ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-TUYE4

MACHINE LEARNING FOR ANOMALY DETECTION AND CLASSIFICATION IN PARTICLE ACCELERATORS*

I. Lobach†, M. Borland, K. Harkay, N. Kuklev, A. Sannibale, Y. Sun
Argonne National Laboratory, Advanced Photon Source, Lemont, IL, USA

Abstract

We explore the possibility of using a Machine Learning (ML) algorithm to identify the source of occasional poor performance of the Particle Accumulator Ring (PAR) and the Linac-To-PAR (LTP) transport line, which are parts of the injector complex of the Advanced Photon Source (APS) at Argonne National Lab. The cause of reduced injection or extraction efficiencies may be as simple as one parameter being out of range. Still, it may take an expert considerable time to notice it, whereas a well-trained ML model can point at it instantly. In addition, a machine expert might not be immediately available when a problem occurs. Therefore, we began by focusing on such single-parameter anomalies. The training data were generated by creating controlled perturbations of several parameters of PAR and LTP one-by-one, while continuously logging all available process variables. Then, several ML classifiers were trained to recognize certain signatures in the logged data and link them to the sources of poor machine performance. Possible applications of autoencoders and variational autoencoders for unsupervised anomaly detection and for anomaly clustering were considered as well.

INTRODUCTION

This contribution investigates the possibility to use unsupervised and supervised Machine Learning (ML) methods for anomaly detection and classification in the Particle Accumulator Ring (PAR) and in the Linac-To-PAR (LTP) transport line in the injector complex of the Advanced Photon Source (APS) [1, 2] at Argonne National Lab. We create intentional perturbations in PAR and LTP, which result in poor injection and extraction efficiencies. Then, these data are used for training and testing of various ML models. We chose PAR and LTP for these studies, because a considerable dedicated study time is available in these parts of the APS complex without interruption of user operations.

DATA ACQUISITION

The data presented in this paper were collected during three studies in November, 2021 and during one study on January 30, 2022. Figure 1 illustrates the intentional perturbations of several Process Variables (PVs) on January 30, 2022 and their effect on the charge extracted from PAR. During our studies, the injection cycle rate was 2 Hz. Therefore, the machine state was also logged at 2 Hz as well. Overall, we logged about 9000 PVs related to PAR, LTP, and the linac. However, later we decided to only use up to 155 most relevant PVs as input for ML models. During the intentional perturbations, the beam charge was always kept above zero in order to use the signals from the Beam Position Monitors (BPMs) and to keep various control laws active. Although, most of the data for this paper were collected manually, the process, illustrated in Fig. 1, can be fully automated. We developed a script and tested it with several PVs.

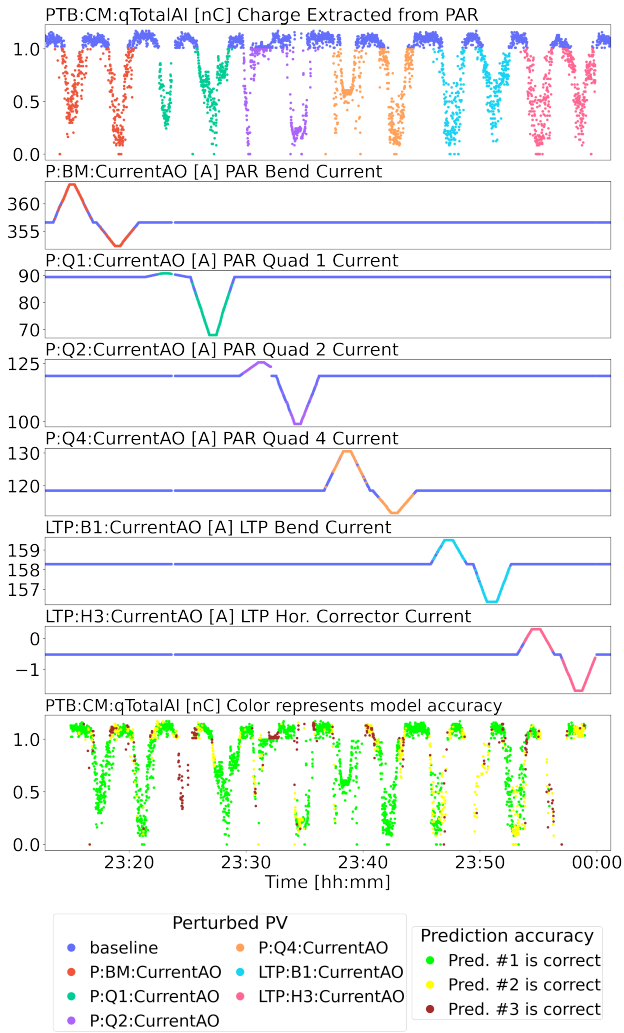

Figure 1: Intentional perturbations of some process variables in PAR and LTP and their effect on the extracted charge. Also, prediction performance of a neural network classifier (bottom panel).

SUPERVISED ML

The machine state snapshots collected at 2 Hz during the intentional perturbations, as in Fig. 1, constitute a labeled

* The work is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.
† ilobach@anl.gov

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-TUYE4

data set. Each poor-performance state can be labeled by the name of the PV that is perturbed at the time. Such data set can be used to train a neural network classifier. The following PVs were included in the input of the neural network: PAR BPM signals and corrector currents, LTP BPM signals and corrector currents, charge in PAR at 10 different timesteps within each injection cycle, charge in LTP and in PTB (PAR-To-Booster), injection and extraction efficiencies, linac trigger timing, PAR kicker timings. The total number of input PVs was 155. The intentionally perturbed PVs (see Fig. 1) were not included in the model input, as it would make the training process trivial. Instead, the goal of this study was to test whether a neural network could learn to recognize and classify certain signatures in the state of the machine, without direct access to the root cause of poor performance. The possible signatures include the reduced charges in different parts of the machine, changed BPM signals, changed charge vs. time within one injection cycle, changed corrector strengths (due to the orbit control law in PAR and due to the trajectory control law in LTP).

The chosen architecture of the neural network was the following (layer sizes): 155 → 80 → Dropout → 80 → Dropout → 40 → 7. Rectified Linear Units (ReLU) were used for activation (except for the last layer, where a SoftMax activation function was used). The Dropout layers' rates were 0.5 [3, 4]. This neural network returns a vector of probabilities for the causes of poor machine performance—six probabilities for the perturbed PVs and one probability, reserved for the baseline state. The Dropout layers were used to avoid overfitting.

One can train the neural network classifier on the data from the beginning of one study, and test it on the data from the end of the same study (a few hours apart). In this case, the prediction accuracy on the test data was above 99 % for each of our study shifts. However, it was important to verify that the model would generalize well on new data, several weeks or months away from the training data, because there are long-term drifts in the machine. The bottom panel of Fig. 1 illustrates the prediction accuracy of the neural network classifier on the data collected on January 30, 2022, while the neural network classifier was trained on the data from November, 2021. Because the classifier returns 7 probabilities for causes of poor performance, on practice, the machine operator can rank them in descending order, and if the best prediction (green) is incorrect, they can check the second-best (yellow), third-best (brown), etc.

To remove outliers in the training data we used Isolation Forest [5]. Before training, the data were scaled by a Robust Scaler [5]. The loss function (cross entropy) was weighted according to the sizes of anomaly classes, because the training data were unbalanced. This is especially important for the baseline class, which is much bigger than others.

UNSUPERVISED ML

One disadvantage of supervised ML is that it requires labeled anomaly data, which is usually scarce. This is why in the previous section we used intentional perturbations to generate it. In this section, we consider an unsupervised ML model, namely, an autoencoder, which can be trained on abundant normal-operation (baseline) data. An autoencoder is a neural network with equal dimensions of input and output layers. It contains one or more hidden layers, and there is always a bottleneck layer in the middle, with a lower size than input and output layers, see Fig. 2. Autoencoders

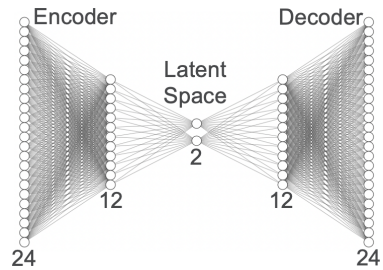

Figure 2: Chosen autoencoder architecture [6].

learn a compressed representation of the training data by minimizing the difference between the input and the output. There are various kinds of loss functions, we will use the Mean Squared Error (MSE). An autoencoder is used for anomaly detection in the following way. First, it is trained on the baseline data, so that it can learn various patterns, typical for the baseline data only. Then, when it encounters an anomalous data sample, it is unlikely to reconstruct it well. Hence, the reconstruction error constitutes an anomaly score. The threshold can be chosen based on the reconstruction error in the training data.

In this section, we consider perturbations in the PAR dipoles and quadrupoles only. We use the horizontal BPM signals and the horizontal corrector currents in PAR (24 PVs) as the autoencoder input. The perturbed PVs are not included in the autoencoder input. The chosen architecture of the autoencoder is presented in Fig. 2. ReLU was used as the activation function for the hidden layers. The autoencoder's ability to detect anomalies was tested on the data with intentional perturbations from one of our study shifts, see Fig. 3. The autoencoder was trained on the baseline data from a 7-day long interval preceding the test data.

Another possible application of autoencoders is anomaly clustering. Indeed, an autoencoder produces a representation of the data in the latent space of smaller dimensionality, where it may be easier to divide the data into distinct groups. This may be useful for preliminary analysis of poor-performance data, namely, to determine if there is a single problem source, or multiple problems happening at different moments in time. To test this idea, 7 days of baseline data were supplemented by one of our study shifts with intentional perturbations. Then, an autoencoder was trained on the combined data set. A 3-dimensional latent space was chosen. The obtained distribution of the training data in the latent space is shown in Fig. 4(a), which demonstrates that a regular autoencoder does not always learn the optimal data representation. The distribution can be sparse and hard to

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-TUYE4

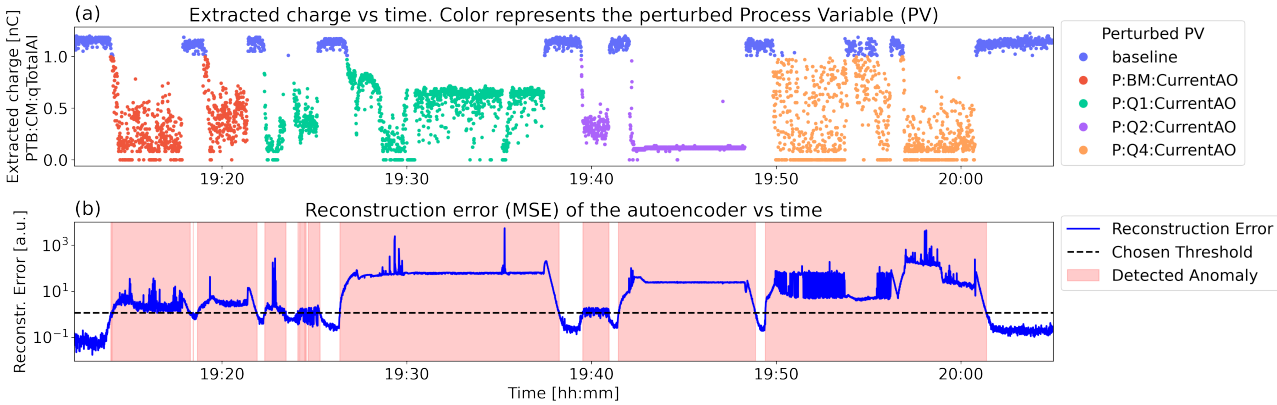

Figure 3: Anomaly detection performance of an autoencoder, which was trained on the normal-operation data from the week, preceding the presented test data. (a) Charge extracted from PAR as a function of time during intentional perturbations (color-coded). (b) Reconstruction error of the autoencoder as a function of time. The anomaly threshold (black dashed horizontal line) was chosen as the 99.9 percentile for the reconstruction error in the baseline data used for training. The red shaded areas represent the detected anomalies, where the reconstruction error exceeds the selected threshold.

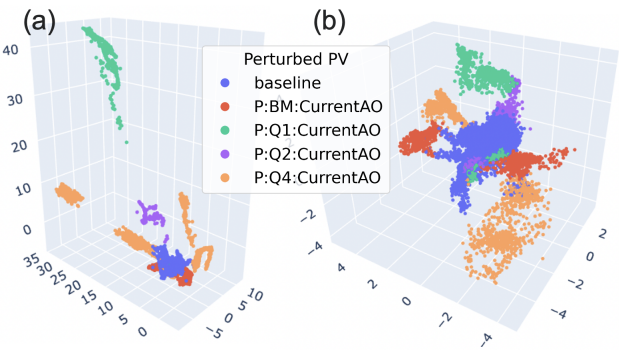

Figure 4: The distribution of the training data in a 3D latent space for (a) an ordinary autoencoder and (b) a β-variational autoencoder.

interpret. An autoencoder may place two "similar" points far apart in the latent space if it minimizes the reconstruction error, because there are no additional constraints on the distribution. Hence, a variational [7] or a β-variational [8] autoencoder may be a better choice, see Fig. 4(b). Variational autoencoders have probabilistic encoders and decoders, which makes the clusters more continuous. Also, they use a zero-mean, unit-variance Gaussian prior for the distribution in the latent space, which constrains it to the desired region. A β-variational autoencoder uses a parameter $\beta > 1$ in the loss function to prioritize the resemblance to the Gaussian prior over the reconstruction precision in order to encourage a disentangled representation [8]. In Fig. 4(b), the baseline cluster is located in the center. Intentional perturbations of each PV are represented by a cluster pair—positive and negative deviations from the baseline value.

DISCUSSION AND CONCLUSIONS

The neural network classifier for the considered anomalies in PAR and LTP is rather accurate and its performance does not degrade significantly with time on a scale of a couple months, see Fig. 1. Even though this was a proof-of-principle experiment, the obtained classifier may be useful in real life, if the current meter for one of the considered magnets (see Fig. 1) becomes faulty and stops reflecting the real magnetic field. In this case, the classifier would point at that magnet as the source of the problem, because the model's decision is based on the signatures created in the BPMs, correctors, etc. One drawback of supervised ML models is that they require labeled data for training. However, it may be possible to develop a script for autonomous collection of the data with intentional perturbations. It can be run after every significant change to the machine state, e.g., after every shutdown.

Autoencoders, trained on normal-operation data, can be effective at detecting anomalies in groups of related PVs, such as the PAR BPM signals and corrector strengths (see Fig. 3), LTP BPM signals and corrector strengths, beam-to-rf phases at different klystrons in the linac, linac injection trigger timing and PAR kicker timings. They can help narrow down the search area for the source of the problem. If during poor machine performance, a larger-than-normal reconstruction error is observed in one of these PV groups, the operator can focus their investigation on that PV group. For example, if the autoencoder for PAR BPM signals and corrector strengths issues an anomaly warning, the problem likely lies in the beam orbit in PAR. Variational and β-variational autoencoders can be used for clustering of anomaly data, see Fig. 4(b). Lastly, the compressed representation of the data in the latent space can be used as input for ML classifiers. This may result in a more robust model, which generalizes better and is less prone to overfitting (not presented in this paper).

ACKNOWLEDGMENTS

We gratefully acknowledge the computing resources provided on Bebop and Swing, high-performance computing clusters operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-TUYE4

REFERENCES

[1] G. Decker, "APS Storage Ring Commissioning and Early Operational Experience," in *Proc. PAC'95*, Dallas, TX, USA, May 1995, pp. 290–292. doi:10.1109/PAC.1995.504639

[2] APS Systems Map, Accessed: 2022-07-26. https://www.aps.anl.gov/About/Overview/APS-Systems-Map

[3] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016. doi:10.48550/ARXIV.1603.04467

[5] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. doi:10.5555/1953048.2078195

[6] A. LeNail, "NN-SVG: Publication-ready neural network architecture schematics," *Journal of Open Source Software*, vol. 4, no. 33, p. 747, 2019. doi:10.21105/joss.00747

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. doi:10.48550/ARXIV.1312.6114

[8] I. Higgins *et al.*, "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017. https://openreview.net/forum?id=Sy2fzU9gl