ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST043

A NOVEL METHOD FOR DETECTING UNIDENTIFIED FALLING OBJECT LOSS PATTERNS IN THE LHC

L. Coyle*, CERN, Geneva, Switzerland
A. Lechner, D. Mirarchi, M. Solfaroli Camillocci, J. Wenninger, CERN, Geneva, Switzerland
F. Blanc, D. Di Croce, T. Pieloni, EPFL, Lausanne, Switzerland

Abstract

Understanding and mitigating particle losses in the Large Hadron Collider (LHC) is essential for both machine safety and efficient operation. Abnormal loss distributions are telltale signs of abnormal beam behaviour or incorrect machine configuration. By leveraging the advancements made in the field of Machine Learning, a novel data-driven method of detecting anomalous loss distributions during machine operation has been developed. A neural network anomaly detection model was trained to detect Unidentified Falling Object events using stable beam, Beam Loss Monitor (BLM) data acquired during the operation of the LHC. Data-driven models, such as the one presented, could lead to significant improvements in the autonomous labelling of abnormal loss distributions, ultimately bolstering the ever ongoing effort toward improving the understanding and mitigation of these events.

INTRODUCTION

To monitor particle losses, the LHC is equipped with over 3000 BLMs placed along the circumference of the machine, see Fig. 1. This vast BLM array provides a very detailed account of the amount, the location and time evolution of particle losses occurring in the LHC at any given moment. The LHC BLM system provides data on many different time scales called Running Sum (RS), ranging from 40 μs to 1.9 s.

Figure 1: BLMs on the LHC

The spatial distribution of particle losses across the LHC is referred to as a loss map. These loss maps provide key information to identify loss mechanisms, but also to ensure a proper alignment of the LHC collimators. They are monitored throughout machine operation and are heavily relied upon throughout the commissioning phase for the setup of the machine protection related accelerator components.

This paper will focus on a specific type of particle loss event referred to as an Unidentified Falling Object (UFO) [1, 2].

Unidentified Falling Objects

UFOs are a very fast and localized loss events caused by micrometer sized dust particles which interact with the particle beams. UFOs have been the cause of many beam dumps in the past and as such, a detection algorithm named UFO-Buster was developed to monitor these events, including those that remain below the dump threshold of the BLM system [1, 3]. A loss map with a UFO event is presented in Fig. 2.

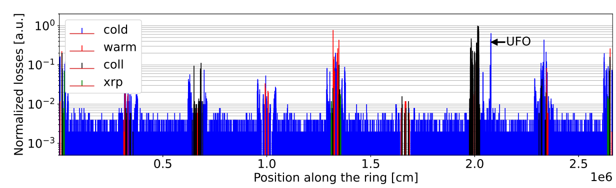


Figure 2: An example loss map with a UFO event indicated, fill 6648 on 2018-05-06 at 20:01:23

UFO BUSTER

Algorithm Description

The UFO-Buster continually monitors the BLM signals and when a series of criteria are met, the event is labeled as UFO candidate and is added to a UFO event database. The main criteria is that 2 BLMs within a distance of 40 m must exceed a dose rate threshold, usually set to $1{\times}10^{-4} Gy/s$, and that the event duration must be on the millisecond timescale, see [1] for details.

UFO Dataset

The UFO-Buster's UFO assignments are used to create a dataset of known UFO events on which to train the model in a supervised manner. For this study, only UFOs occurring in the arc sections of the LHC, during the stable beam mode of physics runs in 2018 were considered. In total 744 UFO events were used with each of these UFO containing loss maps consisting of 3595 individual BLMs, resulting in an initial dataset of shape 744×3595.

MACHINE LEARNING MODELLING

The objective is to create a machine learning model capable of identifying UFOs within operational loss maps, using

* loic.thomas.coyle@cern.ch

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST043

the UFO-Buster dataset. Two different models were trained, a binary classification and a regression model. The distinction between these models lies in the format of their output data, the regression model will be trained on values which indicate how centered the input is on a UFO, whereas the classification model will be provided a binary output, i.e. whether the input contains a UFO or not.

Preprocessing

Before training the model, a significant amount of preprocessing is performed.

Rolling Window Firstly, a rolling window transformation is applied with a window of the typical size of a UFO induced particle shower. A window size of 33 BLMs was chosen. After this transformation we obtain a dataset of shape 2674680×33.

Augmentation In order to help generalize the model, the samples which contain UFOs were augmented by adding their mirrored representations to the dataset. This, in essence, adds the other beam's response to the dataset.

Balancing In order to help with the learning process, it generally is advised to have a relatively balanced dataset. In our case, there are naturally more windows which don't contain UFOs than windows which do. For the regression model, balancing was performed by duplicating the under-represented UFO windows. Whereas for the classification model, sample weighting was used.

Normalization Each window feature of the dataset is normalized so as to obtain a null mean and a standard deviation of 1.

With x_i the data contained in the i^{th} column of the dataset, $\forall i \in [0 \,.\,.\, n_{columns} - 1]$

$$x_{i,normed} = \frac{x_i - \bar{x}_i}{std(x_i)} \tag{1}$$

Label Assignment For the regression model, the labels are assigned following a triangle distribution centered on the UFO with the same width as the rolling window. Resulting in each loss map window being assigned a label based on how centered it is on the UFO, the window perfectly centered on the UFO is assigned the label 1 and decays to zero as the window slides away from the known UFO location.

For the classification model, the labels are easier to compute. Simply, any window which overlaps with the known location of the UFO is labelled as 1 and any which don't are labelled as 0.

See Fig. 3 for the window labels of an example UFO for both models.

Model Description

For both models a 2 layered Convolutional Neural Network (CNN) [6] model was used. The networks were devoloped using the TensorFlow libraries [4, 5]. Each layer is

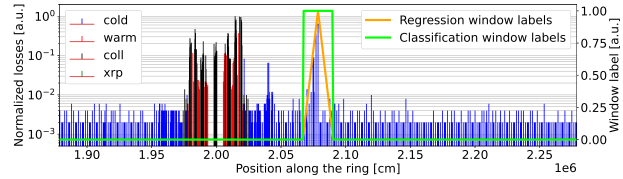
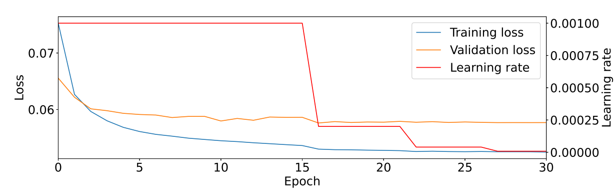


Figure 3: The BLM signals close a UFO event along with the computed window label

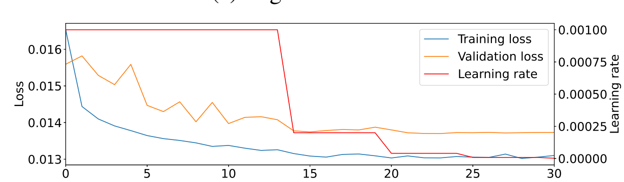
made up of a convolutional and a max pooling step. The convolutional layer learns a multitude of filters which convolve over the spatial dimension of the input data. By stacking multiple convolutional layers it is possible to learn more complex patterns in the input data. The max pooling layer is a down-sampling method which is typically paired with convolutional layers. Max pooling layers keep only the maximal value of non-overlapping subregions of the layer's input. This transformation reduces the complexity of the model, reduces overfitting and provides the model with some translation invariance properties. For both models, the hyperparameters of the convolutional layers were optimized using random search. The regression model's loss function used is Mean Absolute Error and the classification model's loss function is Binary Crossentropy [8]. Both models use the Adam optimizer [7].

Training

The dataset is split into training, validation and testing with ratios 0.6, 0.2, 0.2 respectively. The learning rates of the models are decreased during training and the training is stopped once their loss metrics have converged. The loss functions during training are shown in Fig. 4.



(a) Regression model



(b) Classification model

Figure 4: Training and validation loss functions during training along with the decreasing learning rate in red for both models.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST043

Evaluation and Tuning

Once the models are trained, when provided an input, they output a scalar value which we can interpret as the models' confidence that the provided input contains a UFO event.

The models' output across an entire loss map containing a UFO event is shown in Fig. 5. The models' output is maximal and reaches 1 at the location of the UFO and is lower for nominal BLM signals, i.e. BLM signals of regular losses. For both models, we can also observe some lower peaks throughout the loss map, however the classification model is drastically more noisy.

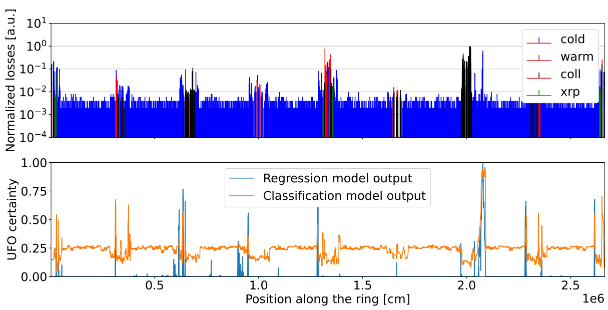


Figure 5: A loss map containing a UFO event (fill 6648 on 2018-05-06 at 20:01:23) along with both models' output across the full loss map.

We are left with the task of determining the optimal sensitivity threshold above which an input sample is said to contain a UFO and below which it does not.

Sensitivity Tuning The trained models' predictions on the test dataset were used to tune the sensitivity. The threshold value was scanned and various performance metrics were used to quantify the models' performances.

Since the regression model has a better performance, i.e. higher accuracy, lower false positive rate and comparable false negative rates than the classification model, as it is possible to verify in Fig. 6, we will focus the last part of this study on the regression model.

A couple working points were considered. Taking the working point with a threshold at 0.40 reduces the false negative rate while maintaining a reasonable false positive rate. However, using a higher threshold such as 0.85 reduces further the false positive rate at the expense of the false negative rate, see Table 1 for specific values.

Table 1: Performance of the Regression Model for 2 Working Points

Sensitivity	False neg. rate	False pos. rate	Accuracy
0.40	0.0134	0.0046	0.9953
0.85	0.0201	0.0001	0.9998

For our application, minimizing the false negative rate is the best approach since a high identification probability is

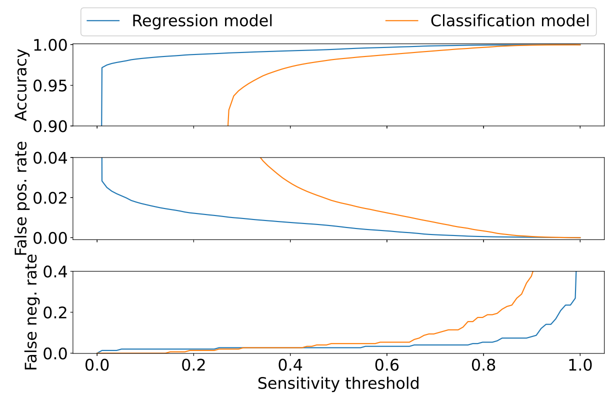


Figure 6: Various performance metrics computed while scanning the sensitivity threshold for both models.

desired to capture as many UFOs as possible, as such the lower sensitivity threshold of 0.40 was chosen.

Width threshold tuning In addition to the sensitivity threshold a second tunable parameter was added to remove some noise in the model's prediction. As UFOs produce highly localized losses, multiple BLMs in close proximity produce high signals when a UFO occurs. To make use of this property, a threshold on the width of consecutive positive assignments was introduced and tuned using a similar approach as the sensitivity threshold. The effect of both thresholds on the final UFO assignment is shown in Fig. 7.

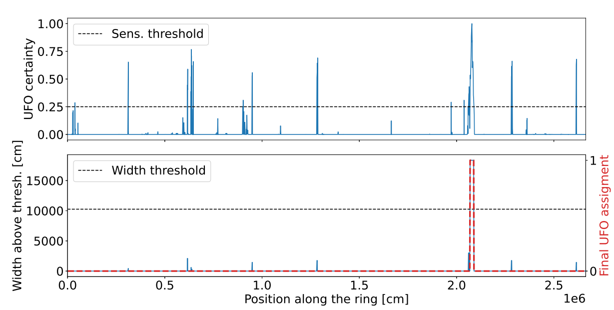


Figure 7: Effect of the thresholds on the final UFO assignment (fill 6648 on 2018-05-06 at 20:01:23).

Using both thresholds the model is able to capture 100% of the UFOs while discarding 98% of the non UFO data.

CONCLUSION

A procedure for utilizing the UFO-Buster's UFO assignments to train a machine learning UFO detection model was developed. Two types of CNN models were trained, tuned, evaluated and compared, of which, the regression based model performed best. Work is ongoing towards further improving the models by optimizing the data preprocessing methodology and increasing the amount of data on which the model is trained through synthetic data generation.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST043

REFERENCES

[1] T. Baer, "Very Fast Losses of the Circulating LHC Beam, their Mitigation and Machine Protection", CERN-THESIS-2013-233.

[2] T. Baer *et al.*, "UFOs in the LHC: Observations, studies, and extrapolations", in *Proc. of the 3rd International Particle Accelerator Conf.*, New Orleans, LA, 2012, THPPP086, pp. 3936–3938.

[3] B. Auchmann et al., *Testing beam-induced quench levels of LHC superconducting magnets*, *Phys. Rev. Accel. Beams*, vol. 18, p. 061002, 2015.

[4] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", 2015. https://doi.org/10.48550/arXiv.1603.04467

[5] https://tensorflow.org

[6] LeCun, Y. *et al.*, "Handwritten digit recognition with a back-propagation network", in *Proc. Advances in Neural Information Processing Systems*, pp. 396–404, 1990.

[7] Diederik P. Kingma *et al.*, "Adam: A Method for Stochastic Optimization", arXiv, 2017. doi:10.48550/arXiv.1412.6980

[8] Good, I. J. *Rational Decisions* Journal of the Royal Statistical Society. Series B (Methodological), vol. 14, no. 1, pp. 107–114, 1952.