Applying Machine Learning to Particle Track Identification in the L1 Trigger of the CMS Detector

Claire Savard
Department of Physics
University of Colorado, Boulder
Boulder, CO 80303
claire.savard@colorado.edu

Abstract

Around 2025, the Large Hadron Collider (LHC) will be upgrading to the High Luminosity (HL) LHC which will prompt significant hardware and software upgrades in the Compact Muon Solenoid (CMS) detector. Among these updates will be the addition of reconstructed particle tracks within the level 1 (L1) trigger system of the detector. As these L1 tracks may contain a mixture of real and fake tracks, it is important to develop an algorithm to efficiently distinguish between these two possibilities. In this work, I train and test a boosted decision tree and neural network on this supervised classification problem and compare performance to current methods used in the LHC physics field. Initial results indicate that both machine learning algorithms perform better than a current technique.

1 Introduction

The Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN) is perhaps the crowning achievement of modern scientific technology. Sitting in a tunnel 100 m under the Franco-Swiss border, this circular particle accelerator stretches 27 km in circumference and is capable of creating collisions of up to 14 TeV, making it the most powerful particle accelerator in the world. The existence of such an apparatus has allowed for fundamental discoveries such as the Higgs boson [1]. To push its capabilities to new frontiers, however, an upgrade is in order.

Around 2025, the LHC will transition to the High-Luminosity LHC (HL-LHC) which will provide up to 10 times as many collisions as the accelerator's nominal design [2]. The consequent increase in particle collisions further obscures interesting interactions hidden within each event. These events will be too complex for current detector technologies to effectively select events of interest. Therefore, both hardware and software must be upgraded in the detectors located around the LHC ring in order to profit from the physics rewards the HL-LHC can provide.

In the Compact Muon Solenoid (CMS) detector, initial track properties will be reconstructed in the level 1 (L1) trigger system to help out with initial event selection [3,4]. An important part of event selection is finding the *fake tracks*, or those that do not originate from a true particle and are errors from the reconstruction process, versus the *real tracks*, or those tracks that do come from true particles. Analysis groups within the CMS collaboration select ranges–which I will refer to as making a cut–on track properties in their samples to minimize fake tracks and maximize tracks of interest. In this paper, I investigate the application of machine learning in identifying tracks as real or fake and compare these computational methods to a current cut-based approach.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

Table 1: The 14 particle track features given to the machine learning algorithms. A stub is a place where the detector senses a particle track has passed through. The CMS coordinate system has *z* along the beam line, *y* up, and *x* toward the center of the ring.

Feature	Description
ϕ	angle in the *x*-*y* plane
η	$-\ln(\tan(\theta/2))$ where θ is the angle relative to the *z* axis
z_0	*z* position of the track vertex
χ^2/dof	measurement of the track fit per degrees of freedom
χ^2_{bend}	measurement of consistency between p_T and stubs
num. of stubs	number of stubs present in the track
num. of stubs in layers 1-6	number of stubs in each detector layer
num. of layers missed	number of detector layers missed (no stubs present) by the track
stubs in PS module	whether the majority of the stubs are in PS or 2S modules [5]

2 Dataset

The real particles reconstructed in the CMS detector can be further identified by their specific particle type, such as muon or electron. Each LHC analysis group weighs the importance of particles differently, and this weight is dependent on what the group is studying. For example, an analysis group studying the Z boson decay into a positron-electron pair wants to ensure all electron candidates pass the identification filter (are labeled real) as it is a key component in their interaction of interest. In order to create an algorithm that each LHC analysis group will benefit from, it is important that all particle types are correctly identified as real. For this reason, I collected the reconstructed particles into three main group which cover some main areas that LHC physicists are interested in: muons, electrons, and hadrons.

Once the particle groups had been decided, the dataset was created using Monte Carlo techniques to simulate the events expected at the HL-LHC including an emulation of the upgraded CMS Level 1 trigger that produces the L1 tracks. The power of using simulated data versus real-world data is gaining the knowledge of which reconstructed tracks are real and fake, making this a supervised problem. The simulated events are from three representative physics samples containing: high-pT jets, dimuon Z boson decays, and dielectron Z boson decays. Overlaid on each event is an average of 200 simultaneous pp collision events. From these simulated tracks, equal amounts of fake tracks, muons, electrons, and hadrons were selected for a training set to emphasize the importance of each group equally. The training set consists of 10000 tracks total, or 2500 in each particle group as well as fake tracks. The total number of tracks in the training set was chosen as 10000 because it provided promising results without taking much time to train the model. The test set consists of all other tracks that are not included in the training set, totaling to 243191 tracks. Note that muons, electrons, and hadrons all fall under the category of real tracks.

The features describing each track are derived from the track properties which will be provided by the L1 trigger system. A full list of these features along with a brief description of each is found in table 1. The transverse momentum of the track (p_T) is not used as it was found that when algorithms were trained with it included, they tended to favor low p_T tracks, which have a naturally low fake rate and disfavor high p_T tracks, which have a naturally high fake rate. As we are generally more interested in high-p_T tracks, this was counterproductive.

3 Results

This classification problem in service of reconstruction lends itself to using a gradient boosted decision tree (GBDT) and neural network[1] (NN). This is largely due to the complexity of the data. A support vector machine and random forest were also tested, but did not perform as well as the GBDT and NN. In this section, I will describe how I compare these algorithms to a current method.

[1]The GBDT was implemented using the scikit-learn software, and the NN was implemented using the Keras software [6,7].

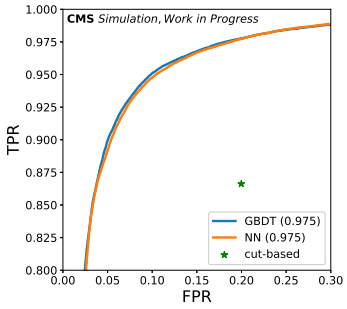

Figure 1: ROC curve showing the GBDT and NN to outperform the current cut-based approach.

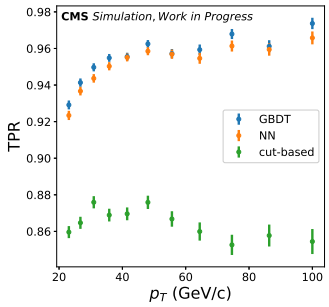

Figure 2: Relationship between TPR and p_T for decision threshold of 0.65.

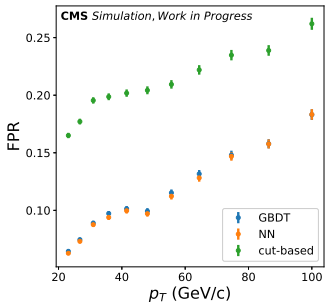

Figure 3: Relationship between FPR and p_T for decision threshold of 0.65.

3.1 Metrics

In order to measure performance of my machine learning models, I focused on three metrics. Each of these metrics is related to the variables *true positive rate* (TPR) and *false positive rate* (FPR) which are defined as

$$\text{TPR} = \frac{\text{\# of real tracks classified as real}}{\text{\# of real tracks}} \qquad \text{FPR} = \frac{\text{\# of fake tracks classified as real}}{\text{\# of fake tracks}}. \quad (1)$$

Another important variable is the *decision threshold*. Both the GBDT and NN output a probability of each track being real, and the decision threshold controls the boundary on this probability that distinguishes real from fake tracks. A decision threshold close to 0 will classify more things as real, and a threshold closer to 1 will classify less as real.

ROC curve and AUC A receiver operator characteristic (ROC) curve looks at the trade off between TPR and FPR as the decision threshold varies from 0 to 1. The area under the curve (AUC) gives a numeric value to the ROC curve in order to compare algorithms against each other. In general, a larger AUC indicates better performance.

TPR/FPR versus p_T This metric shows how the TPR and FPR change as a function of p_T. The particles are binned based on their p_T and TPR/FPR is calculated for each bin. Since particles with higher p_T are more interesting in regards to new physics, more emphasis is placed on whether the higher range is correctly classified.

TPR/FPR versus decision threshold Here, I see how TPR and FPR change as a function of decision threshold. This measure is important to allow different analysis groups to select the threshold that best meets their objectives.

3.2 Analysis

The results of the trained GBDT and NN are shown in figure 1, along with an example of a current cut-based approach[2] that was used by an analysis group to reject high-p_T fake tracks. It is clear that both machine learning algorithms outperform the current approach. For the same FPR, both the GBDT and NN have a 10% larger TPR. We can also see that the GBDT and NN are both performing similarly, with a difference in AUC of only 0.02.

Measuring the TPR and FPR as a function of p_T shows us similar results to the ROC curve; TPR is greater in both the GBDT and NN while FPR is smaller (figures 2 and 3). What is more interesting is that the machine learning algorithms have higher TPR as p_T increases which follows the emphasis that LHC physicists put on correctly identifying real tracks in the high-p_T regions. Unfortunately, the

[2]The cut based approach labeled tracks as real for $P_t > 2$ GeV/c, $|\eta| < 2.4$, $|z_0| < 15$ cm, $\chi^2/\text{dof} < 50$, and $\chi^2_{bend} < 1.75$

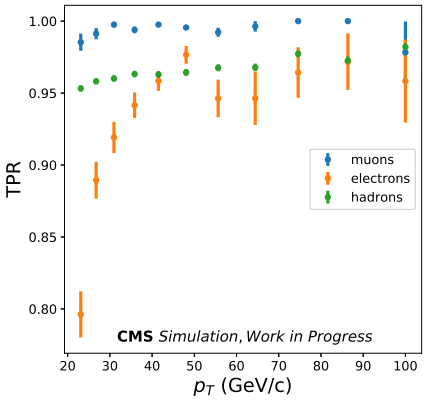

Figure 4: An example of the difference in TPR for various particles with the GBDT algorithm.

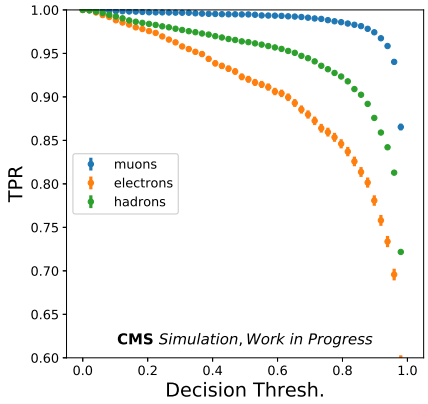

Figure 5: An example of the relationship between TPR and decision threshold for various particles with the GBDT algorithm.

FPR also increases as we look towards higher p_T tracks meaning that more fake tracks get labeled as real.

It is also important to look at the TPR for each individual particle group since many analysis teams only care about a specific particle type. In general, all classifiers do the best on muon classification, followed by hadron and then electron classification as seen in figure 4. This indicates that electrons are harder to classify in general and those looking to reduce the number of electrons being thrown away as fake tracks will need to loosen their decision threshold to increase the TPR.

Depending on what an analysis group is looking for in their sample, they will want to choose a decision threshold that increases the TPR for those particles while keeping the FPR as low as possible. Figure 5 shows an example of how TPR changes as a function of the decision threshold for each particle group classified by the GBDT. Muons are classified so well that one can tighten the threshold to .95 and still have TPR be $\sim$95%, but the same threshold would only give a TPR of $\sim$75% on electrons. The ability to choose a threshold to best suit a certain analysis allows the user to judge their own need for a balance between TPR and FPR.

4 Conclusion and future work

Initial results show that the GBDT and NN machine learning algorithms can significantly outperform current cut-based approaches in classifying real versus fake tracks. These algorithms increase their classification accuracy as track p_T increases which mimics the importance LHC physicists put on higher p_T particle tracks. However, the machine learning algorithms perform better on some particles than others. Therefore, different LHC analysis groups may want to adjust the decision threshold to best classify their specific particles of interest. This work also confirms the results found in an analysis for a different L1 tracking algorithm [8].

With the LHC upgrade increasing the number of particle collisions by a factor of 10, quickly and efficiently rejecting unimportant events in the search for new physics is essential. Current LHC physicists working on upgrading the particle detectors plan to implement new and improved algorithms, such as this one, onto Field-Programmable Gate Arrays (FPGAs) to have them run in real time with the data collection system. This means that the finished algorithm must be adaptable to FPGA software, using HLS4ML [9] for example, and must comply with time and memory restrictions. Hence, once my algorithms are optimized, computational efficiency comparisons will determine which one can be implemented in the detectors.

References

[1] CMS and ATLAS collaborations, *Combined Measurement of the Higgs Boson Mass in pp Collisions at* $\sqrt{s}$*= 7 and 8 TeV with the ATLAS and CMS Experiments*, Phys. Rev. Lett. 114, 191803, 2015.

[2] Apollinari, G., Béjar Alonso, I., Brüning, O., Lamont, M., and Rossi, L. (Eds.) *High-Luminosity Large Hadron Collider (HL-LHC) : Preliminary Design Report*. United States, 2015. doi:10.5170/CERN-2015-005.

[3] CMS collaboration, *The Phase-2 Upgrade of the CMS Tracker*, Technical Design Report, CERN-LHCC-2017-009, CMS-TDR-014

[4] CMS collaboration, *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*, Technical Design Report, CERN-LHCC-2017-013, CMS-TDR-017

[5] Grossman, J., *PS-module prototypes with MPA-light readout chip for the CMS Tracker Phase 2 Upgrade*, Journal of Instrumentation, Vol. 12, C02049–C02049, 2017.

[6] Pedregosa et al., *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830, 2011.

[7] Chollet, F., *Keras*, `https://keras.io`, 2015.

[8] Summers, S.P., *Application of FPGAs to Triggering in High Energy Physics*, CMS-TS-2018-029, CERN-THESIS-2018-248

[9] Duarte. J., et al., *Fast inference of deep neural networks in FPGAs for particle physics*, JINST 13 P07027 (2018), arXiv:1804.06913