Experience with anomaly detection using ensemble models on streaming data at HIPA

Jaime Coello de Portugal[a], Jochem Snuverink[a]

[a] *Paul Scherrer Institut (PSI), Forschungsstrasse 111, 5232 Villigen PSI, Switzerland*

Abstract

Anomaly detection techniques are applied in many industry settings with great success. This paper presents experimental evidence of usefulness of these techniques in the High Intensity Proton Accelerator at the Paul Scherrer Institute. We present an anomaly detection model built as a combination of the Numenta NuPIC model and an LSTM autoregressor that works on streaming data from several beam diagnostic devices, automatically learning complex patterns from these signals after a relatively small adaptation time and small number of false positives. We show how this system could have alerted human experts of the failure of critical beam instrumentation tens of minutes in advance by finding anomalies preceding it, which stood unnoticed for several hours. We also present a full framework to exploit these models, which allows to monitor live data from the diagnostic devices, model results and diagnostic data and show historical data for easier model tuning.

1. Introduction

Particle accelerators instrumentation have very tight operational constraints. Bad readings or failures of critical devices typically lead to avoidable beam interruptions. Particle accelerators control systems handle data from hundreds of thousands of such sensors, like beam current monitors, temperature sensors, beam loss monitors, etc., at different and often very high update rates. In accelerators like the High Intensity Proton Accelerator (HIPA) at the Paul Scherrer

Email addresses: jaime.coello@psi.ch (Jaime Coello de Portugal), jochem.snuverink@psi.ch (Jochem Snuverink)

Institute (PSI), which are operated using such control system but also where users require long-term machine stability or the maximisation of some key parameters, device failure induced beam interruptions can severely harm the performance and safety of the machine.

Human experts can typically spot any misbehaviour of these signals, and perform preventive or mitigating operations to neglect or reduce the impact of the failures. However, it is impractical for human experts to actively monitor thousands of signals. Automated analysis techniques are therefore necessary if one wants to ensure that this kind of failures are detected reliably. As many of the signals have a complicated structure that changes both short term and long term, traditional statistical techniques might not be practical either, as their parameters would have to be firstly tailored to each specific signal and secondly manually changed over time whenever the operation configuration of the machine changes. The use of machine learning (ML) techniques is required to produce models that automatically learn the behaviour and adapt to changes in the input signals, and then detect any spurious deviations.

Anomaly detection is the branch of ML that automatically learns the structure of (parts of) the input data and scores each individual point according to how far they deviate from some notion of *normality*. The definition of this *normality* of the data points is usually one of the main challenges of anomaly detection and requires human expertise and fine-tuning of the models. The only requirements of this notion of *normality* and thus of the anomaly scores is that relatively few points of the data set must be assigned a high score, i.e. the anomalies must be sparse over the data set.

The anomaly detection tasks can be separated in several ways like, for example:

- Supervised vs. unsupervised: Whether the anomalies have been previously labelled (usually by human experts) and the models are trained in a supervised way, using the data under study as input and the targets as outputs, or whether the model has to find the anomalies by automatically learning the notion of *normality* from the input.

- Online vs. offline: In an online setting, the data is fed to the models continuously, can be considered infinite and must provide an evaluation as fast as possible. Offline models are

presented a finite-sized data set at once and can therefore infer a global structure on the data, providing an evaluation only when the whole data set is available.

In this paper we will focus on models that address the online and unsupervised anomaly detection problem.

In an online setting, the signal is sent to the models as soon as it is acquired, and they must provide an evaluation before the next sample is available. Therefore, the required performance of the models depends critically on the frequency of the input stream. These live models must have the ability to quickly adapt to changes in the notion of *normality*. For example, after a change on the operation configuration, a current sensor might exhibit a jump in the measurement and settle at a new value. The anomaly detection models will identify these changes as anomalous, but after certain amount of samples of this new state is available the notion of normality must be adapted.

These types of models must also be able to forget about past learned structures or its memorisation capability might be overwhelmed over time. As once the normality notion is changed, past values are likely to become irrelevant and can typically be forgotten without harming the model performance. This balance between quickly learning new normal sequence and forgetting past irrelevant ones is one of the main tuning problems in these type of models as it depends exclusively on the temporal structure of the input signals.

Many online unsupervised models are constructed by using *autoregressor* models. Autoregressor models are trained to predict future samples of the input sequence by observing past values and learning some inherent structure. This way, a distance metric can be defined to compare the predicted values and the measured ones which gives a score on how unexpected, or anomalous, the newly received sample is.

Another good property of some anomaly detection models is the ability to take into account the context of each new sample (the structure of the previous samples) and, even more importantly, to *doubt* in case of ambiguous context and expect several values at the same time. If the signals can change unexpectedly but these changes might be fairly common, models that expect several possible next values given the same context allow to reduce significantly the number of

false positives.

A typical problem with unsupervised anomaly detection tasks, is the validation of the results and assessment of the performance of the algorithms. Specifically, the identification of false positives. However, as the number of anomalies must be by definition very sparse over time, two or more algorithms of very different nature are very unlikely to report a false positive at the same time. Therefore, by using ensemble learning i.e., using two (or more) anomaly detection models and combining their input in some way that only outputs a high anomaly score when many models agree, the false-positive rate can be kept under control.

Anomaly detection ML models have been successfully used in many industry settings, like credit card fraud detection [1, 2], intrusion detection in cybersecurity [3, 4], or fault diagnostics in industry. Being such complex facilities, anomaly detection models can be an incredibly useful tool to improve the operation and performance of particle accelerators. However, despite their big potential, they are still used only sporadically. Isolation Forest has been used to detect anomalies in the LHC beam position monitors [5], which allows to automatically discard monitors with spurious readings. For the future High Luminosity upgrade of the LHC, anomaly detection models using recurrent neural networks are under study to detect anomalous behaviour in superconducting magnets [6]. Convolutional neural networks have been used at the SOLARIS synchrotron to detect anomalies in a set of signals of the vacuum levels of the accelerator [7]. Finally, an exploratory attempt was made to apply the Hierarchical Temporal Memory algorithm to the RHIC collider at the Brookhaven National Laboratory with moderate success [8]. This is one of the anomaly detection models used in this study (Section 2).

In this paper we present an anomaly detection model built by averaging the scores of two different models, the aforementioned NuPIC Hierarchical Temporal Memory model (Section 2) and an LSTM autoregressor (Section 3). In Section 4 we show how the data from the control system of the accelerator is prepared and the results and diagnostic information of the anomaly detection models are recorded and shown to the human experts for evaluation. In Section 5 several instances of real-time model predictions of the presented model are shown, demonstrating how it learns complex structures in the input signals automatically and how it could have helped

human experts detect device failures tens of minutes in advance.

2. Hierarchical Temporal Memory Detector

The Hierarchical Temporal Memory (HTM) algorithm, was born as a model of the mammal pyramidal neurons' organisation in the neocortex [9, 10]. This model postulates that the thousand of connections between these type of neurons act as a very noise resistant and high capacity sequence memory. Therefore, this algorithm is strongly suited to be used as an anomaly detector, as each new sample in the input (with its context) can be compared with the expected values from previously learned sub-sequences.

The input of the HTM algorithm is a binary vector with a few active bits (typically about 2%), named a sparse density representation (SDR) of the input value. To perform this encoding, the input space is divided into bins and each bin gets a set of bits assigned. This representation allows to include expert knowledge as the active bits are allowed to overlap (or not) to represent *proximity* between input values.

These SDR binary vectors are given to the HTM model, which consists of:

1. The spatial pooler: a large set of cells which identifies patterns in the input SDR.
2. The temporal memory: organises the cells into columns representing the same input pattern but with different connections between columns. These connections between cells of different columns represent the different contexts in which each input pattern can appear.

As a specific implementation of the HTM algorithm, we used the NuPIC library developed by Numenta [11]. This library is written in Python 2 with bindings to C++ to speed up the core computations. Unfortunately, as this library is not being developed anymore and only supports Python 2, the communication with the rest of the Python 3 application is done via remote procedure calls using the standard Python *xmlrpc* library [12].

3. LSTM Detector

Long-Short Term Memory (LSTM) models [13] are a type of recurrent neural network (RNN) models that are designed to overcome the vanishing gradient problem present in RNNs. LSTM

models are sequence-to-sequence models that have been very successfully applied to many real world applications like speech recognition [14] and language translation [15]. LSTMs have been typically used for anomaly detection as autoencoders, like for example in [16].

We build an anomaly detector using an LSTM as shown in Fig. 1: The model takes a window of $w + 1$ samples from the sensor recordings: $x_{t-w-1}, x_{t-w}, ..., x_{t-1}, x_t$, being x_t the most recent (or current) sample. The vector $x_{t-w-1}, ..., x_{t-1}$ of length w is used as input for the LSTM model. The target of the model is to obtain a value close to x_t. In this way the output of the model can be compared with the real output sample and simultaneously compute an anomaly score depending on how far the prediction is from the real sample and train the model on each time step using this newly acquired sample.

As it was described in the introduction, the ability of anomaly detection models to predict several values simultaneously to account for ambiguous context is extremely useful. To allow this behaviour in our LSTM model, the latest value x_t is transformed into a binary vector $\vec{\hat{x}}_t$ by dividing the input space into N *bins* and setting to 1 only the index i corresponding to the bin containing x_t:

$$\hat{x}_{t,i} = \begin{cases} 1, & \text{if } i = \left\lfloor \frac{x_t - min}{max - min} N \right\rfloor \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where *max* and *min* are the maximum and minimum possible readings of the device. Performing a dot product of this vector $\vec{\hat{x}}_t$ with the output of the LSTM $\vec{y}_t$ a value that represents how high the expectation of the model is to find the latest value in its bin. Note that instead of a linear binning also logarithmic or other binning, tailored to the observed sensor, can be used. The anomaly score A for the x_t sample can be written as:

$$A(x_{t-w-1}, x_{t-w}, ..., x_{t-1}, x_t) = 1 - \vec{\hat{x}}_t \cdot \vec{y}_t \tag{2}$$

Since a trivial solution is to always output a vector of ones $\vec{y}_t = \vec{1}$, to force the model away from this solution, the penalty function P for the LSTM model is defined as:

$$P = A + \gamma \left\| \vec{y}_t \right\|_1 \tag{3}$$

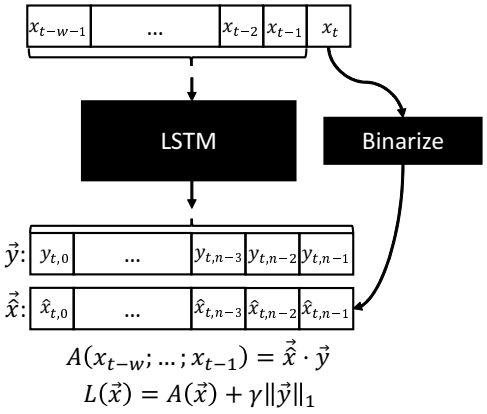


Figure 1: Schematic representation of the LSTM anomaly detector.

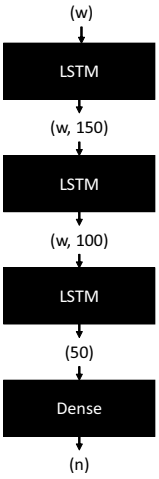


Figure 2: Architecture of the LSTM anomaly detector. Here w is the length of the input window and n the number of output bins.

where $\left\|\vec{y}_t\right\|_1$ is the 1-norm of the model output. The parameter γ controls how strong this penalty is on the model output 1-norm. Higher values will make the model forget faster about values he has not seen in a while. Lower values instead will give a low anomaly score to values that have been present long in the past.

This model was implemented using Keras and Tensorflow and the specific architecture of the LSTM we used is shown in Fig. 2.

4. Data collection and exploitation

We applied the previously described model to data from the HIPA diagnostic devices. HIPA is one of the most powerful cyclotrons in the world, able to provide a 1.4 MW beam to several experiments [17]. To operate and monitor this facility, about 180,000 signals are generated by devices of very different nature, of which about 19,000 are stored in a database (the Archiver).

A framework has been developed in order to take the data from these devices, synchronise and clean the data and present the results to the users. In our case, live data is requested to the control system used at HIPA: EPICS [18]. The framework is also able to take historical data from the Archiver to assess the performance of the models on previously acquired data. The

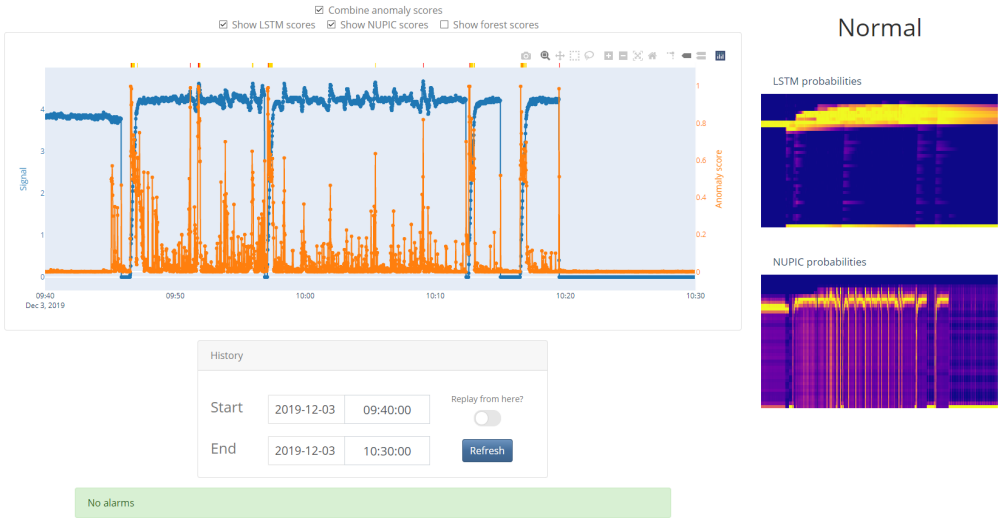


Figure 3: Screenshot of the live GUI, showing in the central plot the signal of a HIPA loss monitor in blue, and the average anomaly score between the LSTM detector and the NuPIC predictor in orange. The plots on the right column show the expectation each model has on the next sample value.

frequency of the data provided by EPICS depends on the monitored device and it is not fixed, a sample is received every time the sensor reading changes its value a certain amount. However, to keep track of the age of each of the samples, the input to the anomaly detection models must have either an indication of the time elapsed between samples or just a constant frequency. The input streams are therefore aligned to a fixed, 1 second frequency time grid by shifting the latest available sample.

Each value read from the input streams is accumulated until a window of length $w + 1$ is filled and then a new window is generated for every new sample by removing the oldest sample and adding the new one. These windows are then sent to the anomaly detection models, in our case the LSTM and the NuPIC detectors. The anomaly scores, input signals and diagnostics information are recorded in a database sample by sample. The database can both be read live to get the streaming anomaly scores and used for offline analysis. The anomaly scores of each model are then averaged to obtain the final anomaly score.

Figure 3 shows a screenshot of the graphical user interface (GUI) of the framework, developed to present a live and historical display of the anomaly scores. It displays the monitored

signal, the anomaly score and an indication of the expectation of the models.

The database can be also monitored by any other system. For example, a small program was also developed that monitors the database and notifies (via email) human experts as soon as an anomaly is encountered, attaching plots and useful information of the event.

5. Results

To test these algorithms, we took as a proof-of-principle one of the loss monitors in HIPA, specifically one of the so-called ”Blende” (aperture) collimators in the beam dump line. This collimator is composed of 4 blades that measure the losses on the top, bottom, left and right of the beam direction.

Fig. 4 shows, in blue, the signal of the device and in orange the ensemble anomaly score assigned by the model, averaging the LSTM and HTM models scores. The horizontal lines show two different thresholds for the anomaly scores: at 0.5 the event is flagged as suspicious and at 0.8 the event is flagged as anomalous. The decision on where to set these thresholds is fairly arbitrary, and depends on how many anomaly reports per unit of time we want to allow. This information can be requested to experts on the devices under study. In any case, these specific models tend to output very low values when the signal behaves normally and large spikes, very close to the maximum score, when anomalies are found. Therefore they are not too sensitive to the selection of the threshold. However, this does not hold in general for other models.

The dips in the device signal are caused by a kicker that diverts the beam to a different beam line to feed the Ultra-Cold Neutron (UCN) experiment [19]. These UCN kicks last a few seconds, and are typically triggered every 5 minutes. The kicker is only active at certain periods of the HIPA operation and therefore the anomaly detection models should adapt fast to the start and end of these UCN sessions.

This kind of behaviour is very challenging for the anomaly detection models as the chosen time windows length of 20 seconds (in this specific case), is much shorter than the 5 minutes space between UCN kicks and therefore these dips are very hard to predict.

One of the main characteristics of these models is their ability to discover patterns in the

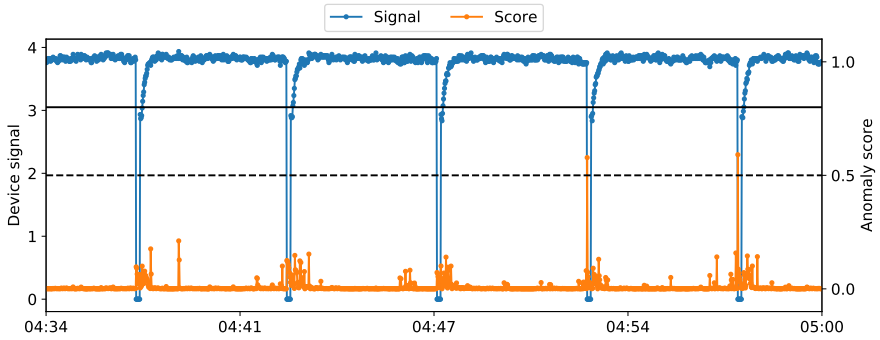


Figure 4: Example of the anomaly detection outcome on a signal from a loss monitor in HIPA. The blue line shows the actual readings of the device. The orange line shows the anomaly score, with the warning and anomaly thresholds shown as the horizontal dashed and solid lines respectively. The model started reading the signal at 00:00.

input signals. In the anomaly scores shown in Fig. 4 the dips in the kicker are not flagged as anomalies (they get a non-zero but generally low anomaly score), this is due to the model having seen many of these events in the past and learning the behaviour, even though it only happens once every 5 minutes or 300 samples.

Figure 5 shows in the bottom plot the individual anomaly scores computed by the NuPIC and LSTM models and the top plot the monitored signal and the combined anomaly score (average of the other two signals). This demonstrates how, as each model rarely outputs a high anomaly score, the combination of both detectors is unlikely to flag false positives in the same time-step, effectively improving the false-positive rate of the system.

An indication of the nature of the predictions of each of the models is shown in Fig. 6. For the NuPIC model, darker colours at a specific time step and a specific bin number, indicate a larger amount of cells expected to be activated in the current timestamp associated with values in that bin, i.e. the model is expecting the current value to land in bins with darker colours. For the LSTM model, this graph shows the $\vec{y}$ vector (see Fig.1), the raw output of the model, which also indicates the expectation that the current value falls in each bin.

These plots show how both models constantly expect the signal to fall to zero, as they saw many of these UCN kicks that unexpectedly drop the signal with no prior indication, but also expect the signal to stay more or less constant over time. The behaviour of the NuPIC model is

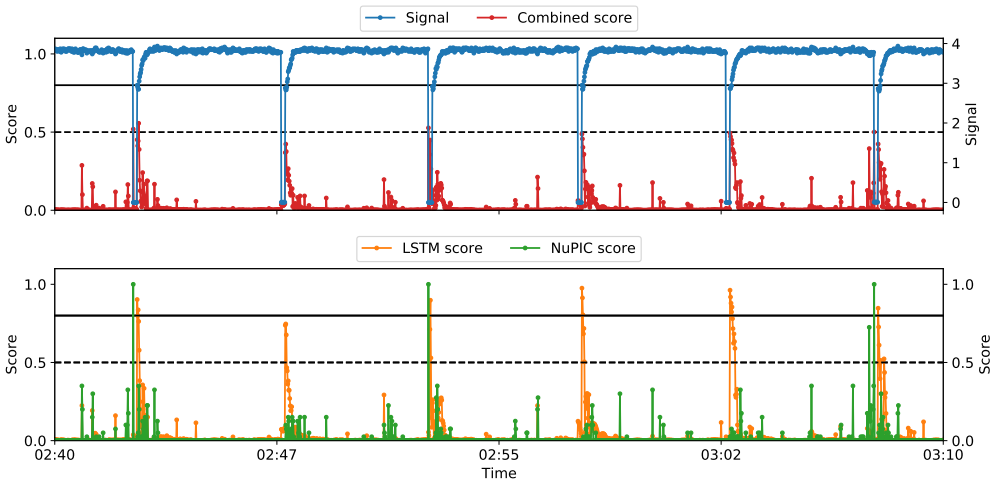


Figure 5: Monitored signal and combined anomaly score (top) and LSTM and NuPIC detectors individual scores (bottom). The combined score is the result of averaging the LSTM score and the NuPIC score.

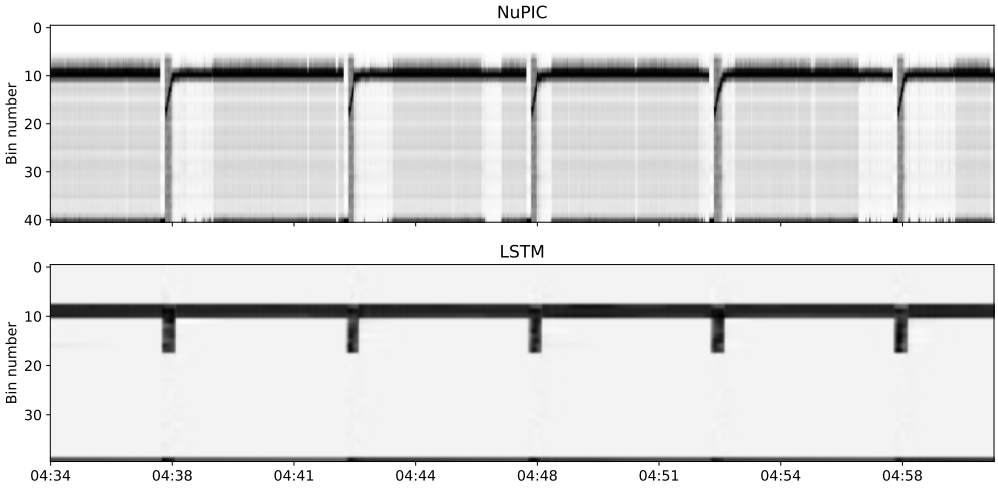


Figure 6: Expectation for the next sample value for the NuPIC model (top) and the LSTM model (bottom). Darker spots represent higher expectation of each model that the next sample will land in that bin and, if the next sample actually lands in that bin, a lower anomaly score.

more complex than the LSTM model, and captures better contextual information. For example, it never expects the signal to drop to zero right after an UCN kick as these never happen right after each other. The LSTM model is more conservative and just learned the broad behaviour of the signal.

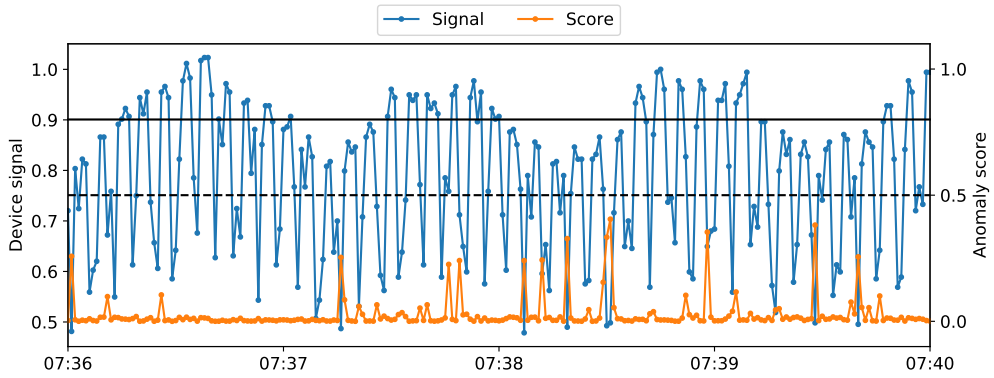
Figure 7a shows the signal and ensemble anomaly score for another Blende loss monitor (MHB34R). This signal has a very clear slow frequency of about one minute and a faster one of a few seconds. The expectation plot, shown in Fig. 7b, demonstrates how the NuPIC detector captures, at least during some periods, the periodic nature of the signal, whilst the LSTM detector treats the signal as purely stochastic over its amplitude.

Figure 8 shows the MHB7R signal and ensemble anomaly score for a period of time when the kicker is not active. The large anomaly score just after 7:00 is an unexplained sudden drop of the device signal that produces a sudden spike in the anomaly score and therefore is detected as an anomaly.

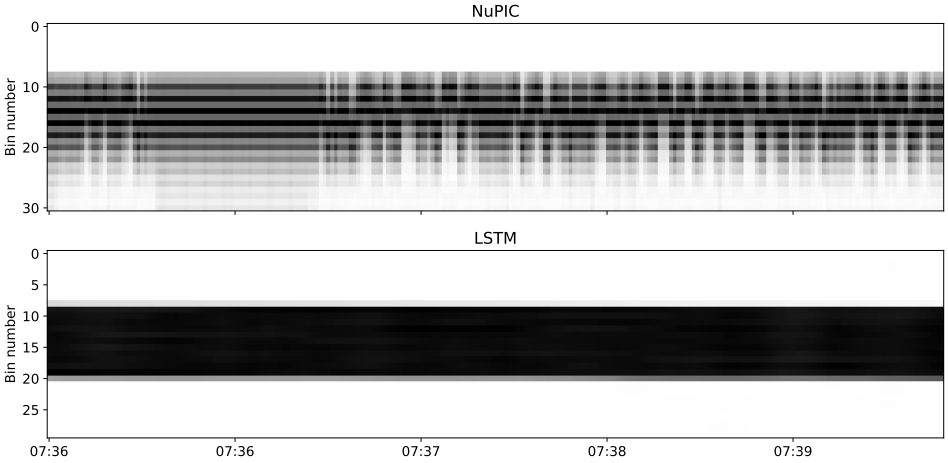
In Fig. 9 another example of this device signal and ensemble anomaly score are shown. At approximately 10:20 the device failed and did not produce any further readings, defaulting to zero value. During this event the beam was going through the device normally with stable current.

The spikes in the anomaly score during the period leading to this breakdown, just after 09:50 and just before 10:10, would have alerted the operators of the strange behaviour of the device and preemptive actions could have been taken in advance. Also, the spike at 10:20 indicates the moment the device stopped measuring and therefore would have alerted the operators in the moment of the breakdown as well.

To asses how many false-positives the model produced during the analysis of this device, the reported anomalies can be manually checked. During the full day of its failure the anomaly detection model, ignoring the first 2 hours of the day as learning phase, reported in total 135 anomalous samples. However, many of theses anomalies are very close together, linked to the same anomalous behaviour. In order to account for this and manually check for false positives, we grouped the scores in 30 second windows counting only one anomaly in each window. Grouping like this, 61 anomalies are detected. One type of anomaly that is often reported is caused by



(a) Signal and anomaly score of the MHB34R loss monitor in the SINQ line.



(b) NuPIC model (top) and LSTM model (bottom) next sample expectation for the same signal.

Figure 7: Signal, anomaly core and models expectation for the MHB34R Blende loss monitor.

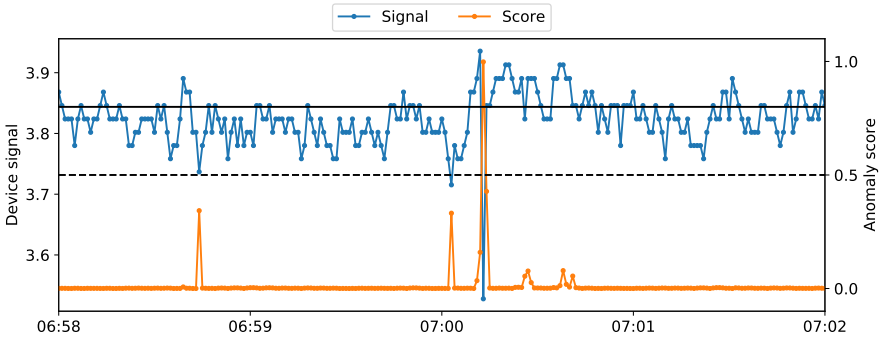


Figure 8: Isolated anomaly detected in the MHB7R loss monitor signal while HIPA was operating with stable current.

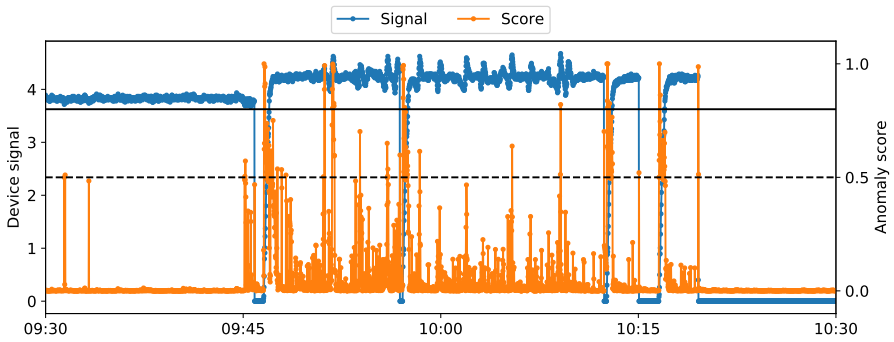


Figure 9: Signal of the MHB7R loss monitor blade and anomaly score calculated for it. The beam current was behaving normally in the period shown here. At the 10:20 mark, the device stopped working completely.

beam interruptions triggered to protect the machine from damage, this type of false-positives are unavoidable as these events are quite rare (about once per hour). However, these are easy to ignore either manually or automatically. Ignoring the anomalies linked to the 35 beam interruptions that happened in this period, results in 26 anomalous samples out of 79200 samples (about 0.03%). Again, many of these anomalies happened due to manual changes of operation (like starting and stopping the UCN kicker system) and can also easily be discarded, by letting the system readapt to the new normality. At the end, the precursors of the failure of the device shown in Fig. 9, some clearly human recognisable anomalies in the signal like the one shown in Fig. 8 and one false positive at an UCN kick were reported by the model.

These results were achieved while performing very little tuning of the two models. The HTM algorithm has a large number of hyperparameters, however the NuPIC package provides a selection of settings that work well in most of the cases, and in these experiments the defaults were used in all cases. The LSTM model network topology was designed after other models found in literature and no significant difference was found between small variations of this architecture. The only signal specific tuning performed was to set the maximum and minimum ranges, the precision of the device, which affects the sensitivity of the models to small changes in the signal and the time-scale parameters of both algorithms, which controls how fast the model forgets previous values. All these parameters are typically well known by the human experts who operate the devices or can be statistically obtained.

6. Conclusions and future work

The anomaly detection model presented in this paper is a combination of two models of very different nature: NuPIC, a HTM model implementation and a LSTM autoregressor. This allows to exploit the natural time sparsity of the anomalies as the two models are unlikely to agree on a false positive, leading to a more robust anomaly detection model than each of the individual models.

A full exploitation system has been presented, where the signals are acquired from EPICS, the control system used at PSI, realigned to a fixed frequency and sent to the anomaly detection models. The results, together with the input signals and diagnostics information, are stored in a database that can be monitored and presented to the user, either in a live setting as soon as the results are available, or offline for historical analysis.

We showed how this system has been used to monitor the MHB7R loss monitor during a full day when a failure of the device happened and went unnoticed by human experts for hours. The model would have alerted the experts of the failure of the device tens of minutes in advance allowing for a more detailed investigation, preemptive maintenance of the device or to take any necessary corrective action. This event was detected while the model only reported a single false positive during the full day, apart from the first 2 hours of learning period, ignoring beam interruption related anomalies and operator changes of machine configuration.

Therefore, we demonstrated how this type of model can be a very useful tool to monitor system critical beam instrumentation, even when these signals exhibit a complex behaviour hard to evaluate using classical statistical methods.

Still, the model presented here has potential to be even more effective by adding additional individual anomaly detection models of other natures, for example, streaming tree-based models [20] or simpler statistical models [21].

Additionally, both of the models presented here are used to monitor individual signals, but are capable of taking an arbitrary number of signals as input. By passing several signals at the same time to the anomaly detection models, these can find unusual behaviour not only in each individual signal, but unusual correlations between them, expanding their detection power.

However, this increases the complexity of the problem too, and much more careful tuning of the model parameters and signal selection is necessary.

Therefore, we consider that the model presented here strikes a good balance of complexity vs. effectiveness, and can be extremely useful tool to improve the maintenance efficiency of particle accelerators.

Acknowledgements

We would like to thank all the members of the Particle Accelerator and Machine Learning (PACMAN) project, funded by the Swiss Data Science Center (SDSC), for their support and great input on the contents of this paper.

References

[1] M. Rezapour, Anomaly detection using unsupervised methods: Credit card fraud case study, International Journal of Advanced Computer Science and Applications 10 (11) (2019). doi:10.14569/IJACSA.2019.0101101.
URL http://dx.doi.org/10.14569/IJACSA.2019.0101101

[2] V. Ceronmani Sharmila, K. K. R., S. R., S. D., H. R., Credit card fraud detection using anomaly techniques, in: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1–6. doi:10.1109/ICIICT1.2019.8741421.

[3] K. Wang, S. J. Stolfo, Anomalous payload-based network intrusion detection, in: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 203–222.

[4] I. Khalkhali, R. Azmi, M. Azimpour-Kivi, M. Khansari, Host-based web anomaly intrusion detection system, an artificial immune system approach, International Journal of Computer Science Issues 8 (09 2011).

[5] E. Fol, R. Tomás, J. Coello de Portugal, G. Franchetti, Detection of faulty beam position monitors using unsupervised learning, Phys. Rev. Accel. Beams 23 (2020) 102805. doi:10.1103/PhysRevAccelBeams.23.102805.
URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.23.102805

[6] M. Wielgosz, M. Mertik, A. Skoczeń, E. De Matteis, The model of an anomaly detector for HiLumi LHC magnets based on recurrent neural networks and adaptive quantization, Engineering Applications of Artificial Intelligence 74 (2018) 166–185. doi:https://doi.org/10.1016/j.engappai.2018.06.012.
URL https://www.sciencedirect.com/science/article/pii/S095219761830143X

[7] M. Piekarski, W. T. Kitka, J. Jaworek-Korjakowska, Deep neural network for anomaly detection in accelerators, in: Proc. 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19), JACoW Publishing, 2019, p. 1379.

[8] T. D'Ottavio, P. S. Dyer, J. Piacentino, M. R. Tomko, Experience using NuPIC to detect anomalies in controls data, in: Proc. 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19), JACoW Publishing, 2019, p. 1619.

[9] S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134 – 147, online Real-Time Learning Strategies for Data Streams. doi:https://doi.org/10.1016/j.neucom.2017.04.070.
URL http://www.sciencedirect.com/science/article/pii/S0925231217309864

[10] J. Hawkins, S. Ahmad, Why neurons have thousands of synapses, a theory of sequence memory in neocortex, Frontiers in Neural Circuits 10 (2016) 23. doi:10.3389/fncir.2016.00023.
URL https://www.frontiersin.org/article/10.3389/fncir.2016.00023

[11] M. Taylor, S. Purdy, breznak, C. Surpur, A. Marshall, D. Ragazzi, S. Ahmad, numenta-ci, A. Malta, P. C. Weinberger, Akhila, M. Lewis, R. Crowder, M. L. Borgne, Yuwei, C. Simons, R. J. McCall, L. Scheinkman, M. Eric, U. Song, keithcom, N. Romano, S. Bolliger, vitaly-krugl, J. Bridgewater, I. Danforth, J. Weiss, T. Silver, D. Ray, zuhaagha, numenta/nupic: 1.0.5 (Jun. 2018). doi:10.5281/zenodo.1257382.
URL https://doi.org/10.5281/zenodo.1257382

[12] Python XMLRPC server and client modules, https://docs.python.org/3/library/xmlrpc.html, [Online; accessed 5-May-2021].

[13] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. arXiv:https://doi.org/10.1162/neco.1997.9.8.1735, doi:10.1162/neco.1997.9.8.1735.
URL https://doi.org/10.1162/neco.1997.9.8.1735

[14] A. Graves, A. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks (2013). arXiv:1303.5778.

[15] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation (2016). arXiv:1609.08144.

[16] M. Said Elsayed, N.-A. Le-Khac, S. Dev, A. D. Jurcut, Network anomaly detection using LSTM based autoencoder, in: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 37–45. doi:10.1145/3416013.3426457.
URL https://doi.org/10.1145/3416013.3426457

[17] D. Reggiani, B. Blau, R. Dölling, P. A. Duperrex, D. Kiselev, V. Talanov, J. Welte, M. Wohlmuther, Improving beam simulations as well as machine and target protection in the SINQ beam line at PSI-HIPA, Journal of Neutron Research 22 (2-3) (2020) 1–11.

[18] L. R. Dalesio, A. Kozubal, M. Kraimer, Epics architecture, Tech. rep., Los Alamos National Lab., NM (United States) (1991).

[19] M. Daum, P. Duperrex, G. Dzieglewski, U. Frei, T. Korhonen, A. Mezger, U. Müller, D. Reggiani, A macro-pulsed 1.2 MW proton beam for the PSI ultra cold neutron source, in: Proceedings of PAC09, 2009, pp. 1748–1750.

[20] S. Guha, N. Mishra, G. Roy, O. Schrijvers, Robust random cut forest based anomaly detection on streams, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 2712–2721.

[21] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.