A Hybrid Approach to Improve the Intrusion Detection Systems Using Generative Artificial Intelligence and Deep Reinforcement Learning

Ines Ben Makhlouf[1], Ghassen Kilani[1], Fehmi Jaafar[2] and Haïfa Nakouri[2,3]
[1]*Mediterranean School of Technology, South Mediterranean University (SMU), Tunis, Tunisia*
[2]*Department of Computer Science and Mathematics, Université du Québec à Chicoutimi (UQAC), Quebec, Canada*
[3]*University of Tunis, LARODEC, Institut Supérieur de Gestion (ISG), Tunisia*

Keywords: Intrusion Detection Systems, Generative AI, Deep Reinforcement Learning, Adversarial Autoencoder, Twin-Delayed Deep Deterministic Policy Gradient, Double Deep Q-Network.

Abstract: In recent years, Artificial Intelligence (AI)-based tools have gained widespread adoption as AI-powered prompts have become increasingly sophisticated. As a result, the rise of AI-integrated websites has created a growing demand for more sophisticated tools to protect devices and networks, especially in light of the emergence of AI-generated malware. Indeed, numerous studies anticipated the threats posed by this type of malware and proposed a variety of solutions to address this issue. In this context, most research introducing generative AI frameworks deals with image-based data, prompting the need to analyze tabular network data. We propose AAE-DRL, an Intrusion Detection System (IDS) that utilizes generative AI and deep reinforcement learning to replicate and predict intrusion behavior. We demonstrate the advantages and limitations of combining reconstruction and adversarial learning objectives with Deep Reinforcement Learning (DRL) in terms of intrusion detection, data generation, and minority sampling. Our approach achieved 89% accuracy, 90% precision, 91% recall, 90% F1-score on the augmented dataset with a 97% Area Under the Curve (AUC).

1 INTRODUCTION

A Network Intrusion Detection System (NIDS) is an essential part of the security infrastructure in an organization, designed to safeguard internal networks and information systems from malicious threats (Sayed and Taha, 2023). With the rapid growth of digital systems and interconnected networks, ensuring network security has become a critical challenge. Intrusions, such as unauthorized access and malicious activities, pose severe threats to sensitive data, system integrity, and availability.

Furthermore, the increase in technological innovation, especially in AI, has driven a demand for AI-powered websites. Numerous websites began integrating generative AI features into their business models to enhance their products and services, such as implementing bots for customer support and personalizing content recommendations. In response to the widespread adoption of generative AI tools, the sophistication of malware generation has notably increased with the emergence of AI-powered malware, showcasing heightened intelligence and the ability to operate autonomously. In a recent study by (Gaber et al., 2024), the authors demonstrated that AI-generated malware can evade traditional analysis methods due to the inherent opacity of neural network decision-making processes.

Many research studies have proposed Deep Learning (DL)-based IDS to address novel threats (Lansky et al., 2021). In particular, adversarial AI showed promising results in IDS applications, exhibiting proven resilience against sophisticated cyberattacks. Algorithms such as Generative Adversarial Networks (GANs) generate diverse network traffic patterns that closely mimic real-world attack variations, enabling a more robust IDS capable of detecting evolving threats (Chiriac et al., 2025). In this context, recent studies, including (H. M. Kotb, 2025), have emphasized the importance of distinguishing between synthetic and real-world malware for effective cybersecurity practices. This differentiation impacts detection methods, threat assessment, and the overall strategy for combating cyber threats.

Additionally, an IDS relies on dataset quality to determine detection accuracy and identify false negatives. However, many intrusion detection datasets are unbalanced due to the inherent nature of network traffic and attack frequency, as most real-world network traffic is normal, leading to data bias. Reconstruction-based models, such as Autoencoders and their variants, showed promising results in learning the under-

Ben Makhlouf, I., Kilani, G., Jaafar, F. and Nakouri, H.
A Hybrid Approach to Improve the Intrusion Detection Systems Using Generative Artificial Intelligence and Deep Reinforcement Learning.
DOI: 10.5220/0013567700003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 467-474
ISBN: 978-989-758-760-3; ISSN: 2184-7711

lying distribution of the data and capturing complex feature interactions, thus generating diverse and realistic samples by capturing the latent space structure.

In terms of combining adversarial models with reconstruction-based models, most studies focused on image-based data because Autoencoders are inherently unsupervised, which can lead to the preservation of noise and irrelevant features in the encoded representation in tabular data.

In this paper, we introduce AAE-DRL, an IDS that incorporates a supervised Adversarial Autoencoder (AAE) and two Deep Reinforcement Learning (DRL) algorithms; Twin-delayed Deep Deterministic Policy Gradient (TD3) and Double Deep Q-Network (DDQN). AAE-DRL offers a novel solution for generating tabular intrusion samples and adversarially predicting the intrusion type.

To the best of our knowledge, AAE-DRL is the first approach to utilize the capabilities of the originally proposed AAE and hybrid DRL algorithms tested on tabular data. We investigate the effectiveness of our approach based on the following:

- Predicting the type of intrusion using a deep learning classifier.
- Optimizing data generation using deep reinforcement learning.
- Generating real intrusion samples for minority classes.

We showcase a significant improvement across five metrics when utilizing AAE-DRL for data augmentation, demonstrating the role of minority sampling in predicting rare intrusion variants. These metrics include: model losses, accuracy, precision, recall, F1-score and AUC. We also address the assumptions and limitations of using AAE-DRL as a fully developed IDS.

The rest of the paper is organized as follows: Section 2 provides the mathematical formulations of the deep learning algorithms used. Section 3 reports related works and their limitations. Section 4 presents the design of the empirical study. Section 5 presents the results and findings. In Section 6, we discuss those findings and their limitations.

2 BACKGROUND

In this section, we present the mathematical structure of the deep learning algorithms used in this study.

2.1 Adversarial Autoencoder

In (Makhzani et al., 2015), the authors proposed Adversarial Autoencoders (AAE), a generative model with a dual objective; reconstruction of error and adversarial training. Specifically, the encoder functions as a generator in the adversarial framework. As the encoder learns to produce a latent representation that aligns with a predefined prior distribution $p(z)$, the adversarial network ensures that the aggregated posterior $q(z)$ matches $p(z)$. In this context, the encoder/generator ensures that $q(z)$ can fool the discriminator into thinking that the hidden code comes from $p(z)$. Thus, the aggregated posterior $q(z)$ is defined as:

$$q(z) = \int_x q(z|x)p_d(x)dx \quad (1)$$

where $q(z|x)$ is the encoding distribution and $p_d(x)$ is the data distribution (Makhzani et al., 2015).

2.2 Twin-delayed Deep Deterministic Policy Gradient

In (Fujimoto et al., 2018), the authors introduced Twin-delayed Deep Deterministic Policy Gradient (TD3) as a variant of Deep Deterministic Policy Gradient. In TD3, the actor is the policy network that maintains a function $\mu(s \mid \theta^{\mu})$ specifying the current policy by deterministically mapping states to an action, where θ^{μ} are the parameters of the actor network. The critics, on the other hand, are Q-networks denoted by $Q(s,a|\theta^{Q})$, learned using the Bellman equation. Moreover, policy smoothing is applied by adding a small amount of random noise to the target policy.

The actor is updated by maximizing the q-value estimated by both critics, while the policy network parameters are updated using a deterministic policy gradient, which is defined as:

$$\nabla_{\phi} J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_{\phi}(s)} \nabla_{\phi} \pi_{\phi}(s) \quad (2)$$

where $\nabla_a Q_{\theta_1}(s,a)$ represent the gradient of the first critic evaluated at $a = \pi_{\phi}(s)$ (Fujimoto et al., 2018). The target critic networks θ' and the policy network ϕ' are both updated by adding a small value to achieve a soft target network update.

2.3 Double Deep Q-Network

In (van Hasselt et al., 2015), the authors introduced Double Deep Q-Network (DDQN), an extension of DQN that addresses overestimation bias in Q-learning in discrete observation spaces. Accordingly, DDQN induces an upward bias using the double estimator method. The target value Y_k^Q is defined as:

$$Y_k^{\text{DQNN}} = r + \gamma Q\left(s', \arg\max_{a \in \mathcal{A}} Q\left(s', a; \theta_k\right); \theta_k^-\right) \quad (3)$$

where r is the reward, γ is the discount factor, Q is the Q-value function, s' is the next state, a is the current action, θ_k is the current weight and θ_k^- is the weights of the target network.

3 RELATED WORK

In this section, we highlight the efforts of intrusion detection using Generative AI.

In (M. Ali and Zhang, 2024), the authors examine the role of generative intelligence, particularly Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), in detecting cyber threats and anomalous network behaviors. This paper is a survey that discusses several datasets in network intrusion detection research, such as NSL-KDD, KDD99, and IoT-23 and compares key findings from existing literature, including performance metrics like accuracy, false alarm rate, and detection rate reported in previous studies that used generative models on these datasets. However, the authors outline several limitations in using generative AI for intrusion detection, including their susceptibility to adversarial attacks.

In (Alabsi et al., 2023), the authors proposed an IDS that leveraged a Conditional Tabular Generative Adversarial Network (CTGAN) to detect attacks in IoT environments and generate synthetic network traffic data to address dataset imbalance. The authors tested on 100,000 records from the Bot-IoT dataset, focusing on the 10 key features. By utilizing the synthetic data generated by CTGAN, the study describes training multiple machine learning and deep learning classifiers, including Logistic Regression (LR), Naive Bayes (NB) and Gated Recurrent Units (GRUs). Accordingly, LSTM achieves the highest detection accuracy of 99.4%, precision of 96.6%, recall of 100% and F1 measure of 99.6%. These results might suggest overfitting, especially with a 100% recall. Moreover, condition algorithms, such as CTGAN, rely on predefined rules and signatures, limiting their effectiveness against unknown threats.

In (Zhao et al., 2021), the authors introduced attackGAN; an improved adversarial attack model based on Wasserstein GAN by adding the feedback of IDS. The authors considered shallow Machine Learning (ML) and Deep Neural Networks (DNN). For benchmarking, the authors considered Fast Gradient Sign Method (FGSM), Project Gradient Descent (PGD), CW attack (CW) and GAN-based algorithms. Using the NSL-KDD dataset, this approach achieved a success rate of 81.37% and an evasion rate of 87.18%, outperforming the GAN-based models. Nonetheless, the paper notes some existing methods do not preserve network traffic features, which leads to invalid traffic data that could be detected by the IDS.

4 EMPIRICAL STUDY

In this section, we describe the process of generating synthetic data and predicting types of intrusion using AAE-DRL. Figure 1 illustrates the overall setup of our approach, starting from data preparation, data generation using AAE, then minority sampling and data augmentation using DRL-integrated AAE. Furthermore, we detail our supervised AAE approach in Figure 2.

4.1 Dataset Description

In this subsection, we consider the UNSW-NB15 dataset to assess the performance of our approach, mainly, the dataset's train and test partition (Moustafa and Slay, 2015). The authors in the Cyber Range Lab of UNSW Canberra created raw network packets of the UNSW-NB15 dataset using the IXIA PerfectStorm tool to generate a hybrid of real modern normal activities and synthetic contemporary attack behaviors. This dataset includes nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. By developing twelve algorithms for the data collected by Argus and Bro-IDS tools, the authors generated 2,540,044 records and 49 features and classes, stored in four CSV files. Moreover, the authors configured a partition from the dataset into a train partition and a test partition. The number of records in the training set is 175,341 and the testing set is 82,332. We follow the standard process of data cleaning and scaling, where we drop null values, encode categorical features using LabelEncoder, and scale continuous features using MinMaxScaler according to the value range in the dataset; exclusively positive. Moreover, we keep the 30 most important features, determined using Recursive Feature Elimination (RFE).

4.2 Data Generation

In this subsection, we explain the role of the AAE component in reconstructing the original data using its decoder. We build our AAE with Linear layers and ReLU activation functions, given the data distribution and range; non-linear and exclusively positive. The overall setup of our AAE follows the original design proposed by the authors in (Makhzani et al., 2015), except for the following:

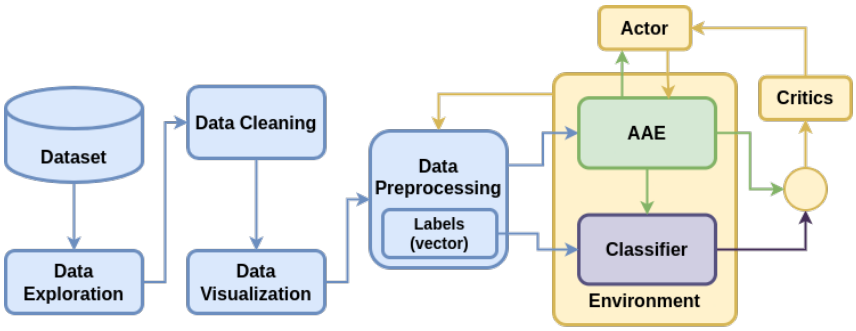

Figure 1: The general setup of our proposed approach.

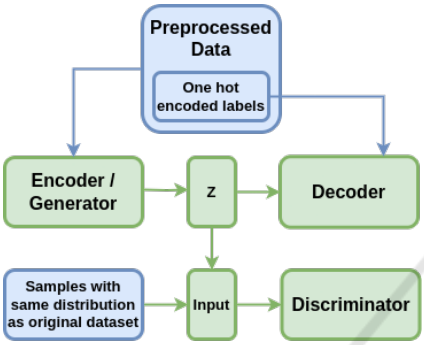

Figure 2: The supervised adversarial autoencoder approach.

- The discriminator uses an attention mechanism to prioritize relevant features
- The decoder's final activation function layer is customized according to the feature type; Rectified Linear Unit (ReLU) if continuous, SoftMax if multi-class, and Sigmoid if binary. The decoder's loss function is also customized; Mean Squared Error (MSE), Cross-Entropy (CE) and Binary Cross-Entropy (BCE) if continuous, multi-class and binary, respectively. These modifications effectively constrain the decoder to operate within a specific range.
- Our AAE is supervised, hence, the decoder inputs the encoded vector as well as the one-hot encoded labels.
- We add a gradient penalty to the discriminator to avoid mode collapse.

Finally, our trained decoder generates data by interpolating between data points in the latent space, allowing our AAE to return unlabeled samples to the classifier.

4.3 Minority Sampling

In this subsection, we describe the process of solving data imbalance with minority sampling. First, we implement a semi-supervised learning technique called pseudo labeling, where we use the TabNet classifier for pre-training on the supervised (original) dataset and then for predicting labels on the unsupervised (synthetic) dataset. Rather than using the pre-trained version from the "pytorch_tabnet" library, we clone another implementation[1] in order to change the BatchNorm1D layers in the classifier's architecture to LayerNorm, as the classifier will be in evaluation mode when implemented in the DRL environment.

Second, we implement TD3 and DDQN hybrid, where we integrate our DDQN into our TD3's critic network. The reason behind using both algorithms is that TD3 and DDQN have in common the structure; two target networks to mitigate bias overestimation, except that TD3 is exclusively used for continuous action spaces and DDQN for discrete action spaces. The workflow of the DRL environment is as follows:

1. The latent vector is optimized according to the action space.
2. The AAE's decoder inputs the optimized latent vector and outputs new samples as input.
3. The AAE's encoder/generator inputs the decoded samples and generates a new latent vector.
4. The AAE's discriminator inputs the encoded vector and calculates the reward.
5. The TabNet classifier inputs the normalized decoded samples and calculates the reward as well.
6. Both rewards are joined; the coefficients of the discriminator and the classifier are set to 0.4 and 0.2, respectively.

Lastly, yet significantly, we force minority sampling by adjusting the weights to favor minority classes when calculating the classifier reward. Thus, our DRL algorithm processes the synthetic data to create a new balanced dataset. We perform pseudo-labeling on the DRL-generated dataset using our classifier, the final supervised dataset is then used to augment our

[1] https://github.com/sourabhdattawad/TabNet

original training data. Subsequently, we train our AAE again on the expanded training set to compare the effect of data augmentation.

5 RESULTS

In this section, we present the results of our approach and compare them to similar studies.

5.1 Class Imbalance

In this subsection, we explain the motivation behind manually balancing classes.

We discover severe data imbalance favoring normal behavior in the train and test partitions. Thus, we add more samples, mainly representing malicious intrusion, from the four main UNSW-NB15 datasets. We also add 677,785 benign samples to adjust the malicious-to-benign ratio.

Additionally, we noticed that the three majority classes in "attack_cat" represent 97.47% of the total sample size. Therefore, we join labels according to their description and define four classes labeled "Malware and Low-level Attacks", "Generic", "Exploits" and "Normal", the first class being the minority class, representing 10% of the data. Therefore, the final dataset contains 616,606 samples, 30 features and 1 target; "attack_cat".

5.2 Comparative Research

To the best of our knowledge, we are the first to test the mentioned models and algorithms on tabular data; thus, we consider approaches tested on image-based data.

Concretely, we chose the following two approaches because of the similarity in architecture: combining an adversarial model and a reconstruction model with a DRL algorithm.

In (Abbasian et al., 2023), the authors developed RL-Controlled GAN; an approach to image-to-image translation that combined an AutoEncoder, a Generative Adversarial Network, a DL classifier, and a Twin Delayed Deep Deterministic Policy Gradient as a backpropagator. The authors provided us with the source code.

In (Fuhl et al., 2020), the authors introduced an approach for eye-tracking data manipulation while protecting user privacy, combining an AutoEncoder and a Double Deep Q-Network (DDQN) as well as two main DL classifiers (A and B) which also represent one of the DDQN agents, named "classification agent".

5.3 Benchmark Classification

We assess all approaches using the following shallow ML classifiers: Random Forest (RF), Extreme Gradient Boosting (XGB), Gradient Boosting (GB), and K-Nearest Neighbor (KNN). We use the described classifiers to evaluate the synthetic data using 5 metrics: accuracy, precision, recall, F1-score, and AUC, and compare results before and after data augmentation performed on the training set.

5.3.1 Benchmark Classification Evaluated on the Unaugmented Dataset

Table 1 presents the results of the benchmark classification on the synthetic data reconstructed by our decoder compared to the data reconstructed by the research studies described above. AAE-DRL reported 86% accuracy, 81% precision, 75% recall, and 94% AUC across almost all classifiers with a 1% interval, meanwhile F1-score ranges from 70% to 74%, compared to 39% accuracy, 27% precision, 26% recall, 19% F1-score and 53% AUC, reported by RL-Controlled GAN and a reported 51% accuracy, 52% precision, 39% recall 38% F1-score and 72% AUC by AE+DQN.

5.3.2 Benchmark Classification Evaluated on the Augmented Dataset

In Table 1, we showcase the results of adding data augmentation generated by our decoder, similarly comparing it with other research studies using their respective decoders. AAE-DRL showed significant improvement, especially in the precision and recall metrics, which increased by 10% and 12% respectively, compared to the 9% and 2% increase reported by RL-Controlled GAN, as well as a 10% increase and a 4% decrease in AE+DQN. However, AE+DQN reported an increase of up to 90% and 89% in precision and recall in the weighted average, which explains the high accuracy and displays the problem of data imbalance.

5.3.3 Effect of Data Augmentation on Minority Class

Before data augmentation, the minority class ("Malware and Low-level Attacks") achieved 58% to 98% precision, 4% to 63% recall, and 7% to 74% F1-score, suggesting that some of the benchmark classifiers struggled with identifying false negatives, as presented in Table 2. After data augmentation, the precision did not change, however, the recall and the F1-score increased to 72-75% and 83-84%, respec-

Table 1: Benchmark classification results.

		Unaugmented data					Augmented data				
		Accuracy	Precision	Recall	F1	AUC	Accuracy	Precision	Recall	F1	AUC
AAE-DRL	RF	0.86	0.81	0.74	0.72	0.94	0.89	0.90	0.91	0.90	0.97
	XGB	0.86	0.81	0.75	0.73	0.94	0.89	0.90	0.90	0.89	0.97
	GB	0.86	0.80	0.74	0.70	0.94	0.89	0.90	0.91	0.90	0.97
	KNN	0.86	0.81	0.75	0.74	0.94	0.89	0.90	0.91	0.90	0.97
RL-GAN	RF	0.39	0.27	0.25	0.14	0.51	0.34	0.36	0.26	0.16	0.54
	XGB	0.39	0.26	0.26	0.19	0.53	0.35	0.34	0.28	0.21	0.55
	GB	0.39	0.10	0.25	0.14	0.50	0.34	0.33	0.26	0.17	0.52
	KNN	0.36	0.23	0.25	0.21	0.50	0.30	0.25	0.25	0.23	0.50
AE+DQN	RF	0.51	0.52	0.37	0.35	0.67	0.91	0.59	0.35	0.39	0.74
	XGB	0.50	0.43	0.39	0.38	0.72	0.91	0.62	0.33	0.37	0.72
	GB	0.46	0.40	0.33	0.30	0.64	0.89	0.52	0.30	0.32	0.66
	KNN	0.37	0.28	0.26	0.23	0.52	0.87	0.35	0.26	0.25	0.66

Table 2: Benchmark classification results of the minority class before and after data augmentation.

	Unaugmented data		Augmented data	
	Precision	Recall	Precision	Recall
RF	0.62	0.08	0.98	0.72
XGB	0.64	0.11	0.98	0.72
GB	0.58	0.04	0.96	0.73
KNN	0.64	0.13	0.95	0.75

tively, suggesting the effect of minority sampling using AAE-DRL.

5.4 Approach Losses

In Table 3, we present the losses reported by the generator, the discriminator, the classifier, and the DRL reward, or in this case, punishment.

5.4.1 Adversarial Training Losses

AAE is trained for 101 epochs with a training batch size of 32, performing evaluation every 10 epochs using a batch size of 64 for the validation set. Then, we test our AAE with the testing set.

Table 3 shows the testing losses of the generator and discriminator without data augmentation at 14.2% and 43.0%, compared to 54.5% and 156.8% generator loss and 68.4% and 36.3% discriminator loss for RL-Controlled GAN and AE+DQN, respectively.

We notice that the authors of AE+DQN considered the AE as a generator and classification agent as a discriminator

5.4.2 Classifier Loss

As mentioned, we train TabNet on the original data for 51 epochs with a batch size of 32, performing evaluation every 10 epochs using a batch size of 64. After generating a balanced dataset with the DRL algorithm, TabNet is evaluated for a second time, where we predict pseudo labels. Moreover, we use CE loss in both cases as well as a confidence level for each label generated. Similarly, we test TabNet's performance with the test set.

Table 3 showcases TabNet classifier loss, which achieved 29.1% on the test set, compared to 34.6% and 36.3% reported by the classifiers implemented in RL-Controlled GAN and AE+DQN, respectively. Moreover, the confidence level for each pseudo label generated ranged from 49% to 90% and averaged 84%, suggesting the accuracy of the model classifier in predicting labels.

5.4.3 RL Reward/Punishment

We set the start time step for exploitation to 50 and the total (maximum) time steps to 4000, as well as the frequency of policy updates to 1000. The training batch size is set to 32 where we perform evaluation every 400 episodes on the validation set. The maximum time step and the training batch size control the dataset shape, as the DRL-enhanced decoder outputs 128,000 samples. Furthermore, the critic network utilizes MSE loss, as for the rewards, we combine the classifier prediction with the discriminator output, where we use CE and BCE, respectively.

Consequently, we iterate through 100 epochs and 64 batches in the test set. We also compute the encoded representation of the previous state and current state, yielding both states in a tuple. As shown in Table 3, AAE-DRL achieved 22.0% testing loss, compared to 217.0% and 6.7% testing loss in RL-Controlled GAN and AE+DQN, respectively.

Table 3: Approach losses.

	Generator loss	Discriminator loss	Classifier loss	DRL reward
AAE-DRL	0.142	0.430	0.291	0.238
RL Controlled GAN	0.545	0.684	0.376	2.170
AE+DQN	1.568	0.363	0.363*	0.067

6 DISCUSSION AND THREATS TO VALIDITY

In this section, we discuss the advantages of applying our approach and its limitations.

6.1 Discussion

In this subsection, we highlight the efficiency of our approach in detecting AI-generated intrusion.

The recall metric in intrusion detection systems (IDS) is crucial because it measures the false negative ratio. False negatives are particularly concerning because they allow real intrusions to go undetected, potentially leading to significant security breaches and data loss.

Our analysis shows that data augmentation significantly impacted detection accuracy and other metrics, especially recall.

In early training trials, our AAE experienced mode collapse, a common problem in GANs, where the model was stuck in the local minima because the generator is focusing on producing a limited set of data patterns. Therefore, we add a gradient penalty to our discriminator loss to encourage our generator to generate diverse samples.

However, AAE-DRL remains susceptible to mode collapse with extended training; specifically, we observed that our AAE consistently stagnates around the 60th epoch. We note that we deliberately maintained the original architecture and penalty parameters to ensure a fair comparison.

Furthermore, we observed that the DRL algorithm continues to generate samples primarily for the "Exploits" majority class while neglecting the "Generic" minority class. This behavior indicates mode collapse, which is further explained by the unchanged recall results presented in Table 4. Therefore, we will deeply investigate methods on how to adapt model complexity and regularization techniques, namely the gradient penalty, to the dataset size, as well as optimizing the minority sampling technique.

On another note, we use shallow machine learning algorithms as benchmarks for several reasons, including practicality, interpretability, and avoiding redundancy, as the TabNet classifier can also provide detection accuracy.

Table 4: Benchmark classification results of the "Generic" class before and after data augmentation.

	Unaugmented data		Augmented data	
	Precision	Recall	Precision	Recall
RF	0.88	1.00	0.89	1.00
XGB	0.88	1.00	0.89	1.00
GB	0.88	1.00	0.89	0.99
KNN	0.88	1.00	0.89	1.00

6.2 Threats to Validity

In this subsection, we discuss the threats to the validity of our approach following a set of guidelines (Wohlin et al., 2012).

Threats to Construct Validity: we identify variables correlated to the labels, however, we are not able to conclude that correlated variables directly cause a change to target or each other. Furthermore, we implement the gradient penalty with a coefficient of 0.2 (less than 1), thus weakening the Lipschitz constraint and breaking the theoretical grounding of adversarial training. The purpose of choosing this coefficient is to balance the AAE losses, as the discriminator becomes over-penalized if the coefficient is greater than 0.5.

Moreover, we convert Conv2D layers and Conv2DT layers to Linear layers + activation function in the comparative research studies in order to test it on tabular data.

We highlight the fact that the results described in Section 5 may vary depending on trials due to the stochastic nature of the training process. To mitigate this threat, we conducted multiple runs with different random seeds and reported their average performance.

***Threats to Internal Validity*:** the results described also depend on the resources available. Specifically, the version mismatch between the CUDA toolkit and Pytorch resulted in training instability. Resource limitations also affect the size of the augmented data as well as the number of neurons in the comparative studies when converting layers.

***Threats to Conclusion Validity*:** in the introduction section, we highlight the role of adversarial learning in providing insights into neural network behavior, however, they do not completely resolve interpretability challenges.

***Threats to External Validity*:** we utilize the

UNSW-NB15 dataset, released in 2015, which is considered outdated. The reason for choosing this dataset is its size (high sample size and manageable feature size) compared to other datasets in the same field.

***Threats to Replicability*:** in this study, we described the process of implementing AAE-DRL. The corresponding code is available on GitHub[2] along with a step-by-step guide to reproduce our approach and the comparative approaches mentioned in the results section.

7 CONCLUSION AND FUTURE WORK

In this paper, we have investigated the role of AI-powered intrusion detection in enhancing the accuracy and efficiency of detecting cyber threats. We have benchmarked our results against state-of-the-art (SOTA) models using four shallow ML classifiers. Our approach showcased the advantages and limitations of generating synthetic data. Our main findings are summarized as follows:

- Our supervised attention-based AAE has outperformed SOTA models in detecting and generating data using real-world data.
- Adapting reinforcement learning to address class imbalance improved recall performance by 17%, minimizing the likelihood of false negatives.
- Despite promising results, challenges such as mode collapse and improving classification prediction remain. Addressing these issues is crucial for deploying generative AI-based intrusion detection in real-world environments.

For future work, we will focus on updating the input dataset with more contemporary examples and automating the process of identifying minority classes.

While AI-driven network intrusion detection holds the potential to transform intrusion detection systems, ongoing advancements and thorough evaluations are essential to ensure its resilience against the evolving landscape of cyber threats.

ACKNOWLEDGMENT

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC), the Mathematics of Information Technology and Complex Systems (MITACS) and the Desjardins Group (Mouvement Desjardins) for their financial support.

[2]https://github.com/anonymousForStudy/AAE-DRL

REFERENCES

Abbasian, M., Rajabzadeh, T., Moradipari, A., Aqajari, S. A. H., Lu, H., and Rahmani, A. (2023). Controlling the latent space of gans through reinforcement learning: A case study on task-based image-to-image translation.

Alabsi, B. A., Anbar, M., and Rihan, S. D. A. (2023). Conditional tabular generative adversarial based intrusion detection system for detecting ddos and dos attacks on the internet of things networks. *Sensors*, 23:1–20.

Chiriac, B.-N., Anton, F.-D., Ioniță, A.-D., and Vasilică, B.-V. (2025). A modular ai-driven intrusion detection system for network traffic monitoring in industry 4.0, using nvidia morpheus and generative adversarial networks. *Sensors*, 25(1):1–23.

Fuhl, W., Bozkir, E., and Kasneci, E. (2020). Reinforcement learning for the privacy preservation and manipulation of eye tracking data.

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods.

Gaber, M. G., Ahmed, M., and Janicke, H. (2024). Malware detection with artificial intelligence: A systematic literature review. *ACM Comput. Surv.*, 56(6).

H. M. Kotb, T. Gaber, S. A. e. a. (2025). A novel deep synthesis-based insider intrusion detection (ds-iid) model for malicious insiders and ai-generated threats. *Nature*, 15(207).

Lansky, J., Ali, S., Mohammadi, M., Majeed, M., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., and Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: A systematic review. *IEEE Access*, 9:101574–101599.

M. Ali, I. Udoidiok, F. L. and Zhang, J. (2024). A review on generative intelligence in deep learning based network intrusion detection. *Cyber Awareness and Research Symposium (CARS)*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders.

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)*. IEEE.

Sayed, M. A. and Taha, M. (2023). Oblivious network intrusion detection systems. *Nature*, 13(22308).

van Hasselt, H., Guez, A., and Silver, D. (2015). Deep reinforcement learning with double q-learning.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Zhao, S., Li, J., Wang, J., Zhang, Z., Zhu, L., and Zhang, Y. (2021). attackgan: Adversarial attack against black-box ids using generative adversarial networks. 2020 International Conference on Identification, Information and Knowledge in the Internet of Things.