Machine Learning for Muon Identification at LHCb

N Kazeev\textsuperscript{1,2,3,4} on behalf of the LHCb collaboration

\textsuperscript{1} Laboratory of Methods for Big Data Analysis, National Research University Higher School of Economics, 3 Kochnovsky Proezd, Moscow 125319, Russia
\textsuperscript{2} The Yandex School of Data Analysis, 11/2 Timura Frunze St., Moscow 119021, Russia
\textsuperscript{3} Sapienza University of Rome, Piazzale Aldo Moro, 5, 00185 Roma RM, Italy
\textsuperscript{4} INFN – Laboratori Nazionali di Frascati, Via Enrico Fermi, 40, 00044 Frascati RM, Italy
E-mail: nikita.kazeev@cern.ch

Abstract. Particle identification is a key ingredient of most of LHCb results. Muon identification in particular is used at every stage of the LHCb trigger. The objective of the muon identification is to distinguish muons from charged hadrons under strict timing constraints. For this task, we use a state-of-the-art gradient boosting algorithm trained with real background-subtracted data. In this proceedings we present the algorithm along with the evaluation of its performance on signal and background rejection.

1. Introduction

Particle identification is instrumental for most physics goals of the LHCb experiment. Muons in particular are present in the final states of many decays sensitive to new physics \cite{1, 2, 3}. The Muon subsystem offers a fast way to select events that contain muons in their final state without reconstructing the whole event. A brief description of the hardware is in section 2, and a detailed one is in \cite{4}. The idea of all muon identification (MuID) algorithms based on the muon subsystem is to check whether the muon chambers contain hits matching the extrapolation of the track in question. The two primary sources of misidentification are combinatorics, where unrelated hits are by chance aligned with a track, and pions decaying in flight into genuine muons which then reach the muon chambers.

The big challenge in the planned five-fold luminosity increase for the LHC Run III (2021–2023) mandates the development of new MuID algorithms to cope with the higher occupancy \cite{5}.

In this paper we investigate the possibility of using the latest Machine Learning techniques to combine the low-level hit information with the output of high-level expert algorithms to achieve the best possible quality while respecting the execution time budget. We describe the current MuID algorithms in section 3 and present the novel ones in 4.

2. LHCb Muon subsystem

The physical idea behind the muon identification is to draw upon the unique penetrating power of muons – if a charged particle is able to pass through the calorimeter and iron absorbers, it is likely a muon. The Muon subdetector consists of 5 sensitive rectangular planes separated
by iron absorbers. The first station, M1, is not used for the offline reconstruction but plays an important role in the hardware trigger (L0). Each station is separated into regions with different granularity according to the particle density in order to keep the occupancy roughly constant. Upgrade will see M1 removed and some additional shielding installed around the beam pipe [6].

3. Current implementation of Muon ID

3.1. IsMuon

The first step is “IsMuon” – a very fast yet discriminating algorithm. For Run I (2009–2013) it kept muon efficiency in the range of 95–98% and background rejection to the level of 99% [1]. Given a reconstructed track in the LHCb tracking system, hits in the Muon stations are searched around the track extrapolation inside a momentum-dependent Field of Interest (FOI). The FOI is determined from the analytical approximation of multiple scattering [7]. If and only if there are hits in enough stations, IsMuon considers the particle a muon.

3.2. muDLL

A likelihood discriminant between muons and combinatorial background is computed as the cumulative probability distribution of the average squared distance significance $D^2$ of the hits in the muon chambers with respect to the linear extrapolation of the tracks from the tracking system [1, 7].

$$D^2 = \frac{1}{N} \sum_{i=0}^{N} \left\{ \left( \frac{x_{\text{closest}} - x_{\text{track}}}{\text{pad}_x} \right)^2 + \left( \frac{y_{\text{closest}} - y_{\text{track}}}{\text{pad}_y} \right)^2 \right\},$$  

(1)

where $N$ is the number of stations containing hits within the FOI, $\{x, y\}_{\text{closest}}$ are the coordinates of the hit closest to the track extrapolation, $\{x, y\}_{\text{track}}$ are the coordinates of the track extrapolation to the muon stations and pad$_{(x,y)}$ are the muon pad sizes that determine the hit coordinates uncertainties.

4. Proposed Machine Learning-based MuID algorithms

4.1. Run I (2009 – 2013) and Run II (2015 – 2018)

To train the models, we used calibration samples [8] taken from $pp$-collision data collected in 2012, where the particle ID is determined using known decays:

- for muons: $J/\psi \rightarrow \mu^+\mu^-$, $4 \cdot 10^5$ tracks
- for pions: $D^{+} \rightarrow D^{0}(\rightarrow K^-\pi^+)\pi^+$, $2 \cdot 10^5$ tracks

The isMuon selection is already applied to these samples. In the training data, the background sample were weighted to match the signal ($p, p_T$) spectrum. As training features, we used the following information about the closest hits in each station:

- space residuals in $x, y$: 
  $$\frac{x_{\text{closest}} - x_{\text{track}}}{\text{pad}_x/\sqrt{12 + \text{MS}_{\text{error}}}},$$

  where MS$_{\text{error}}$ is the track extrapolation error estimated from multiple scattering

- hit time

- $dT$: difference of the hit time read by the vertical and horizontal strips

- crossing: variable that discriminates between horizontal and vertical strips or hits containing both.
4.2. Towards Run III (2021–2023)

The main challenge for Run III is the five-fold occupancy increase. There is not enough data in the calibration samples with high number of primary vertices (nPVs) to accurately emulate the situation after LHCb Upgrade. We used a weighting that adds more emphasis on high-nPVs events while retaining reasonable statistics. The distributions are shown in figure 2.

We used the following data calibration samples collected in 2016 pp collisions to which the IsMuon selection is applied:

![Figure 1. Muon efficiency versus pion rejection after applying IsMuon selection, no kinematic weighting using 2016 calibration data.](image)

![Figure 2. The Run II data was weighted to put more emphasis on events with multiple primary vertices. Note that there is not enough Run 2 events with large nPVs to match the distribution expected in Run III.](image)
• muons: $J/\psi \rightarrow \mu^+ \mu^-$, $8 \cdot 10^6$ tracks
• pions: $D^+ \rightarrow D^{0+}(\rightarrow K^- \pi^+ )\pi^-$, $4 \cdot 10^5$ tracks
• protons: $\Lambda_0 \rightarrow p\pi^-$, $3 \cdot 10^5$ tracks

The non-muon sample was weighted to match the muon sample $(p, p_T)$ spectrum. Both muon and non-muon samples were weighted by nPVs. The calibration samples are selected from the real data and contain a background contribution. To subtract it, the sPlot [14] method is used. sPlot works by assigning weights to samples and some of the weights are by design negative. Negative weights mean that there is no lower bound on the classification loss, which breaks the theoretical foundation of machine learning algorithms. We used the technique from [15] to overcome the problem and include the sWeights in training.

The set of training features was expanded compared to Run II to include the outputs of versions of two physics-based muon identification algorithms that are still under development: the $\chi^2$ algorithm – an extension of muDLL that takes into account hits correlations – and a hit clustering algorithm [16]. The new features list is:

- Experimental high-level variables: $\chi^2$, cluster sizes, number of clusters
- Information about the closest hits in each station: space residuals, coordinates, hit time, hit delta time, whether the hit is a crossed one
- Same information about the hits matched by the $\chi^2$ algorithm
- Momentum and transverse momentum

![Figure 3](image-url)

**Figure 3.** Muon efficiency versus pion rejection after applying IsMuon selection using 2016 calibration data that is weighted in momentum and nPVs. Dev Catboost is the model trained on data weighted to emphasize the high occupancy events as described in subsection 4.2

The algorithm was evaluated via cross-validation on weighted 2016 calibration data, the results comparing muon efficiency and pion or proton rejection are presented on figures 3 and 4 respectively. As expected, the performance is significantly improved compared to the Run II
Figure 4. Muon efficiency versus proton rejection after applying IsMuon selection using 2016 calibration data that is weighted in momentum and nPVs. Dev Catboost is the model trained on data weighted to emphasize the high occupancy events as described in subsection 4.2.

algorithms. We also trained the same algorithm without using the $\chi^2$ and clustering features: the performance difference was on the border of statistical significance. This means that the feature set is excessive and may be safely reduced when the $\chi^2$ algorithm is finalized.

5. Conclusions
Machine learning allows to gain significant improvement of muon identification with the muon subsystem at LHCb, background rejection at 90% signal efficiency improves from 45% to 79% compared to muDLL on weighted data after IsMuon.

Run II models are integrated into the LHCb software and their results are available for analysis for a part of Run II.

For Run II conditions using CatBoost oblivious trees improves the evaluation speed by a factor of 3 compared to TMVA BDT while giving higher discriminating power.

The results for the data with high number of primary vertices are promising. Yet they don’t account for the full Run III occupancy increase and hardware changes. The algorithm evaluation and, possibly, training will need to be repeated when the Monte-Carlo for the upgraded LHCb detector becomes available.

6. Acknowledgments
The research leading to these results has received funding from Russian Science Foundation under grant agreement n 17-72-20127.

7. References
[1] Archilli F et al. 2013 Journal of Instrumentation 8 P10020
[2] Aaij R et al. 2013 Physical review letters 111 101805
[3] Aaij R et al. 2012 Physical review letters 108 181806
[4] AA Alves Jr and others 2013 Journal of Instrumentation 8 P02022
[5] LHCb Collaboration 2011 Letter of intent for the LHCb upgrade Tech. rep. CERN-LHCC-2011-001
[6] LHCb Collaboration 2013 LHCb PID upgrade technical design report Tech. rep. CERN-LHCC-2013-022
[7] Lanfranchi G et al. 2009 Tech. Rep. LHCb-PUB-2009-013. CERN-LHCb-PUB-2009-013 CERN Geneva
[8] Sarti et al. 2010 Tech. Rep. LHCb-PUB-2010-002, CERN-LHCb-PUB-2010-002
[9] Freund Y and Schapire R E 1997 Journal of computer and system sciences 55 119–139
[10] Hoecker A et al. 2007 arXiv preprint physics/0703039
[11] Prokhorenkova L et al. 2018 Advances in Neural Information Processing Systems pp 6638–6648
[12] URL https://catboost.ai/news/best-in-class-inference-and-a-ton-of-speedups
[13] Aaij R et al. 2019 EPJ Techniques and Instrumentation 6 1
[14] Pivk M and Le Diberder F R 2005 NIMA 555 356–369
[15] Borisyak M and Kazeev N 2019 submitted to ACAT2019 proceedings
[16] LHCb collaboration 2018 Rec project v30r1 https://gitlab.cern.ch/lhcb/Rec/tree/v30r3/Muon/MuonID/src/component