmented deep autoencoder," *IEEE Access*, vol. 9, pp. 104695–104706, 2021.
- [19] S. Azmin and A. M. A. A. Islam, "Network intrusion detection system based on conditional variational Laplace autoencoder," in *Proc. 7th Int. Conf. Netw. Syst. Security*, 2020, pp. 82–88.
- [20] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, "Practical GAN-based synthetic IP header trace generation using NetShare," in *Proc. ACM SIGCOMM Conf.*, Aug. 2022, pp. 458–472.
- [21] M. Abdelaty, S. Scott-Hayward, R. Doriguzzi-Corin, and D. Siracusa, "GADoT: GAN-based adversarial training for robust DDoS attack detection," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2021, pp. 119–127.
- [22] B.-E. Zolbayar et al., "Generating practical adversarial network traffic flows using NIDSGAN," 2022, *arXiv:2203.06694*.
- [23] J. Wang, J. Pan, I. AlQerm, and Y. Liu, "Def-IDS: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–9.
- [24] Y. Peng, G. Fu, Y. Luo, J. Hu, B. Li, and Q. Yan, "Detecting adversarial examples for network intrusion detection system with GAN," in *Proc. IEEE 11th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Oct. 2020, pp. 6–10.
- [25] P. Zhang et al., "Real-time malicious traffic detection with online isolation forest over SD-WAN," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2076–2090, 2023.
- [26] L. Li, Y. Lu, G. Yang, and X. Yan, "End-to-end network intrusion detection based on contrastive learning," *Sensors*, vol. 24, no. 7, p. 2122, Mar. 2024.
- [27] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 98–107.
- [28] J. Yu et al., "FastFlow: Unsupervised anomaly detection and localization via 2D normalizing flows," 2021, *arXiv:2111.07677*.
- [29] M. L. D. Dias, C. L. C. Mattos, T. L. C. da Silva, J. A. F. de Macêdo, and W. C. P. Silva, "Anomaly detection in trajectory data with normalizing flows," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
- [30] A. Ryzhikov, M. Borisyak, A. Ustyuzhanin, and D. Derkach, "Normalizing flows for deep anomaly detection," 2019, *arXiv:1912.09323*.
- [31] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016, *arXiv:1605.08803*.
- [32] S. Zhai et al., "Normalizing flows are capable generative models," 2024, *arXiv:2412.06329*.
- [33] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3617–3621.
- [34] OpenAI. (Jul. 2018). *Glow: Better Reversible Generative Models*. [Online]. Available: <https://openai.com/index/glow>
- [35] P. N. Ward, A. Smofsky, and A. J. Bose, "Improving exploration in soft-actor-critic with normalizing flows policies," 2019, *arXiv:1906.02771*.
- [36] D. Fryer, I. Strumke, and H. Nguyen, "Shapley values for feature selection: The good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144352–144360, 2021.
- [37] M. M. Khan and M. Alkhathami, "Anomaly detection in IoT-based healthcare: Machine learning for enhanced security," *Sci. Rep.*, vol. 14, no. 1, p. 5872, Mar. 2024.
- [38] J. F. Crenshaw, Z. Yan, and V. Doster, "jfcenshaw/pzflow: V3.1.2 (v3.1.2)," Zenodo, CERN, Tech. Rep., 2024, doi: [10.5281/zenodo.10636848](https://doi.org/10.5281/zenodo.10636848).
- [39] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
- [40] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2018, pp. 622–637.
- [41] G. Wang, S. Han, E. Ding, and D. Huang, "Student-teacher feature pyramid matching for anomaly detection," 2021, *arXiv:2103.04257*.
- [42] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14298–14308.
- [43] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proc. Int. Conf. Pattern Recognit.*, 2020, pp. 475–489.
- [44] V. Zavrtnik, M. Kristan, and D. Skočaj, "Draem—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 8330–8339.
- [45] N. A. Ahuja, I. Ndiour, T. Kalyanpur, and O. Tickoo, "Probabilistic modeling of deep features for out-of-distribution and adversarial detection," 2019, *arXiv:1909.11786*.
- [46] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc, "Anomalib: A deep learning library for anomaly detection," 2022, *arXiv:2202.08341*.
- [47] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9737–9746.
- [48] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," 2018, *arXiv:1802.09089*.
- [49] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 1135–1144.