

Using deep neural network to reduce mismatch in ALICE Time of Flight

PIERRE-ALEXANDRE DUVERNE - Université Paris-Sud
pierre-alexandre.duverne@u-psud.fr

NICCOLO' JACAZIO - University of Bologna
nicolo.jacazio@cern.ch

MANUEL COLOCCI - University of Bologna
manuel.colocci@cern.ch

04/06/2018 - 31/08/2018

Abstract

The Time Of Flight (TOF) belong to the ALICE experiment and is dedicated to particle identification by computing their time of flight. It works by matching a hit and a track reconstructed by the inner detectors. During that procedure there is some hit/track mismatch that can be reduced by using machine learning techniques. The first part of the internship was dedicated to the study of the problem with an old ROOT library called TMultiLayerPerceptron in order to improve PID with TOF. The second part was dedicated to the use of Keras, a much more recent python library.

I. Introduction

A Large Ion Collider Experiment (ALICE) is one of the four main experiments of the LHC dedicated to the study of the strongly interacting matter created by heavy ion collisions. One of the main purpose of those collision is to recreate a theoretical state of matter : the Quark-Gluon Plasma (QGP). This is an exotic state of hot and dense matter where deconfined quarks and gluons can be observed. This is not possible outside of the thermodynamical conditions created by the collisions because of the color charge that confined quarks and gluons. But studying the evolution of QGP requires to know the species of the particles produced during the interaction. This is the main objective of the Time Of Flight (TOF) that uses the characteristic time of flight of the particles to identify them. It works by associating a track propagated in the TOF and one of the hit in its vicinity. The reconstruction of the track propagated in the TOF is

done for one part by the Inner Tracking System (ITS) located just close to the interaction point. It consists in six layers of silicon around the beam pipe that can give different informations as the track impact parameter or an analogic measurement of the energy loss in the detector, useful for identification of low momentum particles. The second part is done by the Time Projection Chamber (TPC), the main detector of ALICE. However there are some errors in the association between a track and a hit, this is called mismatch (versus goodmatch when the association is right), that degrade performances of the TOF. It is proposed for this work to use deep neural network, because of their efficiency to study non-linear and multi-variables problems as this one, in order to reduce that effect in the datas.

II. Time of Flight

1. How does the TOF works?

The TOF is a cylindrical detector covering the area $|\eta| < 0,9$. It can measure low momentum particles. For the high particle's momentum, it will be identified by another detector of ALICE, for example the High Momentum Particle Identification (HMPID).

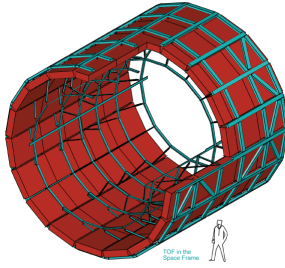


Figure 1: Time of Flight
TOF is a gaseous detector splitted in 18 sectors made of sensitive strips of $120 \times 7 \text{ cm}^2$ [CER18]

This detector can reach a time resolution of 56 ps that allow it to distinguish kaons and pions at 2 GeV/c and protons and kaons at 5 GeV/c [CER00]. To reach that resolution, the detector is based on the **Multi Resistive Plate Chamber (MRPC)** technology. It consists in a double stack of five plates of commercial glass separated with a distance of $250 \mu\text{m}$ by fish line. The gap between the plates is filled with a mix of gas composed of $\text{C}_2\text{F}_4\text{H}_2$ (90%), SF_6 (5%), and C_4H_{10} (5%). There are two external cathodes and one internal anode between the two stacks [CER02]. When a particle passes through a strip, it ionizes the gas, the detector gain creates an avalanche of electrons stopped by the glass. Then it creates an electromagnetic wave which ionizes the next gap, until the electrons reach the internal electrode where they are collected. The final signal is the sum of all the signal created in the previous gaps. This configuration amplifies the signal in the TOF and give a good time resolution. Those clusters are then associated with reconstructed tracks.

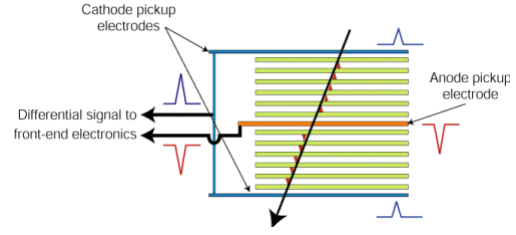


Figure 2: Multi Resistive Plate Gap
The stack of glass plate makes a quick and precise detector. [INF18]

2. Particle identification in the TOF

Particle identification is an important purpose to study heavy ions and QGP for several reasons. On the one hand, knowing the particles produced during an event gives thermodynamical informations, particularly the temperature because QGP happens after a phase transition at $T_{crit} \sim 170 \text{ MeV}$. On the other hand the hard probes used to study the QGP depends on the particles produced at the very beginning of the interaction. Thus, it is necessary to be able to compare the proportions of those particles to the known production rate. [GO18]

Particles are identified in the TOF by plotting the relativistic coefficient versus the momentum. There are several strips visible on figure 3, corresponding to the six species detected in the TOF which are electrons, muons, pions, kaons, protons, and deuterons.

One can see that there is a lot of background, corresponding to the continuous background that needs to be removed from the TOF data to improve the detector performances. Some tracks seems to have a $\beta > 1$ that do not make any physical sense. It can correspond to reflexions of particles on the most external parts of ALICE that are detected by the TOF. This is an exemple of clear mismatch in the detector.

3. TOF performances

The distribution used to identified particles is a centered gaussian with a standard deviation :

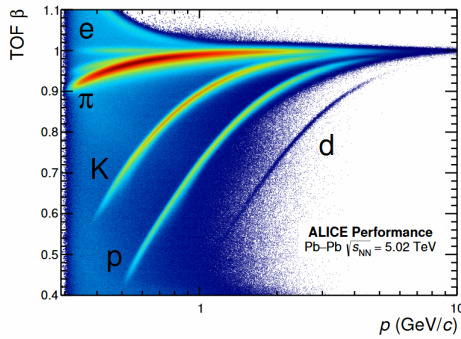


Figure 3: Beta VS Impulsion in the TOF
This was obtained during beam tests for the actual run at the LHC. [Jac18]

$$\sigma_{TOT}^2 = \sigma_{TOF}^2 + \sigma_{trk}^2 + \sigma_{event}^2 \quad (1)$$

where σ_{TOF}^2 is composed of :

- The intrinsic resolution of the MRPC
- The resolution of the readout electronic
- The resolution of the internal clock of the LHC
- The calibration of the TOF

The part σ_{trk}^2 is linked to the resolution on the track's reconstruction and the σ_{event}^2 part is linked to time resolution on the evaluation of the instant when the event occurred. The total time resolution is 56 ps for the TOF. [Car18]

III. Neural networks

1. How does it work

Neural networks were design during the 50's and are based on the biological networks. During the 60's many limitations were pointed out, particularly, it was proven that it was impossible to solve non-linear problem with this first generation of networks, thus for a long time they were forgotten. But during the 80' it changes thanks to the multi-layers perceptrons that solve all the previous problems. That is the kind of network I used during this internship.

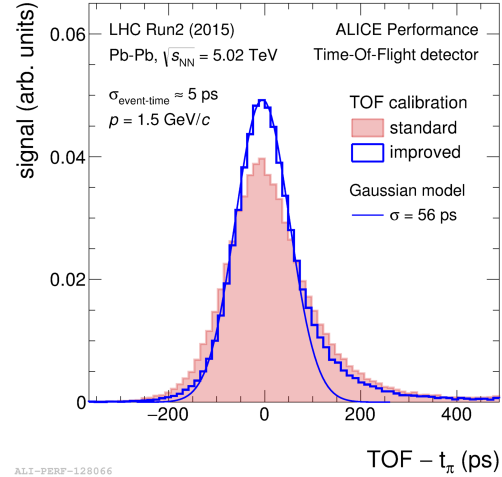


Figure 4: Time Performance of TOF [Pre17]

A neuron is basically a mathematical function described by an activation function which define a threshold that triggers the neuron. Those neurons are bound in layers and each one of them is linked by synapses to all the neurons of the next layer. The synapses are defined by a weight modeling the strength of the interaction between two neurons. For a multi-layer perceptron the computation is made by making the dot product of the entry vector and of the synaptic weight.

The first layer consists of the input parameters of the problem, the next layers are called hidden and the last one is the output layer which give the result of the computation. For the networks I trained the first layer is made of the characteristics of the tracks (see the appendices) and the last one was just made of a neuron that says wether or not the association is good.

2. Interests for the TOF

The problem of the mismatch involves many parameters and their influences on the efficiency has no reason to be trivial. The high number of parameters is mainly due to the fact that one can find up to ten clusters in the vicinity of the track and all of them have to

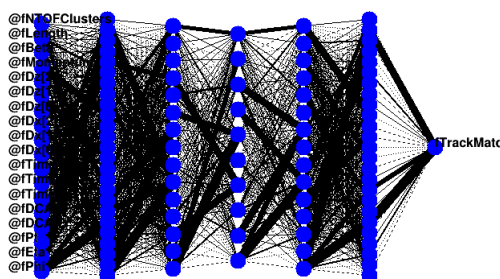


Figure 5: *Shape of a neural network*
A network that was used to analyse the data

be taken into account. This makes the analysis very difficult but neural network were built to study those kind of problems which makes them good candidates to improve particle identification.

To use a network one starts by setting two sets of datas, the first one is used to train the network and the second one is used to check the accuracy after the training phase. During the training, the tracks are presented to the network which tries to minimize the error by backpropagating the gradient, and change at each track the synaptic weights. Then, once the whole training data set is ended, there is a check on the validation set that consists in presenting a track to the network and checking if the answer is right or wrong. Then when it is finished this whole process is iterated, every iteration is called an epoch.

For this internship the data came from simulation GEANT 3 [CD94] of the LHC, able to simulate events in ALICE thanks to a Monte-Carlo algorithm. Thanks to that it was known if

But, the proportion goodmatch/mismatch in those simulated data are 90%/10% and it limits the learning abilities by fixing the weights of the network because it is not facing enough mismatch. So a pre-process was applied on the data to have equal proportions of goodmatch/mismatch. This highly improved the performances of the trained data. One can see an example of a trained network without this process in the appendix to compare with the following results.

IV. Results with TMLP

1. Training

The ROOT library TMultiLayerPerceptron [Del03] was used for the first two third of this work. It is quite old [PR93], but as there are less thing to parametrize, TMLP is simpler and allowed to understand the principles of machine learning. Moreover, as it belongs to the ROOT framework, dedicated to data analysis in particles physics and nuclear physics, it limits the risk of errors in the network structure. For this part the neuron are described by a sigmoid function which is defined as follow :

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (2)$$

except for the output neuron described by a softmax function which is applied to a vector $\vec{x}$ of dimension n that return vector $\vec{o}$ whose components are :

$$\sigma(\overrightarrow{x})_j = \frac{e^{x_j}}{\sum_{i=1}^n e^{x_i}}, \forall j \in \llbracket 1; n \rrbracket. \quad (3)$$

The results presented in this section are given by a network trained during 300 epochs with a set of 3,000,000 tracks. Once the training is over the figure 6 is obtained. This is the distribution of the output neuron which allows to visualize

if the network is able to separate the signal (goodmatch) and the background (mismatch). Thanks to that it is possible to chose a cut. This cut defines if the association is right or wrong according to the network. Here the chosen cut is 0.30.

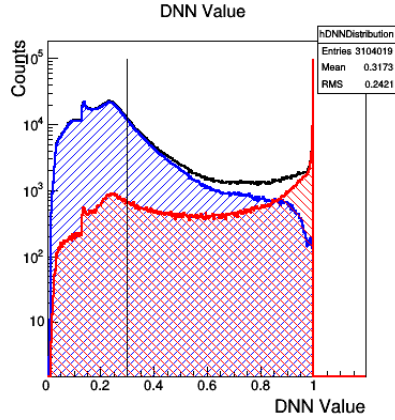


Figure 6: Distribution of the output neuron
The red distribution is for the mismatch. The blue distribution is for the goodmatch.

2. Analysis

After the training, a new set of 1,000,000 tracks is given to the network to check its performances. The figure 7 is the distribution beta versus momentum without any analysis. There are the same problems as the one described for the figure 3 : track with $\beta < 1$, high background a low momentum.

After the analysis, the figure 8 is obtained. There is a clear improvement of the signal, most of the tracks with $\beta < 1$ and the high momentum tracks are also removed, which makes sense as the TOF is designed to identify low momentum particles. Plus there is an improvement in the region $\beta \simeq 0.9$ et $P_T \simeq 1$, even if there is still some mismatch due to the high density of tracks in this region that makes the learning difficult for the network.

However by crosschecking the Monte-Carlo truth of the figure 9 and the analysis by the network it appears that a non negligible part of the signal has to be removed to improve the

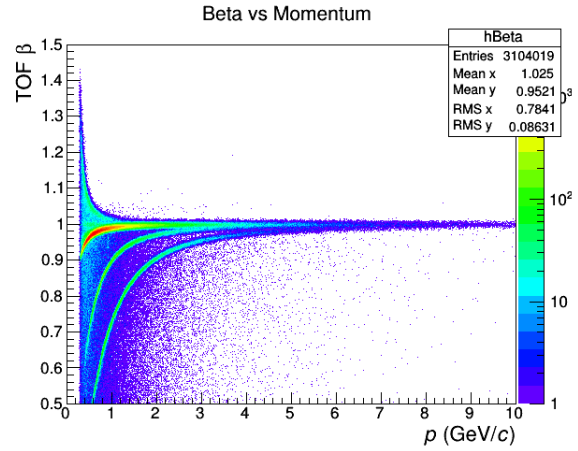


Figure 7: Unselected data
Beta vs Momentum before analysis by the network

quality of the data, which is a high cost for this analysis.

3. Performances of this network

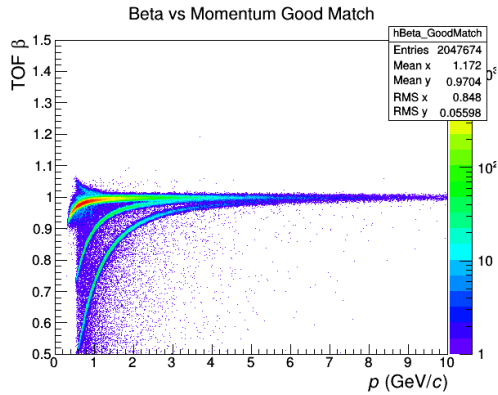
In order to characterise the network, we define the following quantities :

$$Purity = \frac{Good\ tagged\ goodmatch}{Tagged\ goodmatch}, \quad (4)$$

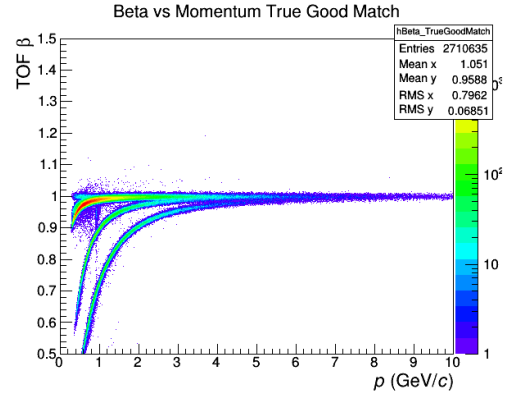
$$Contamination = \frac{Mismatch\ tagged\ goodmatch}{Tagged\ goodmatch}, \quad (5)$$

$$Efficiency = \frac{Tagged\ goodmatch}{Total\ number\ of\ track}. \quad (6)$$

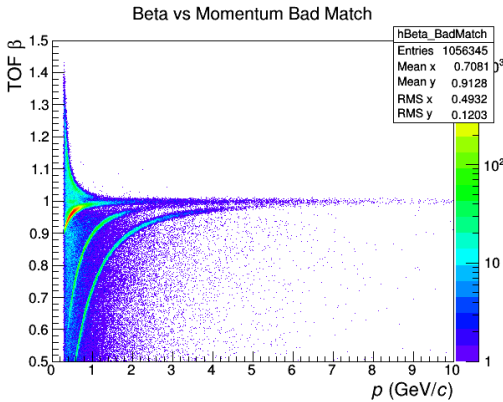
The first quantity is the correct answers rate given by the network, the second one is the rate of false positives and the last one quantify the part of the data that was discarded from the input set. On the figure 10, it appears that the efficiency decreases as the cut increases. This means that more tracks are removed from the unselected data. Moreover, there are more tracks removed at low impulsion because the



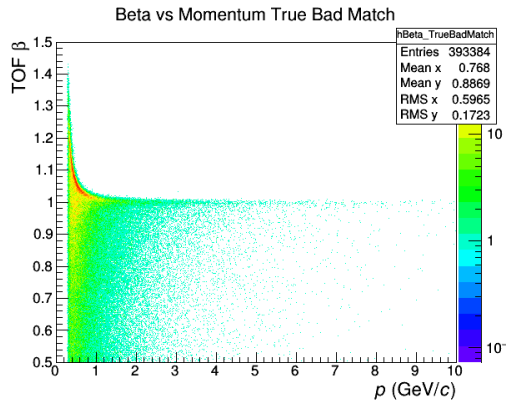
(a) Tagged good match by the network



(a) True Goodmatch



(b) Tagged badmatch by the network



(b) True Badmatch

Figure 8: Analysis of an unknown set of data by the network

Figure 9: Monte-Carlo truth

density is higher in this region. The purity is low in that same region for the same reason, moreover to have a high purity a hard cut is needed, at least 40%. Then it justifies the choice made for the cut at 0.30 and it confirms that to have an improvement part of the signal has to be removed. The same remark is possible for the contamination that needs a hard cut.

4. Conclusion for TMLP

The use of TMLP shows that a reduction of the mismatch is possible to improve the quality of the signal. But to achieve that a non negligible part of the signal has to be sacrificed and t im-

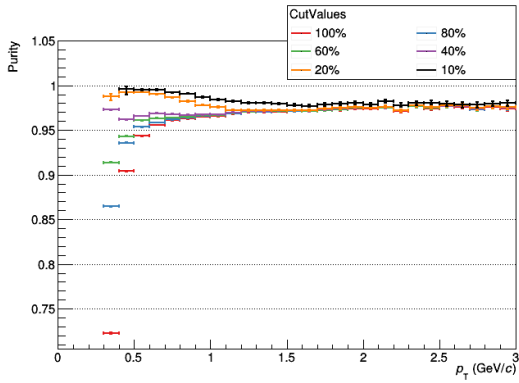
plies that good statistics are needed to have relevant results. Another drawback of the analysis with TMLP is the high computing time needed.

V. Results with Keras

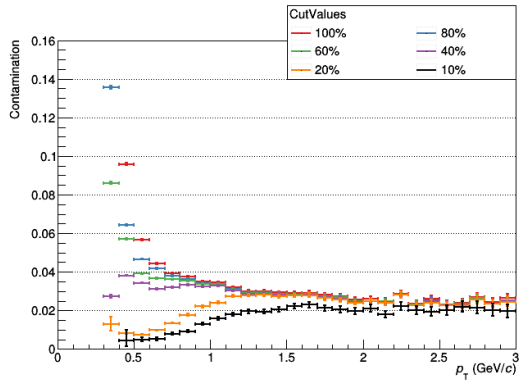
1. Training

Keras is a python library which use Tensorflow's machine learning algorithms. It is more recent than TMLP and it was expected better results. [Cho17]

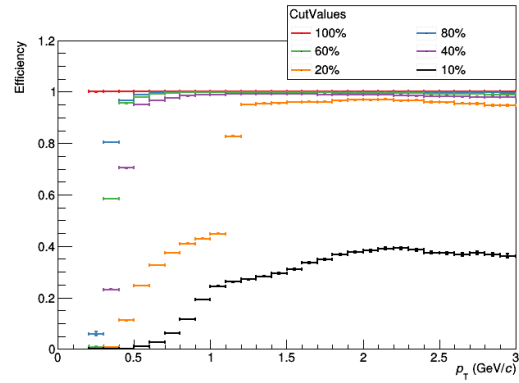
For this section the network was trained with 400,000 tracks during 200 epochs and after that the figure 11(a) is obtained. All the neurons are



(a) Evolution of the purity with the cut



(b) Evolution of the contamination with the cut



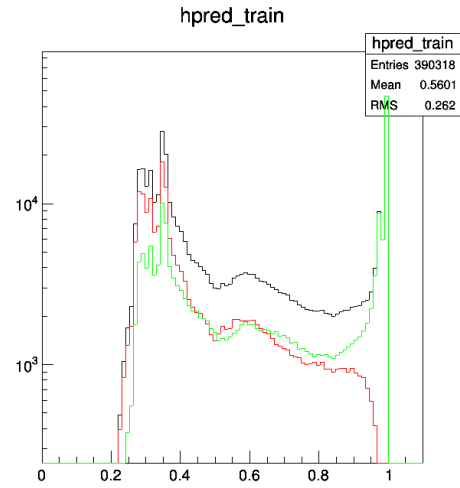
(c) Evolution of the efficiency with the cut

Figure 10: Those quantities characterize the network

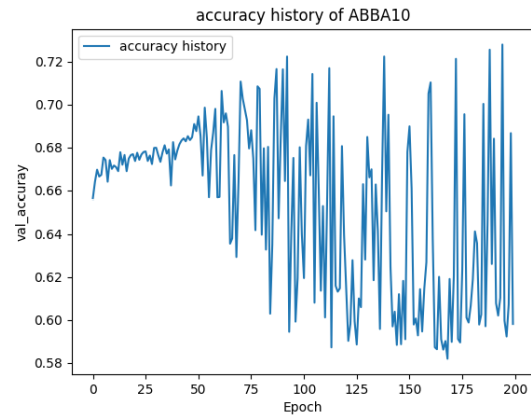
defined by sigmoid function, even the output neuron. There is an actual separation between goodmatch in red and badmatch in green. Particularly the separation is better as the answer

of the network is close to 1. This separation is better than the previous and it took less epoch to be obtained.

Keras also gives access to the accuracy of the network at each epoch and it is in the figure 11(b). One can see that there are important variations of the accuracy after 60 epochs which suggests that it is not necessary to use more epochs. But if the number of epoch is reduced, then the set of data has to be bigger which was not possible here because of memory leak problems that avoided the use of more tracks.



(a) Distribution of the output neuron



(b) Learning curve

Figure 11: Characterisation of the network after training

2. Analysis

For this phase, 500,000 new tracks were used, they are presented in figure 12, and a cut at 0.37 has been chosen. An improvement of the signal is visible in figures 13 and 14. The region $\beta > 1$ is cleaner than before performing the analysis, same as the low momentum region where the characteristic bands are delimited in a better way with the analysis. Particularly the strips of protons and kaons are clearer on the analysed data. Yet it is still difficult to improve the region of relativistic and low impulsion tracks because of the density of the tracks.

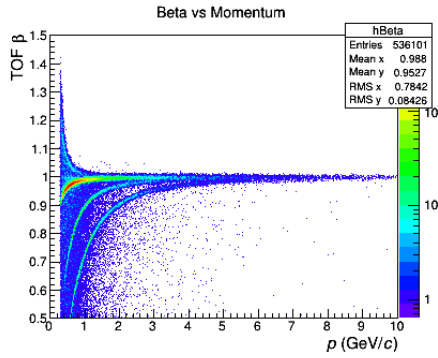


Figure 12: *Unselected data
Beta vs Momentum*

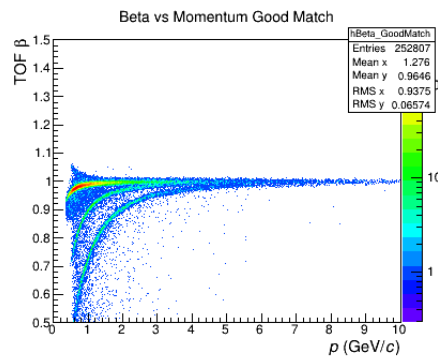


Figure 13: *Tagged goodmatch*

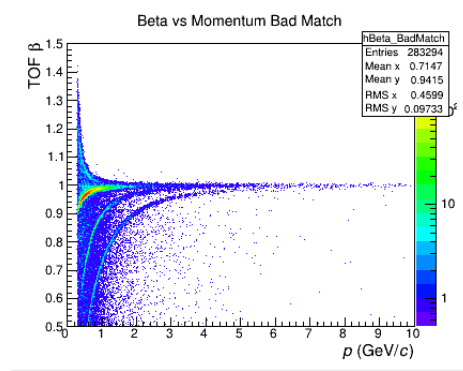
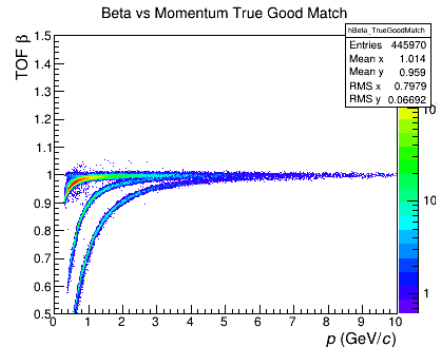
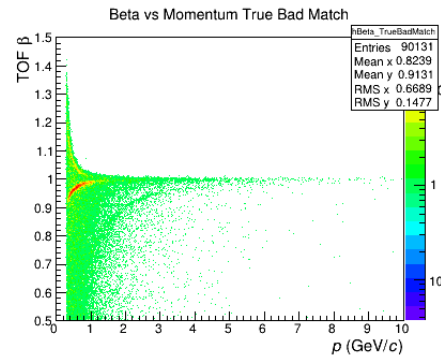


Figure 14: *Tagged badmatch*



(a) *True goodmatch*



(b) *True badmatch*

Figure 15: *Monte-Carlo truth*

But on the other hand it is still necessary to remove an important part of the signal to clean

the initial set has we can see with figure 16. Particularly at low impulsion where the efficiency has to be low to have a high purity and

to limit the contamination.

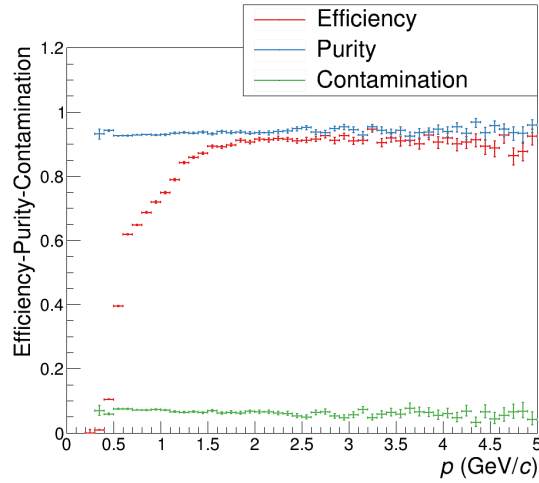


Figure 16: Efficiency, Purity and Contamination
Performances for a cut at 0.37 with the Keras network

Moreover, the figure 18 shows that there is no influence of the analysis with the neural network on the signal shape. Indeed on the figure 18 where the nSigma distribution for the unselected data, the true goodmatch and the true badmatch are plotted and fitted by a gaussian distribution with an exponential tail. The three other plots are the parameters of the fit for the unselected data in black and in blue for the data tagged as goodmatch by the network. The first parameter is the mean and the second is the standard deviation of the gaussian. The third one is the parameter of the exponential tail. Those fits show that there is no apparent bias in the analysis of the network.

3. Conclusion for Keras

Those results show that to obtain equivalent results to the TMLP library, less data and computing time are needed which is a real improvement. Plus with more optimisation it would certainly be possible to obtain better performances even if there will still be the issue of the efficiency loss. Moreover this analysis is relevant as it is not introducing any visible bias on the signal shape.

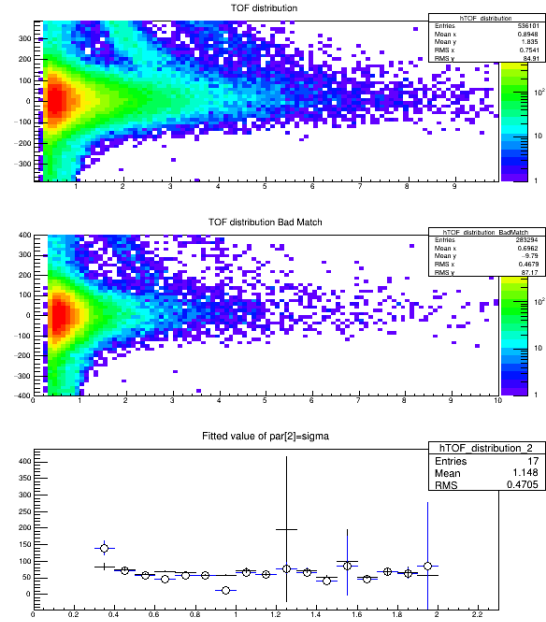


Figure 17: Analysis of potential bias of the network
legende

VI. Conclusion and Outlooks

The purpose of this internship was to analyse data from the TOF, a detector of the ALICE experiment, with deep neural network and understand its feasibility. Indeed, there are some errors remaining that have to be reduced in order to improve the performances. In a first part of the work, the library TMLP of ROOT was used because of its simplicity. The results show a true improvement by removing an important part of the mismatch but part of the signal is eliminated as well. Then high statistics are needed to be relevant, which makes that analysis very costly.

Then Keras has been used and gives similar results but with less data and computing time involved. However, because of missing time some parameters that could improve the training have not been modified, and the network can be optimized. Another way to optimize the training phase is to train one network for each species of particle in the TOF (electron, muons, kaons, pions, protons, deuterons). It

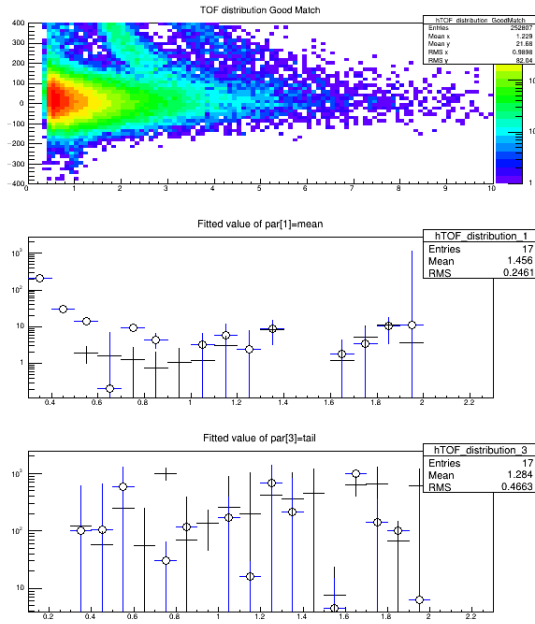


Figure 18: Analysis of potential bias of the network(2)

would be efficient, particularly in the region of low momentum particles where the density is high. Once optimized, those networks might be implemented in the Aliroot framework and will be useful to analyse data that are actually being taken and for the RUN 3 of the LHC.

VII. Acknowledgement

For this internship I would like to thank my supervisor Nicolo' Jacazio who help me so much for my work in many way. I learnt a lot of things thanks to him in computing science, but also in the research field in general. I thank also the whole TOF team who welcome me for those three months, particularly Manuel Colocci who helped me at the very beginning of this work. This also include Sivi and Suzana, two others very nice summer students. I would like to thank the staff of the summer student team who perfectly organized this stay at the CERN.

VIII. Appendix

1. Results for non treated data

This results are from a network that has been trained on a set of data with proportions good-match/mismatch 90%/10%. The results are clearly poorer than the one presented before which justify the choice of the pre-process. This network was trained with 1 500 000 tracks and 100 epochs.

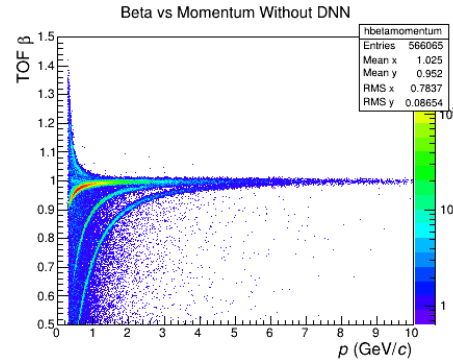


Figure 19: Beta vs Momentum
Data without analysis

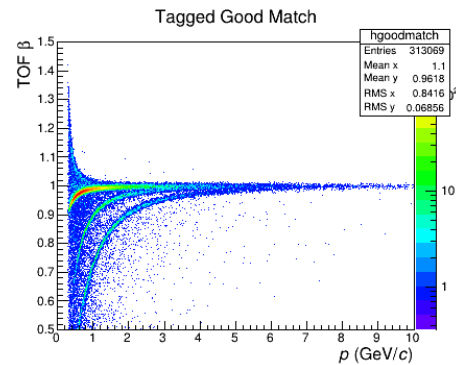


Figure 20: Tagged goodmatch
Data without analysis

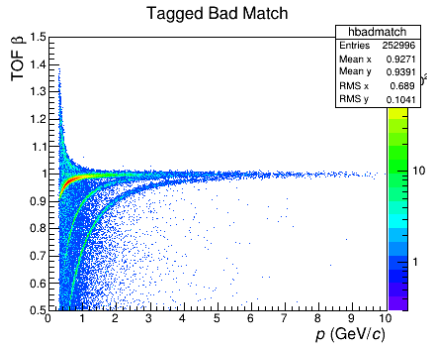
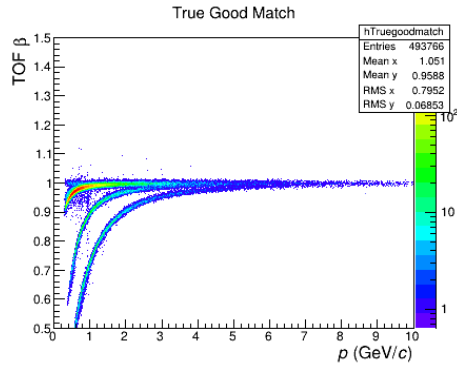
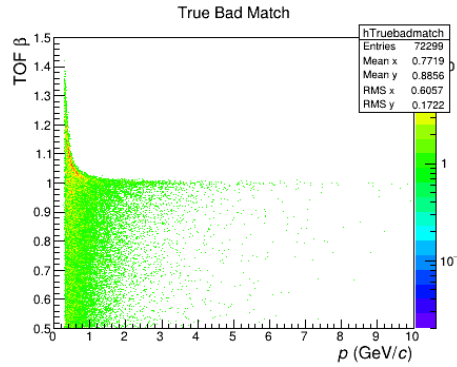


Figure 21: *Tagged badmatch*
Data without analysis



(a) *True goodmatch*



(b) *True badmatch*

Figure 22: *Monte-Carlo truth*

2. Used parameters

Parameter	Description
NTOFClusters	Number of clusters in the vicinity of the tracks
Length	Length of the track
Beta	Relativistic coefficient
Momentum	Momentum of the particle
Dx	Residu on the x axis
Dz	Residu on the z axis
DCAXY	Impact parameter on XY
DCAZ	Impact parameter on Z
T0Trktime	Best estimation of the time of the initial vertex
Eta	Distribution of the Eta parameter
Phi	Distribution of the Phi parameter
ExpTime	Expected time for the six species in the TOF
ExpSigma	Expected resolution for the six species in the TOF
TrackMatchIndex	Index to know if the association cluster/track is right
Track PDG code	Index of the particle detected
TOF Time	Measured time by the TOF
P_T	Transverse momentum

REFERENCES

- [Car18] Francesca Carnesecchi. Performance of the alice time-of-flight detector at the lhc. *arXiv:1806.03825v1*, 11 June 2018.
- [CD94] Application Software Group Computing and Networks Division. *GEANT 3 User Guide*. CERN, Genève, october 1994.
- [CER00] CERN/LHCC. Technical design report of the time of flight system (tof). Technical report, CERN, 16 February 2000.
- [CER02] CERN/LHCC. Addendum to the technical design report of the time of flight system (tof). Technical report, CERN, 24 April 2002.
- [CER18] CERN. Experience ALICE. <https://home.cern/fr/about/experiments/alice>, 2018. [Online].
- [Cho17] François Chollet. Documentation keras. <https://keras.io/>, 2017. [Online].
- [Del03] Christophe Delaere. class tmultilayerperceptron. <https://root.cern.ch/root/html516/TMultiLayerPerceptron.html>, 2003. [Online].
- [GO18] Jan Fiete Grosse-Oetringhaus. Heavy Ion part 3. In *Summer Student lecture*, page 38, July 2018.
- [INF18] INFN. Time of flight for alice experiment. <http://www.bo.infn.it/alice/tof-over/index.html>, 2018. [Online].
- [Jac18] Nicolo' Jacazio. Tof beta vs momentum performance in pb-pb at 5.02 tev. alice info - ALI-PERF-106336, 23 July 2018.
- [PR93] Carsten Peterson and Thorsteinn Rognvaldsson. Jetnet 3.0—a versatile artificial neural network package. *Computer Physics Communications*, CERN-TH.713s/94, December 1993.
- [Pre17] Roberto Preghenella. Tof timing performance improvement. alice info - ALI-PERF-128066, 06 April 2017.