Neuro-symbolic Integration for Open Set Recognition in Network Intrusion Detection

Alice Bizzarri[1], Chung-En Yu[2], Brian Jalaian[2,3], Fabrizio Riguzzi[1], and Nathaniel D. Bastian[4]

[1] University of Ferrara, Ferrara, Italy
{alice.bizzarri, fabrizio.riguzzi}@unife.it
[2] University of West Florida, Pensacola, FL, USA
cy31@students.uwf.edu, bjalaian@uwf.edu.org
[3] Institute for Human & Machine Cognition, Pensacola, FL, USA
bjalaian@ihmc.org
[4] United States Military Academy, West Point, NY, USA
nathaniel.bastian@westpoint.edu

Abstract. Open Set Recognition (OSR) addresses the challenge of classifying inputs into known and unknown categories, a crucial task where labeling is often prohibitively expensive or incomplete. This is particularly vital in applications like Network Intrusion Detection Systems (NIDS), where OSR is used to identify novel, previously unknown attacks. We propose a neuro-symbolic integration approach that combines deep learning and symbolic methods, enhancing deep embedding for clustering with custom loss functions and leveraging XGBoost's decision tree algorithms. Our methodology not only robustly addresses the identification of previously unknown attacks in NIDS but also effectively manages scenarios involving covariance shift. We demonstrate the efficacy of our approach through extensive experimentation, achieving an AUROC of 0.99 in both contexts. This paper presents a significant step forward in OSR for network intrusion detection by integrating deep and symbolic learning to handle unforeseen challenges in dynamic environments.

Keywords: Neuro-symbolic Integration · Deep Embedding for Clustering · XGBoost · Open Set Recognition · Network Intrusion Detection

1 Introduction

Machine learning systems are commonly trained under the closed-world assumption, where it is presumed that every test class corresponds to a training class [16,13,23]. There has been a concerted effort to augment the ability of these systems to recognize and disregard unknown inputs. This effort has been particularly pronounced in the domains of anomaly detection, out-of-distribution (OOD) detection, and open set recognition (OSR). Traditionally, the focus was more on anomaly detection, but recent shifts have prioritized OOD detection and OSR. The fundamental differences between OOD detection and OSR are twofold: firstly, OOD detection involves a greater semantic gap between data

considered outside and within the distribution. Conversely, OSR deals with classifying subsets of data as either within or outside the distribution in the same dataset. Secondly, unlike OOD detection which primarily differentiates between external and internal samples, OSR also assesses classification efficacy on known classes within a closed-world setup [30]. As delineated in [28], a distinction is made between semantic shift and covariate shift. Semantic shift pertains to OOD samples emanating from different classes, whereas covariate shift relates to samples originating from varying domains.

A pertinent example of an OSR challenge is the detection of previously unknown attacks facing by Network Intrusion Detection Systems (NIDS). Our proposed solution, TEX-DEC, integrates Deep Embedding for Clustering (DEC) [27] with XGBoost [7] to address this. DEC extracts pertinent features and clusters them to form a condensed latent space, while XGBoost is utilized to identify novel samples within this space. Notably, our approach employs a neuro-symbolic methodology, merging neural network-based deep learning with symbolic techniques that process data representations symbolically. This fusion enhances the system's adaptability and robustness, enabling it to tackle the diverse and complex challenges presented by OSR effectively. We implemented TEX-DEC in identifying previously unknown attacks in NIDS and in recognizing handwritten images, achieving an impressive AUROC of 0.99 in both applications.

The remainder of this paper is structured as follows: Section 2 reviews relevant literature and prior work concerning OSR and NIDS. Section 3 elucidates key concepts crucial for a comprehensive understanding of our methodology. The details of our proposed approach are elaborated in Section 4. Section 5 describes the experimental setup, datasets used, and the results obtained. An ablation study examining our custom loss function is discussed in Section 6. Finally, Section 7 provides our concluding thoughts and findings.

2 Related Work

Recent advancements in machine learning have significantly enriched the research landscape of OSR. OSR methodologies, pivotal in scenarios with an open-ended or evolving set of possible classes, are generally bifurcated into discriminative and generative approaches [9]. Discriminative models, such as those discussed by Scheirer et al. [25], Hassen and Chan [11], and Bendale et al. [4], utilize probability- or learning-based techniques to distinguish known from unknown classes. Conversely, generative models, exemplified by the works of Neal et al. [22] and Ge et al. [8], deploy generative techniques to identify OSR samples.

Scheirer et al. [25] introduced the Compact Abating Probability (CAP) model, which reduces the probability of class membership as samples diverge from training data towards open space, demonstrating successful application in OSR scenarios. Bendale et al. [4] expanded upon this with OpenMax, incorporating Extreme Value Theory (EVT) to build a CAP model for each class, enhancing robustness by rejecting unknown inputs via thresholding. Although not primar-

ily focused on adversarial inputs, OpenMax exhibits superior resilience compared to traditional softmax models.

Hassen and Chan [11] explored intermediate representations to create a spatial distinction where samples from the same class are clustered together while distinctly separating different classes, enabling the identification of unknown examples through Euclidean distance and predefined thresholds. Neal et al. [22] employed generative adversarial networks to create examples mimicking the training set yet belonging to no known category, training OSR models with these synthetic samples. Ge et al. [8] introduced Generative OpenMax (G-OpenMax), extending OpenMax capabilities to better detect unknown samples.

In our work, we adopt a methodology resonating with the approach of Hassen and Chan [11], enhancing it with a clustered latent space and XGBoost to augment both performance and robustness. Notably, the use of XGBoost obviates the need for threshold-based classification of unknown examples.

The NIDS domain, predominantly using known datasets for attack classification [15,2,3], faces challenges in detecting previously unknown attacks. Traditional anomaly-based methods, which rely on deviations from normative behavior and typically require network flow data, necessitate additional information [24,14,1,29]. Addressing these challenges, we propose a novel DEC and XGBoost approach for packet-level detection of previously unknown attacks, comparing its efficacy against existing packet-based and flow-based systems [19,29].

3 Background

In this section, we provide an overview of key concepts that are needed to understanding our proposed approach. Specifically, we introduce DEC and XGBoost.

3.1 Deep Embedding for Clustering

Cluster analysis holds a crucial role in machine learning and data mining. Deep clustering refers to a set of techniques that combine deep learning with traditional clustering algorithms. Unlike conventional clustering methods, that rely on handcrafted features or distance metrics, deep clustering leverages the representation learning capabilities of DNNs to automatically learn feature representations directly from raw data. Deep clustering takes the feature extraction prowess of deep neural networks to autonomously acquire richer and more representative data representations. This methodology handles high-dimensional and intricate datasets, making it especially suited for scenarios where the inherent structure of the data is not known. By employing gradient-based optimization methods, DNNs can be trained to enhance cluster homogeneity while simultaneously maximizing inter-cluster heterogeneity. The outcome is a resilient and adaptable clustering approach effective on different datasets sourced from various origins and domains.

This study builds upon DEC [27], which employs DNNs to simultaneously learn feature representations and cluster assignments. This is achieved by mapping the data space to a low-dimensional feature space and iteratively optimizing

a clustering objective. DEC comprises encoders responsible for acquiring a latent representation, coupled with a cluster layer. This cluster layer generates a *soft assignment* q for each sample, reflecting the likelihood of its membership in each cluster. Loss is defined as the Kullback-Leibler (KL) divergence between the *soft assignment* q and a target distribution p. DEC initially pre-trains the encoder part of the network using the autoencoder (AE) framework. The purpose is to initialize the weights and significantly reduce the effort required to achieve the clustering objective. A *Cluster layer* is appended to the end of the encoder to generate the *soft assignments*. The *Cluster layer* incorporates centroids as a parameter and uses the output of encoder, z, as an input, subsequently calculating the *soft assignment* in the manner described in Equation 1.

As mentioned above, DEC training is done in several phases, first an AE is pretrained to initialize the parameters. Alternatively, DEC can be trained from scratch, but this requires more effort in terms of training epochs. The AE learns a latent representation that naturally facilitates identifying clustering representations with DEC. The feature space of AE is used as the starting point for training DEC. The algorithm *k-means* is applied to initialize the centroids. After using the AE encoder as the basis for DEC, a clustering layer was added. Both centroids and parameters of encoder, Θ, are now trainable parameters, and Stochastic Gradient Descent (SGD) can be used to learn the feature space and its clustering representation. In [27], the authors used a KL divergence to train the neural network.

In the first step, a *soft assignment* between the embedded points and cluster centroids is computed using Student's t-distribution [18] as a kernel to measure the similarity between the embedded point z_i and the centroid μ_j; in this way we can get the probability q_{ij} that sample i is assigned to cluster j (*soft assignment*).

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'} \left(1 + \|z_i - \mu_{j'}\|^2 / \alpha\right)^{-\frac{\alpha+1}{2}}} \tag{1}$$

where α is the number of degrees of freedom of the Student's t-distribution, we let $\alpha = 1$ for all experiments in accordance with [27].

The second step involves updating both cluster centroids and deep mapping f_Θ parameters by learning from the current high confidence assignments using an auxiliary target distribution. In other words, the loss is obtained by a KL divergence between q and p, where p is a target distribution defined as follows:

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \tag{2}$$

Here $f_j = \sum_i q_{ij}$ is the soft cluster frequency. For more details about target and *soft assignment* please refer to [27]. So, we can use KL divergence as the loss to train the network.

$$\mathcal{L}_{kld} = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

3.2 Extreme Gradient Boosting

XGBoost [7] is a ensemble learning algorithm. Based on the gradient boosting framework, XGBoost constructs a sequence of decision trees, with each subsequent tree aiming to correct the errors made by the previous ones. By iteratively refining the predictions of weak learners, XGBoost effectively captures complex relationships between input features and target variables, leading to high predictive performance and robust generalization capabilities. XGBoost uses classification and regression trees (CART) as weak learners. Trees try to complement each other. Mathematically, we can write our model in the form:

$$\hat{y}_i = \sum_{k=1}^{K} f_k\left(x_i\right), f_k \in \mathcal{F} \tag{4}$$

where K is the number of trees, f_k is a function in space $\mathcal{F}$, and $\mathcal{F}$ is the set of all possible CARTs.

The objective function to be optimized is given by the sum of the losses $l(y_i, \hat{y}_i)$ for all examples plus the sum of the tree complexities $\omega(f_k)$. The latter is used as a regularization term computed on the basis of the number of leaves and the scores assigned to the leaves. More details can be found in [7].

We used *XGBClassifier* with optimization for logistic regression, designed to handle binary classification tasks. XGBClassifier optimizes the logistic regression-specific loss function and uses the logistic activation function to produce predicted probabilities. In addition, XGBoost is known for its ability to efficiently handle problems with large data sets and high data sizes due to its highly efficient implementation and ability to take advantage of computational parallelization.

In conclusion, XGBoost is a powerful machine learning algorithm that combines gradient boosting with decision trees to obtain accurate and generalizable predictive models. With built-in regularization techniques such as tree pruning and column sampling, the model is able to avoid overfitting to training data, ensuring good generalization to new instances.

3.3 Contrastive Learning

Contrastive learning [10] is a self-supervised learning technique that aims at learning useful representations by maximizing the agreement between similar samples and minimizing the agreement between dissimilar ones. By encouraging similar samples to be closer together and dissimilar samples to be farther apart in the learned representation space, contrastive learning enables the discovery of semantically meaningful features that capture underlying patterns in the data. This makes contrastive learning particularly well-suited for tasks such as representation learning, feature extraction, and unsupervised feature learning.

4 Proposed Method

Let D be a dataset comprising pairs (x, y), where $x \in X$ and $y \in C$. Here, X represents the input space and C represents the label set. D is divided into a training set, D_{tr}, and a test set, D_{test}. Additionally, we define two subsets of C: C_k, containing the known classes, and C_u, containing the unknown classes. The objective is to construct a function $f : X \to \{known, novelty\}$ that assigns each input x to one of two categories: *known* if $y \in C_k$, and *novelty* if $y \in C_u$.

We propose Tree EXtreme Gradient Boosting with Deep Embedding for Clustering (**TEX-DEC**) exploits DEC and XGBoost, as shown in Figure 1.

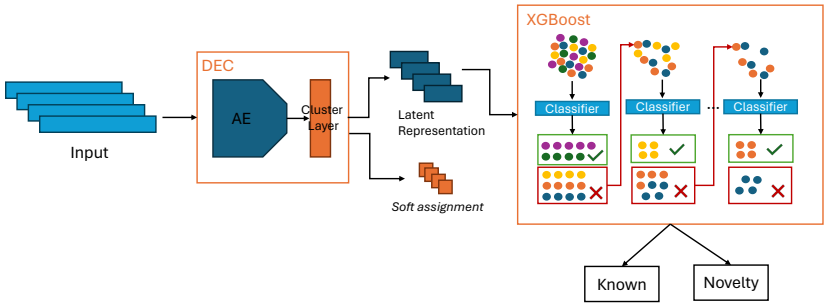

Fig. 1. DEC first extracts the latent representation and performs soft assignment. XGboost then uses the latent representation to detect Known and Novel sample.

We segmented the dataset D into three categories: *Known*, *Novelty 1*, and *Novelty 2*, as illustrated in Figure 2. We denote by *Novelty 1* and *Novelty 2* respectively the examples of novelty classes used for training the XGBoost model and the examples of novelty classes reserved exclusively for the testing phase. The *Known* subset is divided into three parts: one for training DEC, the second, along with *Novelty 1*, for training XGBoost, and the last, along with *Novelty 2*, for test TEX-DEC. One *Novelty 2* class is used at a time, allowing for the training of a separate XGBoost model for each class using a one-vs-all approach. For instance, if we have classes *A, B,* and *C* as novelty, we train three separate XGBoost models, one for each class. Each model uses only one class as *Novelty 2*. For example, if the test class *Novelty 2* is *A*, the remaining classes (*B* and *C*) are treated as *Novelty 1*.

4.1 Deep Embedding for Clustering

We leverage contrastive learning to enhance the distinctiveness of clusters obtained through DEC. By harnessing the complementary strengths of contrastive learning and traditional clustering techniques, we seek to achieve more effective

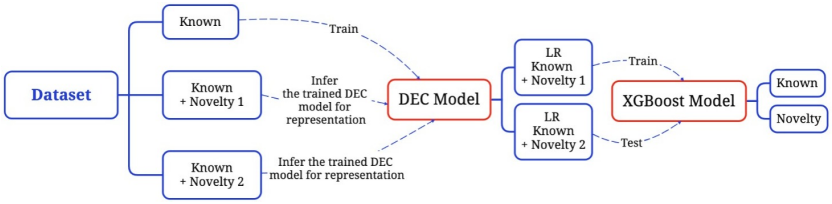

Fig. 2. The dataset is divided into three subsets: *Known, Novelty 1* and *Novelty 2*. The known subset is used to train DEC. A latent representation (LR) is obtained from each of the three subsets (note: DEC parameters are not changed at this stage). These LRs are then used to train (*Known* and *Novelty 1*) and test (*Known* and *Novelty 2*) an XGBoost classifier for sample classification.

data representation and clustering performance in our proposed approach. Our contribution is the addition of a contrastive loss and classification loss to the KL divergence used in DEC. This aids the second part of the architecture in distinguishing between novel and known samples. Below, we describe the loss components used for training.

- *KL Divergence:* as DEC, see Section 3.1, Equations 1, 2 and 3.
- *Contrastive Loss:* We calculate the average $C_{distance}$ of the Euclidean distance between each pair of centroid as follows:

$$C_{distance} = \frac{\sum_{i,j} \|\mu_i - \mu_j\|_2}{k(k-1)} \tag{5}$$

  where k is the number of clusters. This loss measures the mean Euclidean distance between cluster centroids, rather than focusing on individual sample pairs as in traditional contrastive losses [10]. To our knowledge, this specific formulation is novel. The contrastive loss we use is then:

$$\mathcal{L}_{contrastive} = \frac{1}{C_{distance}} = \frac{k(k-1)}{\sum_{i,j} \|\mu_i - \mu_j\|_2} \tag{6}$$

  The aim of this loss is to increase the distance between the centroids.
- *Classification Loss:* Since the clusters should accurately represent the actual classes of known samples. We used the Cross Entropy loss (L_{CE}), define as follow:

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^{i=N} y_i \cdot \log(\widehat{y}_i) \tag{7}$$

  where $\widehat{y}_i$ and y_i are the predicted and real labels, respectively.

The final loss is then:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{kld} + \beta \cdot \mathcal{L}_{contrastive} + \omega \cdot L_{CE} \tag{8}$$

The ablation study in Section 6 shows that each of the three terms is essential for the performance of the system. α, β and ω are used as weights during ablation study to analyze the impact of each loss component.

4.2 XGBoost

After DEC creates a smaller latent representation of the input, XGBoost is used to distinguish between known and novel instances, exploiting both known instances and a small subset of novel examples for training. Notably, our dataset is partitioned into three distinct sets: the first for training DEC exclusively with known instances, the second for training XGBoost with a mix of known and novel examples, and the third for testing, which is composed in a balanced way of novel and known instances. Importantly, the novelty classes employed in XGBoost training differ from those considered as novel in the final test set, ensuring a consistent assessment of novelty detection performance.

5 Experiments

In this section, we present the results of our experiments conducted with TEX-DEC using data described in Section 5.1. The outcomes of these experiments are detailed in Section 5.2.

5.1 Datasets

The system is applied to the task of OSR in network intrusion detection using the CIC-IDS2017 dataset [26]. As the field of NIDS is continually evolving, with the detection of new types of attacks being crucial, as mentioned above, we segmented the dataset into three categories: *Known*, *Novelty 1*, and *Novelty 2*. We also test TEX-DEC using the classical MNIST dataset [5].

NIDS Datasets In domain of NIDS, we used the CIC-IDS2017 and UNSW-NB15 datasets. The CIC-IDS2017 dataset, created by the Canadian Institute for Cybersecurity in 2017, contains packet-based data in packet capture (PCAP) format and flow-based data in CSV format. Both types of data were captured during simulated network traffic in packet-based and bidirectional flow-based formats, including the latest attacks and benign traffic. For our study, we used only packet-based data. The dataset collects simulated traffic information for an acquisition period of five days. The packet-based information in CIC-IDS2017 is unlabeled, making it necessary to use the Payload-Byte tool [6] for the extraction and labeling of network traffic packet capture files using the metadata provided in the dataset. The tool uses the features described in PCAPs to match packets with flow-based labeled data instances. Due to the variability in packet size, Payload-Byte uses a maximum payload length of 1500 bytes, with each byte converted into an 8-bit integer feature. Upon data labeling, any duplicate instances and those devoid of payload data are eliminated. The dataset contains 14 different types of attack and 1 benign class. The dataset was split as shown in Table 1. Three attacks were chosen as novel as in [19], where the authors considered each type of attack in turn as previously unknown, and identified these three attacks as those that showed the greatest performance degradation in terms of detection. Therefore, we also chose to use the same attacks as novelties.

Class	# sample	Subset
Benign	3.328.591	Known
DoS Hulk	2.219.061	Known
DoS Slowhttptest	9.778	Known
Heartbleed	41.283	Known
Brute Force (Web Attack)	28.920	Known
Sql Injection (Web Attack)	45	Known
XSS (Web Attack)	6.767	Known
Bot	5.143	Known
PortScan	946	Known
DoS GoldenEye	34.293	Novelty 1
DoS slowloris	20.877	Novelty 1
DDoS	618.544	Novelty 1
SSH-Patator	181.147	Novelty 1 or 2
FTP-Patator	110.636	Novelty 1 or 2
Infiltration	41.725	Novelty 1 or 2

Table 1. The dataset split into *Known, Novelty 1 & 2*

The UNSW-NB15 dataset [21] is a NIDS dataset developed to identify normal and attack network traffic. The raw network packets were generated by the Australian Centre for Information Security (ACCS) [20]. This dataset was preprocessed in the same manner as the previous one. The Payload-Byte tool [6] was applied to it. This dataset is used as a covariate to test the robustness of our approach, even with datasets from different network configurations. The UNSW dataset is used only during the testing phase and is labeled as *Novelty 2*.

MNIST dataset In our experimental evaluation, we also used the MNIST dataset [5], a well-established benchmark comprising simple handwritten digits. MNIST comprises ten classes representing numbers from 0 to 9. We partitioned this dataset into three subsets: *known*, *Novelty 1*, and *Novelty 2*: digits 0 through 4 were designated as *known*, while digits 5 through 9 were categorized as both *Novelty 1* and *Novelty 2*. During the testing phase, we employed a one-vs-all strategy, wherein a single *Novelty 2* class was utilized at a time, enabling the training of individual XGBoost models for each class.

5.2 Results

As previously mentioned, we tested our approach on different datasets. DEC was trained using different loss configurations, as discussed in Section 6. Additionally, we conducted a grid search on the hyper-parameters of XGBoost, including the number of components and maximum depth, to achieve optimal results. We evaluated the models using the Area Under the Receiver Operating Characteristic Curve (AUROC) because it proves effective in measuring the performance of a binary classification model under various scenarios. AUROC provides a comprehensive assessment of the model's ability to discriminate between positive and negative classes, considering both sensitivity and specificity.

For the CIC-IDS2017 dataset, we compared our results with those of [19] and [29]. Specifically, we compared the AUROC for the *Novelty 2* classes. Additionally, we compared the AUROC for the UNSW-NB15 dataset with [19]. In this case, we did not encounter a semantic shift, but rather a covariate shift. Essentially, we had the same classes but from different datasets and thus different distributions. The results, reported in Table 2, show a significant improvement in the performance of our method compared to previous approaches. For instance, in detecting the previously unknown Infiltration attack, we achieved an AUROC of 0.9843, surpassing the performance of existing methods. Similarly, our method exhibited an AUROC score of 0.9939 for the previously unknown SSH-Patator attack, slightly outperforming [19] (0.9921) and significantly surpassing [29] (0.6787). In the case of the previously unknown FTP-Patator attack, although our AUROC of 0.9950 is slightly lower than [19] (0.9957), it notably outperforms [29] (0.7955). Moreover, when assessing the UNSW-NB15 dataset, which was generated from an entirely different distribution than CIC-IDS2017, our AUROC of 0.9939 surpassed the result reported by [19] (0.9583). This fact underscores the robustness and generalizability of our methodology for detecting novelty across different datasets. While the first three attacks are novel within the same dataset, our method demonstrates high adaptability and detection capability even on a completely different dataset like UNSW-NB15. This generalization ability instills confidence in the validity and utility of our approach across a variety of real-world scenarios.

	AUROC		
Novelty Type	Matejek et al. [19]	Zavrak et al. [29]	TEX-DEC
FTP-Patator	**0.9957**	0.7955	0.9950
Infiltration	0.9742	0.8965	**0.9843**
SSH-Patator	0.9921	0.6787	**0.9939**
UNSW-NB15	0.9583	-	**0.9939**

Table 2. The AUROC results of the methods for specified previously unknown attacks

For the MNIST dataset, during the training phase samples labeled as known were used to train the DEC model, with the aim of establishing a robust representation of these digits. Separate XGBoost models were trained for each number within the Novelty interval (5, 6, 7, 8, 9) using a one-vs-all classification strategy. This training process allowed the development of specialized classifiers to distinguish each novelty class from the normal class. The performance was compared with AAE-II [12] and Isolation Forest [17], the latter two obtaining an AUROC of 0.619, and 0.841, respectively. TEX-DEC achieved 0.935, which significantly exceeded the results obtained in [12]. These results indicate the effectiveness of our approach in detecting novel samples in the MNIST dataset.

6 Ablation Study

As mentioned previously, our work relies on a loss function with three components: KL divergence (L_{kld}), contrastive loss ($L_{contrastive}$), and classification loss (L_{CE}). In this section, we investigate various configurations of this loss to achieve the best results. To conduct this study, we partitioned the dataset consistently and trained only one autoencoder for all configurations, ensuring a common starting point for comparison. The loss function is described in Equation 8. We set the weights α, β, and ω by considering all possible combinations of 0 and 1 to *turn off* individual loss components.

The results are presented in Table 3. Each cell in the table reports the AUROC value for the specific configuration, along with the difference compared to the optimal configuration (highlighted in bold).

α	β	ω	**SSH-Patator**	**Infiltration**	**FTP-Patator**	**UNSW-NB15**	**Average**
1	1	1	99.50	98.43	99.39	**99.39**	**99.28**
1	0	0	99.66	84.54	99.03	98.88	95.53
0	1	0	99.41	98.29	99.52	99.28	99.13
0	0	1	99.57	**98.61**	**99.56**	99.35	99.27
1	1	0	**99.69**	84.52	99.03	98.85	95.52
1	0	1	99.68	84.40	99.05	98.88	95.51
0	1	1	99.51	98.38	99.36	**99.39**	99.16

Table 3. The table displays the AUROC for the different test *Novelty 2*: SSH-Patator, Infiltration, FTP-Patator, and UNSW-NB15.

The results, summarized in Table 3, show discernible trends among different configurations. In particular, configurations in which all loss components are turned on ($\alpha = \beta = \omega = 1$) consistently produce AUROC values close to optimal levels, underscoring the synergistic contribution of each component to overall model performance levels. In contrast, configurations with the KL divergence component active and one or more components at 0 ($\alpha = 1$ *and* $\beta = \omega = 1$ *or* 0) show lower performance, particularly in the case of *Infiltration*, emphasizing the indispensable role of each component in facilitating effective novelty detection.

Intermediate configurations, in which specific loss components are selectively activated, reveal nuances about their respective contributions. They show how the use of classification and contrastive loss lead to improved novelty detection. Systematic exploration of loss function configurations provides valuable insights into the interaction between individual components and their collective impact on novelty detection performance. Such insights are critical in guiding the refinement and optimization of novelty detection systems, thereby advancing the state of the art in novelty detection research.

7 Conclusion

This research was primarily aimed at enhancing the robustness of machine learning by NIDS by improving the detection capabilities for previously unknown attacks, a critical aspect of modern network security. As we navigate through an era of rapidly advancing technological threats, the ability to identify novel, complex attacks becomes imperative. The system presented in this paper, which synergistically combines a neural component (DEC) with a symbolic component (XGBoost), leverages the strengths of neural networks in feature extraction from extensive data sets along with the robust decision-making capabilities of decision tree ensembles. This neuro-symbolic artificial intelligence approach not only enhances the robustness and adaptability of NIDS but also contributes significantly to the domain by improving the system's ability to recognize and react to new threats dynamically. Moreover, the integration of these technologies offers improved explainability and the ability to discern complex relationships within the input data, setting a foundation for addressing more intricate challenges in network security. The effectiveness and innovative aspects of this approach are underscored by its application to both previously unknown attack detection in NIDS and scenario involving handwriting recognition, achieving a high AUROC of 0.99 in both domains. This underscores the versatility and potential of our approach to generalize across different types of data and applications, paving the way for broader implementations in cybersecurity and beyond.

Acknowledgments

This work was supported in part by the Italian Ministry of University and Research through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 del 15/03/2022), Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by the European Union under the NextGeneration EU programme"; and by the U.S. Military Academy (USMA) under Cooperative Agreement No. W911NF-23-2-0108, the U.S. Army Combat Capabilities Development Command Army Research Laboratory under Support Agreement No. USMA 21050, and the Defense Advanced Research Projects Agency under Support Agreement No. USMA 23004. The views and conclusions expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, U.S. Army, U.S. Department of Defense, or U.S. Government.

References

1. Abdalgawad, N., Sajun, A., Kaddoura, Y., Zualkernan, I.A., Aloul, F.: Generative deep learning to detect cyberattacks for the iot-23 dataset. IEEE Access **10**, 6430–6441 (2021)
2. Andresini, G., Appice, A., Di Mauro, N., Loglisci, C., Malerba, D.: Multi-channel deep feature learning for intrusion detection. IEEE Access **8**, 53346–53359 (2020)

3. Asam, M., Khan, S.H., Akbar, A., Bibi, S., Jamal, T., Khan, A., Ghafoor, U., Bhutta, M.R.: Iot malware detection architecture using a novel channel boosted and squeezed cnn. Scientific Reports **12**(1), 15498 (2022)
4. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1563–1572 (2016)
5. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
6. Farrukh, Y.A., Khan, I., Wali, S., Bierbrauer, D., Pavlik, J.A., Bastian, N.D.: Payload-byte: A tool for extracting and labeling packet capture files of modern network intrusion detection datasets. In: 2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT). pp. 58–67. IEEE (2022)
7. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
8. Ge, Z., Demyanov, S., Chen, Z., Garnavi, R.: Generative openmax for multi-class open set classification. arXiv preprint arXiv:1707.07418 (2017)
9. Geng, C., Huang, S.j., Chen, S.: Recent advances in open set recognition: A survey. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3614–3631 (2020)
10. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006)
11. Hassen, M., Chan, P.K.: Learning a neural-network-based representation for open set recognition. In: Proceedings of the 2020 SIAM International Conference on Data Mining. pp. 154–162. SIAM (2020)
12. Hassen, M., Chan, P.K.: Unsupervised open set recognition using adversarial autoencoders. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 360–365. IEEE (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
14. Hwang, R.H., Peng, M.C., Huang, C.W., Lin, P.C., Nguyen, V.L.: An unsupervised deep learning model for early network traffic anomaly detection. IEEE Access **8**, 30387–30399 (2020)
15. Khan, A.S., Ahmad, Z., Abdullah, J., Ahmad, F.: A spectrogram image-based network anomaly detection system using deep convolutional neural network. IEEE access **9**, 87079–87093 (2021)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
17. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
18. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
19. Matejek, B., Gehani, A., Bastian, N.D., Clouse, D., Kline, B., Jha, S.: Safeguarding network intrusion detection models from zero-day attacks and concept drift
20. Moustafa, N.: Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic. Ph.D. thesis, UNSW Sydney (2017)

21. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Information Security Journal: A Global Perspective **25**(1-3), 18–31 (2016)
22. Neal, L., Olson, M., Fern, X., Wong, W.K., Li, F.: Open set learning with counterfactual images. In: Proceedings of the European conference on computer vision (ECCV). pp. 613–628 (2018)
23. Parmar, J., Chouhan, S., Raychoudhury, V., Rathore, S.: Open-world machine learning: applications, challenges, and opportunities. ACM Computing Surveys **55**(10), 1–37 (2023)
24. Sabeel, U., Heydari, S.S., Elgazzar, K., El-Khatib, K.: Building an intrusion detection system to detect atypical cyberattack flows. IEEE Access **9**, 94352–94370 (2021)
25. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE transactions on pattern analysis and machine intelligence **36**(11), 2317–2324 (2014)
26. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
27. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International conference on machine learning. pp. 478–487. PMLR (2016)
28. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334 (2021)
29. Zavrak, S., Iskefiyeli, M.: Anomaly-based intrusion detection from network flow features using variational autoencoder. IEEE Access **8**, 108346–108358 (2020)
30. Zhu, F., Ma, S., Cheng, Z., Zhang, X.Y., Zhang, Z., Liu, C.L.: Open-world machine learning: A review and new outlooks. arXiv preprint arXiv:2403.01759 (2024)