Machine learning based long-lived particle reconstruction algorithm for the LHCb experiment

Adam Dendek
AGH University of Science and Technology
Krakow, Poland
adendek@agh.edu.pl

Tomasz Szumlak
AGH University of Science and Technology
Krakow, Poland
szumlak@agh.edu.pl

Abstract

The LHCb collaboration has been pioneering the employment of machine learning in real-time computing, related to the high energy physics field (HEP), since 2015. The LHCb software trigger exploited a novel machine learning techniques helped to select high-quality physics data in real-time. One of such algorithms is dedicated to reconstructing long-lived particles. In this paper, we present a model that was commissioned and successfully operated during the second data-taking period of the LHCb experiment. The second part of the paper gives a short report on the first application of two selected methods enhancing trust in the model predictions for the LHCb experiment.

1 Introduction

The LHCb (Large Hadron Collider beauty) experiment [1] is one of the four big experiments instrumenting the LHC accelerator, currently operating at CERN. It is dedicated to the indirect search of New Physics beyond the Standard model via the study of particles containing charm and beauty quarks. During Run 1, the LHCb physics program was extended to electroweak, soft QCD, and even heavy-ion physics. The results from LHCb based on data collected during the LHC Run 1 have proven that measurements of excellent quality can be made in the heavy flavor sector. However, the LHCb has not recorded any phenomena beyond the prediction of the Standard Model. To enhance the discovery potential of the LHCb, the collaboration decided to investigate machine learning techniques. Several models were built, tested, and deployed to increase the amount and quality of collected significant, from the physics analysis point of view data.

One such project was the reconstruction of, so-called, Downstream Tracks. The Downstream tracks are created by the daughters of long-lived particles, such as K_s^0 or Λ. Those particles decay outside of the LHCb vertex detector (VELO). Thus associated tracks contain hits collected in two tracking detectors. The first of them, called TT, is located just before the bending magnet, and the second one right after it (T stations).

2 Reconstruction of the Downstream Tracks

Downstream tracks at LHCb are reconstructed in the following way [2]. First, a standalone track reconstruction in the T stations with a dedicated algorithm is performed, creating T tracks also called T-Seeds. After filtering with a machine learning classifier to discard bad candidates, i.e., tracks that do not represent the trajectory of a real particle, these tracks are propagated back through the magnet to the TT stations. After that, the TT clusters are matched to the predicted trajectory, and a Downstream Track candidate is formed. The best track candidates are then being selected by another machine learning classifier.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

The search for the best classifier was performed using a simulated data sample containing signal $B \to J/\Psi K_s^0$ events. Any reconstructed track segment associated with a valid Monte Carlo particle not being an electron was considered a true seed. A critical step in the pre-processing of input data was balancing the composition of the training sample, which should have an approximately equal number of true and fake T tracks. In our case, both training and validation samples contained about 40% of actual tracks and were prepared using the undersampling technique [3].

The model selects the T tracks based on the XGboost[4] library. To get the best possible classification outcome, the model hyperparameters were optimized using the Bayesian Optimization method with the area under the ROC curve as an objective function.

Since the tracking algorithm is a part of the real-time LHCb High-Level Trigger system (HLT), both the execution time and memory footprint are important, so using the full continuous classifier is not an option. Instead, a binned xgboost (called bonsai Boosted Decision Tree - bBDT) classifier that meets the speed and memory criteria of the HLT is used. The detailed description of how the BDT is binned can be found in [5].

The ROC curve of the binned BDT is presented in figure 1. A slight drop in the performance of the binned classifier with respect to the full one has been noted. The figures of merit measured for the bBDT algorithm amounted to 87%. This allows the reduction of about 30% fake T tracks.

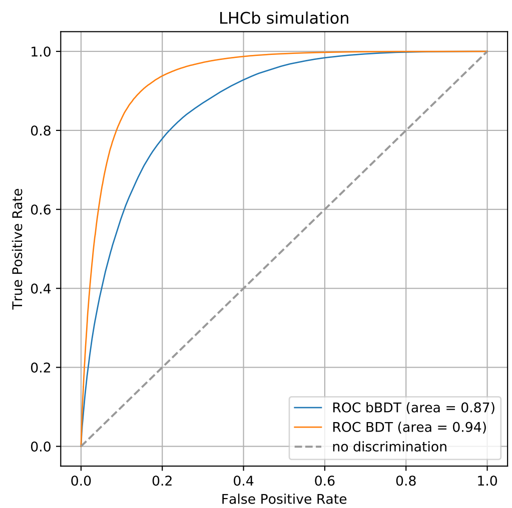

Figure 1: Comparison of ROC curves for selecting true T tracks using a simulated $B \to J/\Psi K_s^0$ sample before and after binning.

3 Understanding model's prediction

One of the crucial problems when building a complicated Machine Learning model is a lack of interpretability of its prediction. This fact raises the question of why the researcher should trust the model. The complex model interpretability is an active research area, and many ideas were recently published [6]. To make sure that the model described in section 2, provides reliable predictions two methods, were proposed: SHAP (Shapley Additive exPlanations) [7] and LIME (Local Interpretable Model-Agnostic Explanations) [8].

3.1 SHAP Shapley Additive exPlanations

Shap is a method that combines game theory and Machine Learning. It is based on a Shapley value. This quantity was introduced in the late '50s [9] as one of the essential concepts in the coalitional games.

The coalitional game is such a game where there are n players, and each of which may or may not participate in the coalition. Generally, contribution to the coalition of each player is not equal; some of them provide more resources than the others. The Shapley value is the solution to the problem of fair share the coalition payoffs ϕ:

$$\phi_i^{shapley}(v) = \sum_{S \subseteq N, s=|S|} \frac{(n-s-1)!s!}{n!}(v(S \cup \{i\}) - v(S))) \tag{1}$$

where: ϕ_i is a payoff that player i should receive, which can be interpreted as an importance of a particular feature i, $n = |N|$ is a maximal number of features that can be used to train a classifier, $v(S)$ is a characteristic function, that describes worth of coalition S, in the context of interpretation of the machine learning model the $V(S)$ is a performance of classifier that were trained using subset of features S. The idea of measuring the importance of each coalition player can be leveraged as a tool to get a better understanding of the importance of the features. In such a case, each of the model's features is represented as an individual player, and the Shapley value can be interpreted as relative feature importance.

The Shap summary plot was introduced to visualize individualized feature attribution to the model's decision. In order to make that kind of plot, the features are sorted by their global impact $\sum_{j=1}^{N} |\phi_i^{(j)}|$, then each of the dots corresponding to the Shap values $\phi_i^{(j)}$ are plotted horizontally. This concept allows achieving a similar effect to so-called violin plots [10], which can be further enhanced by coloring the dots according to the feature's value.

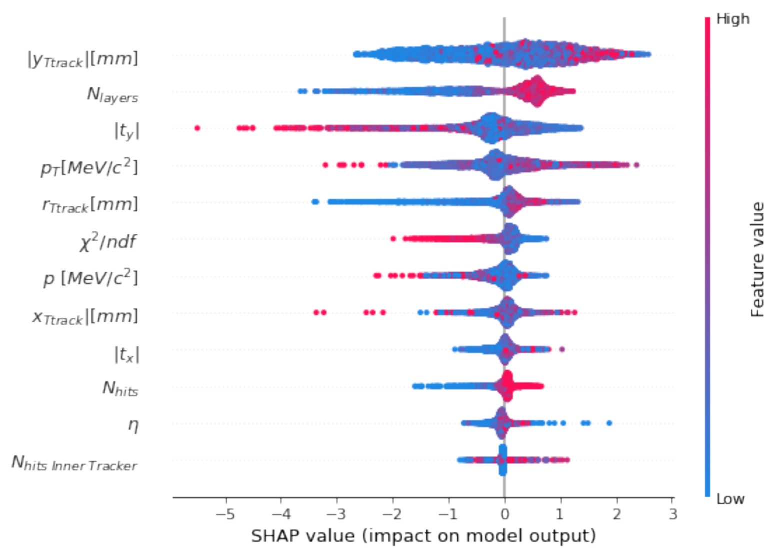

Figure 2: Shap summary plots of all T-Seed classifier features. The plots were obtained using a sample of 50 000 test examples. The interpretation of this plot is straightforward; the higher the Shap value of the feature, the higher the influence on the decision of whether T-Seed is reconstructable. Each dot represents feature importance for every single test example that was run through the model. The feature's value colors dots (red high, blue low). This allows deducing that N_{layers} is one of the most important feature and its small value drives the classifier toward **false** decision.

3.2 LIME Local Interpretable Model-Agnostic Explanations

Another approach to enhance the understanding of the model's prediction is based on two concepts. The first one is a local interpretation, and the second is the concept of a class of the models that are interpretable by the human. An interpretable explanation needs to use a representation that is understandable to humans, regardless of the actual model's architectures or features used to make a decision. For instance, a possible interpretable representation of the text classification is a vector of binary values indicating the presence or absence of a certain word, even though the model may use more complex input features. The locality of the interpretation means that the algorithm analyses the model prediction on a single test example z. Conceptually LIME approximation is very close to the Taylor series approximation, which infinitely differentiable function transforms to a power series around the specific point x_0. The usual choice for interpretable representation $g(x)$ is a linear model, where feature importance is represented by the absolute value of the corresponding weight.

The explanation $g^*(x)$ produced by LIME is obtained by finding the minimum of the following formula:

$$g^*(x) = \underset{g \in G}{\operatorname{argmin}} \, \mathcal{L}(f, g, \pi_x(z)) + \Omega(g) \tag{2}$$

Where: $\mathcal{L}(f, g, \pi_x(z))$ is a loss, which measures how faithful the local approximation g is, the $\pi_x(z) = exp(-D(x, z)^2/\sigma)$ is a proximity measure between an instance z and other test examples implemented as an exponential kernel over some distance function D, which is used to define a local neighborhood of instance z, finally the $\Omega(g)$ is a regularization factor which depends on the complexity of a model g. Tern $\mathcal{L}(f, g, \pi_x(z))$ ensures local fidelity of interpretation while $\Omega(g)$, typically implemented as a ridge loss allows producing sparse linear models as an explanation. The exemplary explanation result is presented in figure 3.

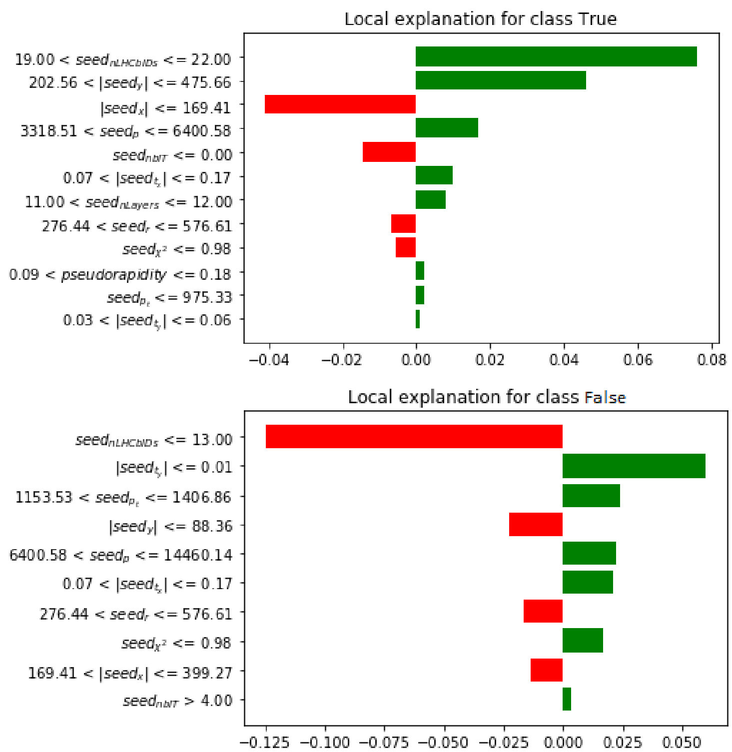

Figure 3: Exemplary outcome of the LIME analysis for two randomly selected examples (true T-seed upper sub-figure and ghost T-seed the bottom one). Those samples were drawn from the test dataset, the one that was never used to train the model. The red color indicates that the features pushed prediction toward fake.

4 Conclusions

The improvement of the track reconstruction is a vital step toward observation of the New Physics. This can be achieved via the application of the novel machine and deep learning models. This paper presents one such project, which was deployed as a part of the LHCb real-time trigger system during LHC Run 2.

We also propose two methods, which allowed us to get a better intuition on how the model treats the input data and why it makes a particular classification decision. We encourage other physicists working on machine learning models to perform a similar analysis to enhance their understanding of the models.

Broader Impact

We believe that this work does not have future societal or ethical consequences.

References

[1] The LHCb Collaboration. The LHCb detector at the LHC. *Journal of Instrumentation*, 3(08):S08005–S08005, aug 2008.

[2] Adam Davis et al. PatLongLivedTracking: a tracking algorithm for the reconstruction of the daughters of long-lived particles in LHCb. Technical Report LHCb-PUB-2017-001. CERN-LHCb-PUB-2017-001, CERN, Geneva, Jan 2017.

[3] Julio Hernandez et al. An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 262–269, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[4] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[5] V V Gligorov and M Williams. Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree. *Journal of Instrumentation*, 8(02):P02013–P02013, feb 2013.

[6] Sara Hooker et al. A benchmark for interpretability methods in deep neural networks, 2018.

[7] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[8] Marco Tulio Ribeiro et al. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[9] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[10] Jerry L. Hintze et al. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.