Enhancing Adversarial Robustness of Anomaly Detection-Based IDS in OT Environments

Andreas Flatscher [*], Branka Stojanović [*]
[*] JOANNEUM RESEARCH Forschungsgesellschaft mbH,
DIGITAL – Institute for Digital Technologies,
Steyrergasse 17, 8010, Graz, Austria
Email: andreas.flatscher@joanneum.at,
branka.stojanovic@joanneum.at

Ozan Özdenizci [†]
[†] Montanuniversität Leoben,
Chair of Cyber-Physical-Systems,
Franz-Josef-Straße 18, 8700, Leoben, Austria
Email: ozan.oezdenizci@unileoben.ac.at,

Abstract—**The increasing use of deep learning approaches, particularly generative models such as autoencoders (AEs), as Intrusion Detection Systems (IDS) in cybersecurity, introduces vulnerabilities to adversarial attacks. These attacks involve small, malicious perturbations to input data that can deceive the system, disguising attacks as normal behavior. In this paper, we investigate the susceptibility of an AE-based IDS deployed in an Operational Technology (OT) environment, specifically a water distribution system. We explore various defense strategies to enhance model robustness against adversarial attacks, focusing on increasing the minimal perturbation required to evade detection. Our study examines both adversarial training and sensitivity-based training, comparing their effectiveness in hardening the system against adversarial attacks with different number of features available to the attacker (100%, 75%, 50%, 25%, 2%). Results show that while both methods have improved the robustness of the model architecture for some scenarios, no method shows clear improvement on all experiments. This work highlights the importance of adversarial robustness in critical infrastructure protection and provides insights into defense mechanisms for enhancing the security of AE-based IDS systems.**

Index Terms—**Intrusion Detection System, Anomaly Detection, Adversarial Attacks, Autoencoders, Adversarial Robustness, Critical Infrastructure Protection.**

I. Introduction

With the increasing use of generative models and deep representation learning based approaches for intrusion detection systems (IDS) [1], a natural security flaw that arises and becomes increasingly relevant are adversarial attacks on said systems [2]. Adversarial attacks are small perturbations added to the input, which are maliciously crafted to fool the model in the system. In the case of intrusion detection systems, this typically means that an attack gets disguised and misclassified as normal behaviour. Generative models, like autoencoders (AEs) or Generative Adversarial Networks (GANs), work very well for detecting anomalies and have been used more frequently in the past years [3]. These deep learning methods are however inherently vulnerable to adversarial attacks due to their oftentimes very complex function parameters and high non-linearity [4]. Crafting adversarial samples that fool a system while still being similar to benign input samples has been investigated thoroughly and is a rather easy task for an attacker, especially when the model architecture and weight parameters are known.

In this paper, we investigate the effectiveness of adversarial attacks on an AE-based IDS, which has proven effective in detecting cyber attacks in an Operational Technology (OT) environment: a water distribution system (WDS). The main contribution of this paper is the examination of various defense strategies, which enhance the models robustness, thereby increasing the minimal modification needed to generate an adversarial sample. The findings indicate that employing these more robust training methods increases the so-called hardness of evasion, a measure of how easily an attack can be disguised. Furthermore, different amounts of features available to the adversary are considered in empirical analysis, since this aligns more closely with practical attack scenarios.

II. Related Work

Adversarial attacks on IDS pose a significant, realistic security challenge, as they involve malicious actors crafting inputs designed to deceive the IDS into misclassifying or failing to detect intrusions. By subtly altering the attack patterns or data, adversaries exploit weaknesses in the detection algorithms, often causing the IDS to overlook real threats or falsely flag benign activities as malicious [5]. This undermines the reliability of the IDS, potentially leaving systems vulnerable to breaches while reducing confidence in automated threat detection solutions. In particular, the access of only a small branch of an OT environment, for instance a single Programmable Logic Controller, might also be available to an adversary to manipulate a significant part of the system [6], which creates a more challenging situation.

Recent attacks on cyber-physical systems (CPS) have shown that with an increase in usage of sensors, actuators, edge devices in automation in smart cities, there is also an increase in vulnerabilities of such systems [7]. One signifcant incident in the domain of water treatment facilities was in 2020, where attackers successfully gained access to a water treatment plant's computer system and tried to increase sodium hydroxide levels of the water to a dangerous level [8]. Only through the attentiveness of a plant operator was the attack halted. These attacks are more and more common and the adversarial

978-3-903176-66-9 ©2024 IFIP

robustness of CPS remains relatively underexplored in current research [9].

Our work examines the susceptibility of an autoencoder-based Intrusion Detection System (IDS) to adversarial attacks, focusing on its effectiveness in detecting cyber threats within an OT setting, particularly in a water distribution system.

Experimental part of the work is based on The BATtle of the Attack Detection ALgorithms (BATADAL) dataset, originally created for an intrusion detection system challenge [10]. Since its introduction, it has been used as a benchmark dataset for WDS IDSs, since it is a rather large dataset with different attack scenarios included [11], [12], [13]. It was artificially created using the epanetCPA water distribution modeling toolkit and has proven itself as a benchmark data set for time series anomaly detection [14]. It consists of a one-year simulation without any attacks and a six-month simulation with seven different (partially labeled) attacks, respectively. It also contains a test dataset which is a three month simulation with seven different attacks.

In our previous work we have tested different model architectures to find the best model to detect anomalies in the time series data [15]. The conclusion was that the most promising approach is a simple autoencoder architecture, which has only been trained on benign data without any anomalies. This method is called One-Class Novelty Detection and has been proven to work well for finding anomalies in various datasets. The model learns to extract important features and how to reconstruct the original data sample based on the extracted features. When finally testing the AE on the test data, which included benign data samples as well as anomaly samples, the reconstruction error has been shown to be relatively small for the benign data samples, while it is high for the anomalies. This is due to the fact that the AE is unfamiliar with the underlying statistical distribution of the anomaly data samples and is unable to successfully reconstruct them. While this method has brought state of the art results for time series anomaly detection with this dataset, as with all current deep learning architectures, the question arises if this is a robust approach against adversarial attacks. For IDSs, adversarial attacks are most often in the form of evasion attacks, where the attacker wants to stay unnoticed by disguising their attack as benign behaviour. This is made possible by altering the input data of the model.

Adversarial attacks are small perturbations added to the input features that are maliciously crafted to fool the underlying system [16]. Much research on how to craft said perturbations and how effective the attacks are on modern machine learning approaches has been done in the past years [17], [4], [9], [18]. However, to the best of our knowledge, this is still an ongoing research topic and is very much relevant for many different deep learning applications, especially in the cyber-security domain. Adversarial attacks can be categorized by white-box or black-box attacks as well as untargeted or class-targeted attacks, depending on the model access capabilities and adversarial purpose of the attacker [19].

In this paper, we focus on the white-box attack scenario, since it results in better crafted adversarial samples, which lead to more successful attacks on the system. In the white-box scenario all model parameters as well as the models architecture are assumed to be known to the attacker. This information is useful for calculating the gradient of the model with respect to the input data. State-of-the-art white-box adversarial attack methods on neural networks, such as Projected Gradient Descent (PGD) [20], use this gradient information to update the input sample by iteratively stepping into the direction of greatest ascent of the loss in every feature. This often results in a perturbed input sample which is very similar to the original sample and barely over the decision boundary resulting in the wrong classification.

It should be noted that black-box attacks, while out of the scope of this paper, are an alternative approach to craft adversarial samples which do not require much knowledge about the models' inner architecture or parameters. Instead, the model output can be obtained for a given input. With a suitable number of queries (which varies depending on type of model, number of parameters, etc.), the classification boundaries can be approximated and adversarial samples which lie on these boundaries can be found as well [21].

III. Methodology

A. One-Class Novelty Detector

One-Class Novelty Detection is a machine learning technique used to identify outliers or novel data points by training a model on a single class of normal data and detecting deviations from this normal class. It is an important task in many different domains like computer vision, cybersecurity, finance, healthcare, industry and production, etc. [22]. Traditional machine learning approaches like One-Class Support Vector Machines (SVM) have recently been outdated by deep learning approaches in terms of performance. In particular, generative models like GANs and AEs have gained popularity in this task since they have many benefits such as nonlinearity, scalability and robustness to noise. Training of the generative models is relatively straightforward: The training data consists of exclusively normal data (in-distribution data), and the models are trained to reconstruct the original input data sample by first reducing and then expanding the feature space again. For inference, the reconstruction error of the data samples are looked at as a score of how likely the input sample is an outlier. Since the model has not seen the outlier data before and is not able to reconstruct it as well as the normal data, the reconstruction error is higher for these data samples.

Autoencoder Architecture. The One-Class Novelty Detector which has achieved state of the art results for the BATADAL dataset is a simple autoencoder architecture [15]. It consists of 4 layers in the encoder and decoder each, which compress input data into a lower-dimensional latent representation and reconstruct the original data from this compressed form again. We used tanh activation functions interleaved between layers. The latent space representations between the encoder and the decoder blocks consisted 18 features. Model parameters were optimized with the Adamax algorithm. In the original paper that introduced this architecture [15], a fixed number of epochs of 100 was used. Since we test different training strategies, the fixed number of epochs hinders the different models abilities to correctly fit to the training data distribution. We introduce an early stopping mechanism that stops training once the validation loss has not improved for 10 epochs, to have an objective comparison of all training strategies. We set the threshold by calculating the optimal F1-score.

B. Adversarial Attacks

Adversarial attacks are a common vulnerability of machine learning methods, especially for deep neural networks. Adversarial samples are data samples that lie at the decision boundary of the model and are therefore classified into the wrong class [16], [23]. Adversarial attacks work best for models that have a decision boundary which differs from the true decision boundary. While all machine learning methods principally have faulty decision boundaries to some degree, highly nonlinear models, such as deep neural networks, oftentimes have more ambiguous decision boundaries and are therefore more vulnerable to adversarial attacks than, for example, more traditional machine learning techniques.

As previously discussed, adversarial attacks can be broadly classified into white-box and black box attack settings. In the white-box setting, it is assumed that the attacker has full knowledge of the model parameters and the specific architecture. This is useful information since the gradient of the models loss function with respect to the input sample can easily be calculated and maximized, until the perturbed sample gets classified wrongly.

1) Projected Gradient Descent: A state-of-the-art white-box adversarial attack algorithm proposed by Madry et al. [20] is Projected Gradient Descent (PGD). The gradient of the loss function with respect to the input sample $\boldsymbol{x}$ gets calculated. The gradient is scaled by a small α and added to the input sample. A projection is done on this perturbed sample, to ensure some constraints given by the task. This is usually an epsilon magnitude clipping to ensure a small l_∞ norm or, as in our case, a projection on the feasible set of features, which can be changed. The exact formula for the iterative update step can be defined as:

$$\tilde{x}_{k+1} = \Pi_{\mathcal{B}_\infty}(\tilde{x}_k + \alpha \cdot \operatorname{sign}(\nabla_{\tilde{x}_k} L(\theta, \tilde{x}_k, y))), \qquad (1)$$

following the notation from [24]. This update rule is done iteratively for a fixed number of steps or until a condition is met. PGD can be seen as an extension of the Fast Gradient Sign Method (FGSM) which was the first adversarial attack [23]. PGD is in principle an iterative version of FGSM.

C. Defenses Against Adversarial Attacks

1) Adversarial Training: The classical approach to increase robustness, especially for defending against adversarial attacks, is the so-called adversarial training [20]. For classification tasks, adversarial training works by generating and adding adversarial samples to the training dataset, with their respective original class. These samples are generated for example using the PGD algorithm explained in Section III-B1 since it is, at the time of writing, one of the most effective ways to generate adversarial samples [25]. At every epoch a certain percentage of data samples get transformed to an adversarial sample, while their target class stays the same. This guides the model to learn to filter out and ignore the adversarial perturbations and is more resilient against these attacks, meaning to generate an adversarial sample, a bigger perturbation needs to be applied to the sample.

However, for One-Class Novelty Detectors adversarial training functions somewhat differently, since we only train the autoencoder in an unsupervised manner and our training data consists purely of benign data samples without any target classes. We therefore use the adversarial loss:

$$\begin{aligned}\mathcal{L}_{AE} &= \|\boldsymbol{X} - \operatorname{Dec}(\operatorname{Enc}(\boldsymbol{X}^*))\|_2^2 \\ &\quad + \lambda \|\operatorname{Enc}(\boldsymbol{X}^*) - \operatorname{Enc}(\boldsymbol{X})\|_2^2,\end{aligned} \qquad (2)$$

where $\boldsymbol{X}$ is the original data sample and $\boldsymbol{X}^*$ is the adversarial sample generated from $\boldsymbol{X}$. The first part of Eq. (2) describes the L2-norm of the original data sample and the reconstruction of the adversarial sample, which was generated live using PGD, meaning for every sample, the adversarial sample is generated for the current model and then used to calculate the loss. The second part is weighed with a λ and describes the difference in the latent representation of the original data sample and the adversarial sample. This term is necessary to filter out the adversarial noise in the encoder. We use $\lambda = 0.091$ and $\alpha = 7 \cdot 10^{-7}$, which were the ideal hyperparameters optimized by Optuna Parameter search algorithm.

The model is expected to filter out any adversarial perturbations. Salehi et. al. have shown that this type of adversarial robust training can increase the so-called hardness of evasion, which is a metric that measures the size of the perturbation needed for a misclassification [26]. In our case, it is calculated via the average L2-norm of the perturbations needed for an attack to stay unnoticed by the system.

2) Sensitivity Based Training: Since the adversarial training requires a lot of computational resources, we have investigated another method that is supposed to increase robustness of the anomaly detector, while still being relatively cheap computationally [27]. This method also uses a custom training loss that is supposed to help the model in learning to filter out the adversarial noise. This sensitivity based loss is described by:

$$\begin{aligned}\mathcal{L}_{sen} &= \|\boldsymbol{X} - \operatorname{Dec}(\operatorname{Enc}(\boldsymbol{X}))\|_2 \\ &\quad + \gamma \|\operatorname{Enc}(\boldsymbol{X}) - (\operatorname{Enc}(\boldsymbol{X} + \Delta h))\|_2,\end{aligned} \qquad (3)$$

where Δh is a vector with the same length as $\boldsymbol{X}$ and every entry has been randomly generated from a Gaussian distribution with $\mu = 0$ and $\sigma = q$. For our experiments, we have found the best results using Optuna hyperparameter search with $q = 1.5 \cdot 10^{-5}$ and $\gamma = 1.0$.

3) Hardness of Evasion: Hardness of Evasion is defined as the average Euclidean distance between the original data sample and the adversarial data sample, which has the smallest perturbation needed to be wrongly classified. In our case this

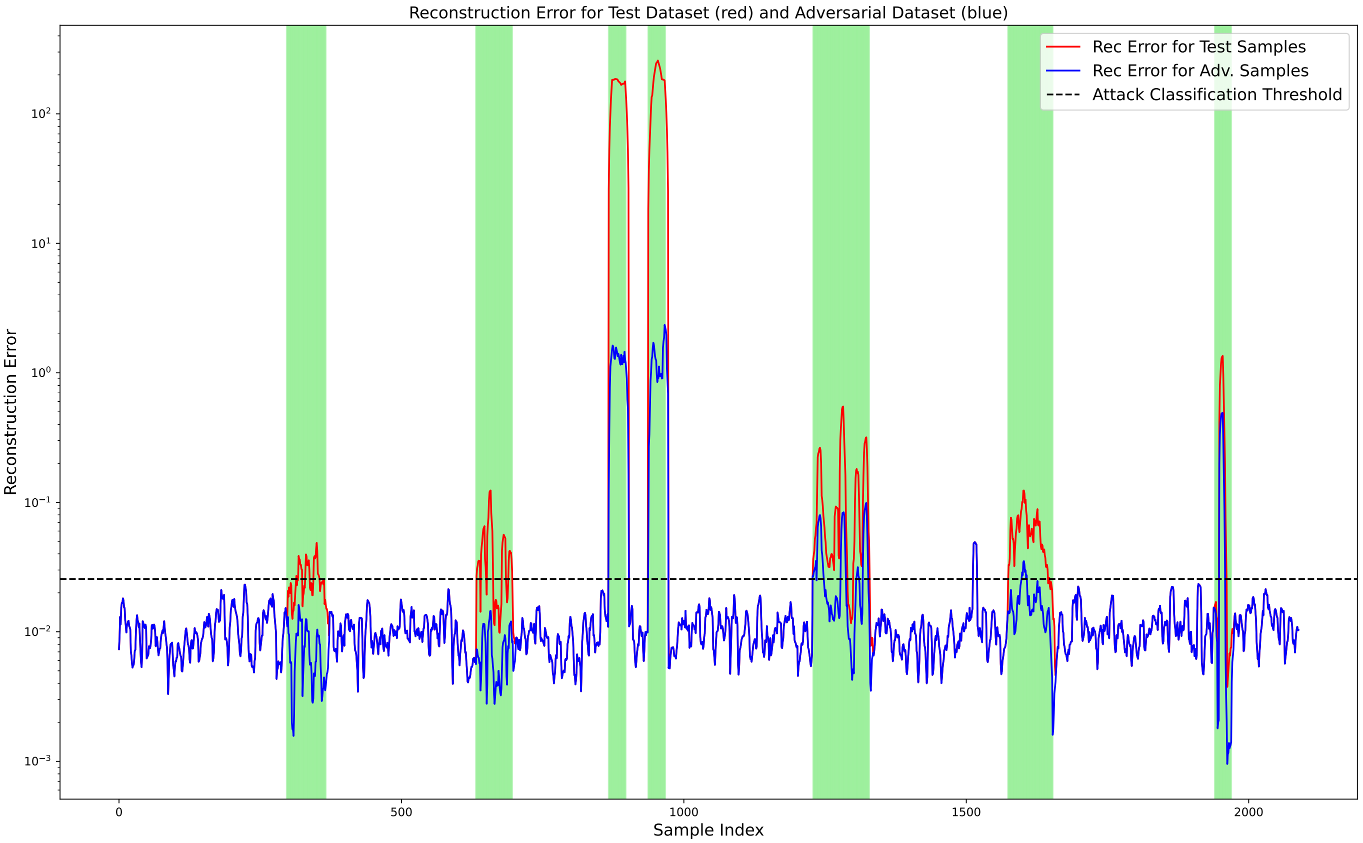

Fig. 1. Comparison of *original test dataset* and the *adversarially generated test dataset* based on the original autoencoder architecture. Green Area marks the attack samples

means a reconstruction error below the threshold and therefore a classification as a benign data sample. We calculated this metric using the PGD algorithm with a very small $\alpha = 10^{-7}$. This is a useful measure for comparing the robustness of different training strategies. The higher the hardness of evasion, the more robust the model is. For our experiments, we have normalised this metric, meaning that we divided the score by the number of features available to the attacker. This was done to have a better comparison between the different attacks.

IV. Experiments

We aim to build upon an existing One-Class Novelty Detection Architecture [15] to evaluate the current robustness of the system and find methods to increase it. We therefore train and compare three models with the same architecture, but different training loss functions: original model with L2-norm as loss, adversarially trained model with the adversarial loss in Eq. (2) and sensitivity trained model with the sensitivity loss in Eq. (3). We keep most hyperparameters of the original architecture the same, to be able to have a fair comparison and not skew the results by introducing more unknown variables. Parameters that we have not changed include number of layers, number of neurons, training- and testing data split, pre- and postprocessing of the data. While the original architecture trains the model for a fixed number of epochs, we had to introduce an early stopping condition since we have changed the loss functions for two experiments and this changes the training drastically. The early stopping condition stops the training when there has been no significant decrease of the validation loss in over 10 epochs. It should be noted that while we use different losses for adversarial- and sensitivity based training, to measure performance on the validation- and test dataset, we use the mean squared error between the reconstructed and original samples.

We also tested the scenario where different numbers of features were allowed to be perturbed by the adversary. This is a more realistic setting, since an attacker is more likely to be able to only change a percentage of features in a cyber-physical system as opposed to all the features. This makes the task drastically more challenging. We examine approximately 100%, 75%, 50% and 25% of the features available for perturbation, as well as only a single feature. We choose the features that have the most effect on the classification, at the time of the first iteration in the PGD algorithm for every sample to cover the worst case scenario.

For testing the robustness of the system, we wanted to generate adversarial data for the attacks in the testing data. In theory, a successful generation of these adversarial samples should decrease the reconstruction error enough to be classified as a benign class. We have used the PGD algorithm

TABLE I
Hardness of evasion scores and percentages of failed attacks, for different numbers of features available to the adversary.

	Hardness of Evasion (normalized)			Number of failed attacks (%)		
# features	*original*	*adversarially trained*	*sensitivity trained*	*original*	*adversarially trained*	*sensitivity trained*
51 (100%)	0.01229	**0.02154**	0.00871	**22.7**	18.3	22.0
37 (75%)	**0.01138**	0.00820	0.00969	**24.7**	18.8	21.5
25 (50%)	0.00688	**0.01069**	0.01038	**26.7**	22.5	25.7
13 (25%)	0.00180	**0.02100**	0.00220	42.1	**44.3**	39.1
1 (2%)	4.7572e-09	2.2749e-14	**5.1661e-08**	60.9	58.2	**61.3**

(explained in detail in Section III-B1) for all attack samples in the testing set, which were 407 out of 2089 samples. It is important to note that while the PGD algorithm tries to increase the loss with respect to the input sample, in our case we want to decrease the reconstruction error to stay unnoticed by the underlying system. This means that instead of going in the direction of the gradient with the update rule, we go into the opposite direction by multiplying the gradient by -1. Other than this simple change, the algorithm stays the same.

Since we want to look at the minimum perturbation needed for the model to misclassify, we have ignored the epsilon clipping which normally guarantees a small l_∞-norm. Through Optuna hyperparameter search library [28], we have achieved the best results with an α of 10e-4 and 10 000 iterations.

In this paper we focus on concealing attacks rather than attempting to manipulate benign data to resemble an attack. Both scenarios are a valid research topic but in the domain of cybersecurity, the former case is more threatening to the integrity of the system than the latter.

We first looked at the original AE architecture and how vulnerable it is to adverarial attacks. In Figure 1 we can see the reconstruction errors of the original (unperturbed) testing dataset (red curve) as well as reconstruction errors for the adversarial testing dataset (blue curve), where PGD was used on the attack samples (marked as green area in the plot). We can see that for most attacks, the reconstruction error could be reduced enough to be classified as benign by the original system. There are some exceptions where the original reconstruction error was too high to begin with and the PGD algorithm converged before the rec. error was reduced enough. However, overall it is very possible to disguise most attacks as benign behaviour, which proves as a serious vulnerability.

V. Results and Discussion

In order to compare the different training methodologies, we have displayed the hardness of evasion metric (explained in Section III-C3) as well as the number of failed attacks (i.e. the percentage of attack samples that could not be disguised as benign data for the given system architecture) in Table I. These metrics were calculated for different amounts of features available to the attacker, since in a real world scenario, this is more practical. We can see that the adversarially trained model clearly outperforms the original architecture, when 100% (51) of features were available, since the score is almost double of the original. This is the case for all attacks except the one with 75% and 2% of available features. The sensitivtiy trained model has better robustness than the original architecture only when less or equal to 50% of features were used.

The adversarially trained model seems to be the best architecture regarding the hardness of evasion metric, since it is the highest across the board. However, while the average hardness of evasion increased by a lot, the number of failed attacks is the lowest for all architectures for 4 out of the 5 attack strategies. It seems that some attacks which could not be disguised on the original- and the sensitivity-trained model, could be perturbed enough to be misclassified on the adversarially trained model. Interstingly, the number of failed attacks metric stands in contrast to the hardness of evasion metric, in the sense that for high number of features available (more than 50%), the original model performs best. However, since in practice the attacker is unlikely to have a high number of features avialable, the more relevant case is for low number of features available. For these attack scenarios, the adversarially trained and sensitivity trained model clearly outperform the original.

While the main goal of this paper was to investigate methods to increase robustness against adversarial attacks, we still want to look at the performance of the three models on the original (unperturbed) test dataset. This can be seen in Figure 2 where the reconstruction errors of the normal and unperturbed test data for our three models has been plotted. They mostly look similar, with the main difference being that the sensitivity trained model has a lower threshold for classification. Other than that, the models performances are mostly the same. This is made more clear when we compare the F1-scores for the architectures: the original architecture has an F1-score of 0.668, the adversarially trained model has a score of 0.678 and the sensitivity trained model only 0.656. The worse score for sensitivity trained model can easily be explained by the fact that we add noise to the training data to increase robustness, which makes it harder to train in a general manner. The adversarially trained model actually has an increased F1 score, meaning we actually have better performance for this model than the original. However, these differences in F1 score are miniscule and do not consistute for actual improvement or worsening, but rather for random chance.

VI. Conclusion

In this paper we wanted to investigate the robustness of an established IDS in an OT environment. We looked at the

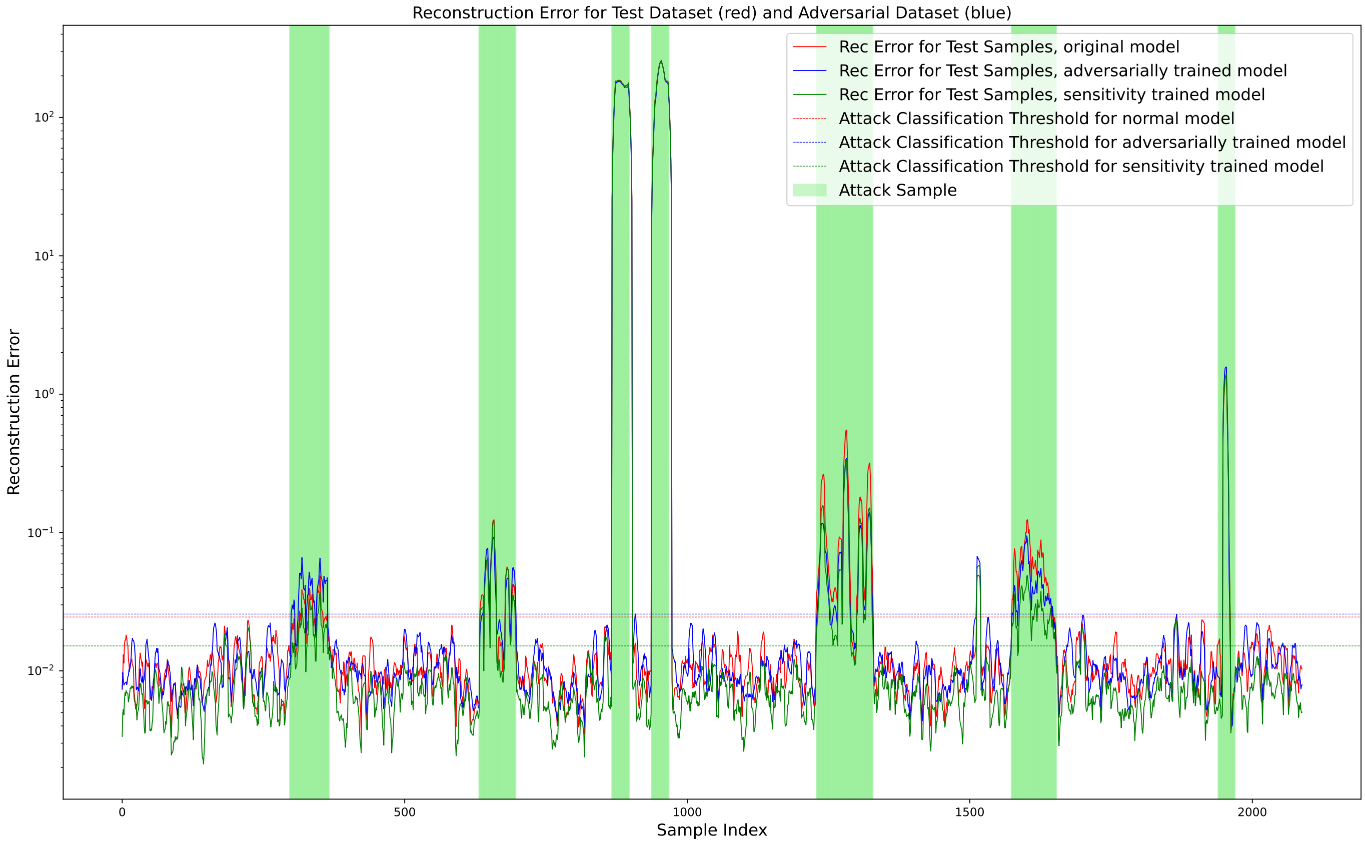

Fig. 2. Comparisons of different model architectures on the unperturbed, original test dataset. Green Area marks the attack samples

vulnerabilities of the original architecture regarding white-box adversarial attacks of varying degrees, and methods to increase the robustness by simply changing the training methology while the model structure stays the same. We have compared two methods which increase the hardness of evasion for adversarial attacks, while the overall performance of the IDS stays fairly similar. However, since the total number of failed attacks has been decreased in 3 out of the 5 attack scenarios, it is not definite that adverarially trained or sensitivity trained models are a safer option in practice. It depends on the kind of resilience that is needed for the practical system. This research underscores the significance of investigating security flaws in modern IDS', which oftentimes utilise modern machine learning techniques which are vulnerable to adversarial attacks.

Further investigation is needed in this domain, particularly for black-box attacks in OT environments, as they are a more realistic setting, where the attacker does not have any knowledge about the inner workings of the model but instead is able to query the model. Ultimately, the topic adversarial robustness of IDS will require more reasearch, especially as the system architectures get more complex.

Acknowledgements

This work was funded by the Austrian Security Research Programme KIRAS of the Federal Ministry of Finance (BMF) under GA No. FO999905309 (project SeRWas) and by the Austrian Research Promotion Agency (FFG) under GA No. FO999899544 (project PRESENT).

References

[1] G. Ketepalli and P. Bulla, "Review on generative deep learning models and datasets for intrusion detection systems," *Revue d'Intelligence Artificielle*, vol. 34, no. 2, 2020.

[2] C. Zhang, X. Costa-Pérez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1294–1311, 2022.

[3] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[4] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *CoRR*, vol. abs/1810.00069, 2018. [Online]. Available: http://arxiv.org/abs/1810.00069

[5] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[6] J. M. Acton, "Cyber warfare & inadvertent escalation," *Daedalus*, vol. 149, no. 2, pp. 133–149, 2020.

[7] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.

[8] Cybersecurity and I. S. A. (CISA), "Compromise of u.s. water treatment facility (aa21-042a)," February 2021, accessed: 2024-08-19. [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/aa21-042a

[9] Z. A. Sheikh, Y. Singh, P. K. Singh, and P. J. S. Gonçalves, "Defending the defender: Adversarial learning based defending strategy for learning based security methods in cyber-physical systems (cps)," *Sensors*,

vol. 23, no. 12, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/12/5459

[10] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks, B. M. Brentan, M. Herrera, A. Rasekh, E. Campbell, I. Montalvo, G. Lima, J. Izquierdo, K. Haddad, N. Gatsis, A. Taha, S. L. Somasundaram, D. Ayala-Cabrera, S. E. Chandy, B. Campbell, P. Biswas, C. S. Lo, D. Manzi, E. Luvizotto, Jr, Z. A. Barker, M. Giacomoni, M. F. K. Pasha, M. E. Shafiee, A. A. Abokifa, M. Housh, B. Kc, and Z. Ohar, "The battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *Journal of Water Resources Planning and Management*, vol. 144, no. 8, p. 04018048, Aug. 2018.

[11] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," *Human-centric Computing and Information Sciences*, vol. 9, pp. 1–22, 2019.

[12] M. Housh, N. Kadosh, and J. Haddad, "Detecting and localizing cyber-physical attacks in water distribution systems without records of labeled attacks," *Sensors*, vol. 22, no. 16, p. 6035, 2022.

[13] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks *et al.*, "Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *Journal of Water Resources Planning and Management*, vol. 144, no. 8, p. 04018048, 2018.

[14] Q. Lin, S. Verwer, R. Kooij, and A. Mathur, "Using datasets from industrial control systems for cyber security research and education," in *Critical Information Infrastructures Security*, S. Nadjm-Tehrani, Ed. Cham: Springer International Publishing, 2020, pp. 122–133.

[15] B. Stojanović, H. Neuschmied, M. Winter, and U. Kleb, "Enhanced anomaly detection for cyber-attack detection in smart water distribution systems," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ser. ARES '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3538969.3543796

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[17] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–39, 2022.

[18] Y. Jia, J. Wang, C. M. Poskitt, S. Chattopadhyay, J. Sun, and Y. Chen, "Adversarial attacks and mitigation for anomaly detectors of cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 34, p. 100452, 2021.

[19] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, 2013, pp. 387–402.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[21] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks." in *CVPR Workshops*, vol. 2, no. 2, 2017.

[22] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou, "A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges," *arXiv preprint arXiv:2110.14051*, 2021.

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[24] O. Özdenizci and R. Legenstein, "Training adversarially robust sparse networks via Bayesian connectivity sampling," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8314–8324.

[25] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International journal of automation and computing*, vol. 17, pp. 151–178, 2020.

[26] M. Salehi, A. Arya, B. Pajoum, M. Otoofi, A. Shaeiri, M. H. Rohban, and H. R. Rabiee, "Arae: Adversarially robust training of autoencoders improves novelty detection," *Neural Networks*, vol. 144, pp. 726–736, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608021003646

[27] P. P. Chan, Z. Lin, X. Hu, E. C. Tsang, and D. S. Yeung, "Sensitivity based robust learning for stacked autoencoder against evasion attack," *Neurocomputing*, vol. 267, pp. 572–580, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217311608

[28] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.