Contrastive Autoencoding with Gaussian Confidence Regions for Concept Drift Detection in IDS

Lincoln Q. Vieira[1], **Ricardo Choren**[1], **Ricardo Sant'ana** [1]

[1]Instituto Militar de Engenharia (IME) – Rio de Janeiro – RJ – Brazil

{lincoln.queiroz,choren}@ime.eb.br, ricksant2003@gmail.com

Abstract. *Intrusion Detection Systems (IDS) are essential for network security; however, the growing complexity of cyberattacks challenges traditional signature-based and anomaly-based approaches, which struggle to detect novel threats while maintaining low false positive rates. In dynamic environments, evolving attack strategies cause concept drift that degrade the performance of static models. To address this, we propose a novel machine learning approach that integrates an autoencoder with contrastive learning and models known attack classes using Gaussian-based confidence regions. Experimental results show that the proposed classifier outperforms the baseline approach, achieving a higher average F1-score (0.39 vs. 0.25) due to the adaptability of hyperellipsoidal confidence regions.*

1. Introduction

Intrusion Detection Systems (IDS) are critical for network security, providing real-time monitoring of suspicious activities. They are typically classified as signature-based, which are accurate for known threats but ineffective against new ones, and anomaly-based, which detect unknown threats but often yield high false positive rates [Liu and Lang 2019]. With the increasing complexity of cyberattacks, IDS must quickly identify diverse threats, demanding more adaptive and advanced solutions [Abdulganiyu et al. 2023]. However, existing IDS methods often fail to keep up, reinforcing the need for improved and innovative approaches [Ozkan-Okay et al. 2021].

In the context of network attacks, where environments are constantly changing or evolving, attackers continuosly develop new methods to by-pass institutional security policies, a phenomenon known as concept drift [Elwell and Polikar 2011]. Concept drift is characterized by a shift in the relationship between input data and the target variable over time, resulting in performance degradation in non-adaptive models [Escovedo et al. 2018]. Machine learning (ML) techniques stand out for concept drifting attacks due to their robustness, resilience to data noise and adaptability [Kocher and Kumar 2021]. Some studies that employ ML techniques for concept drift detection in IDS context include [Yang et al. 2021] and [Kuppa and Le-Khac 2022].

This paper presents a novel machine learning approach for concept drift detection by integrating an autoencoder neural network with contrastive learning. The autoencoder compresses network traffic into a lower-dimensional latent space, while contrastive learning improves representation by separating similar and dissimilar instances. Known attack classes are modeled as Gaussian distributions and samples outside all confidence regions are flagged as concept drift. This method provides a more flexible and accurate classification framework, enhancing the detection and distinction of drifted samples.

This paper is structured as follows: Section 2 reviews contrastive learning concepts relevant to this study. Section 3 introduces the proposed approach. Section 4 presents experimental results and analysis, highlighting the method's effectiveness in detecting concept drift. Section 5 discusses related work on concept drift detection in IDS. Finally, Section 6 outlines conclusions and future research directions.

2. Contrastive Learning

This section presents key concepts about contrastive learning essential for understanding the study.

The main objective of contrastive learning is to project samples into a lower-dimensional space while preserving their semantic relationships. This is achieved using a contrastive loss function that minimizes the distance between samples of the same class and maximizes it between different classes. By enhancing feature discrimination, contrastive learning proves effective in representation learning, facial recognition, recommendation systems and anomaly detection [Le-Khac et al. 2020].

To calculate the contrastive loss, the model receives a pair of samples (X_i, X_j), which are analyzed to determine their similarity relationship. This relationship is represented by a binary value Y, where $Y = 0$ indicates that both samples belong to the same class, while $Y = 1$ indicates that the samples belong to different classes. Each network in the model encodes its respective input sample separately, generating corresponding latent representations or embeddings (z_i, z_j) [Chopra et al. 2005].

The contrastive loss is defined by [Chopra et al. 2005] as:

$$\mathcal{L} = (1 - Y)D^2 + (Y)\{max(0, m - D)\}^2 \tag{1}$$

where D is the Euclidean distance between z_i and z_j, and m is the margin used to bring similar samples closer or push dissimilar samples farther apart.

3. The Proposed Aproach

This section presents an approach combining autoencoders and contrastive learning to cluster same-class samples and separate different classes while preserving class-specific features. Encoded samples are classified by modeling known classes as Gaussian distributions and using confidence regions to define class membership.

The proposed approach integrates an encoder trained using a combination of an autoencoder neural network and contrastive learning. To apply contrastive learning, the training set is split into pairs of samples, each labeled with a similarity value Y, where $Y = 0$ indicates both samples belong to the same class, and $Y = 1$ indicates they belong to different classes. These samples are then encoded into a lower-dimensional latent space using the encoder, and the contrastive loss is computed as defined in Equation 1. Simultaneously, the network reconstructs the input data, as in a standard autoencoder, and calculates the reconstruction loss $\mathcal{L}_a$ using Mean Squared Error (MSE).

Finally, a total loss $\mathcal{L}_t$ is computed in Equation 2 where $\mathcal{L}_c$ represents the contrastive loss and λ is its weighting factor, it was empirically tuned on the validation set

to balance the contribution of the contrastive loss and the reconstruction loss. Its significance lies in controlling the trade-off between clustering quality in the latent space and reconstruction accuracy, both of which affect drift detection performance. The contrastive autoencoder is trained to minimize this total loss, integrating both reconstruction and contrastive objectives.

$$\mathcal{L}_t = \mathcal{L}_a + \lambda \mathcal{L}_c \tag{2}$$

Using the trained encoder, the training data is projected into the latent space, and each class is modeled as a Gaussian distribution. A confidence hyperellipsoid is computed for each class at a confidence level γ, as defined in Equation 3, where z is a k-dimensional vector, μ_i is the mean of the encoded training sample from class i, Σ_i is a $k \times k$ covariance matrix of class i and χ^2 is the chi-squared distribution for k degrees of freedom [Chew 1966].

$$(z - \mu_i)^T \Sigma_i^{-1} (z - \mu_i) \leq \chi^2(\gamma, k) \tag{3}$$

Test samples are classified based on whether their latent representations fall within a class's confidence region. Samples outside all regions are identified as concept drift. Gaussian Mixture Models (GMM) were used to model each attack class individually in the latent space. Experimental results showed that, for most classes, a single-component model outperformed multi-component models.

4. Experiments

This section provides an overview of the experiments conducted to evaluate the proposed approach.

4.1. Experiment Settings

This subsection provides an overview of the experiments configuration, including details of the hardware used, as well as the description of the real-world dataset used in the experiment. To evaluate the proposed approach, an experiment was conducted on a computer equipped with an AMD Ryzen 9 5900X processor, 32 GB RAM memory and NVIDIA GeForce RTX 3070 GPU.

To evaluate the performance of the proposed approach, Contrastive Autoencoder for Drifting detection and Explanation (CADE) [Yang et al. 2021] was used as a baseline model. CADE was chosen as a baseline due to its conceptual and architectural similarity to our method. Both approaches rely on autoencoders and contrastive learning for latent representation and address concept drift detection. CADE employs the same type of contrastive autoencoder network architecture. The primary difference lies in their strategies for concept drift classification. The proposed model classifies samples based on confidence regions derived from Gaussian distributions, whereas CADE utilizes the Mean Absolute Deviation (MAD) approach. In the CADE framework, for each class i, which contains n_i encoded samples z_j in the latent space from the training set, two metrics are computed: d_i and MAD_i, as defined in Equations 4 and 5. Here, c_i represents the centroid of class i in the latent space and b is a constant.

$$d_i = \text{median}(||z_j - c_i||), j = 1, ..., n_i \tag{4}$$

$$\text{MAD}_i = b * \text{median}(| \ ||z_j - c_i|| - d_i \ |), j = 1, ..., n_i \tag{5}$$

To classify a test sample z, which has been encoded in the latent space, the values of A_i are computed for each class i, as defined in Equation 6. If all computed A_i values exceed a predefined threshold, the sample is considered to represent a concept drift.

$$A_i = \frac{(| \ ||z - c_i|| - d_i \ |)}{\text{MAD}_i} \tag{6}$$

For the experiments, the baseline model parameters were kept the same as in the original work.

The proposed approach was evaluated using the CICIDS-2018 dataset, a widely adopted benchmark in cybersecurity research that also enables the simulation of drift scenarios, supporting a comprehensive assessment of model effectiveness. Developed by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick as an evolution of CICIDS-2017, the dataset includes 10 days of network traffic, featuring 80 attributes generated by CICFlowMeter, covering time, packet, byte, and flow statistics, and contains benign traffic alongside 14 attack types grouped into 7 categories: Brute Force, DoS, DDoS, Botnet, Infiltration, Web Attacks and SQL Injection [Sharafaldin et al. 2018]. CICIDS-2018 was selected because it presents a diverse range of contemporary attacks over multiple days, which naturally introduces temporal variation and realistic traffic dynamics. These characteristics make it suitable for simulating concept drift scenarios without artificial data manipulation.

In addition to network statistical attributes, the dataset includes timestamp information with date and time. However, this information should be excluded, as each attack class was collected on a distinct day, which could lead the model to rely on date cues rather than network features for classification.

4.2. Experiments

The dataset was initially split into training/validation and test sets with an 80/20 ratio, preserving class distributions. The training/validation set was then further divided into training (64% of the total data) and validation (16% of the total data) sets, maintaining the same 80/20 split and class balance. Additionally, the benign class underwent random undersampling to 6% of its total size in order to keep it within the same order of magnitude as the other classes, as can be observed in Table 1, which shows the total number of samples from the benign class and from the 14 attack classes, grouped into 7 attack categories. Undersampling the benign class was necessary to balance the training process and ensure the model does not become biased toward the dominant class [Chawla et al. 2002].

To simulate the effect of concept drift, the attack classes are hidden during the training phase and revealed during the testing phase.In this way, as interpreted by [Yang et al. 2021], unknown attacks are treated as manifestations of concept drift, where novel patterns deviate significantly from the latent distributions of known attack classes,

Class	Number of samples	Total
Benign	803,354	803,354
DDoS attack-HOIC	686,012	1,263,933
DDoS attacks-LOIC-HTTP	576,191	
DDoS attack-LOIC-UDP	1,730	
DoS attacks-Hulk	461,912	654,300
DoS attacks-SlowHTTPTest	139,890	
DoS attacks-GoldenEye	41,508	
DoS attacks-Slowloris	10,990	
Botnet	286,191	286,191
Brute Force FTP	193,360	380,949
Brute Force SSH	187,589	
Infiltration	160,639	160,639
Web Attack Brute Force Web	611	841
Web Attack Brute Force XSS	230	
SQL Injection	87	87

Table 1. Number of samples in each class.

so that in a future step the model can be adapted to the new concept. For concept drift detection, the samples from the class hidden during training are considered positive samples, while the other classes are considered negative samples. The proposed approach identifies concept drift based on the confidence regions of known classes, regardless of whether they are benign or malicious. Thus, previously unseen benign traffic (legitimate traffic exhibiting a new pattern) can also be flagged as drifted, just like unknown attacks. To evaluate the model's performance, the accuracy, precision, recall and F1-score metrics will be used.

For the contrastive autoencoder network, the architecture 82-64-32-16-7-16-32-64-82 was used, with 250 epochs and a learning rate of 0.0001. For the contrastive loss, a margin m of 10 and a weight λ of 0.1 were used. For the calculation of the confidence regions, γ values in the range of 0.95 to 0.50 were tested, with the value of 0.85 achieving the best result on the validation set, obtaining an average F1-score of 0.39.

4.3. Experimental Results and Analysis

Table 2 shows the performance in concept drift detection using the metrics mentioned for each attack class hidden during the training phase. Similarly, Table 3 presents the performance of CADE for each hidden attack class. Both classification models are capable of identifying the occurrence of concept drift for the DoS and Brute Force classes, with the proposed model achieving F1-scores of 0.83 and 0.68, respectively, while the baseline reaches F1-scores of 0.67 and 0.72. Comparing the results with the DDoS class hidden, the proposed model significantly outperforms the baseline, achieving an F1-score of 0.66 versus 0.08. For the Botnet class, the proposed model outperforms the baseline, achieving an F1-score of 0.42, compared to 0.26. In the Infiltration class, the proposed model also outperforms the baseline, though with a low F1-score of 0.13 versus 0.02, which is much lower than the detection performance for the previously mentioned hidden classes. This is due to the contrastive autoencoder network's difficulty in representing this class

distinctly from the benign class, as illustrated in Figure 1, where feature 4 of the latent space encoding is the one that most differentiates the two classes, yet both are still heavily overlapping. The Web Attacks and SQL Injection classes, however, were not adequately detected by either model, while the proposed model indeed fails to identify both classes, the baseline is able to detect 12% of the SQL Injection attacks. However, due to the small number of samples from this class in the dataset, this value was diluted among the false positives.

Hidden class	Accuracy	Precision	Recall	F1-score
DDoS	0.80	0.85	0.54	0.66
DoS	0.94	0.88	0.77	0.83
Brute Force	0.93	0.70	0.66	0.68
Botnet	0.92	0.52	0.35	0.42
Infiltration	0.92	0.13	0.13	0.13
Web Attacks	0.98	0.00	0.03	0.00
SQL Injection	0.96	0.00	0.00	0.00

Table 2. Proposed model performance for each hidden attack class.

Hidden class	Accuracy	Precision	Recall	F1-score
DDoS	0.61	0.24	0.05	0.08
DoS	0.86	0.59	0.78	0.67
Brute Force	0.92	0.56	1.00	0.72
Botnet	0.80	0.19	0.45	0.26
Infiltration	0.95	0.10	0.01	0.02
Web Attacks	0.97	0.00	0.06	0.00
SQL Injection	0.80	0.00	0.12	0.00

Table 3. CADE performance for each hidden attack class.

Table 4 presents a direct comparison of the average results obtained by the proposed model and the baseline. The proposed method shows better results in the accuracy, precision, and F1-score metrics, while it is outperformed by the baseline in the recall metric. The classification performance of the proposed model surpasses the baseline due to the flexibility of the confidence regions, which take the shape of a hyperellipsoid, whereas the baseline is restricted to a hypersphere format, in other words, CADE depends solely on the distance of the samples to the class centroid, forming a hypersphere that is less adaptable to the distribution of the samples in the latent space than the hyperellipsoid shape proposed in this work. This leads to situations where, depending on the class distribution, there may be many false positives along with true positives, or many true negatives along with false negatives.

5. Related works

In [Yang et al. 2021], the author proposes CADE, a contrastive autoencoder neural network designed to detect concept drift in IDS and malware classification. The model jointly minimizes contrastive loss and reconstruction error to cluster samples not only by class but also by shared characteristics. For classification, CADE employs a statistical

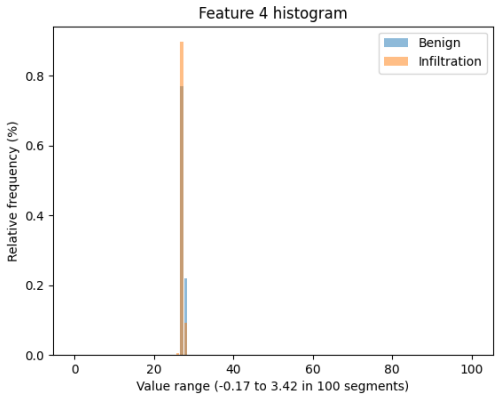

Figure 1. Histogram of the feature in the latent space that best distinguished the Benign and Infiltration classes.

Classification method	Accuracy	Precision	Recall	F1-score
Proposed model	**0.92**	**0.44**	0.35	**0.39**
CADE	0.84	0.24	0.35	0.25

Table 4. Comparison of the average metrics between the proposed model and the baseline.

distance-based method using Mean Absolute Deviation (MAD). However, this classification method is less flexible compared to the one presented in this work.

The study by [Kuppa and Le-Khac 2022] addresses concept drift detection, concept identification, and model adaptation using a contrastive autoencoder with cosine similarity, like [Yang et al. 2021], applied to IDS and URL categorization. It introduces the Nearest Class Mean (NCM) method, which classifies a sample as concept drift if its distance to all class centroids exceeds a threshold. However, this method, like that of [Yang et al. 2021], offers less flexibility compared to the approach proposed in this work.

6. Conclusion and Future Works

The ability to detect zero-day attacks is crucial for enhancing network security. Consequently, identifying changes and variations in known attack patterns is essential. Prior studies indicate that combining autoencoder networks with contrastive learning is a promising strategy for dimensionality reduction and clustering same class samples, enabling spatial classifiers to detect novel attacks by identifying samples that fall outside known classes. The confidence region-based classifier outperformed the MAD-based approach, achieving an average F1-score of 0.39 compared to 0.25, due to the greater flexibility of hyperellipsoidal regions over hyperspherical ones. However, both methods failed to detect concept drift in certain attack classes, suggesting limitations in either the extracted features or the dimensionality reduction technique.

As future work, we plan to compare the proposed model with the Nearest Class Mean (NCM) classifier, evaluate it on additional datasets like UNSW-NB15 [Moustafa and Slay 2015], and assess the impact of retraining with concept drift samples in an online learning setting to verify the model's ability to adapt and associate attack variations with their original classes.

References

Abdulganiyu, O., Ait Tchakoucht, T., and Saheed, Y. (2023). A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, 22:1125–1162.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chew, V. (1966). Confidence, prediction, and tolerance regions for the multivariate normal distribution. *Journal of the American Statistical Association*, 61(315):605–617.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *Neural Networks, IEEE Transactions on*, 22:1517 – 1531.

Escovedo, T., Koshiyama, A., da Cruz, A. A., and Vellasco, M. (2018). Detecta: abrupt concept drift detection in non-stationary environments. *Applied Soft Computing*, 62:119–133. 24 nov. 2024.

Kocher, G. and Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Comput.*, 25(15):9731–9763. 24 nov. 2024.

Kuppa, A. and Le-Khac, N.-A. (2022). Learn to adapt: Robust drift detection in security domain. *Computers and Electrical Engineering*, 102:108239. 24 nov. 2024.

Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934.

Liu, H. and Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20). 24 nov. 2024.

Moustafa, N. and Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6.

Ozkan-Okay, M., Samet, R., Aslan, , and Gupta, D. (2021). A comprehensive systematic literature review on intrusion detection systems. *IEEE Access*, 9:157727–157760.

Sharafaldin, I., Habibi Lashkari, A., and Ghorbani, A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pages 108–116.

Yang, L., Guo, W., Hao, Q., Ciptadi, A., Ahmadzadeh, A., Xing, X., and Wang, G. (2021). Cade: Detecting and explaining concept drift samples for security applications. In *Proc. of USENIX Security*, pages 2327–2344.