DOC-NAD: A Hybrid Deep One-class Classifier for Network Anomaly Detection

Mohanad Sarhan*[1], Gayan Kulatilleke[1], Wai Weng Lo[1], Siamak Layeghy[1], Marius Portmann[1]

[1]University of Queensland, Brisbane, Australia

*Corresponding Author: m.sarhan@uq.net.au

Abstract

Machine Learning (ML) approaches have been used to enhance the detection capabilities of Network Intrusion Detection Systems (NIDSs). Recent work has achieved near-perfect performance by following binary- and multi-class network anomaly detection tasks. Such systems depend on the availability of both (benign and malicious) network data classes during the training phase. However, attack data samples are often challenging to collect in most organisations due to security controls preventing the penetration of known malicious traffic to their networks. Therefore, this paper proposes a Deep One-Class (DOC) classifier for network intrusion detection by only training on benign network data samples. The novel one-class classification architecture consists of a histogram-based deep feed-forward classifier to extract useful network data features and use efficient outlier detection. The DOC classifier has been extensively evaluated using two benchmark NIDS datasets. The results demonstrate its superiority over current state-of-the-art one-class classifiers in terms of detection and false positive rates.

Keywords— One-class classifier, intrusion detection, machine learning, anomaly detection

1 Introduction

With the rapid increase and modification of network attack vectors, the need for dynamic attack detection is inevitable. Network attacks can cause information leakage, tampering, and disruption of organisational networks [1]. Modern network attacks often require sophisticated tactics and techniques that may bypass current security control systems [2]. Therefore, detecting network attacks is vital to maintaining the three principles of information security; confidentiality, integrity, and availability [3]. Network Intrusion Detection Systems (NIDSs) aim to scan and detect network adversaries at the perimeter layer as they penetrate a computer network [4]. There are two main types of NIDSs, i.e. signature- and anomaly-based NIDS [5], that alert network and security administrators of a sign of a threat. Signature-based NIDSs scan incoming network traffic for any known Indicators of Compromise (IOCs), such as IPs, domain names, and hash values [6]. This detection method is highly effective in identifying known threats as it compares traversing packets to a pre-configured list of known malicious signatures.

Anomaly-based NIDSs detect suspicious network behaviours as an anomaly by learning a computer network's normal or benign usage behaviour [7]. The baseline represents how the standard operating network traffic activities look, and any out-of-the-ordinary activity that does not fit the pre-defined profile will trigger a detection. Therefore, anomaly-based NIDSs are highly effective in detecting zero-day attacks compared to signature-based NIDSs [8] as they do not rely on pre-defined malicious signatures. Most modern anomaly-based NIDSs use Machine Learning (ML) techniques to train effective models for detecting network intrusions. ML-based NIDSs are designed with various architectures and are exposed to network data samples in the training phase to extract the semantic attributes of the captured training dataset. The training set is a collection of network data flows captured from the hosting organisation and presented in a flow export format such as NetFlow [9]. During repeated training rounds, the models are optimised to minimise the error in mapping the input data samples to the desired output.

There are two main and widely used types of ML classification; binary- and multi-class [10]. Binary classification is widely adopted in developing ML-based NIDSs, with great success. The model learns the distinctive patterns between benign and malicious network traffic [11]. The output predictions are in a binary format, indicating a safe or an unsafe test data sample. In the development of such models, the training dataset must include both data classes (malicious

and benign) to facilitate a binary classification model [12]. Multi-classification has been used in developing ML-based NIDSs to classify network data traffic into benign, or one of the known attack classes [13]. The model learns the unique patterns of each available data class to identify the test data sample into one of them. Multi-class ML-based NIDSs are mainly used in network forensics to identify the type of exploit used in penetration. In developing multi-class models, the training dataset requires data samples from each data class.

The critical limitation of designing binary- and multi-class classification models is the dependence on the availability of each class data sample in the training phase [14]. The necessity of having a training set consisting of both benign and malicious network data samples has made the development of such NIDSs challenging. Malicious data samples in production network environments are challenging to collect [15] in sufficient amounts for ML training. This is due to most of the exploit attempts being blocked at the perimeter level of an organisation by a current security control such as a firewall or an existing NIDS. In case of a rare zero-day or successful exploit occurrence, accessing and capturing an adequate amount and quality of training data samples is difficult as the exact time of penetration and associated network data logs are unknown. Therefore, the requirement of collecting malicious network logs for ML training has significantly affected the development of ML-based NIDSs [16]. One-class classification is an emerging technique of ML classification used in anomaly detection to overcome the limitations faced by other techniques, where the learning models are trained exclusively on data samples representing a single class [17].

This paper proposes a novel Deep One-Class (DOC) Classifier based on a one-class classification methodology for network anomaly detection. The DOC classifier solves the ever-growing challenge and limitation of other classification techniques of collecting sufficient attack traffic samples for training. The DOC classifier uses a deep one-class learning technique known as Deep Support Vector Data Description (Deep SVDD) [18] to map network data features to an enhanced low-dimensional embedding, which is subsequently used by a Histogram-Based Outlier Score (HBOS) [19] for anomaly detection. The DOC operates by building an accurate representative profile of the benign network standard operating activities and detects outliers as an anomaly. Powerful deep feature extraction and rapid HBOS detection enable the DOC classifier to detect network intrusions effectively and accurately. DOC development requires benign network data samples only for the training process and omits the requirement of attack data collection, which makes it a practically suitable ML-based NIDS.

The paper is structured as follows; In Section 2, key related works that use one-class classification techniques to develop ML-based NIDSs are discussed. In Section 3, the usage of the proposed DOC classifier is motivated, and its architecture is explained. The evaluation methodology of the DOC classifier on two widely used NIDSs is described in Section 4. The results of the DOC detection performance are presented in Section 5 and compared with five key one-class classifiers. The main contributions of this paper are a) the design of a novel DOC classifier for NIDS that does not require the availability of attack data samples for training and b) the extensive evaluation of the DOC classifier across two datasets and comparison with other classifiers demonstrating the superiority of the proposed framework.

2 Related Work

This section reviews some of the key related work proposing one-class classifiers for NIDS. Kind et al. [20] highlighted the promising uses of histograms in the detection of network intrusions. The method proposed in 2009 involves constructing histograms of different network traffic features. Each histogram is embedded into a metric space to position similar histograms together. Data mining techniques are used to model benign network behaviour. The typical patterns are compared with the incoming network traffic feed to identify deviations and detect anomalies. Two real-world and one synthetic datasets were used to evaluate the proposed framework. The results demonstrate the superiority of histograms over entropy-based distribution approximations in detecting a wide range of anomalies.

Zavrak et al. [21] proposed a variational autoencoder system for network anomaly detection. The authors use flow-based features because of their ease of extraction for model training and compare the detection performance with traditional autoencoder and one-class Support Vector Machine (SVM) models. In a one-class classification methodology, models were trained on benign-only data samples obtained from the CICIDS2017 NIDS dataset. It is shown that the variational autoencoder-based anomaly detection system performs better compared to the other considered models. The proposed method achieved a higher detection rate in 9 of the 14 attack groups available in the dataset.

An evaluation study [22] compares the performance of several one-class classifier models using the UNSW-NB15 dataset. The authors highlight the necessity of adopting the one-class classification methodology in developing NIDSs due to the imbalanced nature of the datasets. The evaluation involved developing and evaluating one-class SVM, isolation forest, minimum covariance determinant, and local outlier factor models. The Pearson correlation feature selection technique discarded the highly correlated data features and used the random forest to identify the most relevant ones. The results demonstrate the superiority of one-class SVM over the other considered approaches with an accuracy of 61.9%.

In [23], Verkerken et al. evaluated the performance of various unsupervised ML models using the CIC-IDS-2017 NIDS dataset. The paper highlights the importance of unsupervised techniques in detecting zero-day attack groups. The dataset is transformed using Principal Component Analysis (PCA) for dimensionality reduction. In a one-class classification technique, the models were trained on benign-only data samples and evaluated on merged (benign and malicious) data samples. The results show that autoencoders yield the best detection performance of a 96.16% F1 score, followed by one-class SVM, isolation forest, and PCA classifiers.

Zhang et al. [24] recommend using one-class SVM over the traditional two-class variant due to the ease of constructing the training sets in developing NIDSs. Both classifiers were trained and tested using the KDDCUP99 dataset and compared to the performance of a Probabilistic Neural Network (PNN). The one-class SVM trained on the benign data samples achieved an effective detection rate on the DoS and Probe attack categories; however, the R2L, U2R, and "other" attack groups were unreliably detected. In comparison, the one-class SVM achieved a higher overall detection rate but a lower precision rate due to a more significant number of false positives compared to the PNN and two-class SVM.

In [25], Vasudevan et al. combined the advantages of supervised and unsupervised learning to obtain better intrusion detection performance. The proposed hierarchical model is devised in three stages: Dirichlet Process (DP) clustering based on the underlying data distribution, Local Outlier Factor (LOF) to identify dense local areas and group them into four bins, and finally, a one-class classifier to model the benign instances in each bin. Hybrid density estimation, reconstruction, and boundary methods are used in the design of the one-class classifier. The evaluation of the proposed model is carried out on the KDDCup99 and SSENet-2011 datasets and compared with one-class SVM. The results demonstrate the superiority of the proposed model on both datasets.

3 Hybrid DOC

Current ML-based NIDSs implement ML models in binary- and multi-class classification techniques, which require a training set constructed of benign and malicious data samples. This practice has been deemed challenging in the real world, as the practical collection of sufficient anomaly data samples in a traffic trace is complex. This is because most known attacks are generally blocked at the perimeter by current security controls. In the rare occasion of a successful exploit, the complete set of associated network traffic is unknown, and their real-time capture is difficult. The collection of zero-day attacks for ML training is impossible due to their non-existence at the time of ML-based NIDS development. Furthermore, training and successful evaluation of ML models on known attacks does not indicate a generalisable behaviour on unknown attacks, as illustrated in [8]. Therefore, the practical deployments of effective ML-based NIDSs in binary- and multi-class classification techniques are limited.

One-class classification is an emerging paradigm of ML and is generally associated with anomaly or outlier detection. The models are exclusively trained on "normal" class data samples to build a profile of the accepted usage. One-class classification assumes that the training set is free or insufficiently sampled of the "abnormal" class data samples. This assumption correlates with the real-world networking application, as the benign (normal) class data samples are easy to capture and analyse in an operational network environment. Whereas, on the other hand, it is very challenging to collect a sufficient amount of malicious (abnormal) class data samples. More importantly, due to the rapid increase and modification of network attack vectors, it is impossible to collect malicious network data samples representing the wide range of attack scenarios, such as zero days, that will be observed in production (post-training). Therefore, from practical experience, it is easier to train a one-class classifier compared to binary- and multi-class classifiers in ML-based NIDS models.

The DOC classifier proposed in this paper is designed using a hybrid one-class architecture that utilises the deep extraction of data point embeddings and the construction of histograms of benign network data samples. The training stage involves mapping extracted semantic attributes to an accurate representative profile of the standard operating network environment. Outlier network flows that do not fit within the set profile are detected as anomalies. The DOC architecture consists of two main integrated components: DeepSVDD [18], and HBOS [19], as shown in Figure 1. The integration of the two components complements the benign data deep extraction and mapping process to improve the anomaly detection performance compared to other classifiers.

The first component of DOC is known as DeepSVDD, which uses deep learning to extract relevant features automatically without the substantial feature engineering processes required in shallow learning methods. DeepSVDD is inspired by a kernel-based one-class classification, and minimum volume estimation [18]. During training, the neural model minimises the hypersphere volume that encloses the network representations of the benign data samples in the output space. This technique forces the model to extract the common factors of variation of the data distribution as the data points are closely mapped to the centre of the sphere. Stochastic Gradient Descent (SGD) is used to optimise the parameters of the neural network model using back-propagation.
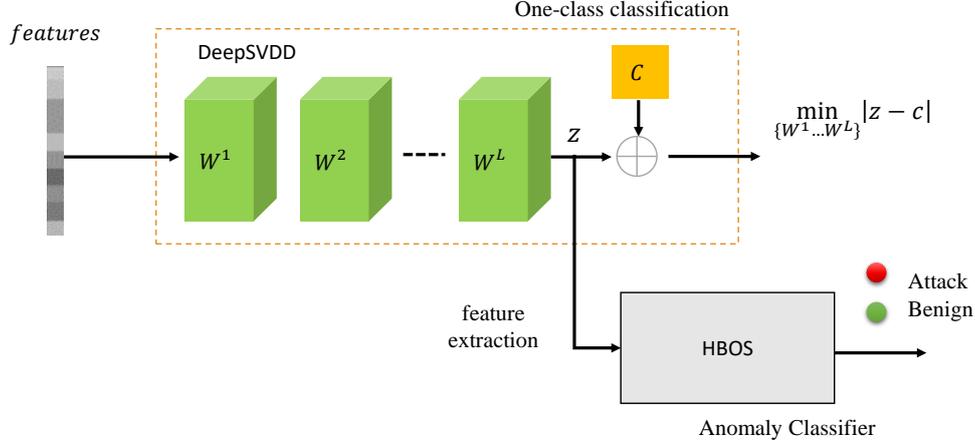
Figure 1: System architecture

The main objective of DeepSVDD is defined in Equation 1, where the input and output spaces are $\mathscr{X} \subseteq \mathbb{R}^d$ and $\mathscr{F} \subseteq \mathbb{R}^p$ respectively, $\phi(\cdot;\mathscr{W}) : \mathscr{X} \to \mathscr{F}$ is a neural network containing $L \in \mathbb{N}$ hidden layers and set of weights $\mathscr{W} = \{W^1, \ldots, W^L\}$. Specifically, $\phi(x;\mathscr{W}) \in \mathscr{F}$ is the learnt representation of $x \in \mathscr{X}$ from network ϕ with parameters $\mathscr{W}$. DeepSVDD aims to learn $\mathscr{W}$ by finding a hypersphere of minimum volume with centre $c \in \mathscr{F}$. The hypersphere is contracted by minimising the mean distance of all data representations to the centre c. Essentially, minimizing $\|\phi(x;\mathscr{W}^*) - c\|^2$ minimizes the volume of the hypersphere. The second term is a weight decay regulariser with hyper-parameter $\lambda > 0$. $\|\cdot\|_F$ denotes the Frobenius norm.

$$\min_{\mathscr{W}} \frac{1}{n} \sum_{i=1}^{n} \|\phi(x_i;\mathscr{W}) - c\|^2 + \frac{\lambda}{2} \sum_{\ell=1}^{L} \left\| W^\ell \right\|_F^2 \quad \text{where } \mathscr{W} = \{W^1 \cdots W^L\}. \tag{1}$$

Optimising Equation 1 results in benign network data points being closely mapped to the centre c of the hypersphere through learning to extract the common factors of variation of the data. While benign data samples will be closely mapped to c, outliers (which lack the common factors) will be mapped further away. DeepSVDD assumes most of the training data $\mathscr{D}_n$ is normal, which is a reasonable assumption in one-class classification tasks and, it penalises the mean distance over all data points forcing the benign data points closer to c. We use the representations z_i from the trained DeepSVDD obtained via Equation 2.

$$z_i = \phi(x_i;\mathscr{W}) \tag{2}$$

The second component of DOC, known as HBOS, maps the features extracted by DeepSVDD into a set of histograms. A histogram is a statistical distribution of several samples over the possible values of the data value. In a NIDS dataset, histograms can present the distribution of a network traffic data feature such as the number of TCP flags, the TTL value or in/out bytes. Histograms model the detailed data features of network traffic, which enables the identification of a broader range of anomalies. HBOS assumes independence of data features and calculates the degree of anomaly by building histograms. This feature makes HBOS computationally efficient compared to multivariate approaches, which makes it suitable for developing NIDS where the detection results are required immediately.

HBOS constructs a univariate histogram for each network data feature. A static number of bins are used using k equal width bins over the value range. The frequency of samples located in each bin is used to estimate the height of the bins. As shown in Equation 3, for each data feature d, a histogram is constructed with a height representing the density estimation. All histograms are normalised to a maximum height of 1.0, ensuring an equal weight of each feature to the anomaly score. Finally, the HBOS of each data sample p is calculated using the corresponding height of the bins where the sample is located.

$$HBOS(z_i) = \sum_{j=0}^{d} \log \left(\frac{1}{\text{hist }_j(z_i)} \right) \tag{3}$$

In Figure 2, the two-phase data point representation generation and classification of the DOC are illustrated. The left side displays the first phase where the deep one-class classification objective of DeepSVDD (Equation 1) is used to generate discriminative lower d dimensional feature representations. Essentially, deep extraction is performed by

4

learning a mapping function ϕ such that the majority benign data samples are closer towards the centre c of the enclosing hyper-sphere of a radius R while the minority anomaly data samples tend to be further away. In the next phase, shown on the right side, learnt representations z_i are used to build a probability distribution for *each* lower dimensional feature $j \in \{1, \ldots, d\}$ of the majority benign representations. This expected set of distributions is shown in green.
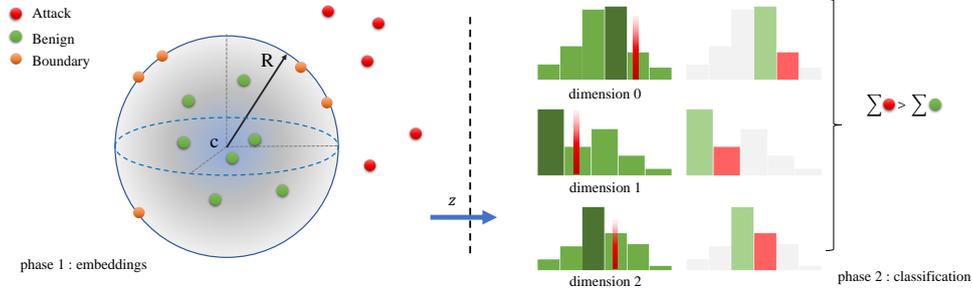

Figure 2: The two-phase Benign vs Anomaly data point representations and classification

Classification is performed via the HBOS score, which is obtained by aggregating each of the feature's inverse occurrence probability obtained from the expected distribution of the majority benign distribution. In a benign sample, the feature values are closer to the expected probability distribution, and the aggregated inverse HBOS score is lower. However, as anomalous samples tend to differ from the majority distribution (as shown in red), and as a result, probabilities of feature values are lower and the aggregated inverse HBOS score is higher.

4 Evaluation Methodology

The DOC model is evaluated on two modern NIDS datasets. Several other one-class classifiers are implemented and evaluated to compare the detection performance on the same datasets. The most common and widely used models are implemented for comparison. The chosen models are Isolation Forest (IF) [26], Principal Components Analysis (PCA) [27], and Variational Autoencoder (VAE) [28]. In addition, we utilise the standalone DeepSVDD and HBOS models as a benchmark to the developed DOC model. The experiments were run in Python using PYOD [29] based on Keras for ML and Pandas for data processing. The default hyperparameters of the PYOD library were used in designing the models. Standard NIDS classification performance metrics are used to evaluate the one-class classifiers, i.e. Detection Rate (DR), False Alarm Rate (FAR), Area Under the Curve (AUC), and F1 score. These metrics are defined based on the numbers of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN), as shown in Table 1.

Table 1: Evaluation Metrics

Metric	Definition	Equation
Accuracy	The percentage of correctly classified samples in the test set.	$\frac{TP+TN}{TP+FP+TN+FN} \times 100$
Detection Rate (DR)	The percentage of correctly classified total attack samples in the test set.	$\frac{TP}{TP+FN} \times 100$
False Alarm Rate (FAR)	The percentage of incorrectly classified benign samples in the test set.	$\frac{FP}{FP+TN} \times 100$
Area Under the Curve (AUC)	The area underneath the DR and FAR plot curve in the test set.	N/A
F1 Score	The harmonic mean of the model's precision and DR.	$2 \times \frac{DR \times Precision}{DR + Precision}$

Two key and widely used NIDS datasets are used to evaluate the one-class classifiers, i.e., NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS2018-v2 [30]. Both datasets are synthetic and created via virtual network testbeds representing organisational network environments. Synthetic datasets are widely used in the literature as they overcome the privacy and security concerns encountered in real-world production networks. Benign network traffic is captured for standard network usage baseline. Moreover, several attack scenarios are conducted, and the corresponding network traffic is collected. Network traffic is captured in its native packet capture (pcap) format. Further processing involves the

extraction of network data features for traffic analysis. The features present information regarding the data flows, labelled malicious or benign records. The network data flows in NF-UNSW-NB15-v2, and NF-CSE-CIC-IDS2018-v2 are presented in NetFlow v9 standard format. NetFlow is a de facto flow network monitoring and analysis standard due to its practicality and ubiquitous deployment. Each used dataset has been generated over different test beds and includes a unique set of benign applications and malicious use cases. This consequently results in a variation of the statistical distributions held by each dataset. Therefore, the datasets used in this paper are non-Independent and Identically Distributed (IID) and extensively evaluate the proposed model.

- **NF-UNSW-NB15-v2 [30]**- A NetFlow dataset based on the UNSW-NB15 dataset has recently been generated and released in 2021. The original dataset was released in 2015 by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The IXIA Perfect Storm tool was configured to simulate benign network traffic and synthetic attack scenarios. The dataset is generated by extracting 43 NetFlow-based features, explained in [30], from the pcap files of the UNSW-NB15 dataset. The nprobe feature extraction tool extracts network data flows, which are labelled using the appropriate data labels. The total number of data flows is 2,390,275, of which 95,053 (3.98%) are attack samples and 2,295,222 (96.02%) benign. There are nine attack groups: Exploits, Fuzzers, Generic, Reconnaissance, DoS, Analysis, Backdoor, Shellcode, and Worms.

- **NF-CSE-CIC-IDS2018-v2 [30]**- An IoT NetFlow-based dataset released in 2021 containing different attack types, such as brute-force, bot, DoS, DDoS, infiltration, and web attacks. The exploits are conducted from an external network to simulate realistic attack scenarios. The dataset is generated by converting the publicly available pcap files of the CSE-CIC-IDS2018 [31] dataset to 43 NetFlow v9 features using the nprobe [32] tool. The total number of data flows is 18,893,708, of which 2,258,141 (11.95%) are attack samples and 16,635,567 (88.05%) benign ones. The source dataset (CSE-CIC-IDS2018) was released by a collaborative project between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) in 2018. Their developed tool called CICFlowMeter-V3 was used to extract 75 network data features. The network testbed simulates a realistic organisational computer network consisting of five departments and a server room.

The publicly available datasets are further processed in this paper for efficient ML operation and reliable evaluation. Initially, the flow identifiers, such as source/destination IPs and ports, are removed to avoid learning bias towards the attacking- and victim-end nodes. This is due to the nature of synthetic network datasets where distinct nodes are used in launching malicious traffic. The benign samples are split into training and testing sets in a ratio of 70% to 30%, respectively. The attack samples are only added to the testing sets to accommodate for one-class classification methods. Finally, the Min-Max Scaler normalisation technique is applied to obtain all values between 0 and 1 for effecting ML training. The normalisation occurs by applying

$$X_* = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4}$$

on each set, where X_* represents the final output value ranging from 0 to 1. X is the original input value, and $X_{\max}$ and $X_{\min}$ indicate the maximum and minimum values for each feature, respectively. To obtain reliable and fair evaluation metrics, a k-fold cross-evaluation technique is adopted with $k = 5$.

5 Results

The proposed DOC model is implemented and evaluated across two NIDS datasets, and the detection performance is compared with five one-class classifiers. In Table 2, the evaluation metrics achieved by the classifiers on the NF-UNSW-NB15-v2 dataset are presented. The accuracy and F1 score metrics of IF, PCA, and VAE are similar; they all achieved around 90%, caused by a low DR and a high FAR. The DeepSVDD and HBOS achieved similar performance with an increased DR compared to the other classifiers. However, the FAR remained high at around 10%, increasing the AUC to 95%. The proposed DOC classifier has achieved a significantly lower FAR than the rest of the classifiers at a rate of 2.01% while maintaining a high DR. The DOC classifier with an F1 score and AUC of 98.26% and 98.89%, respectively, achieved the best evaluation metrics.

Table 3 presents the results of the one-class classifiers achieved on the NF-CSE-CIC-IDS2018-v2 dataset. IF and VAE performance is the poorest across the range of classifiers, caused by low attacks DR with an AUC of 74.90% and 76.87%, respectively. The rest of the classifiers (PCA, DeepSVDD, HBOS, and DOC) achieve similar detection performance, with DOC leading the range with an AUC of 85.52%. The proposed DOC classifier achieves the best performance across all metrics except for the DR, with PCA having a higher attack detection value of 81.60% compared to 76.03%. However, a significantly lower FAR of 5.00% is noticed by the DOC classifier, which makes it a superior classifier overall.

Table 2: NF-UNSW-NB15-v2 results

	Accuracy	F1 Score	AUC	DR	FAR
IF	89.87	90.83	89.45	88.90	10.00
PCA	88.85	89.81	85.22	80.42	9.98
VAE	88.91	89.87	85.54	81.09	10.01
DeepSVDD	91.26	92.19	95.03	100.00	9.94
HBOS	91.23	92.16	95.01	100.00	9.98
DOC	**98.20**	**98.26**	**98.89**	**99.79**	**2.01**

Table 3: NF-CSE-CIC-IDS2018-v2 results

	Accuracy	F1 Score	AUC	DR	FAR
IF	80.60	80.01	74.90	59.80	9.99
PCA	87.39	87.45	85.41	81.60	9.99
VAE	81.83	81.42	76.87	63.72	9.98
DeepSVDD	86.01	86.00	83.59	77.16	9.98
HBOS	85.61	85.57	82.91	75.72	9.91
DOC	**89.09**	**88.87**	**85.52**	**76.03**	**5.00**

The proposed DOC classifier achieves the best-performing detection metrics across two different NIDS datasets compared to the rest of the considered classifiers. A great feature of the DOC classifier is the low FAR value, where a significant decrease of around 80% and 50% were observed on the NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS2018-v2 datasets, respectively. This is an essential improvement considering how a high number of FP alerts would result in security operation teams' fatigue and, eventually lack of trust in the triggered events. While attaining a low FAR, DOC maintained a competitive DR equal to or slightly lower than the rest. Overall, the superior performance of the DOC classifier across the different ranges of classification metrics confirms the efficiency and effectiveness of the proposed architecture.

6   Conclusion

In this paper, a deep one-class classification model is proposed and defined as DOC. The model is evaluated using two key NIDS datasets, and its performance is compared to five standard one-class classifiers. The proposed approach reliably detects unseen attack groups and outperforms the state-of-the-art classifiers. A high level of DRs and significantly lower FARs across the two datasets demonstrate the effectiveness of the DeepSVDD and HBOS integration. The deep extraction and mapping of the benign network environment to univariate histograms is a promising feature of ML-based NIDSs and an effective solution to the lack of malicious data samples challenge and should be further explored and evaluated.

References

[1] R. Nazir, K. Kumar, S. David, M. Ali, *et al.*, "Survey on wireless network security," *Archives of Computational Methods in Engineering*, pp. 1–20, 2021.

[2] C. Zhou, B. Hu, Y. Shi, Y.-C. Tian, X. Li, and Y. Zhao, "A unified architectural approach for cyberattack-resilient industrial control systems," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 517–541, 2020.

[3] S. Samonas and D. Coss, "The cia strikes back: Redefining confidentiality, integrity and availability in security.," *Journal of Information System Security*, vol. 10, no. 3, 2014.

[4] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature extraction for machine learning-based intrusion detection in iot networks," *Digital Communications and Networks*, 2022.

[5] V. Jyothsna, R. Prasad, and K. M. Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.

[6] V. Kumar and O. P. Sangwan, "Signature based intrusion detection system using snort," *International Journal of Computer Applications & Information Technology*, vol. 1, no. 3, pp. 35–41, 2012.

[7] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT)*, pp. 141–147, IEEE, 2017.

[8] M. Sarhan, S. Layeghy, M. Gallagher, and M. Portmann, "From zero-shot machine learning to zero-day attack detection," *arXiv preprint arXiv:2109.14868*, 2021.

[9] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," in *Big Data Technologies and Applications*, pp. 117–135, Springer, 2020.

[10] A. Jha, M. Dave, and S. Madan, "Comparison of binary class and multi-class classifier using different data mining classification techniques," in *Proceedings of International Conference on Advancements in Computing & Management (ICACM)*, 2019.

[11] J. M. Beaver, C. T. Symons, and R. E. Gillen, "A learning system for discriminating variants of malicious network traffic," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, pp. 1–4, 2013.

[12] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "Bodmas: An open dataset for learning based temporal analysis of pe malware," in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 78–84, IEEE, 2021.

[13] G. Singh and N. Khare, "A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques," *International Journal of Computers and Applications*, vol. 44, no. 7, pp. 659–669, 2022.

[14] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.

[15] R. Lohiya and A. Thakkar, "Application domains, evaluation data sets, and research challenges of iot: A systematic review," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8774–8798, 2020.

[16] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection," *Journal of Network and Systems Management*, vol. 31, no. 1, pp. 1–23, 2023.

[17] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[18] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402, PMLR, 10–15 Jul 2018.

[19] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 9, 2012.

[20] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Transactions on Network and Service Management*, vol. 6, no. 2, pp. 110–121, 2009.

[21] S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.

[22] P. Arregoces, J. Vergara, S. A. Gutiérrez, and J. F. Botero, "Network-based intrusion detection: A one-class classification approach," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, IEEE, 2022.

[23] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Unsupervised machine learning techniques for network intrusion detection on modern data," in *2020 4th Cyber Security in Networking Conference (CSNet)*, pp. 1–8, IEEE, 2020.

[24] M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions," in *2015 11th International conference on mobile ad-hoc and sensor networks (MSN)*, pp. 102–107, IEEE, 2015.

[25] A. R. Vasudevan and S. Selvakumar, "Local outlier factor and stronger one class classifier based hierarchical model for detection of attacks in network intrusion detection dataset," *Frontiers of Computer Science*, vol. 10, no. 4, pp. 755–766, 2016.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, pp. 413–422, IEEE, 2008.

[27] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," tech. rep., Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering, 2003.

[28] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[29] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.

[30] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, 2022.

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[32] L. Deri and N. SpA, "nprobe: an open source netflow probe for gigabit networks," in *TERENA Networking Conference*, pp. 1–4, 2003.