# NF-NIDS: Normalizing Flows for Network Intrusion Detection Systems

Meryem Janati Idrissi
College of Computing
*Mohammed VI Polytechnic University*
Benguerir, Morocco
meryem.janati@um6p.ma

Hamza Alami
College of Computing
*Mohammed VI Polytechnic University*
Benguerir, Morocco
hamza.alami@um6p.ma

Abdelhak Bouayad
College of Computing
*Mohammed VI Polytechnic University*
Benguerir, Morocco
abdelhak.bouayad@um6p.ma

Ismail Berrada
College of Computing
*Mohammed VI Polytechnic University*
Benguerir, Morocco
ismail.berrada@um6p.ma

***Abstract*—The rising frequency and complexity of cyber threats have necessitated the development of effective Network Intrusion Detection Systems (NIDS). Anomaly-based detection approaches have gained prominence for their ability to detect unknown and sophisticated attacks. In this paper, we introduce NF-NIDS, a novel approach for anomaly-based network intrusion detection using Normalizing Flows (NFs) to accurately classify network traffic into normal or malicious categories given the assumption of the availability of scarce attack samples. To address the challenge of limited attack samples, we employ two flow-based models, namely Inverse Autoregressive Flow (IAF) and Neural Spline Flows (NSF). These models are used to learn the underlying distribution of normal traffic and generate pseudo-attacks from the tails of the distribution. To evaluate the effectiveness of NF-NIDS, we conducted experiments on three well-known network datasets. The results demonstrate that our approach achieves high performance levels while incurring low to negligible false discovery rates. Specifically, NF-NIDS yields impressive results, with an accuracy of 98.71% for USTC-TFC2016, 94.86% for CIC-IDS2017, and 98.20% for CIC-IDS2018. In terms of the F1-score, NF-NIDS scores 98.72% for USTC-TFC2016, 97.05% for CIC-IDS2017, and 99.51% for CIC-IDS2018.**

***Index Terms*—Network Intrusion Detection, Anomaly Detection, Normalizing Flows**

## I. Introduction

As cyberattacks become increasingly sophisticated and frequent, Network Intrusion Detection Systems (NIDS) [1] play a critical role in safeguarding sensitive data and networks alike. A NIDS is a tool or software that monitors network traffic for potentially malicious activities and alerts security teams if any suspicious behavior is detected. Traditionally, NIDS can be classified into two categories: signature-based (SNIDS) and anomaly-based detection systems (ANIDS). A SNIDS involves comparing specific attack patterns with a database of known signatures. While a SNIDS is effective in detecting known attacks, it falls short when dealing with novel or unknown cyber threats. On the other hand, an ANIDS learns the normal behavior of a system and identifies anything that deviates from it as an anomaly. ANIDS have shown great potential in detecting unknown and even zero-day attacks. However, it could be challenging to build a reliable baseline of normal behavior, which can lead to high false alarm rates. Moreover, the techniques used for ANIDS are often computationally expensive and may affect real-time detection performance. Overall, NIDS remains essential in the cybersecurity landscape to provide a defense against a wide range of cyber threats. The ongoing challenge lies in improving the efficiency and accuracy of anomaly-based systems while reducing false positives to make them more practical for real-world applications.

Generative models, particularly those based on deep learning and probabilistic modeling, have gained considerable attention in the field of anomaly-based intrusion detection. Instead of relying solely on predefined rules or known patterns, generative models have the ability to learn and recognize anomalous behavior based on the underlying distribution of normal data. This comes with the potential to detect new and sophisticated attacks that might not be identified by traditional signature-based methods. The prospects of generative models for IDS have led to extensive research in the era and great interest in exploring the potential of other generative models, in particular, Normalizing Flows (NFs). NFs [2] are a class of generative models that transform a simple probability distribution into a complex one by repeatedly applying invertible transformations to the input data. These models offer several advantages, such as the ability to model complex distributions, handle high-dimensional data, and support fast and exact likelihood computations. This is particularly advantageous in the context of anomaly detection (AD) since they can capture complex patterns of normal data that might not be effectively captured by other models. Additionally, NFs have a generative nature, enabling them to generate synthetic anomalies. By manipulating the learned distribution, NFs can create synthetic instances that mimic anomalous activities. These synthetic

979-8-3503-2967-4/23/$31.00 © 2023 IEEE

anomalies can be used for data augmentation, enhancing the model's ability to detect and generalize to different types of anomalies. NFs have also proven to be effective in the realm of semi-supervised learning, where the goal is to leverage both labeled and unlabeled data for training models. The aim is to improve the performance of machine learning algorithms when only a limited amount of labeled data is available, which is a common scenario in many real-world applications.

In this work, we propose using NFs as generative models for semi-supervised anomaly-based network intrusion detection. We recognize that labeled anomaly data is often limited or expensive to obtain, posing challenges to developing robust NIDS. By leveraging the generative modeling capabilities of NFs, our work aims to explore how semi-supervised learning can be utilized to effectively detect anomalies while making use of both labeled and unlabeled data. Moreover, we aim to investigate the use of synthetic data generated by NFs as a means of augmenting the labeled data, enhancing the model's ability to generalize to unseen anomalies.

**Paper contributions.** We broaden the scope of NIDS with the following contributions:

1) We introduce NF-NIDS, an anomaly-based intrusion detection method that leverages NFs for normal data distribution learning and data augmentation.
2) We design a semi-supervised learning scheme that leverages normal data with available real anomaly records along with pseudo-anomalies sampled using NFs to train a classifier.
3) We conduct experiments on a clean version of three prominent NIDS datasets where more reliable features were extracted using an updated version of CICFlowMeter[1]. The results show that the proposed method achieves excellent performance in terms of different metrics.

The rest of the paper is structured as follows: In Section II, existing studies on AD and NIDS are discussed. Section III provides background on normalizing flow models. Section IV introduces the proposed method and discusses its main components. The implementation details and the evaluation results are provided and discussed in Section V. Finally, we conclude the paper in Section VI.

## II. Related Work

Generative models have been widely used in the field of anomaly detection and have shown great potential for related tasks, in particular, intrusion detection. For instance, ZAVRAK et al. [3] proposed a method based on a variational autoencoder to detect anomalies in network traffic flows. The reconstruction error was used to classify network flows as anomalous or normal. By conducting experimentation on the CIC-IDS2017 dataset, the results are shown to be similar to AE and OCSVM models in terms of ROC metric. Another work introduced by Azmin et al. [4] combines a variational laplace autoencoder (VLAE) and a deep neural network (DNN) for intrusion detection. They improved the existing VLAE by incorporating class labels as input to the autoencoder. The new model named conditional variational laplace autoencoder (CVLAE) is then used to learn the latent variable while the DNN is employed as a classifier. Generative Adversarial networks have also been utilized for anomaly-based network intrusion detection. Li et al. [5] adopted Wasserstein GAN Divergence (WGAN-DIV) for data generation for each class in the dataset. Then, XGBoost was trained on the enhanced data and was used for detection. The use of VAE and GAN models for anomaly-based intrusion detection has been thoroughly investigated in the literature and has shown promising results. However, both approaches have their advantages and limitations. Many other works have proposed the use of VAE and GAN models for NIDS [6], [7], [8], [9], [10]. While variational autoencoders have the advantage of learning a probabilistic latent representation of the data, however, they suffer from intractable marginal likelihoods. GANs can generate more realistic samples but suffer from mode collapse and instability issues. Therefore, there is ongoing research and exploration of alternative generative models for anomaly detection, specifically focusing on the utilization of NFs. NFs attempt to merge the advantages of both VAE and GAN to estimate the exact likelihood of the data distribution and learn the feature representation.

NFs [11], [12], [13] have been widely used in the field of anomaly detection due to their ability to model complex probability distributions. Several studies have successfully utilized NFs for anomaly detection, showcasing promising results. The paper [14] by Ryzhikov et al. proposed a model-agnostic anomaly detection procedure based on NFs. The paper addressed the class-imbalanced classification by incorporating existing anomalous samples in the training phase and reformulating one-class classification as two-class classification. Experimental results show that NFAD outperforms existing AD methods. Dias et al. [15] proposed an unsupervised density estimation method for trajectory anomaly detection using NFs. The authors utilized RealNVP and masked autoregressive flow (MAF) to model the trajectory data and detect anomalies. The results of their experiments indicate that NFs outperformed classical density-based methods such as Local Outlier Factor (LOF) and Gaussian mixture models (GMM). Gudovskiy et al. [16] proposed CFLOW-AD, a method for anomaly detection with localization based on conditional NFs. CFLOW-AD uses NFs to describe the distribution of normal network-based features and estimates accurate data likelihood of the tested features. The proposed model is shown to be faster and smaller compared to prior models. However, CFLOW-AD requires a fine design that is different from vanilla CN-based models. Another work, introduced by Yu et al. [17], has also explored the use of NFs for anomaly detection and localization. The authors proposed FastFlow, a detection method that expands standard NFs to 2D space. A feature extractor is first used to extract visual features for normal samples, the features are next fed into a 2D flow model in order to estimate the probability distribution. FastFlow demonstrates improved performance in terms of both accuracy and efficiency compared to previous

[1]https://github.com/GintsEngelen/CICFlowMeter

methods. Very recently, Guo et al. [18] introduced RobestFlow, an unsupervised method for real-world wear detection and segmentation using NFs and attention mechanism. To mitigate the data-hungry issue, the proposed method only requires wear-free images for training. Additionally, RobestFlow combines the normalizing flow model and the SSPCAB attention mechanism. The former models the distribution of the image pixels, while the latter learns the global feature information of pixels.

Several works have been proposed for anomaly detection using NFs. However, our work method fills a gap in the existing literature on network intrusion detection systems by combining anomaly detection with flow-based models.

## III. Normalizing Flows

NFs are a class of generative models that leverage a series of invertible transformations to map the probability density of a random variable, denoted as $X$ to a well-known base distribution with a simple probability density, represented by $Z$. Let's denote the random variable of observed data as $x \in X$ and the latent variable as $z \in Z$. Given a simple base distribution $p_Z(z)$ and the probability density $p_X(x)$ over $x$, NFs aim to learn a mapping function $f$, such that, $z = f(x)$ and $x = f^{-1}(z)$. This transformation is achieved by composing a sequence of $K$ bijective transformations, i.e., $f = (f_1 \circ f_2 \circ ... \circ f_k)$. Once learned, NFs can be used in both directions. Density estimation can be performed using the forward mapping $X \rightarrow Z$ and sampling (synthetic data generation) can be achieved using the inverse mapping $Z \rightarrow X$. Using the change of variable formula, we can express the probability density of $x$ by:

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

where $\frac{\partial f(x)}{\partial x}$ and $det\left(\frac{\partial f(x)}{\partial x}\right)$ represent the Jacobian matrix of $f$ with respect to $x$ and the determinant of the Jacobian of $f$ respectively. $p_Z\big(f(x)\big)$ denotes the density of $x$ under the base distribution $p_Z$. The determinant of the Jacobian matrix can be numerically challenging to compute, particularly in high-dimensional spaces. Moreover, multiplying multiple determinants together can lead to numerical underflow or overflow issues. To address these challenges, taking the logarithm of the change of variable formula is a common practice. By applying the logarithm, the product of determinants is converted into a sum of logarithms, which is more numerically stable. Taking the logarithm of the change of variable formula yields:

$$log\, p_X(x) = log\, p_Z\big(f(x)\big) + log\left|det\left(\frac{\partial f(x)}{\partial x}\right)\right| \quad (2)$$

If the transformation $f$ is considered to be a composition of a sequence of $K$ successive series of transformations, then the relationship between the observable variable $x$ and the latent variable $z$ can be expressed as follows:

$$\begin{aligned} x = z_0 \xrightarrow{f_1} z_1 \xrightarrow{f_2} z_2 ... \xrightarrow{f_K} z_K = z \\ x = z_0 \xleftarrow{f_1^{-1}} z_1 \xleftarrow{f_2^{-1}} z_2 ... \xleftarrow{f_K^{-1}} z_K = z \end{aligned} \quad (3)$$

From 2 and 3, the log-density of $x$ can be expressed by:

$$log\, p_X(x) = log\, p_Z\big(f(x)\big) + \sum_{i=1}^{K} log\left|det\left(\frac{\partial f(z_i)}{\partial z_i}\right)\right| \quad (4)$$

In practice, $p_Z$ is commonly taken to be a known and simple base distribution (often a Gaussian distribution): $p_Z \sim \mathcal{N}(0, I)$. Furthermore, the negative sign is usually added to the equation 4 to get the negative log-likelihood loss (NLL) used to train the flow-based model. In this work, two NF models were evaluated, namely *Neural Spline Flow* and *Inverse Autoregressive Flow*.

### A. Inverse Autoregressive Flow

Inverse Autoregressive Flow (IAF) [19] is a type of normalizing flow that leverages an autoregressive structure in the latent space. In IAF, the flow is designed such that the transformation from the latent space to the data space is autoregressive. This means that each dimension of the latent variable is transformed sequentially, conditioning on the previously transformed dimensions. More specifically, let's consider a latent variable $z$ with dimensions $z_1, z_2, \ldots, z_d$. In IAF, the transformation from $z$ to the data variable $x$ is performed by the following recursion:

$$x_i = z_i \odot \sigma_i(z_{i:i-1}) + \mu_i(z_{i:i-1}) \quad (5)$$

Where $\sigma$ and $\mu$ are the scale and shift functions, respectively, computed from previous instances $z_{i:i-1}$. IAF has been shown to be effective in modeling complex and high-dimensional distributions. It can capture dependencies and intricate structures in the data distribution by leveraging autoregressive transformation.

### B. Neural Spline Flow

In Neural Spline Flow (NSF) [20], the transformation from $z$ to $x$ is constructed by composing a set of spline segments, each associated with a specific point called *knot*. The transformation function can be written as:

$$x = f^{-1}(z; \alpha, \beta) \quad (6)$$

$f(z; \alpha, \beta)$ is the forward transformation function of the NSF that maps the latent variable z to the data space. $\alpha$ and $\beta$ are the parameters of the spline function, representing the knot locations and spline parameters, respectively. The transformation can be efficiently computed using the spline functions, and the probability density function of $x$ can be evaluated using the change of variable formula. NSF offer a flexible and powerful framework for generative modeling, capturing complex data distributions while maintaining computational efficiency.

## IV. Our Method

We propose NF-NIDS, a novel approach that leverages NFs for network intrusion detection systems. By applying NFs to network traffic data, we can accurately learn the underlying probability distribution of normal network behavior and generate a probability density function to quantify the

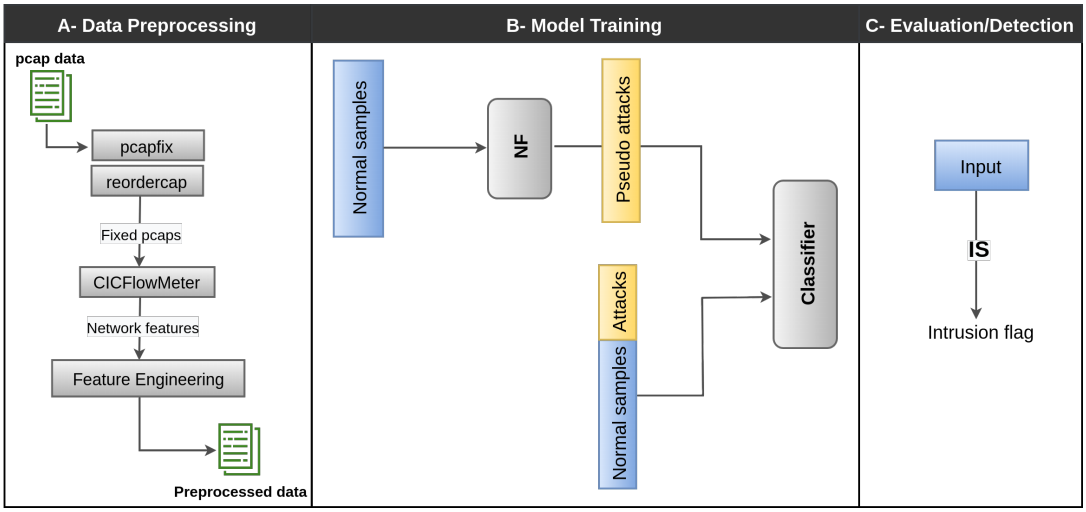



Fig. 1: NF-NIDS general architecture which consists of three main phases, data preprocessing, model training, and a detection phase.

likelihood of observed normal events. Fig. 1 illustrates the general overview of NF-NIDS, which can be divided into three main phases: data preprocessing, model training, and a detection phase. Each step is explained thoroughly in the remainder of this section.

### A. Data preprocessing

For each dataset utilized in this work, the publicly available native packet capture (pcap) files were utilized and preprocessed following the methodology proposed in [21], [22]. Initially, the *pcapfix*[2] tool is used to repair any possible damaged or corrupted pcap files. The resulting pcaps are then run through *reordercap*[3] to ensure that the packets are ordered by timestamp. The output is fixed and ordered pcaps that are ready for further analysis. In order to extract relevant features from the network traffic data, we utilize CICFlowMeter, a widely used tool for feature extraction in network traffic analysis. However, we adopt an improved version proposed in [21], [22], which addresses various issues related to labeling, flow construction, attack simulation, and more. A total of 87 features are extracted and saved into CSV files, including features such as packet length, duration, and protocol type. To prevent learning bias, we drop the flow identifiers {ID, source/destination IP and ports, timestamp} due to their strong correlation with the target label. Network data is generally composed of both categorical and numerical features, hence a *Feature Conversion* step is necessary to convert character data into numeric values. The resulting features have different units and scales, and it is recommended to perform *Feature Normalization* before the learning process. This ensures that all features are on a similar scale, with zero mean and unit variance. In our case, min-max scaling was used to normalize the data: $x' = \frac{x - x_{min}}{x - x_{max}} \in \mathbb{R}^{82}$ where $x_{min}$ and $x_{max}$ are computed on the training dataset. For the labeling of flow data in the CIC-IDS2017 and CSE-CIC-IDS2018 datasets, we adhere to the guidelines[4] provided by [22]. The pipeline of the preprocessing phase is illustrated in Fig. 1.

### B. Model training

This phase involves training two models, a NF model, and a classifier.

*a) Training NF model:* The NF model is trained using only the set of normal network traffic data with the objective of maximizing the log-likelihood of observed normal events, as shown in equation 2. This maximization is equivalent to minimizing the negative log-likelihood (NLL). Therefore, the training objective is to find the best set of hyperparameters of the NF transformation function $f$ that minimizes the overall likelihood over the training dataset $\mathcal{D}$:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} log\, p_X(x) \tag{7}$$

Once the NF model learns the underlying distribution of the normal data, including the characteristics of the tails, it is then used to sample pseudo-attack samples. During the training, the model captures the rare events and outliers present in the data, enabling it to generate samples from the tails of the learned distribution. To create pseudo-anomalies, we sample from the tails of the latent space distribution, obtaining latent vectors that correspond to synthetic anomalies. These latent vectors are then transformed through the inverse transformations of the flow to generate pseudo-anomalous samples in the data space. The NF model serves as a data augmentation method to generate synthetic attack samples, which helps in enhancing the performance of the classifier.

[2]https://github.com/Rup0rt/pcapfix

[3]https://www.wireshark.org/docs/man-pages/reordercap.html

[4]https://github.com/GintsEngelen/CNS2022_Code

*b) Training the classifier:* Most AD solutions for NIDS are designed with the assumption that only normal data is available for training. However, in reality, labeled attack samples are often available but limited. While the scarcity of labeled attack samples may seem like a limitation, it also presents an opportunity to leverage the available limited attack data. Incorporating attack samples, even if they are sparse, can provide valuable insights into the characteristics of specific attack types. Integrating these limited attack samples into the training data allows the model to learn from them and potentially improve the detection of similar attack patterns in the future. Hence, our method aims to augment the attack class data while also benefiting from the available attack samples.

Once the NF model is trained, we proceed to train a classifier using normal samples and a combination of the available real attacks and sampled pseudo-attacks generated by the NF model. During the classifier training, we optimize the model's parameters using a supervised learning approach, where the goal is to accurately classify samples into normal and attack categories.

### C. Detection phase

Once the convergence of both the NF model and classifier training phases is achieved, the detection phase begins. The classifier is utilized to classify incoming network traffic as either normal or attack. First, an Intrusion Score (IS) is computed for the input data, which, in our case, are the probabilities output by the classifier. If the IS exceeds a certain threshold, the traffic is classified as an attack.

$$x_{new} = \begin{cases} Normal, & \text{if IS} \leq \text{thr} \\ Anomaly, & \text{otherwise} \end{cases} \tag{8}$$

## V. Experiments and Results

### A. Datasets

Three datasets are used to validate the NF-NIDS framework.

***USTC-TFC2016*** [23] is a prominent and widely used dataset for network traffic analysis. It is divided into two main parts, each offering distinct types of data. The first part comprises 10 different categories of malware traffic. The second part of the dataset consists of normal traffic, representing ten different genres. In this study, we divide the dataset into training and testing using a 70-30 ratio.

***CIC-IDS2017*** [24] is a state-of-the-art NIDS dataset widely recognized for its reliability and realistic network traffic data. It comprises real-world network traffic data categorized as "Benign" and "Attacks", all captured in the popular pcap format. The dataset spans a five-day period, from Monday to Friday, providing a comprehensive representation of network activity. In our experiments, we adopt a specific data-splitting strategy for training and testing. We select the first four days of the dataset, from Monday to Thursday, for training and the Friday data for evaluation purposes.

***CSE-CSE-CIC-IDS2018*** is an enhanced version of the renowned CIC-IDS-2017 dataset[5]. It represents a significant improvement over its predecessor, incorporating up-to-date attack types and more comprehensive network intrusion scenarios. To ensure a fair evaluation of intrusion detection models, the dataset is structured based on a two-week timeframe. The first week's worth of data is allocated for training, enabling the construction and refinement of intrusion detection models. On the other hand, the second week's data is kept entirely separate and utilized solely for testing.

[5]https://www.unb.ca/cic/datasets/ids-2018.html

### B. Implementation details and evaluation metrics

The IAF and NSF models utilized in this paper consist of 10 flows of neural networks as bijections, where each network is formed with two hidden layers. Both models were trained for 10 epochs with a batch size of 100 and a learning rate of 2E-04. The classifier is modeled as three fully-connected layer network, trained for 3 epochs with a batch size of 100 and a learning rate of 0.001 using the Adam optimizer. We assume the availability of 100, 1000, and 10000 attack samples for USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018 respectively, which represents only 0.04%, 0.08%, and 0.02% of the entire training data size respectively. We empirically determine the threshold by testing the values in the range $\{0.85, 0.86, \ldots, 0.99\}$ and selecting the value that maximizes the F1-score in the testing set.

To evaluate the proposed method, we adopt three commonly used network intrusion detection metrics, namely F1-score, Accuracy, and False Discovery Rate (FDR). With FDR refers to the rate at which the system incorrectly identifies normal or benign events as anomalies. It is a metric used to assess the accuracy and reliability of the intrusion detection system. Furthermore, we consider *Receiver Operating Characteristic* (ROC) curves and the correspondent Area Under Curve (AUC) as additional performance measurements. The higher the value of the AUC the better the model at classifying normal and attack inputs.

The implementation was built upon the publically available code[6] using PyTorch and other libraries such as numpy, pandas, etc. This work was carried out using the African SuperComputing Center HPC service, supported by Mohammed VI Polytechnic University[7].

TABLE I: NF-NIDS testing results on USTC-TFC2016 dataset. The best results are presented in bold.

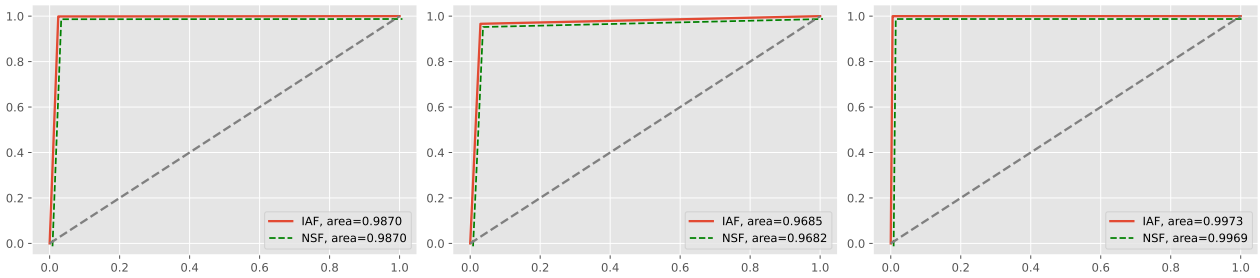
| Metrics | IAF | NSF |
|---|---|---|
| F1-score (%) | **98.71** | **98.71** |
| FDR (%) | 0.14 | **0.09** |
| Accuracy (%) | 98.71 | **98.72** |

### C. Results and discussion

Table I, II and III provide the obtained performance results for USTC-TFC2016, CIC-IDS2017 and CSE-CIC-IDS2018

[6]https://doi.org/10.7717/peerj-cs.757/supp-1

[7]https://ondemand.hpc.um6p.ma/




(a) USTC-TFC2016 (b) CIC-IDS2017 (c) CSE-CIC-IDS2018

Fig. 2: ROC curves and respective AUC/ROC values for both models IAF and NSF for detecting attacks in USTC-TFC2016, CIC-IDS2017, CSE-CIC-IDS2018.

TABLE II: NF-NIDS testing results on CIC-IDS2017.

| Metrics | IAF | NSF |
|---|---|---|
| F1-score (%) | 94.80 | **94.86** |
| FDR (%) | **0.68** | 0.70 |
| Accuracy (%) | 97.00 | **97.05** |

TABLE III: NF-NIDS testing results on CSE-CIC-IDS2018.

| Metrics | IAF | NSF |
|---|---|---|
| F1-score (%) | **98.44** | 98.20 |
| FDR (%) | 8.51E-04 | **5.26E-04** |
| Accuracy (%) | **99.51** | 99.44 |

datasets, respectively. For the three datasets, both IAF and NSF models achieved high performance over all evaluated metrics. The models can separate normal from attack samples with an accuracy of over 98%, 97%, and 99% for USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018 respectively. Similarly for F1-score, the IAF and NSF models achieved scores above 0.96 for both USTC-TFC2016 and CSE-CIC-IDS2018 datasets and above 94% for CIC-IDS2017. While FDR metric reports extremely low values, especially for the CSE-CIC-IDS2018 dataset where the FDR is less than 5.27E-04. These results indicate that NF-NIDS can accurately classify network traffic into normal and attack categories, with high accuracy and a low FDR. The ROC curves and AUC values were also considered throughout the evaluation of our proposed method, further confirming its effectiveness in network intrusion detection. For all datasets, the proposed method achieves the AUC of a minimum of 0.96%. We can also observe that IAF and NSF models show comparable performance. However, it is worth mentioning that IAF is remarkably faster than NSF, especially for large datasets such as CSE-CIC-IDS2018.

When evaluated on the USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018 datasets using the proposed method, our models consistently achieved high performance for network intrusion detection tasks in terms of F1-score, accuracy, and FDR. Furthermore, incorporating the attack samples even when they are scarce, has been shown to improve the overall detection capabilities of the system, highlighting the importance of leveraging limited attack data for better anomaly detection.

In summary, this work outcome highlights the potential of using NFs for NIDS. Hence, further research is warranted to explore the applicability of these models in a wide range of network intrusion detection scenarios. Moreover, other aspects of NFs can be investigated in the case of NIDS, such as interpretability. While the primary purpose of NFs is to model complex data distributions and enable tasks like data generation and density estimation, they can also provide valuable insights into the underlying data and model representations.

## VI. Conclusion

In this paper, we proposed the use of normalizing flows for anomaly-based intrusion detection systems. We introduced NF-NIDS, a novel approach that leverages normalizing flows to accurately classify network traffic into normal and attack categories. Two main models, IAF and NSF, were employed to learn the exact distribution of normal behavior and generate pseudo-attacks from the tails of the distribution. We conducted empirical evaluations on multiple benchmark datasets, including USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018, to demonstrate the effectiveness of our proposed method. Our models consistently achieved higher performance in terms of F1-score, accuracy, and false discovery rate when evaluated on these datasets. The results indicate that normalizing flows can be a valuable tool for network intrusion detection tasks, offering accurate classification and low false discovery rates. This research opens new possibilities for enhancing network security and showcases the potential of leveraging generative modeling techniques in the realm of cybersecurity.

The positive results obtained from our research motivate further exploration of additional approaches utilizing NFs for anomaly-based NIDS. As future directions, we believe that assessing the effectiveness of other flow-based models, such as NICE [11], RealNVP [12], and Glow [25] would offer much-needed insight into the area. Additionally, investigating the generalizability power of these models would also be worthwhile.

## REFERENCES


[1] 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," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.

[2] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[3] S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108 346–108 358, 2020.

[4] S. Azmin and A. B. M. A. A. Islam, "Network intrusion detection system based on conditional variational laplace autoencoder," in *Proceedings of the 7th International Conference on Networking, Systems and Security*, ser. NSysS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 82–88. [Online]. Available: https://doi.org/10.1145/3428363.3428371

[5] D. Li, D. Kotani, and Y. Okabe, "Improving attack detection performance in nids using gan," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 817–825.

[6] D. Liao, S. Huang, Y. Tan, and G. Bai, "Network intrusion detection method based on gan model," in *2020 International Conference on Computer Communication and Network Security (CCNS)*, 2020, pp. 153–156.

[7] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," in *Pacific-asia conference on knowledge discovery and data mining*. Springer, 2022, pp. 79–91.

[8] G. Andresini, A. Appice, L. De Rose, and D. Malerba, "Gan augmentation to deal with imbalance in imaging-based intrusion detection," *Future Generation Computer Systems*, vol. 123, pp. 108–127, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21001382

[9] M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, and I. Berrada, "Fed-anids: Federated learning for anomaly-based network intrusion detection systems," *Expert Systems with Applications*, vol. 234, p. 121000, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423015026

[10] A. Tabassum, A. Erbad, W. Lebda, A. Mohamed, and M. Guizani, "Fedgan-ids: Privacy-preserving ids using gan and federated learning," *Computer Communications*, vol. 192, pp. 299–310, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422002171

[11] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[12] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," *CoRR*, vol. abs/1605.08803, 2016. [Online]. Available: http://arxiv.org/abs/1605.08803

[13] E. Zisselman and A. Tamar, "Deep residual flow for out of distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[14] A. Ryzhikov, M. Borisyak, A. Ustyuzhanin, and D. Derkach, "Normalizing flows for deep anomaly detection," *CoRR*, vol. abs/1912.09323, 2019. [Online]. Available: http://arxiv.org/abs/1912.09323

[15] M. L. D. Dias, C. L. C. Mattos, T. L. C. da Silva, J. A. F. de Macedo, and W. C. P. Silva, "Anomaly detection in trajectory data with normalizing flows," 2020.

[16] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 98–107.

[17] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, "Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows," *CoRR*, vol. abs/2111.07677, 2021. [Online]. Available: https://arxiv.org/abs/2111.07677

[18] Y. Guo, J. Tang, L. Yang, Z. Zhao, M. Wang, and P. Shi, "Robustflow: An unsupervised paradigm toward real-world wear detection and segmentation with normalizing flow," *Tribology International*, vol. 179, p. 108173, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301679X22007447

[19] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[20] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, *Neural Spline Flows*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[21] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the cicids2017 case study," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 7–12.

[22] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018," in *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2022, pp. 254–262.

[23] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 712–717.

[24] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4707749

[25] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.