AADMIP: Adversarial Attacks and Defenses Modeling in Industrial Processes

Vitaliy Pozdnyakov[3,4], **Aleksandr Kovalenko**[3,4], **Ilya Makarov**[3,4], **Mikhail Drobyshevskiy**[1,2,4] and **Kirill Lukyanov**[1,2,4]
[1]Ivannikov Institute for System Programming of the Russian Academy of Sciences
[2]Moscow Institute of Physics and Technology (National Research University)
[3]AIRI, Moscow, Russia
[4]ISP RAS Research Center for Trusted Artificial Intelligence
{pozdnyakov, kovalenko, makarov}@airi.net, {drobyshevsky, lukyanov.k}@ispras.ru

Abstract

The development of the smart manufacturing trend includes the integration of Artificial Intelligence technologies into industrial processes. One example of such implementation is deep learning models that diagnose the current state of a technological process. Recent studies have demonstrated that small data perturbations, named adversarial attacks, can significantly affect the correct predictions of such models. This fact is critical in industrial systems, where AI-based decisions can be made to manage physical equipment. In this work, we present a system which can help to evaluate the robustness of technological process diagnosis models to adversarial attacks, as well as consider protection options. We briefly review the system's modules and describe useful applications. Our demo video is available at: http://tinyurl.com/3by9zcj5.

1 Introduction

Correct Fault Detection and Diagnosis (FDD) allows to increase the efficiency and safety of enterprise production processes. Since the advent of Programmable Logic Controllers (PLC), this problem is usually solved at the hardware level by creating simple logical rules. However, this approach did not allow analyzing the states of technological processes characterized by complex behavior patterns. Nowadays, this issue can be solved by smart manufacturing technologies such as Artificial Intelligence (AI). AI and Deep Neural Networks (DNN) allows to analyze sensor data, increasing the efficiency in FDD task in supervised and unsupervised setting [Lomov *et al.*, 2021; Golyadkin *et al.*, 2023]. However, there are some restrictions on the widespread implementation of such systems for industrial process management. One of the restrictions is the vulnerability of DNN to adversarial attacks (Fig. 1). There is a potential threat in which an attacker will gain access to the data exchange system, and slightly changing the data, e.g., even by using domain knowledge [Ganeeva *et al.*, 2024], will make the DNN predictions incorrect [Pozdnyakov *et al.*, 2024]. Such situation is unsafe if DNN is involved in managing industrial equipment.

From a mathematical point of view, data from sensors is a multivariate time series consisting of observations $x_1, ..., x_n$,

Figure 1: Scenario of an attack on a large industrial process.

where $x_t \in \mathbb{R}^d$ are sensor values at time t and d is the number of sensors. Each timestamp is labeled by a vector $y_t \in \{0,1\}^{m+1}$ which indicates either the normal or one of the m faulty states of the technological process. To predict the state of a process at time t, a certain k-size time interval is usually used. It consists of $x_{t-k+1}, ..., x_t$ observations that form a matrix $X_t \in \mathbb{R}^{d\times k}$. A machine learning model can be represented as a function $f : \mathbb{R}^{d\times k} \to \{0,1\}^{m+1}$ that predicts the state by a given observation matrix.

An adversarial attack is a minimal malicious modification of the input data in order to violate the correct prediction of the model. During the attack, an adversarial sample $X_t^{'}$ is created such that: $f(X_t^{'}) \neq f(X_t)$, where $X_t^{'} = X_t + \mathcal{N}$ and $\mathcal{N} \in R^{d\times k}$ is a perturbation matrix. To make changes in the data invisible to the human eye and other detection systems (Fig. 2), its maximum shift can be constrained by the ϵ parameter: $\|X_t - X_t^{'}\|_\infty \leq \epsilon$. There are two types of adversarial attacks: white-box and black-box. White-box attacks require full access to a machine learning model to create a perturbation matrix $\mathcal{N}$ using auxiliary information, such as gradients of the backpropagation algorithm used in neural networks. In contrast, black-box attacks use only the inputs and outputs of a machine learning model. Potentially, white-box attacks are more dangerous than black-box attacks.

In this paper, we present the Adversarial Attacks and Defenses Modeling in Industrial Processes (AADMIP) system, which allows to evaluate the robustness of diagnosis methods against various types of adversarial attacks by applying different defense techniques. The system consists of 5 modules. Each module is implemented in Python language using popular packages for operating with data and machine learning, such as CatBoost [Prokhorenkova *et al.*, 2018], PyTorch [Paszke *et al.*, 2017]. In Section 2 we briefly review the modules, in Section 3 we consider some useful applications.

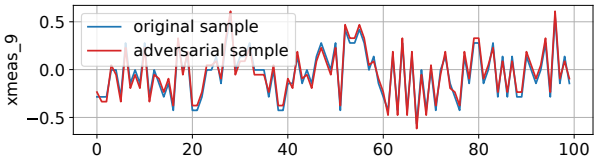

Figure 2: Difference between an original (blue) and an attacked (red) sample of sensor data. The original sample is diagnosed as the normal state of the process, the attacked sample is diagnosed as Fault #16 (Tennessee Eastman Process dataset, sensor xmeas_9).

2 System Overview

2.1 Dataset Module

Dataset Module allows to connect datasets for training and testing diagnosis methods. Each dataset is represented by a set of source files. `df.csv` contains the sequence of sensor values at each point in time. `label.csv` contains the state number of the industrial process, where 0 is the normal state. `train_mask.csv` contains the mask of the training set, where 0 is not a training sample, 1 is a training sample.

Dataset Module allows downloading data from the internet via a public link, unpacking and loading it into memory for later processing. Currently, there are 3 public benchmarks available in the system. `rieth_tep` is a dataset of Tennessee Eastman Process based on the dataset [Rieth *et al.*, 2017]. It contains 52 sensors, 21 states including 20 faults, 21000 runs, each run length is from 500 to 960 observations. `reinartz_tep` is a dataset of Tennessee Eastman Process based on the paper [Reinartz *et al.*, 2021]. It contains 52 sensors, 29 states including 28 faults, 2800 runs, length of each run is 2000 observations. `lessmeier_bearing` is a dataset of an electromechanical drive system based on the paper [Lessmeier *et al.*, 2016]. It contains a single vibration sensor, 3 states including 2 faults, 220 runs, length of each run is 256000 observations. Custom datasets can be easily added using the class inheritance.

2.2 Fault Diagnosis Module

Fault Diagnosis Module allows to manage the configuration of fault diagnosis methods, such as the size of the sliding window and model training parameters: batch size, learning rate, number of training epochs. During training of a machine learning model, data is supplied from *Dataset Module*. The data available for training is determined using a train mask. After training a model, testing on a test set and prediction by an incoming sample of sensor data are available by Application Programming Interface (API).

At this time, there are 5 models available in the module: Linear model [Pandya *et al.*, 2014], Gradient Boosting [Prokhorenkova *et al.*, 2018], Multi-Layer Perceptron (MLP) [Khoualdia *et al.*, 2021], Temporal Convolutional Network (TCN) and Gated Recurrent Unit (GRU) [Lomov *et al.*, 2021].

2.3 Adversarial Attack Module

Adversarial Attack Module allows to manage an adversarial attack parameters and perturb the input data to trick a diagnosis method. The module consists of white-box and black-box attacks. Each attack requires setting the value of the maximum acceptable deviation of the perturbed data from the input. The attacks use predictions of a diagnostic method to attack the data.

Available attacks are: Random Noise [Zhuo *et al.*, 2022] (black-box), Fast Gradient Sign Method (FGSM) [Goodfellow *et al.*, 2014] (white-box), Projected Gradient Descent (PGD) [Madry *et al.*, 2017] (white-box), DeepFool [Moosavi-Dezfooli *et al.*, 2016] (white-box), Carlini and Wagner (C&W) [Carlini and Wagner, 2017] (white-box), Distillation Black Box [Cui *et al.*, 2020] (black-box).

2.4 Defense Module

Defense Module allows to manage parameters of defense methods. During defense, a defense method encapsulates the diagnosis method and can be used for testing and prediction in the same way as the diagnosis method. In this way, a defense method becomes indistinguishable from a diagnosis method from attacker's point of view, allowing defense to be performed invisibly.

Available defense methods are: Adversarial Training [Goodfellow *et al.*, 2014], Defensive Distillation [Papernot *et al.*, 2016], Data Quantization [Xu *et al.*, 2017], Gradient Regularization [Finlay and Oberman, 2021].

2.5 Visualization Module

Visualization Module allows to run an interactive dashboard based on Bokeh [Jolly, 2018] to analyze the performance of adversarial attacks and defenses. To run the dashboard, a set of prepared files is used, where each file represents a combination of a selected dataset, diagnostic method, adversarial attack, and defense method. Each file contains an attacked data and the predicted process state in CSV format.

Currently all combinations are supported for the `rieth_tep` dataset; diagnostic methods: Linear model, MLP, GRU; adversarial attacks: Noise, FGSM, C&W; protection methods: Adversarial training, Quantization, Regularization.

3 Use Cases

3.1 Benchmarking Fault Diagnosis Methods

To decide on the effectiveness of a particular defense method, it is necessary to compare their quality under different adversarial attacks for different diagnosis methods. For example, let us consider a diagnosis methods based on MLP, GRU, TCN neural networks. A model trained on the `reinartz_tep` dataset is subjected to C&W attack with different levels of distortion of the original signal. We measure the quality of each diagnosis method at each distortion level and each defense method. The results obtained (Fig. 3) show that Gradient Regularization is suitable for defense of TCN, while it is significantly less effective for other models. In addition, it can be observed that Data Quantization shows the best protection for all models and for almost all distortion levels. The only defense that shows better performance is Adversarial training at distortion levels close to 0.1. From this we can conclude that Adversarial training is best for cases when we expect an attack at 0.1 level, for all other cases Data Quantization is better.

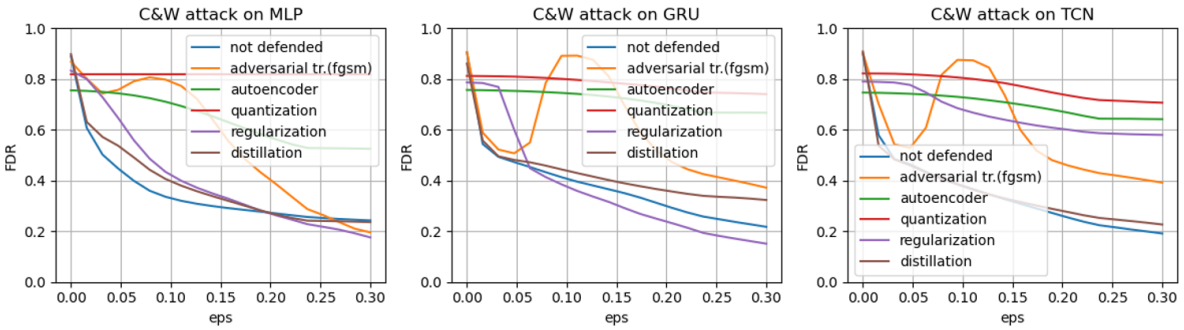

Figure 3: Results of benchmarking C&W attack with different distortion level of attacking MLP, GRU, TCN.

3.2 Interactive Dashboard

After preparing data for visual analysis, we can run an interactive dashboard to analyze the results. On the top left there are switches for diagnosis methods, adversarial attacks, and defense methods (Fig. 4). There are also sliders for selecting a unique process run and sensors displayed below. On the top right there is a scheme of the process. In the center we see current sensor values for a sliding window of size 30. On the left are the original values, on the right the attacked values. The data is updated dynamically, simulating real-time equipment monitoring. At the bottom, the real and predicted process state are available for analysis. By switching attacks and protection methods, we can analyze how sensor values and model predictions change.

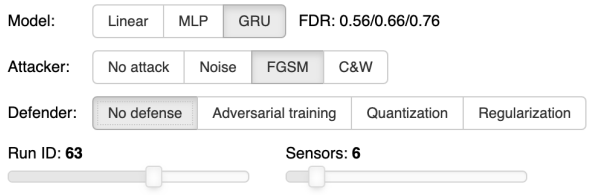

Figure 4: Interactive dashboard.

3.3 Defending Fault Diagnosis Using API

The developed API allows to use diagnosis methods and defense methods available in the system both in isolation and together with each other. Let us consider an example of isolated use of Data Quantization defense method. Suppose there is some diagnosis method that is already used in the process monitoring system. To connect the *Defense Module*, we need to add this diagnosis method as a child class of the base model, implementing the interface of training and prediction. After that, when creating a Data Quantization object, we need to pass a diagnosis method in the initialization parameters. When the object is created, the model is retrained taking into account Data Quantization. After training, the object can be used for prediction using the `predict` method, which takes a numpy [Harris *et al.*, 2020] array of dimension $[B \times L \times D]$, where B is the size of the batch of data, L is the size of the sliding window, D is the number of sensors. An overview of the defending interface is shown in Fig. 5.

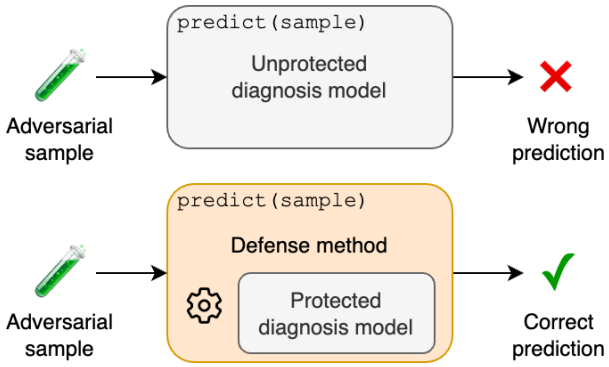

Figure 5: Overview of the defending API. At the top: a wrong prediction received by the `predict` method of an unprotected model. At the bottom: a correct prediction is received from the `predict` method of a defense method that operates on a diagnosis model.

4 Conclusion

When integrating AI into industrial processes, it is necessary to at least estimate the existing potential threats. In this demo we present the AADMIP system that simulates adversarial attacks on sensor data and evaluate defense methods. The system has an open license for freely commercial use and can be integrated into various industrial processes through an API.

Acknowledgements

The work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Ivannikov Institute for System Programming of dated November 2, 2021 No. 70-2021-00142.

References

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[Cui *et al.*, 2020] Weiyu Cui, Xiaorui Li, Jiawei Huang, Wenyi Wang, Shuai Wang, and Jianwen Chen. Substitute

model generation for black-box adversarial attack based on knowledge distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 648–652. IEEE, 2020.

[Finlay and Oberman, 2021] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *Machine Learning with Applications*, 3:100017, 2021.

[Ganeeva *et al.*, 2024] Veronika Ganeeva, Kuzma Khrabrov, Artur Kadurin, Andrey Savchenko, and Elena Tutubalina. Chemical language models have problems with chemistry: A case study on molecule captioning task. In *The Second Tiny Papers Track at ICLR 2024*, 2024.

[Golyadkin *et al.*, 2023] Maksim Golyadkin, Vitaliy Pozdnyakov, Leonid Zhukov, and Ilya Makarov. Sensorscan: Self-supervised learning and deep clustering for fault diagnosis in chemical processes. *Artificial Intelligence*, 324:104012, 2023.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Harris *et al.*, 2020] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[Jolly, 2018] Kevin Jolly. *Hands-on data visualization with Bokeh: Interactive web plotting for Python using Bokeh*. Packt Publishing Ltd, 2018.

[Khoualdia *et al.*, 2021] Tarek Khoualdia, Abdelaziz Lakehal, Zoubir Chelli, Kais Khoualdia, and Karim Nessaib. Optimized multi layer perceptron artificial neural network based fault diagnosis of induction motor using vibration signals. *Diagnostyka*, 22, 2021.

[Lessmeier *et al.*, 2016] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, volume 3, 2016.

[Lomov *et al.*, 2021] Ildar Lomov, Mark Lyubimov, Ilya Makarov, and Leonid E Zhukov. Fault detection in tennessee eastman process with temporal deep learning models. *Journal of Industrial Information Integration*, 23:100216, 2021.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[Pandya *et al.*, 2014] D_H Pandya, Sanjay H Upadhyay, and Suraj Prakash Harsha. Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform. *Soft Computing*, 18:255–266, 2014.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Pozdnyakov *et al.*, 2024] Vitaliy Pozdnyakov, Aleksandr Kovalenko, Ilya Makarov, Mikhail Drobyshevskiy, and Kirill Lukyanov. Adversarial attacks and defenses in automated control systems: A comprehensive benchmark. *arXiv preprint arXiv:2403.13502*, 2024.

[Prokhorenkova *et al.*, 2018] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[Reinartz *et al.*, 2021] Christopher Reinartz, Murat Kulahci, and Ole Ravn. An extended tennessee eastman simulation dataset for fault-detection and decision support systems. *Computers & Chemical Engineering*, 149:107281, 2021.

[Rieth *et al.*, 2017] Cory A Rieth, Ben D Amsel, Randy Tran, and Maia B Cook. Additional tennessee eastman process simulation data for anomaly detection evaluation. *Harvard Dataverse*, 1:2017, 2017.

[Xu *et al.*, 2017] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[Zhuo *et al.*, 2022] Yue Zhuo, Zhenqin Yin, and Zhiqiang Ge. Attack and defense: Adversarial security of data-driven fdc systems. *IEEE Transactions on Industrial Informatics*, 19(1):5–19, 2022.