ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB028

USING MACHINE LEARNING TO IMPROVE DYNAMIC APERTURE ESTIMATES*

F. F. Van der Veken†[1], M. Giovannozzi, E. H. Maclean[1], C. E. Montanari[2], G. Valentino[1]
CERN Beams Department, Geneva, Switzerland
[1]also at University of Malta, Msida, Malta
[2]also at Bologna University, Bologna, Italy

Abstract

The dynamic aperture (DA) is an important concept in the study of nonlinear beam dynamics. Several analytical models used to describe the evolution of DA as a function of time, and to extrapolate to realistic time scales that would not be reachable otherwise due to computational limitations, have been successfully developed. Even though these models have been quite successful in the past, the fitting procedure is rather sensitive to several details. Machine Learning (ML) techniques, which have been around for decades and have matured into powerful tools ever since, carry the potential to address some of these challenges. In this paper, two applications of ML approaches are presented and discussed in detail. Firstly, ML has been used to efficiently detect outliers in the DA computations. Secondly, ML techniques have been applied to improve the fitting procedures of the DA models, thus improving their predictive power.

INTRODUCTION

Machine Learning (ML) is the process of building a data-driven mathematical model to make predictions or decisions without being explicitly programmed [1]. Following the successful use in high-energy physics [2], ML techniques were also introduced in accelerator physics. Beam diagnostics and beam control systems were among the first domains in which ML applications were applied [3, 4], and recently, substantial progress has been made (see [5–12] for a sample of recent applications of ML to accelerator physics topics).

At CERNs Large Hadron Collider (LHC) [13], the inherent complexity in terms of number of hardware systems, amount of data collected and available for on-line or off-line analyses, variety of beam dynamics configurations, such as optical configurations, and beam dynamics phenomena, make it ideal for ML applications (see [14] for an overview of recent results). The focus of this paper is the application of ML to Dynamic Aperture (DA). DA is the extent of the simply-connected region of phase space in which the particle's motion remains bounded over a finite number of turns. Such a volume is shaped by, amongst others, the nonlinear imperfections in the magnetic fields.

As detailed knowledge of the magnetic field errors is difficult to gather, the DA is computed for various realisations of the error distributions. Then, the distribution of DA values needs to be carefully analysed, in particular taking potential outliers into account. ML techniques can be used to automatise the detection of the latter. Another hurdle to overcome in the numerical evaluation of the DA is the huge amount of CPU time required to estimate it accurately for a realistic time scale. Apart from using distributed computing systems [15], a key approach to tackle this issue uses analytical scaling laws based on general theorems of dynamical systems theory [16], which depend on a limited number of parameters (two or three). This allows to use the results of numerical simulations to evaluate the parameters in the scaling laws to extrapolate the DA for a much larger number of turns. ML techniques can be used to obtain robust and reliable parameter estimates in view of using such models for extrapolation purposes, and will be essential for the on-going studies for the HL-LHC Project [17] and the Future Circular Collider (FCC). The models use the following equations to describe the time dependence of the DA:

$$D(N) = \rho_* \left(\frac{\kappa}{2\mathrm{e}}\right)^{\kappa} \frac{1}{\ln^{\kappa} \frac{N}{N_0}}, \tag{1a}$$

$$D(N) = \rho_* \left[-\mathrm{e}\, \mathscr{W}_{-1}\left(-\frac{1}{\mathrm{e}}\left(\frac{N}{N_0}\right)^{-\frac{2}{\kappa}}\right)\right]^{-\kappa}, \tag{1b}$$

where ρ_*, κ, N_0 are the model parameters (see [16] for a derivation of these models and the meaning of the parameters) and $\mathscr{W}_{-1}$ is the negative real branch of the Lambert function. From the original version of the Nekhoroshev theorem [18], there exists an estimate for N_0, which implies we can devise two variants of the model, one with three and one with only two parameters, namely ρ_* and κ.

OUTLIER IDENTIFICATION IN DA SIMULATIONS

For a given angle in the transverse plane, the stable amplitude may differ considerably from seed to seed, creating a spread in estimated DA values. The two variables that are of particular relevance are $\mathrm{DA}_{\mathrm{av}}$ (the DA integrated over angles and averaged over seeds), and $\mathrm{DA}_{\mathrm{min}}$ (the minimum over angles and seeds). While the former is a rather robust quantity, it is clear that the latter might be strongly impacted by existing outliers. These may be due to the excitation of particular resonances as a result of the distribution of nonlinear magnetic errors, which is highly seed-dependent. These low DA values are not necessarily a physical representation of the machine performance, as in reality the operational conditions would be steered away from these. For this reason, it is considered appropriate to remove these outliers from the analysis.

* Work supported by the HL-LHC Project
† frederik.van.der.veken@cern.ch

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB028

Outlier detection is done on an angle-by-angle basis. The set of stable amplitudes $r_{i,j}$ for a given angle j over the different seeds i, is re-scaled to lie in the interval [0,1]. This ensures that the detection is done in a consistent way for the different angles. Several ML approaches were investigated. In the supervised learning approach, the detection is treated as a classification problem, and a Support Vector Machine (SVM) algorithm [19] is used to discriminate between normal and abnormal points. The Radial Basis Function (RBF) kernel [20] with a penalty factor $C = 1$ were identified as the best hyperparameters for the SVM model.

Two unsupervised learning (UL) approaches were also considered: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [21] and the Local Outlier Factor (LOF) [22] method. In both approaches, 75 % of the data set was used for training and 25 % was used for validation. For the DBSCAN method, the maximum distance between two samples for one to be in the neighbourhood of the other is set to 1, and the minimum number of samples in a neighbourhood for a point to be considered a core point, including the point itself, is set to 3. For LOF, the number of neighbours used to measure the local deviation of density of a given sample with respect to the same neighbours is set to 58, and the contamination (the expected fraction of outliers in the data set) is set to 0.001.

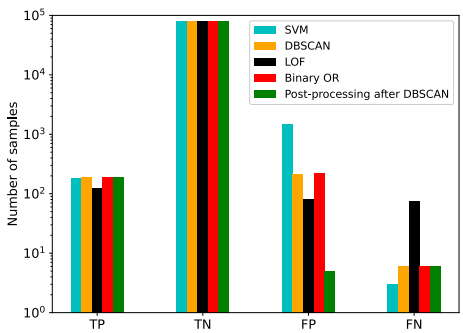

Figure 1: Results from outlier detection using SVM, DBSCAN, LOF, a binary OR between DBSCAN and LOF, and post-processing following DBSCAN methods.

A comparison between the performance of the SVM, DBSCAN, and LOF algorithms trained using the scikit-learn [23] implementation is shown in Fig. 1, listing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The labels determined by the DBSCAN and LOF algorithms were also combined through a binary OR operation to produce a fourth set of labels. While FN are to be avoided as much as possible, as they represent outliers that are missed by the algorithm, too many FP are not desirable either, as they represent regular points that are wrongly flagged as outliers. The latter requires a lot of manual intervention to un-flag these points, defeating the purpose of automatising the outlier detection. As it can be seen in Fig. 1, SVM has the lowest number of FN, but at the expense of an unacceptably high amount of FP (ten times more than the other methods). Even though the other methods still achieve low amounts of FN, it would still be desirable to lower their rate of FP.

For this reason, a fifth set of labels is created following an initial labelling by DBSCAN by removing FP through a statistical method to determine whether this approach would add to the robustness of the original prediction. For a point originally flagged by DBSCAN as an outlier to be considered a true outlier, three additional criteria should be fulfilled: the distance from the mean should be $> 3\,\sigma$ (mean and standard deviation calculated over the points not flagged as outliers); the distance to the nearest non-outlier point should be > 0.15 in absolute units and > 34 % of the total spread of the non-outlier points. See Fig. 2 for an example DA result where the post-processing correctly removes an FP.

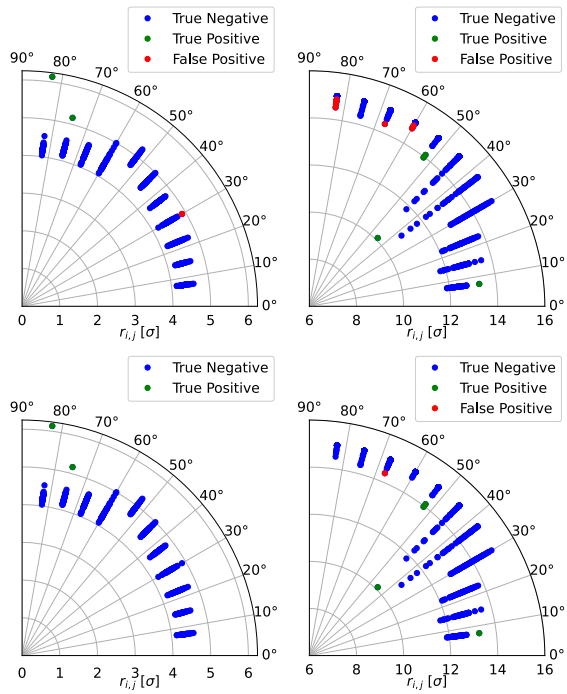

Figure 2: Top row: DA simulations for two LHC configurations (markers represent the 60 seeds). Left: Example with two outliers correctly flagged (green). Right: Example of a false positive (red). Bottom row: idem, but after post-processing, showing the improvement in outlier detection.

FITTING THE DA AS A FUNCTION OF NUMBER OF TURNS

Another domain where ML techniques were applied is the modelling of DA as a function of turn number. The scaling laws introduced earlier would allow for predictions of the DA for $N \gg N_{\text{max}}$ that would be inaccessible to numerical simulations due to the computing time needed (see [16], where the extrapolation error has been considered as a key figure of merit to qualify the DA models).

As ML is not well-suited to extrapolation problems, a different strategy is used, which consists in training a Gaussian Process (GP) [24] on the DA evolution data to generate

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB028

synthetic, though realistic, interpolated points to increase the density of points used to fit the model.

Six DA simulations were performed for the LHC at injection energy up to 10^7 turns to probe the prediction power of the DA model. This is done by setting the value of $N_{\max}$ used to fit the DA model, then using the model to extrapolate the DA up to 10^7 turns, and finally computing the mean squared error (MSE) over the full set of numerical data. All this is repeated, varying $N_{\max}$. The same procedure is applied when the GP is used to improve the quality of the DA model. In this case, 75 synthetic points are uniformly distributed between $10^4 \leq N \leq N_{\max}$ when $N_{\max} = 5 \times 10^5$, and the number of synthetic points is increased up to 750 as $N_{\max}$ increases up to 5×10^6. The synthetic points are disregarded when computing the MSE for the GP-based fit, ensuring a fair comparison between the original and GP-based fits. Note that the GP analysis is repeated 200 times and the minimum MSE over the 200 iterations is used.

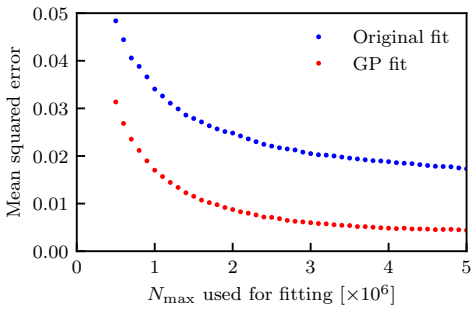

Figure 3: The MSE as a function of $N_{\max}$ of the DA model fit (1b) for the original and the fit including GP-generated synthetic points. Shown here is the fit with two parameters; the results for the three-parameter fit are very similar.

The results of the two-parameter fit are shown in Fig. 3. The MSE for the GP-based fit is always better than that of the original fit. Some saturation in the decrease of the MSE is visible for $N_{\max} \approx 2 - 5 \times 10^6$ for the three-parameter fit, indicating that the simulations performed up until any of the turn numbers in that interval allow a good extrapolation up to 10^7 turns. The GP is more efficient in improving the two- than the three-parameter fit, as the MSE for $N_{\max} = 5 \times 10^6$ is reduced from 1.7×10^{-2} (original fit) to 4.5×10^{-3} (GP fit) for the two-parameter case (74 % reduction), and from 8.5×10^{-3} (original fit) to 5.8×10^{-3} (GP fit) for the three-parameter case (32 % reduction).

The behaviour of the ML-based method was also probed on a large set of DA simulations: 3090 cases of the LHC at injection energy for various configurations of the Landau octupole strength and linear chromaticity. The fits were performed with $N_{\max} = 1 \times 10^5$ and then extrapolating the fitted function up to 10^6 turns and evaluating the MSE and the difference in MSE, i.e. $\Delta\text{MSE} = \text{MSE}_{\text{originalfit}} - \text{MSE}_{\text{GPfit}}$ (see Fig. 4). Whenever the GP is used, 50 iterations were applied (a trade-off between the improvement achieved by the iterations of the GP and CPU-time).

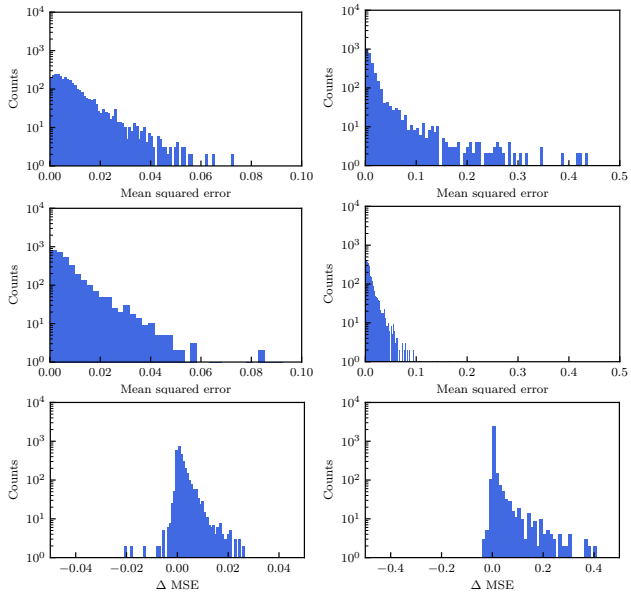

Figure 4: Left column: MSE distribution for the DA model (1b) fit with three parameters (top), after applying 50 times a GP to the DA numerical data (middle), and ΔMSE for the two approaches (bottom). Right column: Same, but for two parameter model.

Globally, the three-parameter fit MSE is smaller than that of the two-parameter fit, and the fit with GP has a better performance in terms of MSE than the original one, and is visible for both the two- and three-parameter fits. The positive part of the distribution of ΔMSE shows how many DA simulations were improved through the GP fit, whereas the negative part shows cases in which the GP fit has degraded the DA model. Although this degradation occured for some DA simulations, it is worth mentioning that this set corresponds only to 13 % and 8 % for the three- and two-parameter fits, respectively. Furthermore, this is likely to improve when using more than 50 GP iterations. It should be finally stressed that this does not represent a real degradation of the result (as one can choose to keep the original fit).

CONCLUSIONS

This paper presented results of ML techniques applied to the analysis of nonlinear beam dynamics in the LHC, namely outlier detection in dynamic aperture simulations, and the improvement of the DA models fitted to simulation data. In both cases, the ML techniques prove to be a good approach to improve existing tools. Outliers can be effectively identified and rejected, while analysis of DA evolution with the number of turns also showed considerable improvement by using a Gaussian Process to add synthetic data to simulations. This improves considerably the reliability of fits of models for DA evolution, and aids in the extrapolation of such simulations to timescales relevant to collider operation.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB028

REFERENCES

[1] T. M. Mitchell, *Machine Learning*. New York, USA: McGraw-Hill, 1997.

[2] A. Radovic *et al.*, "Machine learning at the energy and intensity frontiers of particle physics", *Nature*, vol. 560, pp. 41–48, 2018. doi:10.1038/s41586-018-0361-2

[3] E. Bozoki and A. Friedman, "Neural Networks and Orbit Control in Accelerators", in *Proc. 4th European Particle Accelerator Conf. (EPAC'94)*, London, UK, Jun.-Jul. 1994, pp. 1589–1592.

[4] E. Meier, G. LeBlanc, and Y. E. Tan, "Orbit Correction Studies using Neural Networks", in *Proc. 3rd Int. Particle Accelerator Conf. (IPAC'12)*, New Orleans, LA, USA, May 2012, paper WEPPP057, pp. 2837–2839.

[5] Y. Li, W. Cheng, L. H. Yu, and R. Rainer, "Genetic algorithm enhanced by machine learning in dynamic aperture optimization", *Phys. Rev. Accel. Beams*, vol. 21, no. 5, p. 054601, 2018. doi:10.1103/PhysRevAccelBeams.21.054601

[6] C. Emma, A. Edelen, M. J. Hogan, B. O'Shea, G. White, and V. Yakimenko, "Machine learning-based longitudinal phase space prediction of particle accelerators", *Phys. Rev. Accel. Beams*, vol. 21, no. 11, p. 112802, 2018. doi:10.1103/PhysRevAccelBeams.21.112802

[7] D. Vilsmeier, M. Sapinski, and R. Singh, "Space-charge distortion of transverse profiles measured by electron-based Ionization Profile Monitors and correction methods", *Phys. Rev. Accel. Beams*, vol. 22, no. 5, p. 052801, 2019. doi:10.1103/PhysRevAccelBeams.22.052801

[8] J. Wan, P. Chu, Y. Jiao, and Y. Li, "Improvement of machine learning enhanced genetic algorithm for nonlinear beam dynamics optimization", *Nucl. Instrum. Meth. A*, vol. 946, p. 162683, 2019. doi:10.1016/j.nima.2019.162683

[9] S. C. Leemann *et al.*, "Demonstration of Machine Learning-Based Model-Independent Stabilization of Source Properties in Synchrotron Light Sources", *Phys. Rev. Lett.*, vol. 123, no. 19, p. 194801, 2019. doi:10.1103/PhysRevLett.123.194801

[10] X. Xu, Y. Zhou, and Y. Leng, "Machine learning based image processing technology application in bunch longitudinal phase information extraction", *Phys. Rev. Accel. Beams*, vol. 23, no. 3, p. 032805, 2020. doi:10.1103/PhysRevAccelBeams.23.032805

[11] A. Edelen *et al.*, "Opportunities in Machine Learning for Particle Accelerators", 2018. arXiv:physics.acc-ph/1811.03172

[12] G. Azzopardi, B. Salvachua, G. Valentino, S. Redaelli, and A. Muscat, "Operational results on the fully automatic LHC collimator alignment", *Phys. Rev. Accel. Beams*, vol. 22, no. 9, p. 093001, 2019. doi:10.1103/PhysRevAccelBeams.22.093001

[13] O. S. Bruning *et al.* "LHC Design Report", CERN Yellow Reports: Monographs, CERN, Geneva, 2004.

[14] P. Arpaia *et al.*, "Machine learning for beam dynamics studies at the CERN Large Hadron Collider", *Nucl. Instrum. Methods Phys. Res. A*, vol. 985, p. 164652, 2021. doi:10.1016/j.nima.2020.164652

[15] J. Barranco *et al.*, "LHC@Home: a BOINC-based volunteer computing infrastructure for physics studies at CERN", *Open Eng.*, vol. 7, p. 378–392, 2017.

[16] A. Bazzani, M. Giovannozzi, E. H. Maclean, C. E. Montanari, F. F. Van der Veken, and W. Van Goethem, "Advances on the modeling of the time evolution of dynamic aperture of hadron circular accelerators", *Phys. Rev. Accel. Beams*, vol. 22, no. 10, p. 104003, 2019. doi:10.1103/PhysRevAccelBeams.22.104003

[17] G. Apollinari *et al.*, "High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1", CERN Yellow Reports: Monographs, CERN, Geneva, 2017.

[18] N. Nekhoroshev, "An exponential estimate of the time of stability of nearly-integrable Hamiltonian systems", *Russ. Math. Surv.*, vol. 32, p. 1, 1977. doi:10.1070/RM1977v032n06ABEH003859

[19] B. Boser, I. Guyon, and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", in *Proc. 5th Annual Workshop on Computational Learning Theory (COLT '92)*, Association for Computing Machinery: New York, NY, USA, 1992, p. 144–152.

[20] J. Vert, K. Tsuda, and B. Schölkopf, "A Primer on Kernel Methods", in *Kernel Methods in Computational Biology*, Cambridge, MA, USA: MIT Press, 2004, pp. 35–70.

[21] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, 1996, pp. 226–231.

[22] M. Breunig *et al.*, "LOF: Identifying Density-based Local Outliers", in *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104.

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. http://jmlr.org/papers/v12/pedregosa11a.html

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.