

A Novel Contrastive Loss for Zero-Day Network Intrusion Detection

Jack Wilkie, Hanan Hindy, Craig Michie, Christos Tachtatzis, James Irvine, Robert Atkinson

Abstract—Machine learning has achieved state-of-the-art results in network intrusion detection; however, its performance significantly degrades when confronted by a new attack class—a zero-day attack. In simple terms, classical machine learning-based approaches are adept at identifying attack classes on which they have been previously trained, but struggle with those not included in their training data. One approach to addressing this shortcoming is to utilise anomaly detectors which train exclusively on benign data with the goal of generalising to all attack classes—both known and zero-day. However, this comes at the expense of a prohibitively high false positive rate. This work proposes a novel contrastive loss function which is able to maintain the advantages of other contrastive learning-based approaches (robustness to imbalanced data) but can also generalise to zero-day attacks. Unlike anomaly detectors, this model learns the distributions of benign traffic using both benign and known malign samples, i.e. other well-known attack classes (not including the zero-day class), and consequently, achieves significant performance improvements. The proposed approach is experimentally verified on the Lycos2017 dataset where it achieves an AUROC improvement of .000065 and .060883 over previous models in known and zero-day attack detection, respectively. Finally, the proposed method is extended to open-set recognition achieving OpenAUC improvements of .170883 over existing approaches.¹

Index Terms—Internet of Things, Network Intrusion Detection, Machine Learning, Contrastive Learning

I. INTRODUCTION

NETWORK intrusion detection systems (NIDS) are a crucial component in any cyber defence strategy. They monitor communications within a network and flag potentially malicious traffic (cyber-attacks) to a CyberSecurity Operations Centre (CSOC) for further investigation. Machine Learning (ML) has become the dominant paradigm in NIDS [1]. When trained on a sufficiently large compilation of examples, ML-based approaches are able to learn the distributions of benign and malicious traffic and accurately discriminate between them [2]. However, this approach has its own limitations: they require high volumes of training data, moreover traditional supervised classifiers are able only to identify traffic classes on which they have been trained, and so are vulnerable to zero-day attacks [3].

Methods based on anomaly detection, such as Autoencoders and Support Vector Machines (SVMs) [4] [5], help alleviate

Jack Wilkie, Craig Michie, Christos Tachtatzis, James Irvine and Robert Atkinson are with the Department of Electronics and Electrical Engineering, University of Strathclyde, Glasgow, United Kingdom.

Hanan Hindy is with Faculty of Computer and Information Sciences, Ain Shams University, Egypt.

¹The implementation and experiments are open-sourced and available at: <https://github.com/jackwilkie/CLOSR>

this shortcoming of supervised classifiers. These methods operate by learning the distribution of exclusively benign traffic, and flag traffic which significantly differs from the learned distribution as potential intrusions. As these models are trained only on benign data, which is abundantly available, they relieve the burden of collating a significant volume of labelled examples. Furthermore, by training exclusively on benign examples, these models are theoretically equally able to detect well-known and zero-day attacks alike. However, they are also unable to differentiate between new benign traffic subclasses, which leaves them prone high false positive rates, resulting in them being unsuitable for practical applications. While Open-Set Recognition (OSR) models such as Deep Open Classification [6] and OpenMax [7] aim to extend supervised classifiers to also detect novel classes, they have substandard performance in detecting zero-day attacks when applied to NIDS.

Recently, contrastive learning has emerged as a promising approach for NIDS. Methods such as Siamese Networks [8], [9] and Triplet Networks have shown promising results, motivated by their performance on imbalanced training data. Network traffic and hence training data is inherently imbalanced: perhaps a fraction of 1% of which represents an attack. This class imbalance can significantly affect the performance of classically trained machine learning models; contrastive approaches have shown great resilience to imbalanced training data [8]. Despite their ability to train well on highly imbalanced data, like traditional models, contrastive models suffer performance degradation when faced with zero-day attacks [10].

In this work, Contrastive Learning for Anomaly Detection (CLAD) is proposed as a framework capable of modelling the distribution of benign network traffic in embedded space as a von-Mises Fisher (vMF) distribution. Unlike anomaly detectors it learns this distribution by training on both benign and malicious traffic giving it enhanced attack detection at low false-positive rates. As CLAD does not explicitly model the distribution of malicious traffic (as is typical of binary classifiers), it is able to generalise to zero-day attacks. The CLAD framework is then extended to OSR giving the Contrastive Learning for Open-Set Recognition (CLOSR) framework. This is achieved by independently modelling the distribution of each known class in a distinct embedded subspace allowing it to perform multiclass classification. Furthermore, zero-day attacks manifest as vectors orthogonal to the class centroids allowing to them to be easily identified.

The contributions of this work are threefold: (i) a contrastive loss for binary classification that exclusively learns the benign distribution while still training on malicious traffic; (ii) an

extension of this loss to open-set recognition, enabling multiclass classification with unknown-class rejection; and (iii) a comprehensive empirical comparison against anomaly detection, supervised, and OSR baselines on Lycos2017, demonstrating consistent gains on both known and zero-day attacks.

II. RELATED WORKS

A. Anomaly Detection

Anomaly detection involves understanding the typical patterns of a dataset during training and then identifying deviations from this pattern during inference. In NIDS, this approach is used to train a model on the normal behaviour of network traffic, allowing it to recognise any deviations as malicious. Since these models are trained solely on benign data, they can theoretically identify zero-day as well as known attacks. These models can be used as stand-alone classifiers or alternatively can be combined with traditional rules based or signature-based approaches [11], [12].

Distance-based methods represent one of the simplest approaches to anomaly detection. These methods typically calculate the centroid of benign training data and identify anomalies based on their Minkowski distance from this centroid [13]. The greater the distance, the more likely a data point is to be an anomaly. Alternative distance measures, such as Frobenius and Grassmannian distances, have also been employed [14]. Instead of comparing a samples distance to a centroid, other works have used a samples local outlier factor and the distance to its K th nearest neighbour [15]. Most recently, DeepSVDD has attempted to parameterise these methods by learning a mapping from raw feature space to an embedded space in which benign data is constrained to a hypersphere [16]. However, these methods have been shown to require a large amount of data to be effective [17]. Furthermore, selecting a distance metric can be challenging and choices often don't generalise to zero-day attacks [18].

Another approach extends discriminative models to perform anomaly detection. One class SVMs extend SVMs to anomaly detection by learning a boundary which encompasses benign traffic during training. During inference, samples appearing inside this boundary are deemed to be benign; whilst samples outside the boundary are considered to be malicious [4] [5]. Additionally, Isolation Forests (IF), an extension of tree-based classification models, can be used to identify anomalies as samples which require many branches to isolate from the training dataset [19].

Deep learning methods have been extensively utilised for anomaly detection, with numerous studies employing various techniques [20], [21]. One common approach involves the use of autoencoders, which take an input and encode it into a lower-dimensional representation, before reconstructing it back to the original dimensionality. The model is trained using a reconstruction objective under the assumption that malicious samples will have a higher reconstruction error at test time [22].

Several variations of the standard autoencoder have been proposed to enhance performance. Sparse autoencoders remove the bottleneck layer and apply regularisation to prevent the model from learning the identity function [17]. Deep Unsupervised Anomaly Detection (DUAD) trains an autoencoder

while iteratively clustering the data to remove outliers [23]. DAE-LR adds a regularisation term to ensure that benign data remains compact in latent space [24]. Another variant, AutoSVM, applies an SVM to the bottleneck layer to perform classification [25]. Despite the widespread application of autoencoders in NIDS, challenges remain as certain attacks often exhibit low reconstruction errors, which can prevent them from being detected without a high false positive rate [22].

B. Open-Set Recognition

Open-Set Recognition is a subset of ML which aims to extend traditional classifiers, which discriminate between a closed-set of known classes, to also identify samples of class distributions which were not represented in the training data. This is applicable to NIDS where it is beneficial for a classifier to be able to identify zero-day attacks in addition to identifying the type of traffic amongst a set of known classes.

One approach to OSR, is to employ a multistage classification framework, which trains both an anomaly detector and closed set classifier. Samples are then either flagged as out of distribution (OOD) or assigned their predicted closed-set class label based on thresholds set on both the anomaly detector and the classifier's predictions. In NIDS, an autoencoder and random forest have found to be effective for the anomaly detector and classifier, respectively [26]. Deep Open Classification (DOC) instead trains a classifier for OSR in an end-to-end manner [6]. Each known class in the dataset is assigned its own logit and sigmoid activation in the output layer, with each being trained in a one-vs-rest fashion. After training, a Gaussian distribution is fit onto the sigmoid activations for each class, with a sample's likelihood under these distributions being used to detect zero-day attacks at test time.

Another line of work aims to extend pretrained closed-set classifiers to OSR. OpenMax does this by fitting a Weibull distribution to the distances between the activation vectors from the penultimate layer of a neural network and the mean activation vector for each known class [7]. During inference, an OOD score is calculated using the CDFs of the Weibull distributions. CROSR is an extension of OpenMax where a deep hierarchical reconstruction network (DHRN) is trained instead of a standard MLP [27]. G-OpenMax, another extension of OpenMax trains a neural network with synthesised data as the unknown class [28].

Finally, OSR can be performed using statistical analysis in the feature space. Centroid methods compare the features representation to the mean values using a similarity metric [29]. Open set nearest neighbours considers the ratio of the distance from a sample to its nearest class relative to that of the second-nearest class [30]. Weibull SVMs calibrate the predictions of one class SVMs using Weibull distributions [31]. Extreme value machines derive a density function from extreme values theory and use this to model class belongingness probabilities [32].

C. Contrastive Learning

Contrastive learning is a technique in machine learning that trains models to learn an embedded representation of the input data in which similar samples are close together, whilst being

separated from dissimilar samples. The simplest form of this technique is exemplified by Siamese Networks, which utilise a contrastive loss function designed to minimise distances between similar pairs and maximise those between dissimilar pairs [33]. In NIDS, a Siamese network was initially trained in a supervised fashion, where tabular samples of the same class were treated as similar and different classes were dissimilar [8]. It was later extended to few-shot classification, where the Siamese network was able to identify novel attacks without retraining, given a small number of reference samples [9]. Since then, contrastive learning has been shown to be effective when dealing with class imbalance [34] and datasets of limited size [10].

Due to its effectiveness, contrastive learning has been gaining popularity in NIDS. RENOIR is a contrastive model which employs two autoencoders, one of which has been trained on benign traffic and one which has been trained on malicious traffic. The distance between a sample and its corresponding reconstruction is minimised, whilst that between the sample and the other reconstruction is maximised [35]. FeCo trains an autoencoder using the InfoNCE loss function in a federated learning regime where the model can learn from multiple clients [36]. In online learning regimes contrastive learning has been used to pseudo-label traffic flows, these are then used to periodically retrain the model [37]. Other approaches have used image representations in place of tabular data [38].

Finally, contrastive self-supervised learning (SSL) approaches are capable of learning features from unlabelled or benign traffic which can then be used for anomaly detection or fine-tuned for classification. SSCL-IDS minimises the distance between two masked views of an input samples whilst maximising it between other samples [39]. Conflow minimises the distance between two views of a sample generated by passing it through a model twice, each time using a different dropout mask [34]. CLDNN trains a contrastive model using feature masking to similar pairs and using other samples as dissimilar pairs [40].

III. PROPOSED APPROACH

A. System Overview

This section introduces Contrastive Learning for Anomaly Detection (CLAD) and its extension, Contrastive Learning for Open-Set Recognition (COSR), designed for detecting zero-day attacks in binary classification and OSR scenarios, respectively. As illustrated in Figure 1 (top), CLAD trains a neural network to model benign network traffic as a vMF distribution in embedded space. After training, the centroid of this distribution is computed and its cosine distance from test samples is used as a binary classification score at inference time. By leveraging contrastive learning, CLAD avoids explicitly modelling malicious traffic and thus does not make the closed world assumption, despite training on malicious samples. This results in significantly improved known and zero-day attack detection over traditional supervised classifiers and anomaly detectors.

COSR, shown in Figure 1 (bottom), extends CLAD to the OSR setting by introducing class-wise linear projections. These

projections are optimised independently using the CLAD loss function, enabling the modelling of each known class as an independent vMF distribution in a distinct embedded subspace. Multiclass classification is then performed by identifying the class centroid with the least cosine similarity to the test embedding. Importantly, zero-day attacks manifest as vectors orthogonal to class centroids, making them easily distinguishable from known classes.

B. Contrastive Learning for Anomaly Detection

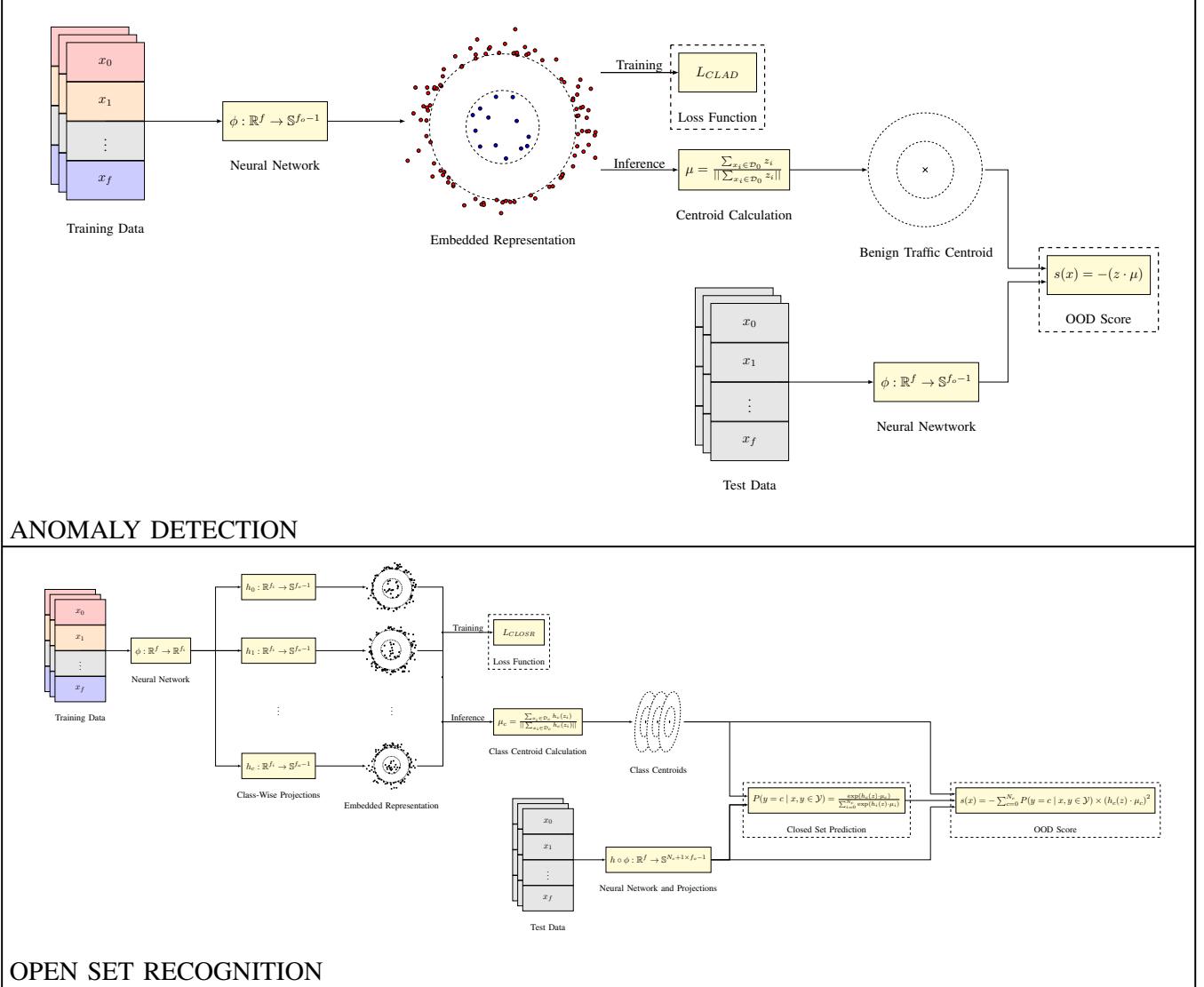
In NIDS the objective of machine learning based binary classifiers is to leverage a training dataset to learn decision boundaries between benign and malicious samples, such that malicious traffic can be identified during inference. Concretely, given a training dataset $\mathcal{D}_{\text{train}} := \{(x_i, y_i) \mid x \in \mathbb{R}^f, y \in \mathcal{Y}\}_{i=1}^{N_{\text{train}}}$ containing $N_{\text{train}} \in \mathbb{Z}^+$ recorded network flows—where x is a set of $f \in \mathbb{Z}^+$ tabular features representing each flow and $\mathcal{Y} := \{0, \dots, N_c\}$ is set of class labels representing benign traffic ($y = 0$) and $N_c \in \mathbb{Z}^+$ known malicious classes of traffic ($y \neq 0$)—a decision function must be learned to predict the binary class labels $y' := \mathbb{1}[y \neq 0] \in \{0, 1\}$ of unknown flows at test time. Here $\mathbb{1}$ represents an indicator function which returns 1 when the condition is true and returns 0 otherwise.

The most common solution is to train a parameterised neural network, $\phi_\theta : \mathbb{R}^f \rightarrow \mathbb{R}$ with parameters θ , to minimise the binary cross-entropy (BCE) loss function, which is equivalent to maximising the log-likelihood of the conditional distribution $P(y' \mid x; \theta)$ under a Bernoulli model. In practice, it is calculated over a batch $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ sampled from the training dataset as shown in Equation 1, where $\hat{y}_i := \sigma(\phi(x_i))$ is the predicted probability that flow x_i is malicious and $\sigma : \mathbb{R} \rightarrow [0, 1]$ is the sigmoid activation function. The model parameters θ have been omitted for simplicity.

$$L_{\text{BCE}} = -\frac{1}{|\mathcal{B}|} \sum_{(x_i, y_i) \in \mathcal{B}} [y'_i \log \hat{y}_i + (1 - y'_i) \log(1 - \hat{y}_i)] \quad (1)$$

NIDS trained using the BCE loss aim to directly model the conditional label distribution $P(y' \mid x)$. However, when faced with imbalanced data, these models often exhibit poor performance due to implicit biases toward the majority class, reflecting the marginal label distribution $P(y')$ in practice. Moreover, this approach typically assumes a symmetry between known and zero-day attacks where $P(y' \mid x, y \in \mathcal{Y}) = P(y' \mid x)$. This symmetry does not hold in practice, limiting the model's ability to generalise to zero-day attacks.

To circumvent the implicit learning of $P(y')$, supervised contrastive learning utilises the pairwise interactions between samples to estimate the density ratio $\frac{P(x|y)}{P(x)}$ from which a value proportional to $P(y \mid x)$ can be recovered under a uniform prior for $P(y)$. The supervised contrastive loss function is given in Equation 2 [41]. Here $\phi : \mathbb{R}^f \rightarrow \mathbb{S}^{f_\phi-1}$ is a neural network mapping the data from feature space to an embedded representation of dimensionality $f_\phi \in \mathbb{Z}^+$ such that $\|\phi(x)\|_2 = 1 \quad \forall x \in \mathbb{R}^f$. $\mathcal{P}(i) := \{x_j \in \mathcal{B} \mid j \neq i, y_j = y_i\}$ denotes the set of positive samples for anchor x_i , and $\mathcal{A}(i) := \{x_j \in \mathcal{B} \mid j \neq i\}$ denotes all other samples in the batch. Finally, $\tau \in \mathbb{R}^+$



OPEN SET RECOGNITION

Fig. 1: Wholistic overview of the proposed approach. **Top:** The CLAD loss function learns the distribution of benign traffic as an embedded vMF distribution. During inference the cosine distance between test representations and the centroid of this distribution is used to identify anomalous traffic. **Bottom:** The CLOSR framework extends CLAD to OSR by modelling the distribution of each known traffic class as an independent vMF distribution in a class specific embedded subspace. Closed set can be inference can then be performed by identifying the distribution with the highest cosine similarity to the test embedding, while zero-day attacks manifest as orthogonal vectors. Here the vMF distributions have been flattened for illustrative purposes.

is the temperature hyperparameter. The notation $z_i := \phi(x_i)$ is used for simplicity.

$$L_{\text{SupCon}} = \frac{1}{|\mathcal{B}|} \sum_{(x_i, y_i) \in \mathcal{B}} \frac{1}{|\mathcal{P}(i)|} \sum_{x_p \in \mathcal{P}(i)} -\log \left(\frac{\exp(z_i \cdot z_p / \tau)}{\sum_{x_a \in \mathcal{A}(i)} \exp(z_i \cdot z_a / \tau)} \right) \quad (2)$$

While contrastive learning has achieved improved performance in NIDS tasks due to its ability to model imbalanced data; existing methods maintain the symmetry assumption between known and unknown malicious classes and thus still suffer from reduced performance when faced with zero-day attacks. This work instead proposes the CLAD framework,

which learns the density ratio $\frac{P(x | y=0)}{P(x | y \neq 0)}$ while explicitly modelling $P(x | y = 0)$ as a vMF distribution, which has been shown in computer vision literature to be well suited for OOD detection [42]. As the density ratio is asymmetrically computed over benign traffic, the model exclusively learns the distribution of benign traffic, with malicious samples being used only to normalise the benign distribution. This relaxes the closed world assumption akin to anomaly detectors (i.e. $P(x | y' = 0, y \in \mathcal{Y}) = P(x | y' = 0)$). Furthermore, by explicitly modelling the data as vMF distributions, a structure is forced on embedded representation of the data allowing for zero-day attacks to be detected.

Additionally, it should be noted that when a shared embedding space is used for many classes, the optimal arrangement of class centroids on the hypersphere approaches a regular simplex configuration, in which the pairwise cosine similarity is fixed

at $\frac{1}{N_c+1}$. This optimal packing reduces the angular separation between centroids as the number of classes increases, which in turn lowers the margin between known and zero-day samples. By contrast, learning the distribution of only benign traffic maximises the margin between benign samples and zero-day attacks.

The CLAD loss function is derived by minimising the negative log-likelihood of benign samples under the density ratio $\frac{P(x|y=0)}{P(x|y \neq 0)}$, using vMF class-conditional distributions as shown in the loss function given in Equation 3. Here $\mu_0 \in \mathbb{S}^{f_o-1}$ and $\mu_{-0} \in \mathbb{S}^{f_o-1}$ are the centroids of the class-conditional distributions $P(x | y = 0)$ and $P(x | y \neq 0)$, respectively. The vMF normalisation terms $C_{f_o}(\kappa_0)$ and $C_{f_o}(\kappa_1)$ have been omitted due to forming the additive constant $-\log(\frac{C_{f_o}(\kappa_0)}{C_{f_o}(\kappa_1)})$, which is eliminated upon differentiation. The notations $\mathcal{B}_c := \{x_j \in \mathcal{B} \mid y_j = c\}$ and $\mathcal{D}_c := \{x_j \in \mathcal{D}_{\text{train}} \mid y_j = c\}$ are used to represent the subset of samples in the batch and training dataset belonging to class $c \in \mathcal{Y}$, respectively. Finally, $d : \mathbb{S}^{f_o-1} \times \mathbb{S}^{f_o-1} \rightarrow [0, 1]$ is the rescaled cosine distance function shown in Equation 4 for two unit vectors z and z' .

$$L = \frac{-1}{|\mathcal{B}_0|} \sum_{x_i \in \mathcal{B}_0} \log \left(\frac{\exp(-\kappa_0 d(z_i, \mu_0))}{\exp(-\kappa_1 d(z_i, \mu_{-0}))} \right) \quad (3)$$

$$d(z, z') = \frac{1 - (z \cdot z')}{2} \quad (4)$$

To circumvent the need for computing centroids over the entire training dataset during optimisation, the relationship between an embedding and samples drawn from a vMF distribution, as defined in Equation 5, is leveraged. This relationship enables the centroids of the vMF distributions to be approximated via Monte Carlo sampling using samples within the current batch, as illustrated in Equation 6, where $\mathcal{N}(i) := \{x_j \in \mathcal{B} \mid j \neq i, y_j \neq y_i\}$ denotes the set of negative pairs for anchor x_i and $\kappa_{y'} \in [0, \infty)$ is class-specific concentration parameter of vMF distribution associate with binary class label y' .

$$d(z, \mu) \propto \mathbb{E}_{z' \sim \text{vMF}(\mu, \kappa)} [d(z, z')] \quad (5)$$

$$L = \frac{-1}{|\mathcal{B}_0|} \sum_{x_i \in \mathcal{B}_0} \log \left(\frac{\exp \left(\frac{-\kappa_0}{|\mathcal{P}(i)|} \sum_{x_p \in \mathcal{P}(i)} d(z_i, z_p) \right)}{\exp \left(\frac{-\kappa_1}{|\mathcal{N}(i)|} \sum_{x_n \in \mathcal{N}(i)} d(z_i, z_n) \right)} \right) \quad (6)$$

Finally, simplifying Equation 6 gives the final CLAD loss function given in Equation 7. Here the concentration terms have been tied such that $\kappa_0 = \kappa_1$ and removed as a multiplicative constant, which is absorbed into the learning rate during hyperparameter optimisation. In this formulation, the distance terms are squared to enhance the separation between benign and malicious traffic in the embedded space.

$$L_{\text{CLAD}} = \frac{1}{|\mathcal{B}_0|} \sum_{x_i \in \mathcal{B}_0} \left[\frac{1}{|\mathcal{P}(i)|} \sum_{x_p \in \mathcal{P}(i)} d(z_i, z_p)^2 + \sum_{x_n \in \mathcal{N}(i)} \frac{1}{|\mathcal{N}(i)|} (1 - d(z_i, z_n))^2 \right] \quad (7)$$

The CLAD loss is reminiscent of the hinge-based contrastive loss function [33], however, differs in several fundamental ways. First, CLAD is evaluated exclusively on benign anchor samples, whereas the contrastive loss optimises learns the distribution of each known class symmetrically. This removes the closed-world assumption inherent in contrastive learning and aligns CLAD with anomaly-style training objectives. Second, CLAD can be viewed as fixing the contrastive loss' margin term to $m = 1.0$ which forces benign and malicious traffic to converge to antipodal regions of the hypersphere and eliminates the hinge regularisation. Third, unlike the geometric motivation of contrastive loss, CLAD admits a probabilistic interpretation via a vMF likelihood ratio, yielding a density-ratio estimator rather than a geometric separation loss

It is pertinent to note that squaring the distance metric preserves the bound range of the rescaled distance metric i.e. $d^2 : \mathbb{S}^{f_o-1} \times \mathbb{S}^{f_o-1} \rightarrow [0, 1]$. By squaring the distance metric the gradient magnitudes are scaled by factors of $2|d(z, z')|$ and $2|d(z, z') - 1|$ for positive and negative pairs, respectively. This results in larger gradients for positive pairs with large separations and for negative pairs with poor separation, giving a more discriminative embedded representation. This is similar to the temperature scaling used by the SupCon loss function [41], where sharper gradients improve class separability.

C. Outlier Detection

CLAD models benign network traffic as a vMF distribution in embedded space. This allows for inference to be performed by evaluating the likelihood of a test embedding under the learned benign vMF distribution. To facilitate this the mean direction, $\mu_0 \in \mathbb{S}^{f_o-1}$, of the benign distribution is calculated as the centroid of benign data, normalised to lie on the unit hypersphere as shown in Equation 8.

$$\mu = \frac{\sum_{x_i \in \mathcal{D}_0} z_i}{\|\sum_{x_i \in \mathcal{D}_0} z_i\|} \quad (8)$$

The cosine similarity between a test embedding and the distribution centroid is proportional to the likelihood $P(x | y = 0)$. In principle, this can be converted to a conditional probability $P(y = 0 | x)$ using Bayes' theorem; however, computing the marginal likelihood $P(x)$ requires integrating over all possible class-conditional distributions, which is only possible under a closed world assumption. Thus, CLAD instead uses the negative cosine similarity between a test embedding and the centroid as an OOD score, $s(x) \in [-1, 1]$, given by Equation 9. This score is proportional to the negative log-likelihood of the sample under the vMF distribution, up to an additive constant. This is in line with prior work in OOD detection for computer vision [43], where log-likelihoods are widely used as effective OOD scores.

$$s(x) = -(z \cdot \mu) \quad (9)$$

Binary classification can be then made by placing a threshold, $\tau \in [-1, 1]$ on the OOD score. As shown in Equation 10: test samples with an OOD score less than this threshold are deemed to be malicious, while those with an OOD score less than or equal to the threshold are considered benign. It is important to note that this work employs threshold independent metrics such as AUROC and FPR@95 and thus does not calibrate the value of the threshold. In practical scenarios the network's CSOC would determine an acceptable false positive rate based on available resources and select a corresponding value using a held-out validation set of benign traffic.

$$\hat{y} = \begin{cases} 0, & \text{if } s(x) < \tau; \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

D. Extension to Open-Set Recognition

The CLAD loss function learns to discriminate the distribution of benign traffic from other classes, resulting in binary predictions indicating whether the unknown sample is benign or malicious. In practical scenarios, it is often beneficial not only to know that traffic is malicious, but also the type of attack being performed. To achieve this, CLAD is first modified to learn the distribution of an arbitrary class, $c \in \mathcal{Y}$, as shown in Equation 11. To facilitate this, the neural network, $\phi : \mathbb{R}^f \rightarrow \mathbb{R}^{f_i}$, projects the data to an intermediate representation, $z_i = \phi(x_i)$, which is then mapped to distinct unit hyperspheres using class-wise linear projection heads, $\{h_c : \mathbb{R}^{f_i} \rightarrow \mathbb{S}^{f_o-1}\}_{c=0}^{N_c}$. These heads are independently optimised to map samples onto a unique hypersphere, where its associated class distribution is modelled using a vMF distribution.

$$\begin{aligned} L_{CLAD}(c) = & \frac{1}{|\mathcal{B}_c|} \sum_{x_i \in \mathcal{B}_c} \left[\frac{1}{|\mathcal{P}(i)|} \sum_{x_p \in \mathcal{P}(i)} d(h_c(z_i), h_c(z_p))^2 \right. \\ & \left. + \sum_{x_n \in \mathcal{N}(i)} \frac{1}{|\mathcal{N}(i)|} (1 - d(h_c(z_i), h_c(z_n)))^2 \right] \end{aligned} \quad (11)$$

The CLOSR loss function can then be defined as the CLAD loss computed over each class, where each class employs its distinct linear projection. The CLOSR loss is formally described by Equation 12.

$$L_{CLOSR} = \sum_{c \in \mathcal{Y}} L_{CLAD}(c) \quad (12)$$

After training, a mean direction (centroid) is computed for each class, $\mu \in \mathbb{S}^{N_c+1 \times f_o-1}$; where μ_c is the centroid of class c . As shown in Equation 13, this is calculated as the mean of all embeddings of a given class in the training data, in its respective linear projection, and then normalised to lie on a unit hypersphere.

$$\mu_c = \frac{\sum_{x_i \in \mathcal{D}_c} h_c(z_i)}{\|\sum_{x_i \in \mathcal{D}_c} h_c(z_i)\|} \quad (13)$$

By comparing the cosine similarity between an unknown test embedding and the centroid of each class's vMF distribution, unnormalised class-conditional likelihoods $P(x | y = c)$ can be computed for each class. These can be converted to a closed-set prediction under a uniform prior with Bayes' theorem using the softmax function as shown in Equation 14.

$$P(y = c | x, y \in \mathcal{Y}) = \frac{\exp(h_c(z) \cdot \mu_c)}{\sum_{i \in \mathcal{Y}} \exp(h_i(z) \cdot \mu_i)} \quad (14)$$

As the model is not explicitly trained on zero-day attacks, these samples can be assumed to be more uniformly distributed over the unit hypersphere than those from known attack classes. Under this assumption, zero-day embeddings exhibit many weakly correlated dimensions, allowing the cosine similarity between a zero-day sample and a class mean to be approximated as a sum of independent random variables, each weighted by the corresponding component of the class centroid. By the Central Limit Theorem, this sum converges to a Gaussian distribution with zero mean and variance $1/f_o$, as formalised in Equation 15.

$$h_c(z) \cdot \mu_c = \sum_{i=1}^{f_o} h_c(z)_i \mu_{c,i} \sim \mathcal{N}\left(0, \frac{1}{f_o}\right) \quad (15)$$

Following Equation 15, an OOD score, $s(x) \in [-1, 0]$, is defined in Equation 16. This is proportional to the log-likelihood of a test embedding belonging to the Gaussian distribution up to an additive constant. A high score is expected to indicate a zero-day attack.

$$s(x) = - \sum_{c=0}^{N_c} P(y = c | x, y \in \mathcal{Y}) \times (h_c(z) \cdot \mu_c)^2 \quad (16)$$

While this weighted sum assumes isotropy and independence amongst the cosine similarity measurements; these assumptions are explicitly supported in this setting due the construction of the embedding space and the properties of the hyperspherical distributions. Isotropy is ensured under the assumption of zero-day embeddings being uniformly distributed over the unit hypersphere. In this case, the projection of a random unit vector onto any fixed unit direction follows a Gaussian distribution with zero mean and constant variance $1/f_o$, regardless of the class. Furthermore, the independence assumption is supported by the architecture: each class-wise linear projection head is independently optimised using a one-vs-rest strategy allowing for them to be treated as independent. This is a common assumption made by ensemble modelling and mixture of experts architectures [44], [45], [46].

Finally, a threshold, $\tau \in [-1, 0]$, is placed on the OOD score. As shown in Equation 17, samples with an OOD score above the threshold are predicted to be the closed-set class with the highest predicted probability; whilst those with an OOD score equal to or less than the threshold are predicted as belonging to a previously unseen class ($\hat{y} = -1$).

$$\hat{y} = \begin{cases} -1, & \text{if } s(x) > \tau; \\ \text{argmax}(P(y | x, y \in \mathcal{Y})), & \text{otherwise.} \end{cases} \quad (17)$$

It should be noted that, similar to the binary classification threshold, the threshold τ is not calibrated in this work due to the use of threshold independent evaluation metrics. In practical deployments, this would be tuned on a held-out closed-set validation set based on acceptable error rates defined by the corresponding cybersecurity operations centre (CSOC).

The CLAD loss function is a special case of the CLOSR loss. Given an $N = N_c + 1$ way classification task, CLOSR explicitly models N_c known class distributions with the remaining class being treated as OOD samples. For an $N = 2$ way classification task, a single benign class distribution is learned (with known malicious and OOD traffic being considered a single class), yielding the CLAD training objective exactly.

E. Network Architecture

The CLAD and CLOSR loss functions are parameterised by a Multi-Layer Perceptron (MLP), notated as ϕ . The proposed architecture initially projects the input feature vector to a higher-dimensional space of dimensionality $d_{model} \in \mathbb{Z}^+$ using a linear transformation. Subsequently, a non-linear function is learned through a series of sequential MLP blocks. Each block comprises a linear transformation, which maintains the width d_{model} , followed by a ReLU activation function. By using a fixed width across all MLP blocks, the model architecture is simplified, requiring only a single width instead of a distinct width for each layer.

Finally, a linear projection head is used to downsample the data to its embedded dimensionality and project it onto a unit hypersphere. While the CLAD loss function employs a single projection head, the CLOSR loss function introduces multiple projection heads in order to facilitate the training of class-wise subspaces in a one-vs-rest manner. It should be noted that while the CLAD loss function introduces sampling noise through its Monte Carlo approximation, training remained stable without introducing normalisation or variance reduction techniques to the architecture.

IV. COMPARISON TO BASELINE MODELS

A. Experimental Procedure

This work trains and evaluates models using the Lycos2017 dataset [2], which contains 1,789,954 flows across 14 classes of traffic, including both malicious and benign samples. The dataset is highly imbalanced, with benign traffic consisting of over 1,000,000 samples, and malicious classes ranging from as few as 11 samples to as many as 100,000 samples. The dataset was split into a 50%-50% train-test sets, ensuring that each set contained an even number of samples of each class. The Web Attack (SQL Injections) and Heartbleed classes were an exception, as they were only contained in test set to simulate zero-day attacks.

In this section CLAD and CLOSR are evaluated and compared to baseline models from anomaly detection, supervised classification, and OSR. To ensure fair evaluation, the proposed approaches and baseline models were optimised and evaluated using an identical procedure: machine learning models were trained for 200 epochs using the AdamW optimiser [47]. The learning rate was scheduled using a linear warmup for 20

epochs followed by cosine annealing [48]. Weighted class balancing was also used to sample batches. The base learning rate, batch size, weight decay, dropout rate and the model's width and depth were tuned as hyperparameters. The cosine distance metric was used for all models.

The hyperparameters of gradient based approaches were optimised using 200 iterations of random search, with each iteration employing 5-fold of cross-validation across the training data. Here 200 iterations were used for each model due to the available compute budget. Standard search ranges were used for each parameter, with search ranges being kept constant across models. Model specific hyperparameters were searched across parameter ranges suggested in the original works. For gradient-free approaches it was feasible to perform hyperparameter optimisation using a grid search across typical hyperparameter ranges. For fair comparison with the contrastive loss function, CLAD and CLOSR were trained using an equivalent margin and hinge regularisation term allowing for an identical search space, however, all optimised models converged to a margin value of $m = 1.0$.

After hyperparameter optimisation, the configuration with the greatest mean AUROC across each cross-validation fold was selected for training on the entire training dataset and evaluation on the test set. Mean test set results over 20 training and evaluation runs are reported. Additionally, statistical testing was performed between CLAD/CLOSR and the leading baseline model using a Wilcoxon signed-rank test, with statistical significance being indicated where the p-value was below the 0.01 threshold.

B. Comparison to Binary Classification Models

The efficacy of CLAD when compared to baseline models in binary classification was primarily evaluated by measuring the model's area under the ROC curve (AUROC). This metric represents the expectation of the model assigning a higher classification score to a malicious sample than to a benign one, thereby quantifying the trade-off between true positive and false positive rates. The results are given in Table I (top) for known attack classes and Table I (bottom) for zero-day attack classes. Similarly, the FPR@95 metric, which is the model's false positive rate when classifying an attack class with a 95% recall, is reported for CLAD and baselines models for known traffic in Table II (top) and for zero-day attacks in Table II (bottom).

On known attack classes, anomaly detectors expectedly demonstrated poor performance when compared to supervised classifiers. As such, CLAD was found to significantly outperform anomaly detectors on known attacks. When compared to supervised classifiers CLAD achieved statistically significant performance improvements in AUROC. While CLAD outperformed supervised classifiers on known attacks in FPR@95; this improvement was not found to be statistically significant.

On zero-day attack detection, supervised classifiers suffer from significant performance degradation, indicating overfitting on known traffic. In contrast, anomaly detectors have circumvented this issue by training only on benign traffic, consequently outperforming the supervised classifiers. Remarkably, despite

TABLE I

AUROC comparison of CLAD and baseline models when detecting known (top) and zero-day (bottom) attacks. CLAD significantly outperforms supervised classifiers and anomaly detectors in zero-day attack detection. Best performance is shown in **bold** and * indicates statistical significance.

| Class | CLAD | DUAD | DAE-LR | Deep SVDD | AE | SVM | AutoSVM | IF | RENOIR | MLP | Siamese Network |
|----------------------------|-----------------|---------|---------|-----------|---------|---------|---------|---------|---------|---------|-----------------|
| Botnet | .999980 | .888620 | .720118 | .626321 | .648185 | .637679 | .662214 | .696588 | .999990 | .999982 | .999992 |
| DDoS | .999995 | .987811 | .998323 | .971817 | .957563 | .889780 | .722085 | .778166 | .999996 | .999995 | .999881 |
| DoS (Golden Eye) | .999585 | .930244 | .954869 | .740828 | .878592 | .846840 | .796121 | .727087 | .999425 | .999397 | .998539 |
| DoS (Hulk) | .999995 | .988089 | .993563 | .899340 | .950815 | .894898 | .793412 | .781731 | .999991 | .999985 | .999778 |
| DoS (Slow HTTP Test) | .999575 | .911725 | .975763 | .659927 | .978112 | .963021 | .925308 | .848820 | .999831 | .999543 | .999474 |
| DoS (Slow Loris) | .999876 | .947156 | .980669 | .692552 | .938116 | .896824 | .776275 | .775622 | .999928 | .999937 | .999826 |
| FTP Patator | .999986 | .982829 | .952138 | .822762 | .779748 | .736828 | .702073 | .759742 | .999991 | .999987 | .999948 |
| Portscan | .999993 | .943170 | .974972 | .732919 | .871651 | .741269 | .584936 | .742066 | .999986 | .999962 | .999823 |
| SSH Patator | .999930 | .961050 | .961583 | .671834 | .824424 | .799072 | .789865 | .816086 | .999951 | .999980 | .999978 |
| Web Attack (Brute Force) | .999754 | .836997 | .731959 | .635410 | .801017 | .767771 | .744257 | .748669 | .999312 | .998011 | .996850 |
| Web Attack (XSS) | .999732 | .909319 | .773056 | .640942 | .801203 | .761677 | .734927 | .768263 | .999286 | .998353 | .996466 |
| Closed Set Mean | .99985* | .935183 | .910637 | .735877 | .857221 | .812333 | .748316 | .767531 | .999790 | .999557 | .999141 |
| Heartbleed | .995557 | .987390 | .999798 | .985815 | .995161 | .993468 | .988778 | .955030 | .780736 | .071957 | .692232 |
| Web Attack (SQL Injection) | .997696 | .884098 | .777752 | .717808 | .686157 | .745087 | .767333 | .721814 | .991721 | .995762 | .997427 |
| Open Set Mean | .996627* | .935744 | .888775 | .851812 | .840659 | .869277 | .767333 | .838422 | .886228 | .533859 | .844829 |

TABLE II

FPR@95 comparison of CLAD and baseline models when detecting known (top) and zero-day (bottom) attacks. CLAD significantly outperforms supervised classifiers and anomaly detectors in zero-day attack detection. Best performance is shown in **bold** and * indicates statistical significance.

| Class | CLAD | DUAD | DAE-LR | Deep SVDD | AE | SVM | AutoSVM | IF | RENOIR | MLP | Siamese Network |
|----------------------------|-----------------|---------|---------|-----------|---------|---------|---------|---------|---------|---------|-----------------|
| Botnet | .000020 | .133028 | .334342 | .705342 | .382650 | .393574 | .367771 | .732462 | .000015 | .000024 | .000011 |
| DDoS | .000009 | .023125 | .004296 | .155205 | .068373 | .170113 | .323883 | .586338 | .000009 | .000008 | .000157 |
| DoS (Golden Eye) | .001110 | .187053 | .151281 | .750513 | .309403 | .252365 | .247377 | .554949 | .001085 | .002366 | .001829 |
| DoS (Hulk) | .000011 | .025627 | .014094 | .192490 | .068418 | .144633 | .230494 | .433950 | .000020 | .000051 | .000283 |
| DoS (Slow HTTP Test) | .000818 | .181274 | .038792 | .867094 | .081772 | .181849 | .231434 | .441208 | .000335 | .000738 | .000819 |
| DoS (Slow Loris) | .000036 | .202757 | .074781 | .860426 | .244468 | .365951 | .604420 | .702358 | .000091 | .000036 | .000175 |
| FTP Patator | .000013 | .017932 | .051922 | .493948 | .221201 | .263549 | .299008 | .664956 | .000006 | .000004 | .000057 |
| Portscan | .000008 | .125238 | .051744 | .702921 | .175783 | .306717 | .474476 | .634320 | .000013 | .000016 | .000218 |
| SSH Patator (Brute Force) | .000112 | .042739 | .042120 | .722980 | .185131 | .204481 | .211120 | .420270 | .000057 | .000031 | .000029 |
| Web Attack (Brute Force) | .000295 | .923579 | .795793 | .699538 | .315095 | .264761 | .275640 | .671678 | .000813 | .003889 | .005108 |
| Web Attack (XSS) | .000298 | .923851 | .795767 | .682162 | .206077 | .246975 | .273129 | .600777 | .000831 | .003952 | .005132 |
| Closed Set Mean | .000248 | .253291 | .214085 | .621147 | .205306 | .254088 | .321705 | .585752 | .000298 | .001010 | .001256 |
| Heartbleed | .007264 | .019323 | .000848 | .094894 | .022762 | .067708 | .105113 | .110316 | .339468 | .993871 | .326209 |
| Web Attack (SQL Injection) | .003331 | .218930 | .343934 | .601108 | .346540 | .286927 | .256526 | .657605 | .027203 | .005510 | .006520 |
| Open Set Mean | .005297* | .119126 | .172391 | .348001 | .184651 | .177318 | .180819 | .383960 | .183336 | .499691 | .166364 |

TABLE III

Comparison of the proposed CLOSR framework with baseline models in OSR. CLOSR outperforms existing models in both open set AUC and OpenAUC. Best performance is shown in **bold** and * indicates statistical significance.

| Model | Closed-Set Acc | Open-Set AUC | OpenAUC |
|-------------------------|-----------------|-----------------|-----------------|
| CLOSR | .995276 | .974022* | .969420* |
| MultiStage | .996612 | .801251 | .798537 |
| DOC | .995536 | .570263 | .567717 |
| OPENMAX | .995615 | .720174 | .717016 |
| CRSOR | .994940 | .748295 | .744509 |
| Siamese Network | .997722* | .720811 | .719167 |
| CLOSR + Siamese Network | .997722 | .974022 | .971803 |

TABLE IV

Closed set classification performance comparison between CLOSR and baseline models. Best performance is shown in **bold** and * indicates statistical significance.

| Model | Precision | Recall | PR-AUC | F1 Score | FP Rate |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| CLOSR | .818063 | .929404 | .870178 | .853196 | .005340 |
| MultiStage | .972191* | .884658 | .140567 | .889541 | .001893 |
| DOC | .832814 | .940472* | .873167* | .861716 | .004897 |
| OPENMAX | .835872 | .937882 | .865545 | .861687 | .004746 |
| CRSOR | .821883 | .938760 | .830984 | .853783 | .005658 |
| Siamese Network | .910641 | .897597 | .855822 | .903406* | .000712* |

training on malicious samples, CLAD maintains its strong performance against zero-day attacks. By leveraging the representations learned from both benign and malicious traffic, CLAD surpasses all baselines with statistically significant performance improvements on both AUROC and FPR@95. These results highlight the applicability of CLAD to the binary classification of both known and zero-day attack classes.

C. Comparison to Open-Set Recognition Models

The OSR performance of CLOSR was benchmarked against a range of baseline models, including state-of-the-art approaches

from both the NIDS and OSR literature. Models were evaluated using closed-set accuracy, open-set AUC, and OpenAUC [49], where OpenAUC is defined as the product of the former two metrics. In this context, open-set AUC quantifies the probability of the model correctly distinguishing test samples from unknown classes against those from the known class distributions. The results, reported in Table III, highlight the strong open-set performance of CLOSR. Notably, CLOSR achieves a substantially higher open-set AUC than all baseline methods. While this improvement is accompanied by a decrease in closed-set classification accuracy; CLOSR achieves the highest overall performance by a significant margin, as reflected

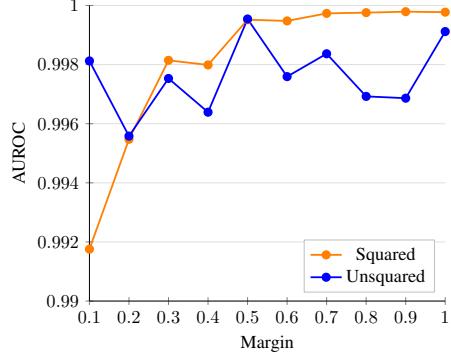


Fig. 2: AUROC of CLAD as the margin hyperparameter is varied in the range $[0.1, 1]$, with (orange) and without (blue) the distance terms being squared.

in its superior OpenAUC score.

It should be noted that CLOSR can be used as an independent out of distribution detector and paired with a supervised classifier which performs closed-set classification. In this regime, the combination of the CLOSR and the Siamese network achieves the open-set performance of CLOSR and the closed-set performance of the Siamese network, matching the best performance in both open and closed-set classification. For completeness, additional multiclass classification metrics are reported in Table IV. These results are consistent with the above findings: CLOSR exhibits slightly lower closed-set performance compared to baselines, yet its effectiveness in open-set scenarios makes it the best OSR model in practice.

V. ADDITIONAL RESULTS AND ABLATIONS

A. Experimental Procedure

In this Section results from additional experiments and ablation studies are reported. To facilitate this, the dataset was split into training and test sets identical to those used in Section IV. To avoid data leakage from the zero-day classes, the training data was further divided 80%-20% into training and validation sets, with models being trained on the training data, and validation performance being reported.

In order to enable fair comparison between CLAD and the contrastive loss function, and to ensure that ablations were unbiased by the design choices used during hyperparameter optimisation, models were trained using a fixed architecture and set of commonly used hyperparameters. An exception was made for Section V-G which reports the computational requirements of the optimised models.

B. CLAD Loss Ablations

To empirically evaluate design choices used by the CLAD loss function, a margin based hinge regularisation term was introduced equivalent to the contrastive loss function. CLAD models were then trained with the margin value being swept in the range $[0.1, 1]$ in increments of 0.1, with and without the distance terms being squared. The results, shown in Figure 2, demonstrate that squaring the distance terms in the CLAD loss yields smoother and more stable improvements in AUROC across margins, whereas the unsquared formulation exhibits

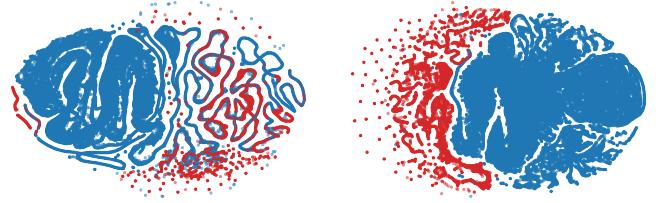


Fig. 3: T-SNE plots of the validation embeddings produced by **left**: the contrastive loss function and **right**: the CLAD loss function. Both loss functions produce distinct clusters for benign traffic (blue) and malicious traffic (red). The contrastive loss function embeddings have overlap between the clusters; while the CLAD loss function creates a clear separation between clusters.

fluctuations in performance. Additionally, when the distance terms were squared, there was an increasing trend in the model's performance as the margin value increases towards $m = 1.0$. This suggests that the performance is greatest when the model is trained to maximally separate benign and malicious traffic such that the respective samples lie antipodal on the unit hypersphere.

To evaluate the impact of tying the concentration terms such that $\kappa_0 = \kappa_1$, a hyperparameter $\alpha = \frac{\kappa_0}{\kappa_0 + \kappa_1}$ was defined, resulting in a loss function where positive-pair distances were scaled by α and negative-pair distances were scaled by a factor of $1 - \alpha$. The α parameter was then swept in the range $[0.1, 0.9]$. The mean AUROC across runs was found to be 0.999574 with a standard deviation of 0.000570 suggesting that CLAD is largely insensitive to the ratio of the concentration terms.

C. Embedding Analysis

A qualitative evaluation of the CLAD loss function was conducted by examining the output embeddings from the model's contrastive head. In Figure 3, the embeddings generated by CLAD are visualized using t-SNE dimensionality reduction and compared against those produced using the contrastive loss. Visual inspection reveals that both loss functions are able to separate benign (blue) and malicious (red) traffic into distinct clusters. However, the contrastive loss exhibits some overlap between classes, whereas CLAD achieves a much clearer separation.

This distinction reflects the core advantage of the CLAD loss: by explicitly modelling benign traffic as a vMF distribution, the model is encouraged to structure the embedding space such that the centroid becomes a statistically meaningful and robust class proxy. Rather than relying on incidental spatial structure, CLAD imposes a probabilistic constraint that promotes both intra-class compactness and inter-class separation. While t-SNE offers qualitative insight, the improved detection performance and lower false positive rate quantitatively support the effectiveness of this representation.

D. Alignment Analysis

A quantitative comparison between the CLAD and contrastive loss functions was conducted by evaluating their respective abilities to classify known and zero-day network traffic throughout training. To facilitate this analysis, two linear classification probes were appended to the model alongside the contrastive projection head. One probe was trained on known

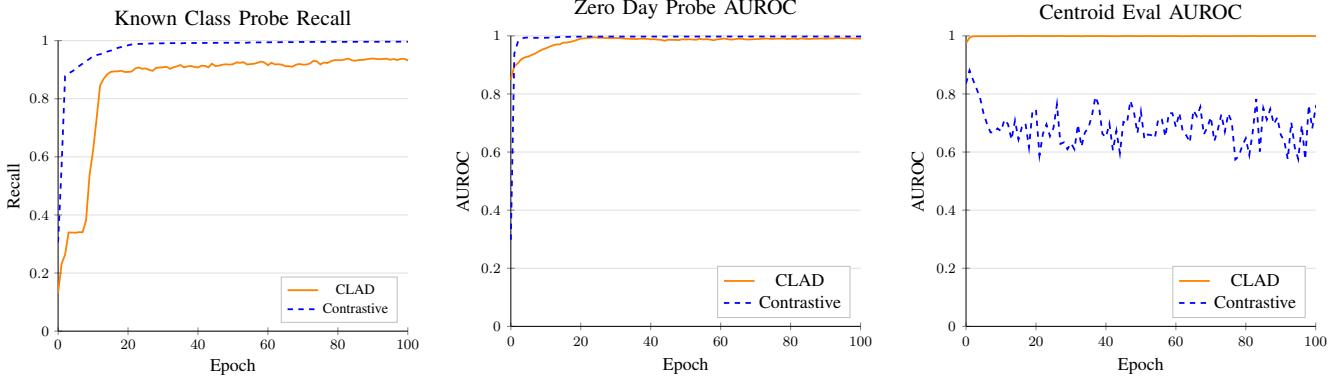


Fig. 4: Alignment analysis of the contrastive (blue) and CLAD (orange) loss functions when detecting known and zero-day attacks. **Left:** Balanced recall of the contrastive and CLAD loss functions amongst classes included in the training data when evaluated using a linear classification probe. **Centre:** Mean AUROC of the contrastive and CLAD loss functions when evaluated on zero-day using a linear probe. **Right:** Mean AUROC of the contrastive and CLAD loss functions throughout training when evaluated by measuring the distance between validation samples and the centroid of benign traffic.

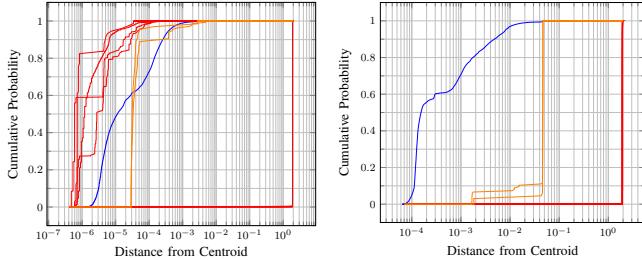


Fig. 5: Cumulative distribution function of the cosine distance between data points and the centroid of benign traffic when training a model using the CLAD and contrastive loss functions. **Left:** The contrastive loss function learns a representation in which malicious classes (red) are often closer to the centroid than benign samples (blue). **Right:** the CLAD loss function learns an embedded representation of the data in which benign samples (blue) appear close to the centroid, malicious samples (red) are far from the centroid, and zero-day attacks (orange) are orthogonal to the centroid.

attack classes in multiclass classification, whilst two additional malicious classes were separated from the training data and used to train the other probe as a binary classifier to identify zero-day attacks. Both probes were trained on embeddings after the gradient information had been detached, allowing them to evaluate the efficacy of the features learned by the model without impacting its weights. The performance of each probe was evaluated on the validation set. The performance of the contrastive head was measured by inferring class labels from the distance of validation samples to the centroid of benign embeddings in the training data, this method of evaluation is hereby referred to in the remainder of this work as centroid evaluation.

The results, given in Figure 4, indicate that optimising both the contrastive and the CLAD loss functions yields embedded representations where both known and zero-day classes are linearly separable. Notably, CLAD’s representations of different malicious distributions are linearly separable, despite being learned using binary class labels. These findings were further validated by employing a KNN classifier to assess the performance of the contrastive heads, where CLAD and the contrastive loss were found to have comparable mean recall scores of .8877 and .8769 respectively.

While both models learned representations where benign and malicious traffic was linearly separable, only CLAD’s contrastive head consistently improved in performance throughout training. In contrast, little correlation was observed between the performance of the contrastive loss function’s centroid evaluation performance and its training progress, indicating that malicious embeddings often appear closer to the benign class centroid than benign embeddings.

This phenomenon was further highlighted by plotting the cumulative distribution functions of the cosine distances from validation samples of each class to the benign centroid for both fully trained models, as shown in Figure 5. The contrastive loss function learned an embedded representation in which many known malicious (mean = 1.157, std = 0.926) and zero-day samples (mean = 2.458×10^{-5} , std = 4.415×10^{-5}) lie closer to the benign centroid than a substantial fraction of benign traffic (mean = 0.00185, std = 0.0587), explaining its poor performance when using centroid-based inference. Conversely, CLAD learned an embedded space in which benign traffic was tightly clustered around the centroid (mean 0.00378, std = 0.0634), with malicious traffic being maximally separated (mean = 1.922, std = 0.0263). Zero-day attacks occupied the intermediate region (mean = 0.0431, std = 0.0112), which is consistent with the orthogonal OOD score derived for CLOSR in Section III-D.

These findings demonstrate that while both the CLAD and contrastive loss functions generate discriminative features for identifying both known and zero-day attacks, the contrastive loss does not guarantee separation between malicious traffic and the benign class centroid. Hence, reference samples are required by the classifier to infer a zero-day attack’s position in embedded space. This can be achieved via training a linear probe, or without re-training by using a KNN classifier. Conversely, the CLAD loss function ensures a high expectation of malicious samples appearing further from the benign centroid than benign samples, allowing zero-day attacks to be successfully detected.

TABLE V
AUROC comparison of CLAD using various class proxies.

| Class Proxy | Mean AUROC |
|--------------------|----------------|
| Centroid | .999633 |
| Median | .999622 |
| Trimmed Mean | .999627 |
| Medoid | .999627 |
| Neighbour Distance | .645342 |

TABLE VI
Open-set AUC of CLOSR using various OOD scoring functions.

| OOD Score | Open-set AUC |
|-------------------|----------------|
| Energy | .979926 |
| Gaussian | .952326 |
| Weighted Gaussian | .993768 |

E. Class Proxy Ablation

CLAD relies on the distance between a test sample and the centroid of benign traffic to perform inference, however, centroid-based classification methods can be sensitive to noisy training data and adversarial samples. In Table V, the performance of CLAD is reported with alternative distance measures—specifically, the distance to the median, trimmed mean, medoid, and nearest neighbour. Amongst these, the centroid yields the best performance.

Interestingly, the nearest neighbour distance performs worse than the centroid distance in this context, which contrasts with previous contrastive and self-supervised learning approaches. This can be attributed to CLAD explicitly modelling the benign traffic as a vMF distribution, unlike other non-parametric approaches such as the SupCon loss function. While these results suggest the centroid remains effective under clean conditions, future work will aim to evaluate the robustness of CLAD to adversarial samples and polluted benign samples in the training dataset.

F. CLOSR Inference Ablation

To identify zero-day attacks in the embedding space, CLOSR computes an OOD score by evaluating the likelihood of a test sample under a set of Gaussian distributions defined over the class-specific subspaces. This formulation is motivated by the Central Limit Theorem and the assumption that zero-day attacks occupy a broader region of the hypersphere relative to known classes. To empirically validate this assumption, two malicious classes were withheld from the training data to simulate zero-day attacks. A CLAD model was then trained, and the output embedding matrices were analysed. The known-class validation embeddings exhibited a normalised rank of 0.0625, indicating that known traffic concentrates within a small subspace of the hypersphere. In contrast, the zero-day embeddings yielded a substantially larger normalised rank of 0.4375, supporting the hypothesis that their distribution is significantly more diffuse.

To evaluate the suitability of the weighted Gaussian likelihood used by CLOSR as an OOD score, an ablation study compared this score against an energy-based score proportional to the negative joint log-likelihood of a test sample under all known class distributions [50] and a score proportional to the

likelihood of the sample under a Gaussian orthogonal to the centroid of the sample’s predicted closed-set class label in its corresponding subspace. The resulting open-set AUC values are reported in Table VI. The energy-based score outperformed the unweighted Gaussian score, suggesting that aggregating information across all class-specific subspaces provides more reliable zero-day detection. Finally, the soft-weighted Gaussian likelihood achieved the highest open-set AUC, indicating that a Gaussian-based formulation is a more appropriate OOD scoring mechanism than those derived from the vMF distributions used for modelling known classes.

G. Computational Cost and Scalability

The computational requirements of CLAD, CLOSR, and baseline models during inference are summarised in Table VII, reporting parameter count, VRAM usage, computational complexity (MFLOPs), latency, and throughput. To ensure a fair comparison, all models were evaluated using the architectures obtained through hyperparameter optimisation. Measurement conditions were standardised: a fixed batch size of 1 was used for assessing computational complexity and latency, a batch size of 1024 was used for measuring VRAM usage, and throughput was determined using optimal batch sizes selected from $\{2^i \mid i \in \mathbb{Z}, 1 \leq i \leq 20\}$. Latency and throughput were measured on an NVIDIA RTX 3090 GPU and Intel Xeon W-2255 CPU under Ubuntu 22.04, using PyTorch 2.0.1 (FP32). Each latency value is the mean over 20 timed runs after 5 warmups. FLOPs were estimated with the PyTorch profiler under identical model configurations.

In binary classification, the optimised CLAD architecture exhibited a substantially higher parameter count than the baseline models, reflecting its scalability to larger model sizes without overfitting. Despite this increased complexity, CLAD achieved lower latency and higher throughput than RENOIR, demonstrating clear efficiency advantages. Although it was less efficient than other baselines, its balance of low latency and comparatively high throughput highlights its suitability as a practical real-time NIDS solution.

In OSR, CLOSR was found to be less computationally efficient than non-contrastive baselines, which can partly be attributed to the hyperparameter search selecting a relatively large architecture. This reflects CLOSR’s robustness against overfitting. When compared to the Siamese network, CLOSR was markedly more efficient, achieving superior results across all computational metrics. The inefficiency of the Siamese network stems from its reliance on the neighbours-distance scoring function, which requires an expensive search over the training set. CLOSR’s combination of latency and throughput makes it practical for NIDS.

VI. DISCUSSION AND CONCLUSIONS

In this work the contrastive loss for anomaly detection (CLAD) was introduced to bridge the gap between supervised classifiers, which perform well on known traffic but struggle to detect zero-day attacks, and anomaly detectors, which do not degrade in performance when detecting zero-day attacks but suffer from high false positive rates. The CLAD

TABLE VII

Computational requirements of CLAD, CLOSR, and baseline models during inference for binary classification (top) and open-set recognition (bottom).

| Model | Parameters (k) | VRAM (MiB) | Compute (MFLOPs) | Latency (ms) | Throughput (Ms^{-1}) |
|-----------------|----------------|------------|------------------|--------------|--------------------------|
| CLAD | 3318.80 | 33.06 | 6.61 | 0.39 | 2.31 |
| Autoencoder | 5.91 | 9.91 | 0.01 | 0.20 | 80.32 |
| DUAD | 536.28 | 18.70 | 1.06 | 0.53 | 7.78 |
| Deep SVDD | 87.10 | 12.36 | 0.17 | 0.24 | 40.08 |
| Renoir | 1181.28 | 31.01 | 7.03 | 2.10 | 2.25 |
| MLP | 667.16 | 19.08 | 1.33 | 0.20 | 10.06 |
| Siamese Network | 200.14 | 11.90 | 0.40 | 0.45 | 18.57 |
| CLOSR | 3048.20 | 33.18 | 6.07 | 0.40 | 2.68 |
| DOC | 277.13 | 22.22 | 0.55 | 0.31 | 18.72 |
| OPENMAX | 655.24 | 20.92 | 1.31 | 2.57 | 10.94 |
| CROSER | 81.76 | 12.38 | 0.16 | 2.80 | 22.32 |
| Siamese Network | 3374.15 | 3392.09 | 109.82 | 19.06 | 2.29 |

loss function exploits contrastive learning to exclusively model benign traffic as a von-Mises Fisher distribution in embedded space. As CLAD is only evaluated on benign anchor samples, the closed world assumption made by existing supervised classifiers is relaxed allowing for effective generalisation to zero-day attacks despite training on known malicious classes.

CLAD was experimentally evaluated and compared to leading anomaly detectors and supervised classifiers. It was found that by training on malicious samples, CLAD outperformed both existing supervised classifiers when detecting known attack classes. Furthermore, CLAD was able to effectively generalise to zero-day attacks, exploiting patterns learned from both known attacks and benign traffic to significantly outperform anomaly detectors when faced with previously unseen classes.

The extension of CLAD to open-set recognition gave rise to the Contrastive Loss for Open-Set Recognition (CLOSR) framework, which enabled multiclass classification over a set known malicious classes whilst simultaneously identifying zero-day attacks. This was achieved by learning the distribution of each known class within a distinct embedded subspace, with classification performed by evaluating the likelihood of a test sample belonging to each distribution. Zero-day attacks manifested as embedded vectors orthogonal to the distributions' centroids allowing easy identification. Experimental results demonstrated that CLOSR not only maintains strong closed-set classification performance but also generalises effectively to zero-day attacks, thereby outperforming existing OSR approaches for network intrusion detection systems.

While the ability of CLAD and CLOSR to generalise to zero-day attacks represents a significant step toward practical machine learning-based network intrusion detection systems, several limitations remain. The primary drawback of CLOSR is its reliance on a distinct linear projection head for each class in the training data, leading to a model whose size and computational cost scales with the number of classes. This design choice may hinder cross-domain applicability, particularly in fields such as computer vision where datasets can contain thousands of classes. Moreover, the proposed approaches have not yet been evaluated against adversarial attacks or under conditions involving noisy training data,

leaving their robustness in such scenarios uncertain. Addressing these gaps with future work could enhance both the efficiency and resilience of the methods. Finally, extending CLAD and CLOSR to support cross-network generalisation and few-shot learning on limited datasets would further strengthen their practicality and broaden their applicability.

REFERENCES

- [1] H. Hindy *et al.*, “A taxonomy of network threats and the effect of current datasets on intrusion detection systems,” *IEEE Access*, vol. 8, pp. 104650–104675, Jun. 2020. [Online]. Available: <http://dx.doi.org/10.1109/ACCESS.2020.3000179>
- [2] A. Rosay, F. Carlier, E. Cheval, and P. Leroux, “From CIC-IDS2017 to LYCOS-IDS2017: a corrected dataset for better performance,” in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, New York, NY, USA, 2022, pp. 570–575. [Online]. Available: <https://doi.org/10.1145/3486622.3493973>
- [3] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, p. 20, Jul. 2019. [Online]. Available: <https://doi.org/10.1186/s42400-019-0038-7>
- [4] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, and X. Bellekens, “Utilising deep learning techniques for effective zero-day attack detection,” *Electronics*, vol. 9, no. 10, Oct. 2020, Art. no. 1684. [Online]. Available: <https://www.mdpi.com/2079-9292/9/10/1684>
- [5] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines,” in *Proc. 2002 Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, May 2002, pp. 1702–1707. [Online]. Available: <https://doi.org/10.1109/IJCNN.2002.1007774>
- [6] L. Shu, H. Xu, and B. Liu, “DOC: deep open classification of text documents,” in *Proc. 2017 Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, M. Palmer, R. Hwa, and S. Riedel, Eds., Copenhagen, Denmark, Sep. 2017, pp. 2911–2916. [Online]. Available: <https://aclanthology.org/D17-1314>
- [7] A. Bendale and T. E. Boult, “Towards open set deep networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1563–1572. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.173>
- [8] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, and X. Bellekens, “Developing a siamese network for intrusion detection systems,” in *Proc. 1st Workshop Mach. Learn. Syst.*, Apr. 2021, pp. 120–126. [Online]. Available: <https://doi.org/10.1145/3437984.3458842>
- [9] H. Hindy *et al.*, “Leveraging siamese networks for one-shot intrusion detection model,” *J. Intell. Inf. Syst.*, vol. 60, pp. 407–436, Nov. 2022. [Online]. Available: <https://doi.org/10.1007/s10844-022-00747-z>
- [10] H. Hindy, “Intrusion detection systems using machine learning and deep learning techniques,” Ph.D. dissertation, Abertay Univ., Dundee, U.K., 2021.

[11] I. Sarker, A. Kayes, S. Badsha, A. Hamad, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J. Big Data*, vol. 7, Jul. 2020, Art. no. 41. [Online]. Available: <https://doi.org/10.1186/s40537-020-00318-5>

[12] H. Wang, J. Yang, and Y. Lu, "A logical combination based application layer intrusion detection model," in *Proc. 2020 Int. Conf. Cyberspace Innov. Adv. Technol. (CIAT)*, New York, NY, USA, Jan. 2021, pp. 310–316. [Online]. Available: <https://doi.org/10.1145/3444370.3444590>

[13] M. Hassen and P. K. Chan, "Learning a neural-network-based representation for open set recognition," in *Proc. 2020 SIAM Int. Conf. Data Min. (SDM)*, Cincinnati, OH, USA, Feb. 2020, pp. 154–162. [Online]. Available: <https://doi.org/10.1137/1.9781611976236.18>

[14] J. Rivero, B. Ribeiro, N. Chen, and F. S. Leite, "A grassmannian approach to zero-shot learning for network intrusion detection," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, Guangzhou, China, Oct. 2017, pp. 565–575. [Online]. Available: https://doi.org/10.1007/978-3-319-70087-8_59

[15] S. Kim, C. Hwang, and T. Lee, "Anomaly based unknown intrusion detection in endpoint environments," *Electronics*, vol. 9, no. 6, Jun. 2020, Art. no. 1022. [Online]. Available: <https://www.mdpi.com/2079-9292/9/6/1022>

[16] L. Ruff *et al.*, "Deep one-class classification," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, Jul. 2018, pp. 4393–4402. [Online]. Available: <https://proceedings.mlr.press/v80/ruff18a.html>

[17] Z. Zhang, Q. Liu, S. Qiu, S. Zhou, and C. Zhang, "Unknown attack detection based on zero-shot learning," *IEEE Access*, vol. 8, pp. 193981–193991, Oct. 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.3033494>

[18] M. Hassen and P. K. Chan, "Unsupervised open set recognition using adversarial autoencoders," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Miami, FL, USA, Dec. 2020, pp. 360–365. [Online]. Available: <https://doi.org/10.1109/ICMLA51294.2020.00064>

[19] S. Snehaa, G. Shwetha, and B. Priya, "Network intrusion detector based on isolation forest algorithm," in *Proc. 1st Int. Conf. Comput. Sci. Technol. (ICCST)*, Chennai, India, Nov. 2022, pp. 932–935. [Online]. Available: <https://doi.org/10.1109/ICCST55948.2022.10040395>

[20] A. Nagisetty and G. P. Gupta, "Framework for detection of malicious activities in iot networks using keras deep learning library," in *Proc. 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Erode, Tamil Nadu, India, Mar. 2019, pp. 633–637. [Online]. Available: <https://doi.org/10.1109/ICCMC.2019.8819688>

[21] Y. Lai, K.-Z. Zhou, S.-R. Lin, and N.-W. Lo, "Flow-based anomaly detection using multilayer perceptron in software defined networks," in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2019, pp. 1154–1158. [Online]. Available: <https://doi.org/10.23919/MIPRO.2019.8757199>

[22] H. A. Dau, V. Ciesielski, and A. Song, "Anomaly detection using replicator neural networks trained on examples of one class," in *Proc. Int. Conf. Simulated Evol. Learn. (SEAL)*, Dunedin, New Zealand, Dec. 2014, pp. 311–322. [Online]. Available: https://doi.org/10.1007/978-3-319-13563-2_27

[23] T. Li, Z. Wang, S. Liu, and W.-Y. Lin, "Deep unsupervised anomaly detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Online, Jan. 2021, pp. 3925–3934. [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/html/Li_Deep_Uncsupervised_Anomaly_Detection_WACV_2021_paper.html

[24] D. K. Nkashama *et al.*, "Deep learning for network anomaly detection under data contamination: evaluating robustness and mitigating performance degradation," 2024, arXiv:2407.08838. [Online]. Available: <https://arxiv.org/abs/2407.08838>

[25] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with svm for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, Sep. 2018. [Online]. Available: <https://doi.org/10.1109/ACCESS.2018.2869577>

[26] M. Verkerken *et al.*, "A novel multi-stage approach for hierarchical intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 3, pp. 3915–3929, Mar. 2023. [Online]. Available: <https://doi.org/10.1109/TNSM.2023.3259474>

[27] R. Yoshihashi *et al.*, "Classification-reconstruction learning for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4011–4020. [Online]. Available: <https://doi.org/10.1109/CVPR.2019.00414>

[28] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," 2017, arXiv:2407.08838. [Online]. Available: <https://arxiv.org/abs/1707.07418>

[29] G. Fei and B. Liu, "Breaking the closed world assumption in text classification," in *Proc. 2016 Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol. (NAACL-HLT)*, San Diego, California, Jun. 2016, pp. 506–514. [Online]. Available: <https://aclanthology.org/N16-1061>

[30] P. Júnior *et al.*, "Nearest neighbors distance ratio open-set classifier," *Mach. Learn.*, vol. 106, pp. 1–28, Mar. 2017. [Online]. Available: <https://doi.org/10.1007/s10994-016-5610-8>

[31] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014. [Online]. Available: <https://doi.org/10.1109/TPAMI.2014.2321392>

[32] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, "The extreme value machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 762–768, Mar. 2018. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2017.2707495>

[33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, San Diego, CA, USA, Jul. 2005, pp. 539–546. [Online]. Available: <https://doi.org/10.1109/CVPR.2005.202>

[34] L. Liu, P. Wang, J. Ruan, and J. Lin, "Conflow: Contrast network flow improving class-imbalanced learning in network intrusion detection," 2022. [Online]. Available: <https://doi.org/10.21203/rs.3.rs-1572776/v1>

[35] G. Andresini, A. Appice, and D. Malerba, "Autoencoder-based deep metric learning for network intrusion detection," *Inf. Sci.*, vol. 569, pp. 706–727, Aug. 2021. [Online]. Available: <https://doi.org/10.1016/j.ins.2021.05.016>

[36] N. Wang, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "FeCo: boosting intrusion detection capability in iot networks via contrastive learning," in *Proc. IEEE INFOCOM*, London, United Kingdom, May 2022, pp. 1409–1418. [Online]. Available: <https://doi.org/10.1109/INFOCOM48880.2022.9796926>

[37] X. Zhang *et al.*, "AOC-IDS: autonomous online framework with contrastive learning for intrusion detection," 2024, arXiv:2402.01807. [Online]. Available: <https://arxiv.org/abs/2402.01807>

[38] Z. M. Wang, J. Y. Tian, J. Qin, H. Fang, and L. M. Chen, "A few-shot learning-based siamese capsule network for intrusion detection with imbalanced training data," *Comput. Intell. Neurosci.*, Sep. 2021, Art. no. 7126913. [Online]. Available: <https://doi.org/10.1155/2021/7126913>

[39] P. Golchin *et al.*, "SSCL-IDS: enhancing generalization of intrusion detection with self-supervised contrastive learning," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, Thessaloniki, Greece, Jun. 2024, pp. 404–412. [Online]. Available: <https://doi.org/10.23919/IFIPNetworking62109.2024.10619725>

[40] Y. Yue, X. Chen, Z. Han, X. Zeng, and Y. Zhu, "Contrastive learning enhanced intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 4232–4247, Nov. 2022. [Online]. Available: <https://doi.org/10.1109/TNSM.2022.3218843>

[41] P. Khosla *et al.*, "Supervised contrastive learning," 2020, arXiv:2004.11362. [Online]. Available: <https://arxiv.org/abs/2004.11362>

[42] Y. Ming, Y. Sun, O. Dia, and Y. Li, "How to exploit hyperspherical embeddings for out-of-distribution detection?" 2023, arXiv:2203.04450. [Online]. Available: <https://arxiv.org/abs/2203.04450>

[43] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based out-of-distribution detection," 2020, arXiv:2010.03759. [Online]. Available: <https://arxiv.org/abs/2010.03759>

[44] T. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Mult. Classif. Syst. (MCS)*, Cagliari, Italy, Jun. 2000, pp. 1–15. [Online]. Available: https://doi.org/10.1007/3-540-45014-9_1

[45] R. M. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Dec. 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13391792>

[46] J. Puigcerver, C. Riquelme, B. Mustafa, and N. Houlsby, "From sparse to soft mixtures of experts," 2024, arXiv:2308.00951. [Online]. Available: <https://arxiv.org/abs/2308.00951>

[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019, arXiv:1711.05101. [Online]. Available: <https://arxiv.org/abs/1711.05101>

[48] ———, "Sgdr: Stochastic gradient descent with warm restarts," 2017, arXiv:1608.03983. [Online]. Available: <https://arxiv.org/abs/1608.03983>

[49] Z. Wang, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang, "Openauc: Towards auc-oriented open-set recognition," in *Adv. Neural Inf. Process. Syst.*, vol. 35, New Orleans, LA, USA, Nov. 2022, pp. 25 033–25 045. [Online]. Available: <https://proceedings.neurips.cc/paper/2022/file/9f73d65a4186198152357be871345771-Paper-Conference.pdf>

[50] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based out-of-distribution detection," 2021. [Online]. Available: <https://arxiv.org/abs/2010.03759>