Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 1566–1571

Time-Series Deep Learning Anomaly Detection for Particle Accelerators

Davide Marcato [*,**] **Damiano Bortolato** [*]
Valentina Martinelli [*] **Giovanni Savarese** [*]
Gian Antonio Susto [**]

[*] *Legnaro National Laboratories, INFN, 35020 Legnaro, Italy. (e-mail: davide.marcato@lnl.infn.it).*
[**] *Department of Information Engineering, University of Padova, 35131 Padova, Italy.*

Abstract: High energy particle accelerators rely on superconducting radio frequency cavities to transfer energy and accelerate the beam. Such particle accelerators are complex and expensive systems prone to failures which lead to downtime of the whole experimental facility: it is thus of primary importance to anticipate and prevent these faults to improve the uptime and cost-effectiveness of particle accelerators. Data-driven methods are especially fit for this task as they can leverage all the data recorded and archived by a typical control system. Previous works used classical machine learning (ML) models for anomaly detection to detect early signs of an upcoming fault. We propose here a different approach based on deep learning (DL) models, exploiting the temporal correlation of the raw data. Three different models are tested on data from the ALPI (Linear Accelerator for Ions) linear accelerator in INFN (National Institute for Nuclear Physics) Legnaro National Laboratories in Italy and they are compared with the classical ML approach.

Keywords: AI methods for FDI, Anomaly Detection, Deep Learning, Time Series Forecasting.

1. INTRODUCTION

Particle accelerators are complex experimental plants which deliver high energy particle beams to a target for physics experiments or industrial and medical applications. They are composed of different and heterogeneous systems such as the ion source, the beam transport magnets and electrodes or the beam diagnostics, along with supporting infrastructure such as the water cooling and cryogenic systems. Given the complexity of each subsystem and the unpredictability of the interactions between them, it's difficult to avoid faults and operating errors.

In the ALPI (*Acceleratore Lineare per Ioni*, which stands for Linear Accelerator for Ions) particle accelerator at INFN Legnaro National Laboratories (LNL) one of the most critical systems is the radio frequency (RF) which is responsible to transfer energy to the beam and thus accelerate it. It uses superconductive RF cavities which are extremely sensitive to external factors such as noise coming from the cryogenic lines. The RF control system is able to correct small perturbations and keep the cavity operating as intended most of the time, but it still can stop working when anomalous conditions arise. These conditions are hard to predict and to take into account; when one of such failures happens on a single cavity, the beam trajectory is impacted so that the ions don't reach the target and thus the physics experiment is paused. Being able to predict these faults would mean to being able to act early and to prevent such conditions. Furthermore, a good prediction model can also be used to better understand the different causes of each fault and to identify the best preventive action to perform. Such an Anomaly Detection model (Maggipinto et al. (2022)) can be part of a broader pipeline of engineering tools with the aim of maximizing the total operating time of the accelerator, leading to increasing its scientific output and to optimizing its large operational costs.

With the advent of Machine Learning (ML), data-driven methods have been very successful at modeling complex systems and thus they are promising candidates for this kind of tasks. Furthermore, particle accelerators usually collect and record the trends of many process variables (PVs) from the thousands of sensors and actuators installed, meaning that a large dataset of raw signals can be available or can be easily collected during a run of the accelerator. This creates the possibility to adopt data-intensive approaches such as Deep Learning (DL) models, which are usually able to reach great performance by paying a penalty on training times and data efficiency. For these reasons, ML and DL models have been adopted by the particle accelerator community for different purposes, such as physics data analysis, beam dynamics optimization or to develop advanced and proactive control systems (Edelen et al. (2018)).

This paper presents a method to predict runtime faults on particle accelerations by means of DL Anomaly Detection models; the approach was tested and validated on the ALPI RF control system. The prediction is performed by

[1] This work was partially funded by the TeRABIT project for Marcato, Savarese and by the SEED BADA3 project for Susto.

10.1016/j.ifacol.2023.10.1855

a neural network based forecasting model for time series. Different approaches have been proposed in this field: Li et al. (2021) presented a classification model to predict safety interlocks based on Recurrence Plots and Convolutional Neural Networks (CNN). An anomaly detection method based on Recurrent Neural Networks is used by Sulc et al. (2022) to predict faults on the RF system. In this case the facility accelerates a photon beam and the faults are due to quenches, when a cavity loses its superconductive status. Li and Adelmann (2022) offer a review of time series forecasting methods in the field of particle accelerator, highlighting both linear and non-linear models and their importance in different applications in particle accelerators. An application on RF cavity faults based on a CNN classificator is presented, while methods based on recurrent neural networks are only theorized and left for future works. In Marcato et al. (2021) a first approach to the fault prediction problem and its application on the RF control system of the ALPI accelerator was presented; such work was based on classical ML models for Anomaly Detection, such as Cluster Based Local Outlier Factor (CBLOF) which don't natively take into account the time correlation of subsequent samples.

More advanced Anomaly Detection approaches for time series can be found in literature applied to other fields. For example in Liu et al. (2023) a Generative Adversarial Network (GAN) is used to reconstruct the original data, with the idea that anomalies will not be reconstructed correctly. This paper builds upon this idea, the usage of the residual reconstruction error of a deep learning model as an anomaly score, with a focus on the specific domain of particle accelerators data.

The paper is organized as follows: Section 2 will describe the ALPI RF control system and the nature of the faults to be predicted while Section 3 explains the data preprocessing and the model development. Finally, in Section 4 some experimental results are presented and a comparison between ML and DL performance is presented, highlighting the strengths and weaknesses of the two approaches. We finally draw conclusions on Section 5.

2. ELEMENTS OF PARTICLE ACCELERATORS CONTROL SYSTEM: THE ALPI RF CASE STUDY

The ALPI linear accelerator is part of the accelerator complex at INFN LNL, used to accelerate proton beams for nuclear physics experiments. It is based on superconductive RF Quarter Wave Resonators (QWR) operating at 80MHz and 160Mhz. To reach the superconductive state, the cavities are cooled down using liquid helium to a temperature of about $4K$ and they are placed inside a cryostat cooled with liquid nitrogen in groups of four. The specific geometry of these cavities is designed to generate an oscillating electric field in the direction of the beam. A charged particle travelling in the same direction is thus accelerated thanks to the Lorentz force. Since the beam is not a continuous flow of protons but is divided into bunches, the frequency and phase of the electric field oscillation must be synchronized with the beam bunches, so that when a bunch of protons enters the cavity the field is in the correct direction to accelerate the bunch instead of decelerating it. Thus the RF system must

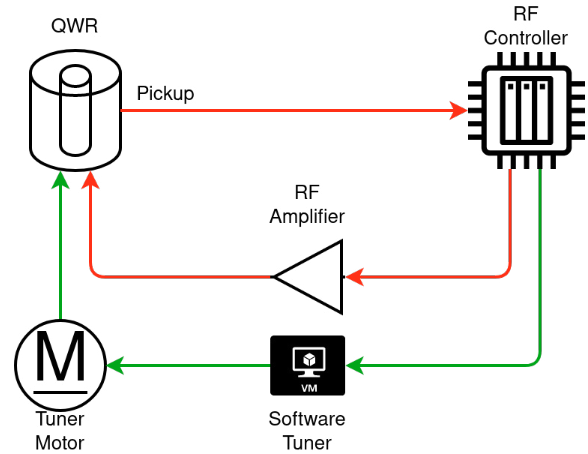

Fig. 1. RF feedback control system

control the cavity electric field through a power RF signal which must have fixed frequency and phase, as well as the desired amplitude setpoint. When all these conditions are respected, the cavity is considered *locked*.

To achieve these goals the RF control system implements two feedback control loops (see Bortolato et al. (2018)), as highlighted in Fig. 1: the electric field in the cavity is sampled through an opening, the *pickup*, and digitized by the RF controller. This a custom built board which includes ADCs, DACs and an FPGA, used to calculate a feedback signal to be amplified and then fed back into the cavity. This system is very fast and accurate, but can only correct small errors on the RF frequency and phase. The second loop is thus used to perform bigger but slower corrections: the frequency error sampled by the RF controller is read by the software control system running on a remote server and, if it gets bigger than a certain threshold, the tuner motors are moved to reduce it. In fact, the native resonance frequency of a cavity depends on its geometry and thus, by moving a plate at one end of the cavity it is possible to change it. This control loop is much slower and less precise than the first one, but is able to correct slow drifts.

Theoretically, these two systems would be able to keep the RF signal phase, frequency and amplitude fixed at their setpoint. In practice, multiple faults are typically observed, which results in *unlock* events. Such faults are due to the complexity of the real world, where each cavity is not built exactly the same as another one and it behaves differently (especially at cryogenic temperatures), motor movements are not completely linear and suffer from hysteresis and external systems introduce unknown variables. Unfortunately, when a single cavity is *unlocked*, the beam energy changes, and thus it will loose the synchronization with subsequent RF cavities causing a complete loss of beam particles, meaning that the beam will no longer reach the target and the physics experiment is paused. From the point of view of the experiment, the whole accelerator experience a downtime and a manual intervention is required. Since ALPI has about 80 cavities and a single cavity fault is enough to block the whole accelerator, it is mandatory to reduce the probability of each fault to the minimum. This paper uses the data from

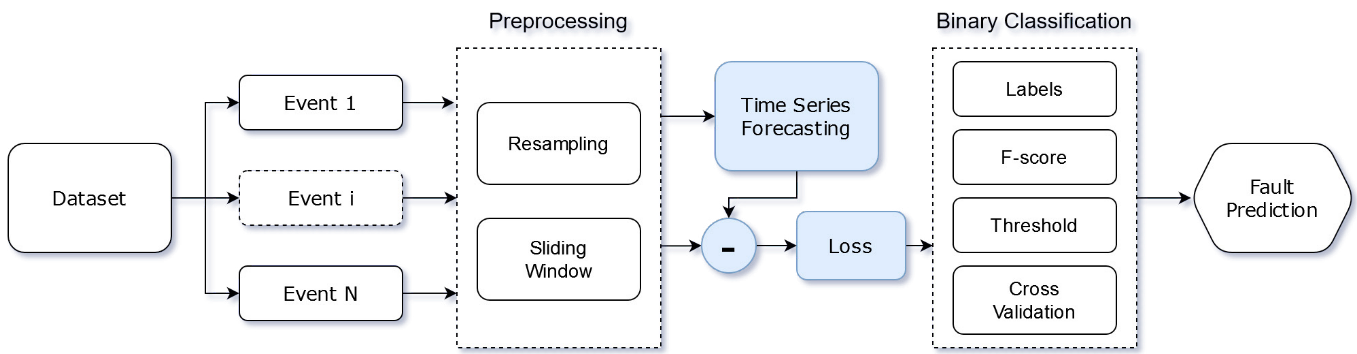

Fig. 2. A schema illustrating the main steps of the proposed approach to build an anomaly detection model. The anomalies are then used as indicators for fault prediction.

a one month long run of the ALPI accelerator in February 2020, which include a total of 24 cavities operating at 80MHz and a total number of 169 registered unlock events.

3. ANOMALY DETECTION

This section describes the anomaly detection model developed for fault prediction, including a brief description of the available dataset, its pre-processing and the evaluation metrics. Figure 2 shows a flowchart which summarizes the proposed fault prediction pipeline. The historical data is first divided into events, each including a single fault on a single cavity and all the relative process variable time series during the normal operation preceding the fault. Then, a preprocessing phase is used to normalize and resample the time series to a constant sample frequency and to apply a sliding window to obtain the final training dataset. This is then used to fit a time-series forecasting model, which is able to predict the next value given an observation from the past. The main idea is that a good prediction model should be able to forecast the future value with a small error during normal operation, when all the variable behave as expected, while the prediction error could be much bigger for anomalous values. Thus, by comparing the prediction with the next value, as soon as it is available, it's possible to compute the prediction loss and use it as an outlier score. A high outlier score is used as an indicator of an upcoming fault, since we expect to observe anomalous behaviours in the process variables before a fault. So, by applying a threshold on the prediction loss it's possible to obtain a binary classification which distinguish normal trends from upcoming faults.

This pipeline builds upon the work presented in Marcato et al. (2021); with respect to the original work, the main differences are highlighted in blue: previously an anomaly detection model was run on a set of manually extracted features to obtain the outlier score, now the score is calculated from the prediction loss of a time series forecasting model fit on the original data. The following paragraphs explain all these steps in greater detail.

3.1 Dataset preparation

The data is composed of the evolution over time of multiple process variables recorded by the RF control system. The raw data is structured as multiple concurrent time series: each data point includes the PV name, the acquisition time and the value, for a total of 670 million data points and 20GB of CSV files. Among these, a few PVs for each cavity are selected to be used:

- *Lock* and *Lock Failed* statuses;
- Forward RF power;
- Tuning motor direction and moving flag;
- Cryostat liquid helium tank pressure.

These variables are chosen among the ones most correlated with the events, thanks to the experience from previous works. The *Lock* status indicates the normal operation of a cavity, when the cavity is correctly accelerating the beam. The *Lock Failed* status indicates that the cavity is no longer in the Locked status due to a failure, as opposed to an intended operation. These are used to split the data into *events*, where the cavity is working correctly up to the next fault event. The other PVs include the RF power, which is directly correlated with the frequency and phase error of the cavity, the tuning motors status, which are moved to adjust the residual frequency error, and the helium pressure in the cryogenic tank which is used to cool the cavity. For example in Fig. 3 the trend of the RF power and Helium Pressure is shown during a time window

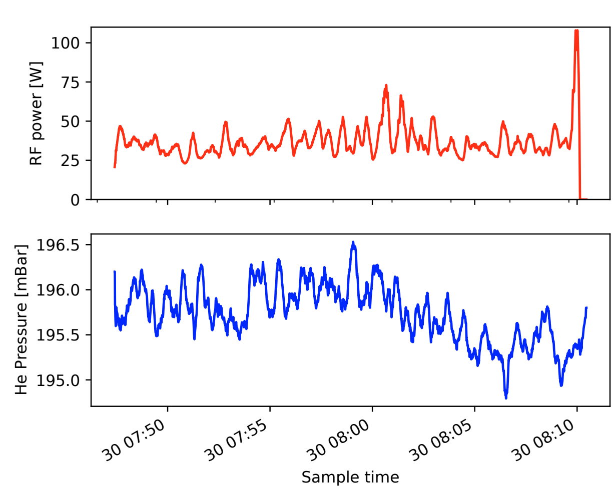

Fig. 3. The trend of the RF power and He Pressure before an unlock event.

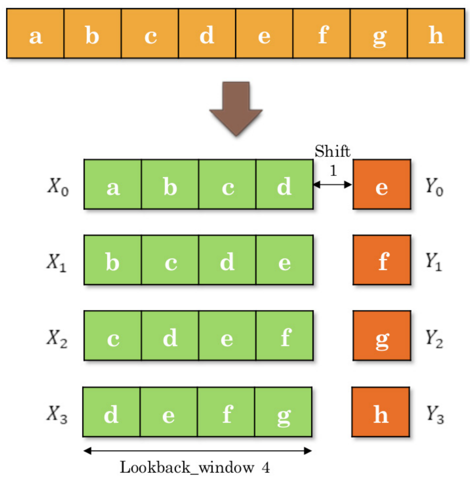

Fig. 4. A sliding window example.

preceding a fault event on a cavity. A strong perturbation is clearly visible in the RF power just before the event.

After splitting the data into events, three transformations are applied before using them. In fact, as can be seen in Figure 3, different PVs have different engineering units and different scales, which is not ideal to be used as input to a neural network because it may result in a slow or unstable learning process. For this reason, the data is first normalized using the Standard Scaler, which imposed the data to be centered in zero, with unit variance. The standard scaler is used in this case as the data do not have an obvious maximum value and thus it's not trivial to normalize it based with respect to the maximum value.

The second transformation is the resampling operation. In fact, time series model expect to have an input with a constant and aligned sampling rate. Instead, the data was recorded during the operation in an event driven way, meaning that each value is written to the database as soon as it changes. This is useful not to loose any value and to save storage space for slow PVs, but is not very convenient during the analysis of the data. Thus all the data was resampled to the same sampling rate, simply using the mean value in the sampling period as the new value. When no data was recorded in the sampling period, the last valid value was propagated. The resampling period is an important parameter to choose: choosing a very fast sampling rate the dataset size and the execution time increases exponentially, without adding any meaning information, but a very long sampling period means loosing a lot of information due to the averaging.

The final step is the creation of the actual dataset for forecasting using a sliding window approach. This is a crucial part of the process to build the model, as it can have a direct impact on the training of the neural network and the final result. An example of sliding window application with a single feature is shown in Figure 4. From the initial sequence of values of the time-series we want to create a set X where each sample is a subsequent portion of the original data (with the correct time order), and a set Y with the future value of each window. Given an element from the X set, our model will be trained to predict the corresponding value in Y. There are many parameters that can be adjusted:

- **input PVs**: this indicates which PV to include in the X samples;
- **output PVs**: which PV to include in the Y samples;
- **lookback window**: the length of the window;
- **shift samples**: how *far* in the future to predict in number of samples;
- **batch size**: the number of samples in a training batch;
- **step size**: how many samples to slide between one window and the next;
- **window sampling rate**: optionally subsample the values in the window to reduce its size while keeping old values;

These parameters determine the forecasting horizon, which is selected aiming both at proving accurate predictions in useful time and at considering typical dynamics of a failure (ie. the amount of time on which anomalous behaviour start to verify before the unlock event). Furthermore, the length of the window is adjusted to be able to capture trends far in the past, while subsampling is then applied to limit the number of points in the window. This enables the use of smaller models, which have a limited number of nodes in the input layer. By applying the subsampling after the windowing, the number of samples in (X, Y) is not artificially reduced. A side effect of the sliding window operation is the explosion of the size of the data, since each original point is repeated many times in the observation windows. For real world use case, like the one presented in this paper, this can become a problem as it's easy to incur into memory limitation issues, even when using modern hardware. In fact, with the naive approach the dataset is fully loaded on the device memory and can exceed its size, in this case the 16GB of memory of the available GPU. For this reason the implementation of the dataset should use streaming data type which loads the data in memory in chunks to avoid hitting the memory limit.

3.2 Forecasting Models

Given the dataset described in the previous paragraph, three Neural Networks were tested as forecasting models. The first one is based on Long Short Term Memory (LSTM) layers (Hochreiter and Schmidhuber (1997)). This is a Recurrent Neural Network (RNN) architecture, meaning that it can propagate a "state" between inputs so that the output depends both on the input and on the state, effectively exploiting the time dependency between the data points. In particular, LSTM layers are designed so that the output does not depend only on the state from the last few nodes, but can have long term dependencies.

As shown in Fig. 5, we chose to use a Neural Network with 2 stacked LSTM layers with 64 units each, both returning the whole sequence of outputs of each unit. Thus, the second layer receives in input these output and reprocesses them sequentially to obtain new outputs. These are finally fed into a fully connected layer with a single output to perform regression on the prediction value. No activation function is used on the output neuron.

The second model was built using a Temporal Convolutional Network (TCN) from Lea et al. (2016): this is based on 1D causal convolution layers arranged so that

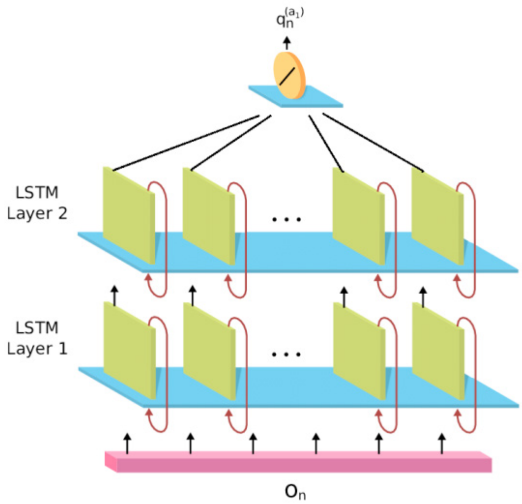

Fig. 5. The LSTM based neural network architecture.

the output depends only on the inputs from earlier in the sequence. Furthermore, by stacking the layers it's possible to obtain a large receptive field, meaning that the input sequence can be long and the data dimensionality is reduced by the convolution. Skip connections can be added to propagate the gradient between deep stacks of layers. In this case the complete network is composed by a stack of 3 layers with dilations of 1, 4 and 16, 64 filters and skip connections. After that a fully connected layer with 16 nodes precedes the single output node with linear activation.

The aforementioned models both use the trend of multiple input time-series to forecast a single value: the RF power. The choice of the single variable to forecast is arbitrary and can have a big impact on the final results. An alternative approach is to forecast multiple values at once and then combine the losses of each prediction to obtain a more robust outlier score. Thus, we developed a third model , based on the TCN network, to predict the forward and reverse RF power and the helium pressure. The dataset is changed to include the three variables in Y and the output layer of the network is increased to 3 nodes. Finally, the average of the prediction loss in the three variables is used as the final score.

The hyper-parameters of the three model were chosen empirically balancing accuracy and fit times, since the final goal is not to obtain a perfect forecasting model but to produce a good anomaly score. These models are trained with the Adam optimizer for 10 epochs using the Mean Squared Error (MSE) loss function. The parameters of the dataset presented in the previous section are chosen as shown in Table 1. This means that the LSTM model predicts 30s in the future by looking at 1 value every 30 seconds in the last 256 minutes, while the TCN models look at one value every 10s in the last 30 minutes and predict 10s in the future.

The three models achieve similar performance, with the LSTM network obtaining a MSE 0.723 and a Mean Absolute Error (MAE) of 0.5222, while the TCN achieved a MSE of 0.702 and MAE of 0.5229. The TCN predicting multiple variable reaches a MSE of 0.503 and a MAE of 0.4044. By computing the absolute value of the difference between the predicted value and the actual value, it's possible to obtain an outlier score for each point. In fact, we expect the forecasting model to be less accurate on anomalous data point. This can be observed in Figure 6, where the time series of all the prediction errors in the last 300 samples before each event are shown. The prediction errors explode in the last few samples before an event, indicating the presence of anomalous data and thus anticipating the fault event.

Table 1. Dataset preprocessing parameters

	LSTM	TCN	TCN Multi
Data sampling period	0.5s	1s	1s
Input PVs	All	All	All
Output PVs	RF power	RF power	RF power, He pressure
Lookback window	30720	1800	1800
Shift samples	60	10	10
Batch size	128	128	128
Step size	3	1	1
Window sampling rate	1/60	1/10	1/10

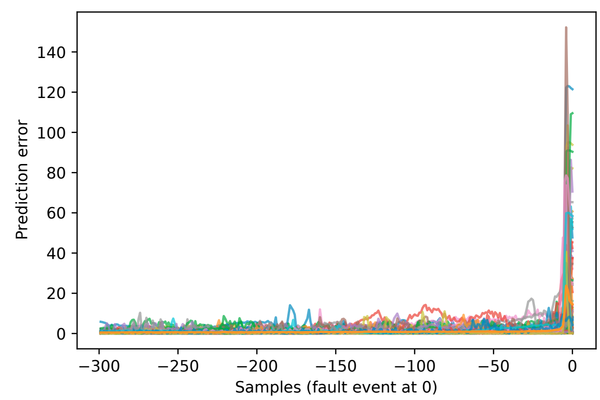

Fig. 6. The prediction errors on the last 300 samples of all the events, aligned with fault at sample 0.

4. EXPERIMENTAL RESULTS

The outlier score obtained from the prediction error, as presented in the previous paragraph can thus be used as an outlier score to predict faults. To do so, a simple threshold function is used to classify each point between anomalous or not. The value of the threshold must be chosen to maximize the number of true positive while avoiding false positives and false negatives.

For this purpose artificial labels were computed from the Y dataset: all the datapoints which are far from the next events are assigned a label of -1, meaning that they are considered non-anomalous. The samples in the last 5 seconds are assigned a non-negative value, identifying the corresponding event, to indicate that they are considered outliers. After that a threshold is applied on the prediction loss and all the points are classified as either 0 or 1, where 1 is considered an anomalous point. Now, a fault is considered correctly classified if at least one point in the last 5 seconds before its occurrence is classified as anomalous. Vice versa, a false positive is counted for each sequence of 1s in the predicted labels which are not in the last 5 seconds before the event. Finally, the correct threshold must be chosen. The F-score is used in this case as a single metric to be maximized. In fact the F-score depends both on the True Positives (TP), the False Positives (FP) and the False Negatives (FN), and it's able to find a trade-off between them. The best threshold is chosen by testing the F-score at different threshold levels, and choosing the one that minimized it. Since the dataset is highly unbalanced, meaning that there are few faults among many normal points, the True Negatives (TN) is close to the total number of points, which is not really interesting and thus it's not calculated.

With this setup it's possible to obtain the following results for the three models:

Table 2. Fault prediction results

	LSTM	TCN	TCN Multi
Fscore	0.578 ± 0.004	0.642 ± 0.006	0.691 ± 0.006
Precision	0.713 ± 0.025	0.743 ± 0.043	0.831 ± 0.023
Recall	0.419 ± 0.019	0.507 ± 0.036	0.516 ± 0.019
TP	25.6 ± 1.20	66.4 ± 4.92	67.6 ± 2.57
FP	10.4 ± 1.85	23.6 ± 6.65	13.8 ± 2.78
FN	35.4 ± 1.20	64.4 ± 4.71	63.4 ± 2.57

The results show that these models are able to correctly predict many faults with few false positives. When compared, TCN models are more robust and obtain a higher F-score than the LSTM one, and a further advantage is gained by predicting multiple variables. The Precison-Recall plot in Fig. 7 shows the behaviour of the models when changing the threshold level, and the TCN Multi outperforms the other on the whole range of threshold. Moreover, the graph highlights that by accepting a lower recall value it's possible to reach high levels of precision, which may be useful depending on the use case.

These results are comparable with the ones of a few classical ML models presented in Marcato et al. (2021), trained on the same dataset, even though they do not reach the F-score of 0.87 of the best one. Even so, DL models have the unique advantage of being usable without any manually crafted feature, which means that they are directly portable to to different environments and different kinds of faults. Furthermore, while the dataset includes millions of data points, not all of them are usable to build the training models, and they pertain only to a short time span of 1 month of operation, meaning that they did not include much variability. Even if the DL training did not take into account the faults, the data included less than 200 events, a very small number to build a confident model. This is probably an advantage for simpler models, but by collecting more data and recording more events DL models are expected to reach better performance.

5. CONCLUSIONS

We presented an anomaly detection approach for fault prediction in the field of particle accelerators, based on DL time series forecasting Neural Networks. Three different networks are developed to predict future values of the process variables of the accelerator. The prediction loss is used as an outlier score and a high value is considered as an indicator of an upcoming fault. The F-score on the fault prediction is used to select the best threshold on the outlier score to be able to discriminate normal point from anomalous ones.

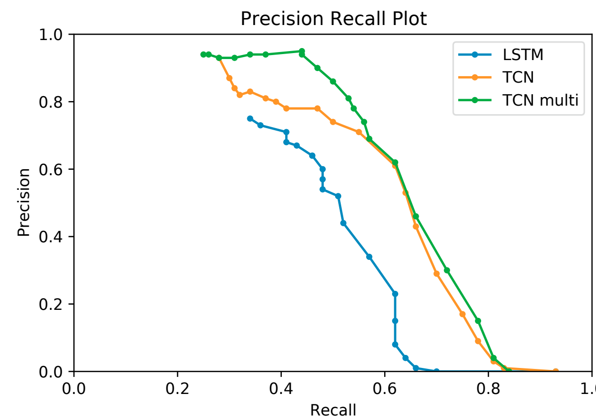

Fig. 7. The Precision-Recall plot of the three models.

The models are tested on the RF control system of the ALPI particle accelerator, where they are able to correctly predict many unlock events on the RF cavities while returning few false positives. A comparison with previous results from classical ML anomaly detection models show a performance advantage for earlier models, but DL models have the advantage of reaching good results without requiring manually crafted features. Furthermore, by collecting more data and more faults during different runs of the accelerator, these model could prove useful where simpler models may scale worse.

REFERENCES

Bortolato, D. et al. (2018). Upgrade of the LLRF Control System at LNL. In *Proc. of ICALEPCS 2017*. JACoW.

Edelen, A. et al. (2018). Opportunities in machine learning for particle accelerators.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Lea, C. et al. (2016). Temporal convolutional networks: A unified approach to action segmentation. doi: 10.48550/ARXIV.1608.08242.

Li, S. and Adelmann, A. (2022). Review of Time Series Forecasting Methods and Their Applications to Particle Accelerators.

Li, S. et al. (2021). A novel approach for classification and forecasting of time series in particle accelerators. *Information*, 12(3).

Liu, S. et al. (2023). Time series anomaly detection with adversarial reconstruction networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(4).

Maggipinto, M., Beghi, A., and Susto, G.A. (2022). A deep convolutional autoencoder-based approach for anomaly detection with industrial, non-images, 2-dimensional data: A semiconductor manufacturing case study. *IEEE Transactions on Automation Science and Engineering*.

Marcato, D. et al. (2021). Machine learning-based anomaly detection for particle accelerators. In *2021 IEEE Conference on Control Technology and Applications*, 240–246.

Sulc, A. et al. (2022). A Data-Driven Anomaly Detection on SRF Cavities at the European XFEL. In *Proceedings of IPAC 2022*. JACoW.