Contrastive Self-Supervised Network Intrusion Detection using Augmented Negative Pairs

Jack Wilkie*, Hanan Hindy†, Christos Tachtatzis*, Robert Atkinson*
*University of Strathclyde, Glasgow, UK
†Ain Shams University, Cairo, Egypt

***Abstract*—Network intrusion detection remains a critical challenge in cybersecurity. While supervised machine learning models achieve state-of-the-art performance, their reliance on large labelled datasets makes them impractical for many real-world applications. Anomaly detection methods, which train exclusively on benign traffic to identify malicious activity, suffer from high false positive rates, limiting their usability. Recently, self-supervised learning techniques have demonstrated improved performance with lower false positive rates by learning discriminative latent representations of benign traffic. In particular, contrastive self-supervised models achieve this by minimising the distance between similar (positive) views of benign traffic while maximising it between dissimilar (negative) views. Existing approaches generate positive views through data augmentation and treat other samples as negative. In contrast, this work introduces Contrastive Learning using Augmented Negative pairs (CLAN), a novel paradigm for network intrusion detection where augmented samples are treated as negative views—representing potentially malicious distributions—while other benign samples serve as positive views. This approach enhances both classification accuracy and inference efficiency after pretraining on benign traffic. Experimental evaluation on the Lycos2017 dataset demonstrates that the proposed method surpasses existing self-supervised and anomaly detection techniques in a binary classification task. Furthermore, when fine-tuned on a limited labelled dataset, the proposed approach achieves superior multi-class classification performance compared to existing self-supervised models.**

***Index Terms*—Network Intrusion Detection Systems, Anomaly Detection, Self-Supervised Learning, Contrastive Learning, Machine Learning**

I. Introduction

MACHINE Learning (ML) models have become the leading approach in Network Intrusion Detection Systems (NIDS). These models have achieved State-of-the-art (SOTA) performance [1] and, unlike traditional methods, do not require experts to painstakingly identify patterns, known as signatures, or develop classification rules for known intrusions. However, ML models traditionally require large labelled datasets, containing many examples of each class the system aims to classify, to be effective. Unfortunately, in network intrusion detection acquiring such datasets is often non-trivial, necessitating that experts identify and label numerous instances of malicious network flows. To exacerbate this issue, traffic taken from other networks, either real or simulated, does not generalise well to new networks [2]. This leaves ML-based NIDS non-implementable in newly established networks where there are no labelled instances of malicious traffic. Furthermore, fully trained ML classifiers struggle to detect novel "zero-day" intrusions for which there are no labelled instances.

One solution to this problem is to exploit the abundance of benign network flows through the application of anomaly detection algorithms, such as Support Vector Machines (SVMs) [3] and autoencoders [4]. These algorithms learn the distribution of benign network traffic during training and flag non-conforming traffic as malicious during inference. While widely researched, the lack of malicious samples during training results in traditional anomaly detectors exhibiting false positive rates that are too high for practical deployment.

Recently, Self-Supervised Learning (SSL) has emerged as a promising approach for NIDS, enabling models to learn meaningful latent representations from unlabelled data [5]–[9]. This label independence allows it to be applied to NIDS to learn meaningful representations of network traffic from only benign flows [10]. Contrastive learning is one such method, where models are trained by minimising a distance metric between similar (positive) pairs of samples while simultaneously maximising it between dissimilar (negative) pairs. This is typically achieved by generating augmented views of the same sample, which are treated as positive pairs, resulting in a latent representation which is robust to the chosen perturbations. Several works have successfully leveraged contrastive SSL to improve intrusion detection performance [6]–[8], however, existing SSL approaches learn a distinct distribution for each sample and its augmented views in latent space. This limits their ability to model benign traffic wholistically and introduces challenges in distinguishing between benign and malicious traffic effectively.

This work proposes Contrastive Learning using Augmented Negative Pairs (CLAN), which presents a change in paradigm: instead of treating augmented samples as positive pairs, they are treated as negative. It is shown that this change results in the model learning a fundamentally different latent representation of the data: while existing approaches learn a distinct latent distribution for each sample; CLAN instead learns a single distribution of benign traffic. Not only does this allow for more efficient inference, but by learning the distribution of benign traffic wholistically CLAN achieves improved performance when both deployed as an anomaly detector or fine-tuned on a limited dataset to perform multi-class classification. The core contributions of this work can be summarised as follows:

DOI: https://doi.org/10.1109/CSR64739.2025.11129979 Link to IEEE Xplore: https://ieeexplore.ieee.org/document/11129979

1) A contrastive SSL framework is proposed for NIDS. In contrast to traditional approaches, where samples and their augmented versions are treated as positive pairs, this work instead aims to model the class level distribution of benign traffic by viewing augmented samples as belonging to another, potentially malicious, distribution. Thus, they are treated as negative pairs.
2) The framework is extended to perform binary classification without fine-tuning allowing it to function as an anomaly detector. Experimental results show the proposed approach outperforms existing anomaly detection and SSL algorithms.
3) It is shown that the priors learned by pretraining can be exploited for supervised classification, allowing the proposed model to be performative when fine-tuned on a limited quantity of labelled samples. It is experimentally shown to outperform existing SSL approaches in this setting.

This work is arranged as follows: Section II begins by describing work related to the proposed approach, including NIDS, anomaly detection, and SSL. Section III introduces the proposed framework and extends it to anomaly detection, with Section IV providing an in-depth comparison to existing SSL approaches. Experimental evaluation is preformed in Section V where the model is compared to existing models in anomaly detection and fine-tuned performance. Finally, Section VI concludes this work with relevant discussion and conclusions based on the experimental findings in its preceding section.

II. Related Works

This section details work the background literature related to the CLAN. Section II-A introduces NIDS and the challenges in building an adequate training dataset, with Section II-B summarising attempts at anomaly detection based solutions. Finally, Section II-C details self-supervised learning and its application to NIDS.

A. Network Intrusion Detection Systems

Network Intrusion Detection Systems are used to monitor network traffic to prevent unauthorised access or attacks. Traditional NIDS relied on signature-based detection, where experts manually crafted features and classification rules to uniquely identify each attack class. While these systems achieved high precision, they faced significant scalability challenges, as extending them to accommodate the exponentially growing number of attacks required substantial human effort. Additionally, signature-based systems are ineffective against novel intrusions, leaving networks vulnerable to zero-day attacks.

To address these limitations, ML has emerged as the dominant approach for reducing the manual effort required to develop effective NIDS. These models operate by learning statistical patterns from historical network traffic data to classify and detect attacks. Gradient-free approaches such as decision trees and Support Vector Machines (SVMs) [3] were initially employed to partition input features into regions of benign and malicious traffic. Since then, deep learning architectures such as multi-layer perceptrons (MLPs), Convolutional Neural Networks (CNNs) [11] and Long Short-Term Memory (LSTM) networks [12] have been achieved SOTA performance by training parameterised models to learn non-linear decision boundaries.

Despite their success, ML-based classifiers require large amounts of labelled training data, which can be difficult and expensive to obtain. Furthermore, their performance deteriorates significantly when confronted with a zero-day attack.

B. Anomaly Detection

Anomaly detection-based approaches have been employed to mitigate the challenges associated with acquiring labelled datasets for training ML-based NIDS. These methods learn the distribution of exclusively benign network traffic during training and flag non-conforming traffic as malicious. Since they do not rely on labelled malicious samples, anomaly detection techniques are able to detect all attacks, both known and zero-day, equally well.

Statistical anomaly detection methods rely on distance metrics or probabilistic models to distinguish between normal and anomalous traffic. Distance-based approaches compute the centroid of benign traffic in the training dataset and classify test samples based on their Minkowski distance from this centroid [13]. This approach has been extended to alternative distance metrics, such as the Frobenius and Grassmannian distance measures [14]. Other statistical methods, including local outlier factor and nearest-neighbour distance-based techniques, detect anomalies by examining the density of local neighbourhoods rather than relying on global statistics [15]. Another category of statistical methods extends traditional discriminative models to anomaly detection, such as one-class SVMs [3] and isolation forests (IF) [16], which adapt SVMs and tree-based models, respectively.

More recently, deep learning-based anomaly detection has gained popularity due to its ability to learn more complex, non-linear decision boundaries than statistical methods. Autoencoders (AE) [4] are a widely used approach where the model is trained to reconstruct input samples from a compressed latent representation. The reconstruction error is then used as an anomaly score, with higher errors indicating deviations from normal traffic. Variants such as sparse autoencoders [17], deep unsupervised anomaly detection (DUAD) [18], and DAE-LR [19] build on this principle with various modifications. Hybrid approaches, such as AutoSVM [20] and Deep Support Vector Data Descriptor (Deep SVDD) [21], and Deep Gaussian Mixture Models (DAGMM) [22] integrate deep learning methods with statistical techniques to enhance anomaly detection performance. Finally, generative approaches such as GANs [23] and variational autoencoders [24] have been used to train a classifier to identify artificially generated samples as malicious distributions.

While anomaly detectors enable classifiers trained on only benign traffic, the lack of malicious examples at train time results in them having false positive rate too high to be used in practice [4].

C. Self-Supervised Learning

Self-supervised learning has emerged as a promising approach for future NIDS. SSL enables models to learn semantic representations from unlabelled data by leveraging pretext tasks such as masked autoencoding [25] and restoration [26]. Exploiting this could allow NIDS to be trained on only benign data whilst maintaining an acceptable false positive rate.

One branch of SSL learns semantic representations by minimising a distance metric between positive views of a sample. These views are often generated through a series of augmentations, with the resultant model learning a latent representation of the data which is invariant to the chosen augmentations. However, this can often lead to the degenerative solution where all inputs are mapped to a point, known as dimensional collapse. Contrastive learning methods such as SimCLR [27] and infoNCE [28] attempt to prevent dimensional collapse by simultaneously maximising the distance between an input sample and its negative views, most often other samples. Similarly, knowledge distillation methods such as BYOL [29] and SimSiam [30] prevent dimensional collapse using architectural tricks such as momentum encoders. Finally, the canonical correlation family of techniques, including models such as VICReg [31] and Barlow Twins [32] prevent dimensional collapse by maximising a lower bound on the information in the output matrices.

Several existing works have applied SSL to NIDS, with contrastive learning being the most common approach. For instance, SSCL-IDS [8] minimises the distance between positive pairs generated using cutmix while maximising the distance between other samples. Similarly, Conflow [6] and CLDNN [7] generate positive pairs using different dropout masks and feature masking, respectively. Research has also explored the use of knowledge distillation methods like BYOL and SimSiam, as well as canonical correlation-based models such as VICReg and Barlow Twins, as these approaches have been shown in other domains to train effectively with lower batch sizes [10]. These methods have demonstrated promising results by outperforming traditional anomaly detection techniques.

While SSL-based NIDS has shown promise, existing methods use augmentations to generate positive pairs. This work instead treats them as negative pairs, whilst using other samples as positive pairs. It is shown that this allows the model to explicitly model the distribution of benign traffic resulting in improved performance and more efficient inference.

III. Proposed Approach

This section introduces the proposed Contrastive Learning using Augmented Negatives (CLAN) framework. A wholistic overview of this approach is illustrated in Figure 1. The CLAN loss function, described in Section III-A, provides the optimisation objective for training a neural network using a combination of benign traffic samples and their augmented variations. By minimising this loss, the model learns a latent representation where benign traffic forms a single latent distribution, from which other traffic types are distinctly separated. As detailed in Section III-B, the centroid of this latent distribution can be computed and cached after training. During inference, a binary class label of an unknown test sample can then be inferred by evaluating the probability that it was sampled from the latent distribution based on the distance between the test sample's latent representation and this centroid.

A. Contrastive Learning using Augmented Negatives

The objective of self-supervised learning for network intrusion detection systems is to leverage a dataset of benign network traffic in order to learn a robust representation of the data, such that binary classification can be performed during inference. Concretely, given a set of possible class labels $\mathcal{Y} := \{0, 1, \ldots, N_C - 1\}, \quad N_C \in \mathbb{Z}^+$ and a training dataset containing only benign traffic, defined as $\mathcal{D}_{\text{train}} = \{(x_i, y_i) \mid x_i \in R^f, y_i = 0, i = 1, \ldots, N_{\text{train}}\}, N_{\text{train}} \in \mathbb{Z}^+$ where $y = 0$ corresponds to the benign class label and $f \in \mathbb{Z}^+$ represents the number of tabular features extracted for each sample, the conditional distribution of benign traffic must be learned such that a binary class label $\hat{y} \in \{0, 1\}$, corresponding to whether a test sample is predicted to be benign ($y = 0$) or malicious ($y = 1$), can be determined for samples of a test dataset defined as $\mathcal{D}_{\text{test}} = \{(x_i, y_i) \mid x_i \in R^f, y_i \in \mathcal{Y}, i = 1, \ldots, N_{\text{test}}\}, N_{\text{test}} \in \mathbb{Z}^+$.

To achieve this the model must use the benign training examples to learn the conditional label distribution $P(y = 0|x)$. This can be expressed in terms of a joint probability $P(y = 0, x)$, and a partition function $Z(x) \in \mathbb{R}^+$ as shown in Equation 1.

$$P(y = 0|x) = \frac{P(x, y = 0)}{Z(x)} = \frac{P(x, y = 0)}{\sum_{c=0}^{N_C - 1} P(x, y = c)} \qquad (1)$$

In this work a parameterised neural network ϕ_θ is used to map the input data to a latent representation $z_i := \phi_\theta(x_i) \in \mathbb{R}^q$. It is assumed that each the class in the latent space follows a homoskedastic Gaussian distribution—i.e., each class i has a distinct mean μ_i but shares the same isotropic covariance $\sigma^2 I$. Formally, the latent class distributions are modelled as $P(z \mid y = i) = \mathcal{N}\big(z;\, \mu_i,\, \sigma^2 I\big)$. Under these assumptions the conditional probability $P(y = 0|x)$ can be expressed as Equation 2, where $d(\cdot, \cdot) \in \mathbb{R}^+$ computes the squared Euclidean distance between two latent representations.

$$P(y = 0|x) = \frac{e^{\frac{-d(z, \mu_0)}{2\sigma^2 I}}}{e^{\frac{-d(z, \mu_\psi)}{2\sigma^2 I}}} \qquad (2)$$

To learn a mapping from feature space to latent space, the negative log-likelihood of this distribution is minimised. After simplification, the negative log-likelihood across the training dataset is given by Equation 3, where $z_a := \phi_\theta(x_a), \forall x_a \in \mathcal{D}_{\text{train}}$.

$$-\log L(\theta) \propto \sum_{a=0}^{N_{\text{train}-1}} \left[d(z_a, \mu_0) - \sum_{c=0}^{N_c - 1} d(z_a, \mu_n) \right] \qquad (3)$$

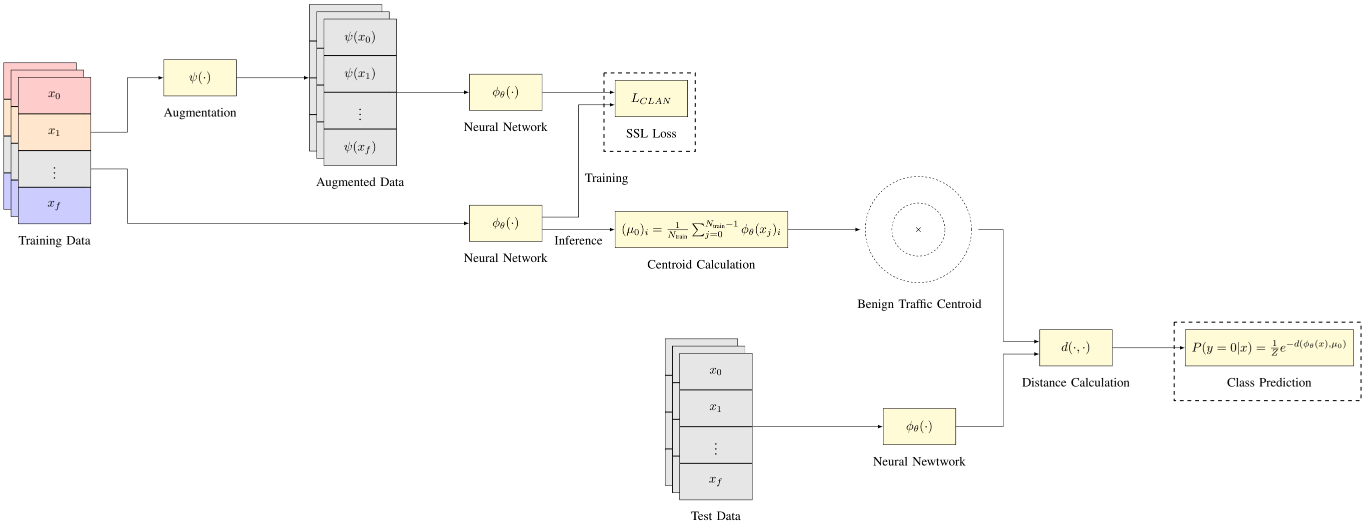

Fig. 1. Overview of the CLAN framework. A neural network is trained on both genuine benign and augmented network traffic to learn the distribution of benign traffic and map it to a single distribution in latent space. Evaluating the probability of a test sample belonging to this distribution can then be used to infer its label during inference.

However, at training time, the number of malicious classes—and consequently their latent mean vectors—is unknown. Additionally, computing the mean of the benign distribution dynamically during training is computationally expensive. To address this, the fact that the distance between a sample and the centroid of a Gaussian distribution is proportional to the expected distance between the sample and samples drawn from the Gaussian is exploited. This is stated formally in Equation 4 for a latent vector z and a second latent vector z' drawn from class *i*. This allows for the distance between a latent representation and a distribution centroid to be estimated via Monte Carlo sampling.

$$d(z, \mu_i) \propto \mathbb{E}_{z' \sim \mathcal{N}(\mu_i, \sigma^2 I)}\left[d(z, z')\right] \tag{4}$$

Since malicious samples are unavailable during training, a surrogate distribution $\psi(x) \in \mathbb{R}^f$ is used instead. Specifically, this surrogate is constructed by resampling features of x uniformly within the range $[-b, b]$ with probability $p_{\text{resample}} \in (0, 1]$, where $b \in \mathbb{R}^+$ and p_{resample} are hyperparameters. It assumed that an even number of samples of each class, both benign and malicious, are drawn from this distribution. Substituting Monte Carlo distance estimation into Equation 3 gives the negative log-likelihood expression shown in Equation 5. Here $\omega_a := \phi_\theta(x_a)$ and $\tilde{\omega}_a := \phi_\theta(\psi(x_a))$ are the latent representations of original and augmented samples, respectively, for a set $\mathcal{S} = \{x_i\}_{i=0}^k$ of k samples drawn from $\mathcal{D}_{\text{train}}$.

$$-\log L(\theta) \propto \sum_{a=0}^{N_{\text{train}-1}} [\sum_{p=0}^{k-1} d(z_a, \omega_p) - \sum_{n=0}^{k-1} d(z_a, \tilde{\omega}_n)] \tag{5}$$

Finally, calculating this over a batch, $x \in \mathbb{R}^{B \times f}$, and introducing a hinge regularisation term with margin hyperparameter $m \in (0, 1]$ results in the proposed CLAN loss function given in Equation 6.

$$\begin{aligned} L_{CLAN}(x) = \frac{1}{B}[\sum_{a=0}^{B-1}[\sum_{\substack{p=0 \\ p \neq a}}^{B-1} d(\phi_\theta(x_a), \phi_\theta(x_p)) \\ + \sum_{n=0}^{B-1} \max(0, m - d(\phi_\theta(x_a), \phi_\theta(\psi(x_n))))]] \end{aligned} \tag{6}$$

The CLAN loss function jointly optimises the latent space to model benign traffic as a Gaussian distribution while simultaneously performing maximum likelihood estimation to learn the mapping from feature space to latent space. This results in a representation where benign traffic is clustered around a centroid, while malicious traffic is pushed away from this cluster. It can be shown that using the cosine distance metric optimises a similar objective while replacing the Gaussian assumption with a von Mises-Fisher distribution assumption. This was found to improve performance and is thus used in the experiments in Section V.

B. Probabilistic Inference

CLAN models benign traffic as a distribution in latent space. Analysing the loss function reveals that the unnormalised joint probability of sampling both the input sample and the benign distribution exhibits an exponential decay with respect to the distance metric optimised and the distribution's centroid, as expressed in Equation 7. Here $d(\cdot, \cdot) \in \mathbb{R}^+$ represents the distance metric optimised by the loss function.

$$\tilde{P}(x, y = 0) = e^{-d(\phi_\theta(x), \mu_0)} \tag{7}$$

During inference, the model parameters remain fixed, allowing the centroid of the benign traffic distribution to be precomputed as the geometric mean of latent representations in the training dataset, as shown in Equation 8.

$$\mu_0 = \frac{1}{N_{\text{train}}} \sum_{x \in \mathcal{D}_{\text{train}}} \phi_\theta(x) \tag{8}$$

The conditional probability of a test sample being benign is then computed by normalising the distance between its latent representation and the benign centroid using a partition function, as defined in Equation 9.

$$P(y = 0|x) = \frac{1}{Z} e^{-d(\phi(x),\mu_0)} \tag{9}$$

Here, the partition function, $Z \in \mathbb{R}^+$, is treated as a constant hyperparameter that controls the trade-off between false positive rate and recall. The final classification decision assigns a predicted label $\hat{y} \in \{0, 1\}$ based on a probability threshold: a test sample is classified as benign ($\hat{y} = 0$) if $P(y = 0|x) > 0.5$, and as malicious ($\hat{y} = 1$) otherwise, as defined in Equation 10.

$$\hat{y} = \begin{cases} 0, & \text{if } P(y = 0|x) > 0.5; \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

IV. Comparison to Existing SSL Approaches

The proposed loss function follows the same general form as existing contrastive loss functions and can be described under the unified contrastive loss framework by setting specific parameters and incorporating the hinge regularisation term [33], which was included in the original contrastive loss function formulation [34], [35]. Furthermore, by removing the hinge function, the CLAN loss function becomes equivalent to the NTXent loss [36] with a temperature value of one. Existing SSL approaches for NIDS implement similar methodologies. For example, SSCL-IDS [8], CLDNN [7], and Conflow [6] all optimise a loss of the form shown in Equation 11, where $\psi(x)$ represents an arbitrary augmentation function, and $\tau \in \mathbb{R}^+$ is a temperature hyperparameter.

$$L(x) = -\frac{1}{B} \sum_{a=0}^{B-1} \log\left(\frac{e^{\frac{-d(\phi(x_a),\phi(\psi(x_a)))}{\tau}}}{\sum_{j=0}^{B-1} e^{\frac{-d(\phi(x_a),\phi(x_j))}{\tau}}}\right) \tag{11}$$

The key distinction between CLAN and existing approaches lies in how augmented samples are treated. While CLAN considers augmented samples as negative pairs, existing approaches treat them as positive pairs. This difference arises from the underlying assumptions in the derivation of their respective loss functions. CLAN performs maximum likelihood estimation under the assumption that benign traffic forms a single distribution in the latent space. In contrast, existing approaches can be derived following CLAN's derivation, except under the assumption that each sample and its augmented versions each form a distinct distribution in latent space. This distinction is illustrated in Figure 2, assuming that the loss functions optimise the squared Euclidean distance under the Gaussian assumption.

CLAN learning a single latent distribution of benign traffic provides several advantages over other SSL methods, which model the latent space as a mixture of distributions. Firstly, CLAN directly learns the overall distribution of benign network traffic, whereas existing SSL methods attempt to improve robustness by mapping noise-induced variations from augmentation to a structured representation. This distinction results in significant performance improvements in both binary classification and fine-tuned performance, as demonstrated in Section V. Notably, this approach is only viable under the assumption that the majority of samples belong to a single class distribution (benign traffic), which has not been exploited by prior SSL methods primarily developed for image-based tasks.

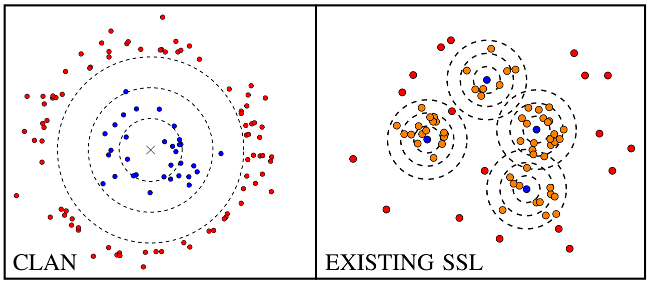

Fig. 2. Comparison of the latent representations learned by the CLAN loss function to those learned by existing self-supervised loss functions. **Left:** CLAN learns a single distribution corresponding to benign traffic (blue) whilst malicious traffic (red) appears outwith this distribution. **Right:** Existing self-supervised loss functions learn a distribution for each benign sample and its augmented views (orange).

CLAN also offers advantages in computational efficiency during inference. Since classification is performed by evaluating the probability of a test sample belonging to the latent benign distribution, only a single distance measurement between the test representation and the centroid is required. Assuming the centroid of the training dataset representations is computed and cached post-training, inference incurs a fixed computational cost, resulting in a complexity of $\mathcal{O}(1)$. In contrast, existing SSL approaches learn a separate distribution for each training sample. During inference, these methods evaluate the probability of a test sample belonging to the nearest learned distribution by thresholding the nearest-neighbour distance. However, identifying the closest distribution requires a nearest-neighbour search, leading to a computational complexity of $\mathcal{O}(N_{\text{train}})$. Consequently, the CLAN framework is significantly more scalable to the demands of modern network NIDS, which may need to monitor millions of flows per day.

V. Experimental Results

This section experimentally compares the CLAN framework to existing approaches in literature. Initially, the experimental procedure is described in Section V-A. The effectiveness of CLAN is then evaluated by comparing its performance in binary classification to existing SSL approaches in Section V-B, and to anomaly detectors in Section V-C. Finally, the performance of CLAN is evaluated when fine-tuned on a limited datset to perform multi-class classification in Section V-D.

A. Experimental Procedure

In this work, the CLAN loss function was used to train a modified MLP architecture. The architecture begins with a linear transformation that projects the input features from $\mathbb{R}^f$ to $\mathbb{R}^{d_{\text{model}}}$, where $d_{\text{model}} \in \mathbb{Z}^+$. This is followed by a sequence of fully connected layers, each followed by ReLU activation functions. A final linear transformation projects the data down to $\mathbb{R}^{d_{\text{head}}}$ where $d_{\text{head}} \in \mathbb{Z}^+$ such that $d_{\text{head}} < d_{\text{model}}$. The network's width and depth were treated as hyperparameters and optimised accordingly. Additionally, the uniform resampling rate and range used in CLAN were treated as hyperparameters. Baseline models were trained using the architectures specified in their respective original implementations.

To evaluate the effectiveness of CLAN compared to baseline models, models were trained and tested on the Lycos2017 dataset [37], an improved version of the CICIDS2017 dataset [38] that addresses various feature extraction and labelling errors. The dataset consists of 1,789,954 network flows across 14 classes, including benign traffic. It is highly imbalanced, with benign traffic accounting for over 1,000,000 samples, while certain malicious classes contain as few as 11 samples.

The dataset was partitioned train and test splits using a stratified sampling approach. Specifically, for each class, 50% of the available samples were randomly selected for the training set, and the remaining 50% were assigned to the test set: ensuring balanced representation across both splits. Exceptions were made for the Heartbleed and SQL Injection classes, which were included exclusively in the test set due to their limited sample size.

In the binary classification comparisons given in Section V-B and Section V-C models were optimised using 200 iterations of random search, with each iteration employing 5-fold stratified cross-validation, with malicious traffic being discarded from each training partition. The best performing configuration was subsequently retrained on benign traffic across the entire training dataset and evaluated on the test set. Models were trained for 200 epochs using the AdamW optimizer with a warm-up cosine learning rate schedule. The base learning rate, batch size, weight decay, and model-specific parameters were treated as hyperparameters.

In the fine-tuned performance comparisons given in Section V-D the weights learned during pretraining in Section V-B were fine-tuned on a limited subset of the training data which was generated through stratified sampling for 100 epochs using the AdamW optimiser with a learning rate of 10^{-6} and batch size of 64. Due to the limited size of the fine-tuning dataset, reported results were averaged over 10 runs, with a different seed being used to sample the subset of training data each time.

B. Comparison to SSL Approaches

In this section, CLAN is pretrained on benign network traffic and deployed as a binary classifier without fine-tuning. Several existing contrastive learning approaches were selected as baselines, including SSCL-IDS [8], CLDNN [7], and Conflow [6], each of which employs distinct augmentation strategies. Additionally, BYOL, SimSiam, Barlow Twins, and VICReg were chosen as non-contrastive SSL baselines and trained using the model architectures and training protocols outlined in previous works [10].

The class-wise AUROC scores, along with the mean AUROC for each model, are presented in Table I. The results demonstrate the effectiveness of CLAN, which achieves significant performance improvements over existing SSL models by learning a holistic representation of benign traffic. In contrast, conventional SSL approaches model a separate latent distribution for each individual sample, limiting their ability to generalise effectively in a network intrusion detection setting.

C. Comparison to Anomaly Detectors

Next, CLAN was compared against several baseline anomaly detection methods under the same evaluation setting as the SSL experiments. The baselines were selected to represent a range of approaches: gradient-free (Isolation Forrest [16] and SVM [3]), deep reconstruction (Autoencoder [4], DAE-LR [19], DAGMM [22] and DUAD [18]), and deep one-class learning (autoSVM [20] and Deep SVDD [21]). The AUROC scores for each baseline are reported in Table II. The results further demonstrate the effectiveness of CLAN, which significantly outperforms all anomaly detection baselines.

D. Fine-tuning Comparison

Finally, to assess the effectiveness of CLAN's learned representations for multi-class classification, a linear layer was appended to each SSL model, which was then fine-tuned using a limited number of samples per class. The mean macro-averaged F1 scores for each model, averaged over 10 runs for each training set size, are presented in Table III.

The results demonstrate that CLAN outperforms all baseline models across all training set sizes, except for the case where the training set consists of 256 samples per class, where it is marginally outperformed by BYOL. However, this appears to be an isolated occurrence, as CLAN consistently outperforms BYOL on all other training set sizes. These findings underscore the effectiveness of CLAN in learning robust priors that facilitate fine-tuning for downstream classification tasks.

VI. Discussion and Conclusions

This work introduced Contrastive Learning using Augmented Negative Pairs (CLAN), a novel self-supervised learning framework for network intrusion detection systems. Unlike conventional contrastive learning approaches that treat augmented views as positive pairs, CLAN treats augmented samples as negative pairs, belonging to a potentially malicious distribution. While existing SSL approaches learn a distinct latent distribution for each training sample, CLAN instead learns a single cohesive distribution of benign network traffic. This paradigm shift results in improved performance and computational efficiency when applied to both anomaly detection and supervised classification tasks.

TABLE I

AUROC comparison of CLAN and existing SSL baselines when performing binary classification without fine-tuning.

Class	CLAN	CLDNN [7]	SSCL-IDS [8]	ConFlow [6]	Barlow Twins [10]	SimSiam [10]	BYOL [9], [10]	VICReg [10]
Botnet	0.915536	0.951724	0.953530	0.927866	0.977046	0.996176	0.985893	0.946107
DDoS	0.996584	0.989344	0.972851	0.925385	0.999179	0.999191	0.999782	0.999400
DoS (Golden Eye)	0.931653	0.979910	0.841126	0.732904	0.896085	0.899778	0.905229	0.924664
DoS (Hulk)	0.977893	0.979486	0.982326	0.910926	0.994941	0.997693	0.990773	0.997170
DoS (Slow HTTP Test)	0.991794	0.613476	0.523339	0.522115	0.633869	0.542329	0.508903	0.522023
DoS (Slow Loris)	0.992513	0.873563	0.966009	0.953790	0.990351	0.982546	0.995689	0.984311
FTP Patator	0.944212	0.996560	0.996554	0.997778	0.993760	0.999653	0.998454	0.996132
Portscan	0.989271	0.849846	0.994321	0.993277	0.979313	0.990024	0.997439	0.985080
SSH Patator (Brute Force)	0.956259	0.958037	0.784271	0.903715	0.914556	0.864209	0.905103	0.840080
Web Attack (Brute Force)	0.907790	0.784891	0.721126	0.817545	0.678314	0.786032	0.755342	0.654605
Web Attack (XSS)	0.963409	0.849181	0.800105	0.880714	0.753717	0.765806	0.728249	0.690359
Heartbleed	0.997604	0.942537	0.993278	0.988757	0.999930	0.999278	0.999655	0.999871
Web Attack (SQL Injection)	0.897170	0.948995	0.966933	0.910350	0.916327	0.963398	0.954639	0.951767
Mean	**0.958591**	0.901350	0.884290	0.881933	0.902107	0.906624	0.901935	0.883967

TABLE II

AUROC comparison of CLAN and existing anomaly detectors when performing binary classification. Columns ordered by mean performance.

Class	CLAN	DUAD [18]	DAE-LR [19]	Deep SVDD [21]	AE [4]	IF [16]	AutoSVM [20]	SVM [3]	DAGMM [22]
Botnet	0.915536	0.819026	0.759901	0.751007	0.671062	0.629887	0.641687	0.637679	0.605228
DDoS	0.996584	0.979402	0.997996	0.996513	0.908072	0.945021	0.933722	0.889780	0.876049
DoS (Golden Eye)	0.931653	0.951141	0.952424	0.892923	0.855621	0.923275	0.887754	0.846840	0.736878
DoS (Hulk)	0.977893	0.967905	0.994860	0.993388	0.906273	0.955478	0.926223	0.894898	0.589771
DoS (Slow HTTP Test)	0.991794	0.954526	0.981080	0.975533	0.962552	0.966122	0.968576	0.963021	0.899879
DoS (Slow Loris)	0.992513	0.929777	0.986823	0.970622	0.897970	0.904747	0.880789	0.896824	0.955010
FTP Patator	0.944212	0.962427	0.959192	0.968324	0.765372	0.762553	0.753264	0.736828	0.748862
Portscan	0.989271	0.982759	0.983505	0.948716	0.720184	0.803987	0.668389	0.741269	0.517609
SSH Patator (Brute Force)	0.956259	0.889312	0.964803	0.962553	0.857118	0.845189	0.874968	0.799072	0.887558
Web Attack (Brute Force)	0.907790	0.864899	0.594903	0.673249	0.784379	0.713374	0.736061	0.767771	0.825337
Web Attack (XSS)	0.963409	0.942706	0.564499	0.658500	0.780935	0.702785	0.730008	0.761677	0.810560
Heartbleed	0.997604	0.979357	0.999801	0.998676	0.989079	0.999801	0.994511	0.993468	0.978630
Web Attack (SQL Injection)	0.897170	0.830627	0.848514	0.744588	0.731559	0.752289	0.745230	0.745087	0.653157
Mean	**0.958591**	0.927221	0.891408	0.887276	0.833090	0.838808	0.826245	0.821093	0.775733

TABLE III

Macro averaged F1 scores of CLAN and SSL baselines when fine-tuned on a limited dataset to perform multi-class classification.

Samples	CLAN	CLDNN [7]	SSCL-IDS [8]	Conflow [6]	Barlow Twins [10]	SimSiam [10]	BYOL [9], [10]	VICReg [10]
8	**0.496316**	0.379488	0.469348	0.480401	0.361244	0.482876	0.460014	0.389542
16	**0.538254**	0.438815	0.509693	0.501118	0.455801	0.470977	0.521448	0.440088
32	**0.544964**	0.483641	0.532057	0.532863	0.490422	0.512560	0.540308	0.513705
64	**0.589188**	0.535440	0.574097	0.571705	0.534474	0.555894	0.582854	0.528189
128	**0.628799**	0.589808	0.612588	0.617402	0.581268	0.603954	0.626497	0.593119
256	0.655416	0.644582	0.650703	0.651041	0.634673	0.645438	**0.657744**	0.631990
512	**0.710183**	0.683213	0.693791	0.705925	0.669987	0.684013	0.703609	0.665208
1024	**0.738838**	0.729141	0.731739	0.735170	0.719939	0.726593	0.735062	0.728573

Through experimental evaluation on the Lycos2017 dataset, CLAN was compared to existing approaches in anomaly detection and self-supervised learning in a binary classification task, where it was found to outperform the leading approaches by an AUROC improvement of 0.031370 and 0.056484 respectively. Additionally, when fine-tuned with a limited quantity of labelled samples, CLAN demonstrated improved multiclass classification performance over existing self-supervised learning models, highlighting its effectiveness in real-world scenarios where labelled data is scarce.

Beyond its performance benefits, CLAN also offers advantages in computational efficiency. By modelling benign traffic as a single distribution, inference requires only one distance measurement giving a resultant complexity of $\mathcal{O}(1)$, making it highly scalable for large-scale deployments. In contrast, existing SSL approaches require a nearest-neighbour search over training samples, leading to a complexity of $\mathcal{O}(N_{\text{train}})$, which becomes impractical in high-throughput network environments.

One limitation of the current approach is that it assumes that all training data is benign, which may not hold in real-world environments where the training data may be polluted with malicious samples. In future work, CLAN's robustness will evaluated under such conditions. Overall, CLAN represents a significant step forward in self-supervised learning for network intrusion detection systems, providing more efficient and effective classification. It is hoped that future works will build upon the CLAN framework by also treating augmented samples as negative pairs when training self-supervised models.

References

[1] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, p. 104650–104675, 2020. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2020.3000179

[2] S. Layeghy and M. Portmann, "Explainable cross-domain evaluation of ml-based network intrusion detection systems," *Computers and Electrical Engineering*, vol. 108, p. 108692, May 2023. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2023.108692

[3] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, and X. Bellekens, "Towards an effective zero-day attack detection using outlier-based deep learning techniques," 06 2020.

[4] H. A. Dau, V. Ciesielski, and A. Song, "Anomaly detection using replicator neural networks trained on examples of one class," in *Simulated Evolution and Learning*, G. Dick, W. N. Browne, P. Whigham, M. Zhang, L. T. Bui, H. Ishibuchi, Y. Jin, X. Li, Y. Shi, P. Singh, K. C. Tan, and K. Tang, Eds. Cham: Springer International Publishing, 2014, pp. 311–322.

[5] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," 2023. [Online]. Available: https://arxiv.org/abs/2304.12210

[6] L. Liu, P. Wang, J. Ruan, and J. Lin, "Conflow: Contrast network flow improving class-imbalanced learning in network intrusion detection," 04 2022.

[7] Y. Yue, X. Chen, Z. Han, X. Zeng, and Y. Zhu, "Contrastive learning enhanced intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4232–4247, 2022.

[8] P. Golchin, N. Rafiee, M. Hajizadeh, A. Khalil, R. Kundel, and R. Steinmetz, "Sscl-ids: Enhancing generalization of intrusion detection with self-supervised contrastive learning," in *2024 IFIP Networking Conference (IFIP Networking)*, 2024, pp. 404–412.

[9] Z. Wang, Z. Li, J. Wang, and D. Li, "Network intrusion detection model based on improved byol self-supervised learning," *Security and Communication Networks*, vol. 2021, no. 1, p. 9486949, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/9486949

[10] H. Fard, T. Schalau, and G. Wunder, "An investigation into the performance of non-contrastive self-supervised learning methods for network intrusion detection," EasyChair Preprint 14457, EasyChair, 2024.

[11] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42 210–42 219, 2019.

[12] P. Kottapalle, "A cnn-lstm model for intrusion detection system from high dimensional data," *Journal of Information and Computational Science*, vol. 10, pp. 1362–1370, 03 2020.

[13] M. Hassen and P. K. Chan, *Learning a Neural-network-based Representation for Open Set Recognition*, pp. 154–162. [Online]. Available: https://epubs.siam.org/doi/abs/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 *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds. Cham: Springer International Publishing, 2017, pp. 565–575.

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

[16] S. S, S. G, and B. Priya, "Network intrusion detector based on isolation … forest algorithm," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, 2022, pp. 932–935.

[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. 193 981–193 991, 2020.

[18] T. Li, Z. Wang, S. Liu, and W.-Y. Lin, "Deep unsupervised anomaly detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2021, pp. 3925–3934. [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/html/Li_Deep_Unsupervised_Anomaly_Detection_WACV_2021_paper.html

[19] D. K. Nkashama, J. M. Félicien, A. Soltani, J.-C. Verdier, P.-M. Tardif, M. Frappier, and F. Kabanza, "Deep learning for network anomaly detection under data contamination: Evaluating robustness and mitigating performance degradation," 2024. [Online]. Available: https://arxiv.org/abs/2407.08838

[20] 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. 52 843–52 856, 2018.

[21] 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*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4393–4402. [Online]. Available: https://proceedings.mlr.press/v80/ruff18a.html

[22] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi, "Deep autoencoding gmm-based unsupervised anomaly detection in acoustic signals and its hyper-parameter optimization," 2020. [Online]. Available: https://arxiv.org/abs/2009.12042

[23] Z. Liu, S. Li, Y. Zhang, X. Yun, and Z. Cheng, "Efficient malware originated traffic classification by using generative adversarial networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[24] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42 169–42 184, 2020.

[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[26] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *CoRR*, vol. abs/1603.08511, 2016. [Online]. Available: http://arxiv.org/abs/1603.08511

[27] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[28] E. Rusak, P. Reizinger, A. Juhos, O. Bringmann, R. S. Zimmermann, and W. Brendel, "Infonce: Identifying the gap between theory and practice," 2024. [Online]. Available: https://arxiv.org/abs/2407.00143

[29] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020. [Online]. Available: https://arxiv.org/abs/2006.07733

[30] X. Chen and K. He, "Exploring simple siamese representation learning," 2020. [Online]. Available: https://arxiv.org/abs/2011.10566

[31] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," 2022. [Online]. Available: https://arxiv.org/abs/2105.04906

[32] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021. [Online]. Available: https://arxiv.org/abs/2103.03230

[33] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," *CoRR*, vol. abs/2102.06810, 2021. [Online]. Available: https://arxiv.org/abs/2102.06810

[34] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR'06, vol. 2, 2006, pp. 1735–1742.

[35] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR'05, vol. 1, 2005, pp. 539–546 vol. 1.

[36] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[37] A. ROSAY, F. CARLIER, E. CHEVAL, and P. LEROUX, "From cic-ids2017 to lycos-ids2017: A corrected dataset for better performance," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, ser. WI-IAT '21. New York, NY, USA: Association for Computing Machinery, 2022, p. 570–575. [Online]. Available: https://doi.org/10.1145/3486622.3493973

[38] I. Sharafaldin., A. Habibi Lashkari., and A. A. Ghorbani., "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP,*, INSTICC. SciTePress, 2018, pp. 108–116.