# Particle Identification Algorithm in Level 2 Trigger

Lara Kamal

Phil Rubin (Supervisor)

August 2020

**Abstract**

The NA62 experiment is primarily a search for an extremely rare kaon decay $K^+ \to \pi^+\nu\bar{\nu}$ transitioning into a singly charged pion, a neutrino, and an anti-neutrino. This transition occurs once in every 10 billion decays therefore, this measurement requires a strong suppression of the background to detect. Sifting through massive amount of data is necessary and it needs to be done with a high level of sophistication. In this effort, we implemented a Boosted Decision Tree (BDT) algorithm that combines information from multiple detectors for a particle identification task using CERN's TMVA package. The BDT technique provides a perfect tool to tackle this problem efficiently and within the time constraint in the L2 (level 2) trigger.The algorithm classifies three particle types: muons, pions, and electrons (intended as both e+ and e), thus making this task a three-class classification problem.

## 1 Introduction

The NA62 experiment adopts a trigger strategy based on three stages:

- Level 0 trigger (L0): A hardware level that contains six detectors: CHOD, NewCHOD, RICH, LAV12, MUV, and LKr. The L0 trigger rate can be up to 1 MHz, resulting in about 1 ms processing time for event selection [3].
- Level 1 trigger (L1): A software trigger that elaborates a restricted set of information due to the time constraints on L0. L1 reduces event rate below 100kHz limit imposed by the calorimeter readout [3].
- Level 2 trigger (L2): A software trigger that allows a selection of high-level event variables and expected to reduce the data rate by a factor 10, below the 10kHz.

Identifying particles is a critical task in this experiment and needs to be implemented at the L2 trigger. Since the typical number of events passing the L1 (level 1) algorithm is between 200 K to 300 K events per burst, L2 trigger can average to 25 to 40 millisecond per event [1], which puts the L2 trigger under time constraints.

## 2 Training Dataset

The construction of a decision tree requires a large set of training data. Training dataset is a set of already classified tracks along with their discriminating variables. To obtain a training set, we utilized Monte Carlo (MC) simulated data to train the BDT model. We simulated $\sim 61,000Ke2tracks(electrons), \sim 48,000K\mu 2$ tracks (muons), and $\sim 100,000K\pi\nu\nu$ tracks (pions).

### 2.1 Discriminating Variables

The variables employed in the particle identification are chosen by taking into account the time constraint at the L2 Trigger. We utilized the RICH (Ring Imaging CHerenkov) detector, the LKr (The Liquid Krypton) calorimeter, and other detectors to choose 13 different discriminating variables:

- 5 RICH Likelihoods: Background, electron, muon, pion, kaon likelihoods from adapted Spectrometer RICH Association algorithm using CHOD-associated track seeds.

  The background and kaon likelihood discriminating variables were omitted since there was no data from the MC simulation and therefore these discriminating made no contribution to the model construction. Figure 1 shows the distribution of the RICH Likelihoods from the training data for the three classes of particles: muons, electrons, and pions.

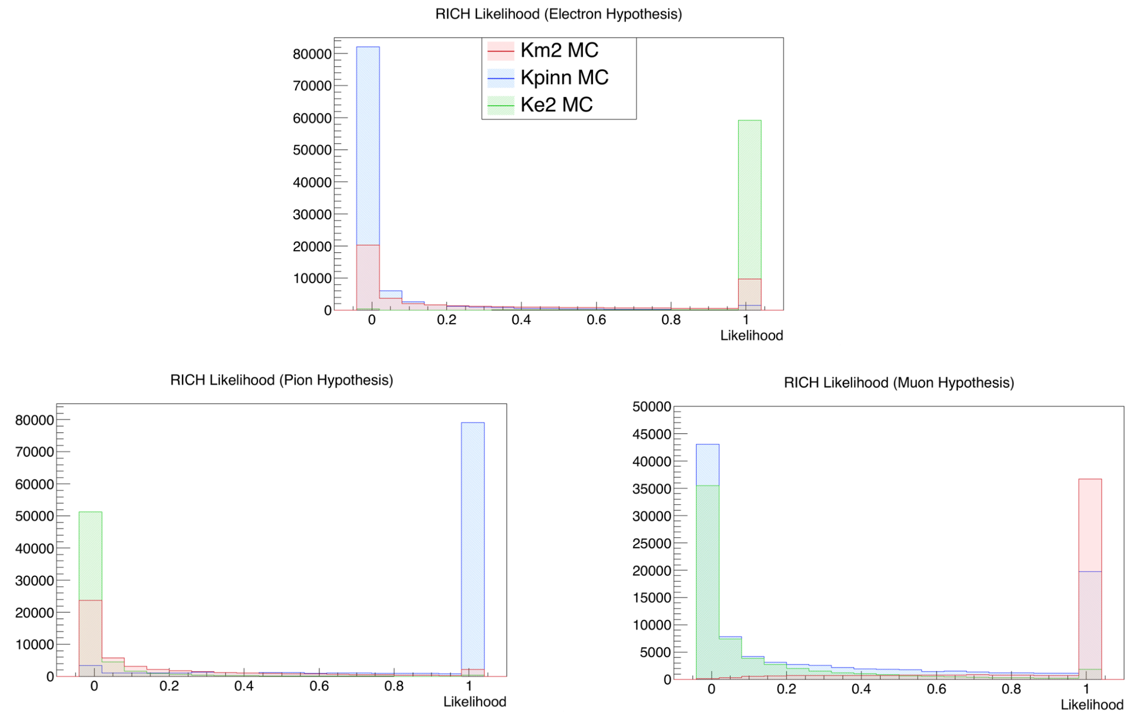


Figure 1: Monte Carlo simulated RICH Likelihoods for Muons, Electrons, and Pions.

- 5 RICH Ring Hits: Measures how much light is generated for background, electron, muon, pion, and kaon.

  The background and kaon ring hits were also omitted for the same reason as the likelihood. Figure 2 shows the distribution of the RICH Ring Hits from the training data for the three classes of particles: muons, electrons, and pions respectively. As seen in Figure 2, these histograms are stacked on top of each other making muons, electrons, and pions separation difficult.

- LKr E/P: The ratio of the energy in the LKr and the track momentum.

- LKr E/Total E: The ratio of the energy in the LKr and the total energy in the calorimeter.

- Total E/P: The ratio of the total energy in the calorimeter and the track momentum.

  Figure 3 shows the ratio of energy distributions for each particle type. It is obvious that there is more of separation in the training data among these particles, however, there is still an overlap. As a result, it is best to use an approach that combines all of the discriminating variable to identify these particles.

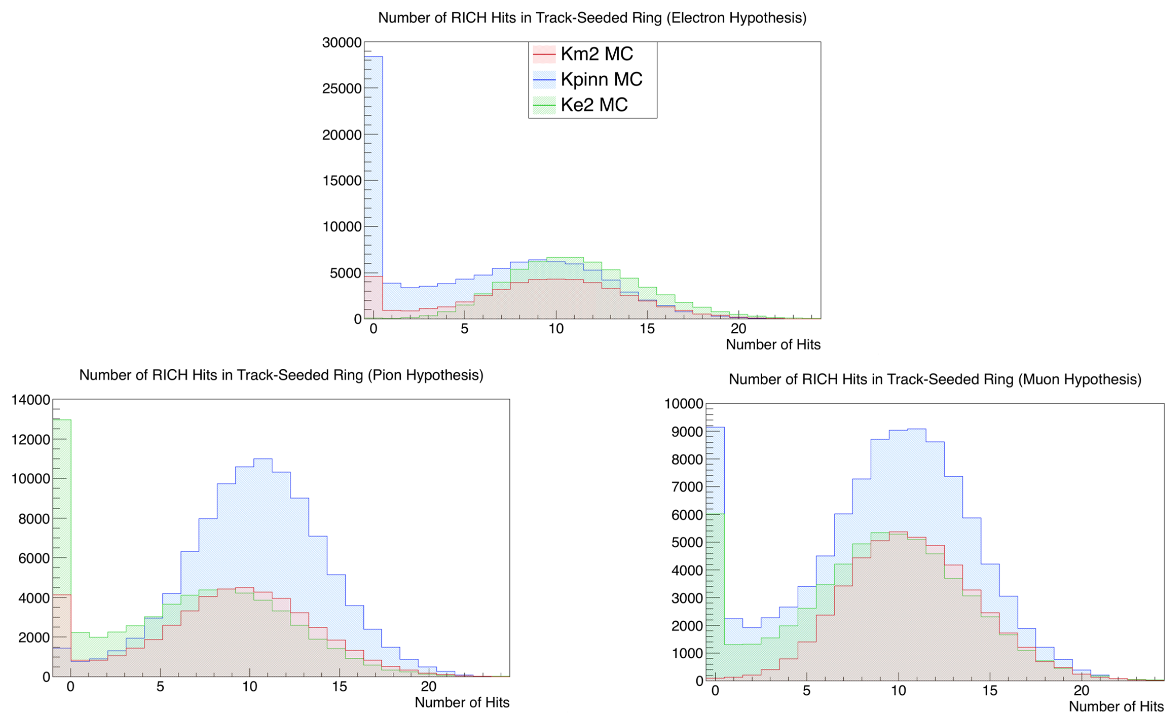


Figure 2: Monte Carlo simulated RICH Ring Hits for Muons, Electrons, and Pions.

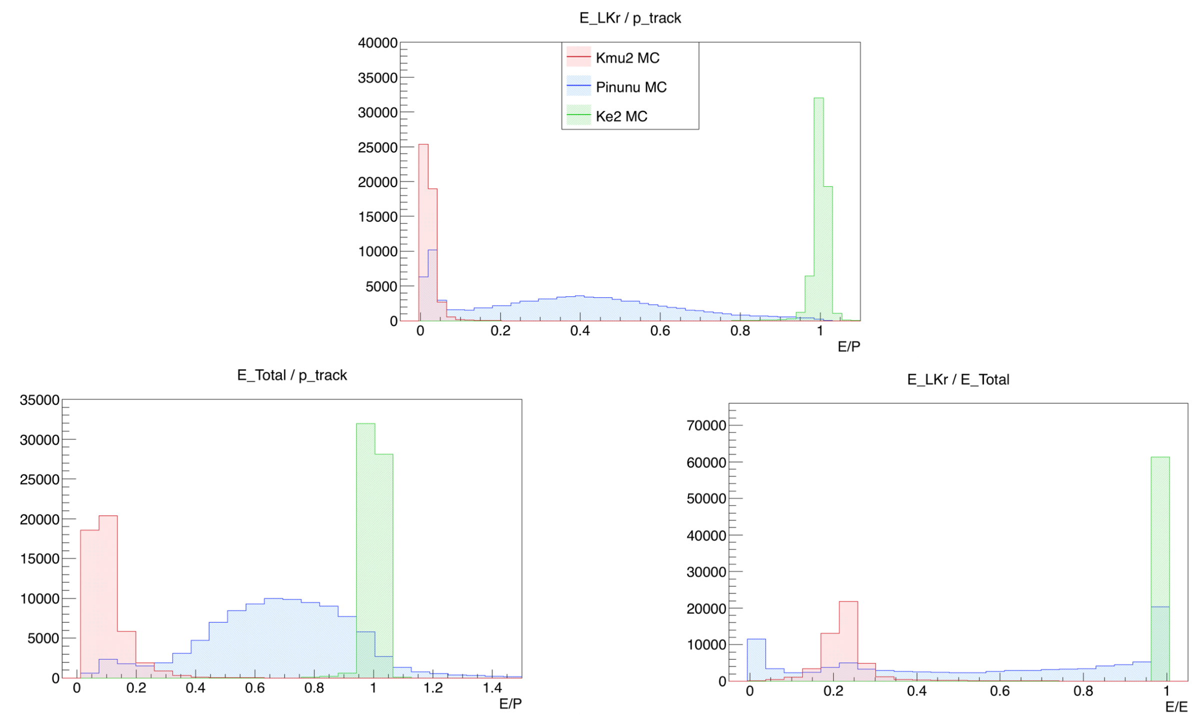


Figure 3: Monte Carlo simulated Energy Ratios for Muons, Electrons, and Pions.

## 3 The Decision Tree

Root provides a toolkit for Supervised Machine Learning (SML) techniques called TMVA (Toolkit for Multivariate Data Analysis) in C++. This toolkit hosts a large variety of multivariate classification and regression algorithms. The applications of machine learning are still limited in physics, however, methods such as Artificial Neural Networks or Boosted Decision Trees have become more common.

Decision Trees (DTs) turn complex problems like classification of events into consecutively smaller tasks. Each node in the DT consists of a discriminating variable which is further split into more nodes as we traverse down the tree. The DT in TMVA uses the Gini Index as a default, which measures the purity of a node, to determine which discriminating variable to split on when constructing the model. The Gini Index is defined as [4]:

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [\, p(i|t) \,]^2. \tag{1}$$

The DT is then built hierarchy until reaching a pure node (class probability = 1), which is a leaf node that provides the event classification. Often times, it is best to prune the DT to avoid over-fitting of the training data and therefore improve the performance of the classifier. This results in having impure nodes as the leaf nodes, which outputs relative probabilities for every class type that sum up to 1. The relative probabilities are the likelihood by particle type produced by the BDT algorithm.

## 3.1 Decision Tree Forest

The development of a DT has drawback of being extremely sensitive to statistical fluctuation of the training dataset, which leads to over-fitting. Over-fitting happens when a DT learns to recognize a pattern that is primarily based on the training sample and that is nonexistent when looking at the testing set [3]. This results in poor performance of the DT model.

To address this problem, a more effective approach has been developed that combines several DTs to classify an event. The ensemble of a large number of DTs is called a forest. By employing the decision tree forest, the classification of an event is no longer based on a single tree, but instead, is obtained as a majority vote among a large number of decision trees. The impact of statistical fluctuation is significantly reduced leading to a more stable classifier.

## 3.2 Boosted Decision Tree

The Boosted Decision Tree (BDT) algorithm is based on the weighting of several cut analyses [3]. The theory of boosting states that the best performance is achieved by combining several weak classifiers rather than few strong ones. Therefore, each DT in a forest should be constrained to a limited number of node splits.

Gradient boost is based on a binomial log-likelihood loss function. It is implemented in the TMVA package and extended to a multi-class classification [3]. The binomial log-likelihood loss function is defined as:

$$L(F, y) = ln(1 + e^{-2F(\mathbf{x})y}). \tag{2}$$

The Multi-class gradient boost is used in the development of the particle identification algorithm.

## 3.3 Implementation of the Boosted Decision Tree Algorithm

The TMVA framework provides several parameters to tune the BDT model. The variation of those parameters results in change in the BDT algorithm performance. The optimal BDT configuration is obtained by varying the following parameters:

- NTree: This parameter represents the number of trees built in the forest. The optimal value of this parameter is NTree = 300, which is determined by cross-validation.
- MaxDepth: This parameter represents the maximum depth of a DT in a forest. The optimal value of this parameter is MaxDepth = 3, which is determined by cross-validation.

- nCut: This parameter represents the number of events tested to split the tree. The optimal value of this parameter nCut = 20, which is suggested by the TMVA User Guide [2].
- MinNodeSize: This parameter represents the minimum number of training events in percentage required in a leaf node. The optimal value of this parameter MinNodeSize = 5
- UseBaggedBoost = True.
- BaggedSampleFraction: This parameter represents Relative size of bagged event sample to original size of the data sample. The optimal value of this parameter is BaggedSampleFraction = 0.50, which is determined by cross-validation.
- BoostType = Grad (gradient).
- Shrinkage: This parameter represents the learning rate for BDT algorithm. The optimal value of this parameter is Shrinkage = 0.10, which is determined by cross-validation.

The TMVA framework provides Graphical User Interface (GUI) macro to display training, testing, and evaluation results for a classification Task. The classification GUI provides many functionalities including training analysis information such as: linear correlation matrices for the input variables, correlation ratios, variable ranking and many more [2]. Table 1 shows variable ranking for the classification task that was determined by the BDT algorithm.

**Variable Importance Ranked by the BDT Algorithm**

| Rank | Variable | Variable Importance |
|---|---|---|
| 1 | $E_{Total}/P_{track}$ | 2.097e−01 |
| 2 | Muon RICH Likelihood | 1.387e−01 |
| 3 | $E_{LKr}/E_{Total}$ | 1.272e−01 |
| 4 | $E_{LKr}/P_{track}$ | 1.255e−01 |
| 5 | Pion RICH Likelihood | 9.249e−02 |
| 6 | Pion RICH Ring Hits | 9.240e−02 |
| 7 | Electron RICH Likelihood | 8.012e−02 |
| 8 | Muon RICH Ring Hits | 7.176e−02 |
| 9 | Electron RICH Ring Hits | 6.211e−02 |

Table 1: The input variables are ranked by the BDT algorithm based on the frequency the variable is used to split the decision tree nodes.

# 4 Performance

To evaluate the BDT algorithm, the performance was tested in two different ways:

1. Creating a testing set from MC simulation and computing the classification accuracy for the model, accuracy by class, error rate, and creating a confusion matrix.
2. Using raw data gathered from previous experiments and running it through different filters.

## 4.1 MC Simulation Performance

For the testing set, we simulated $\sim 61,000 Ke2tracks(electrons), \sim 48,000 K\mu 2$ tracks (muons), and $\sim 53,000 K\pi\nu\nu$ tracks (pions) to evaluate the BDT model. Like the training data, the testing data contains the class label for every event and their discriminating variables. The evaluation of the performance of the

**Confusion Matrix for the BDT Algorithm**

| TMVA Decision | Muons | Pions | Electrons | N |
|---|---|---|---|---|
| K$\mu$2 | 99% | 1% | 0% | 48,297 |
| K$\pi\nu\nu$ | 4% | 96% | 0% | 53,106 |
| Ke2 | 0% | 1% | 99% | 61,211 |

Table 2: The BDT algorithm makes the correct decision 99% of the time for muons and electrons, and 96% of the time for pions.

BDT model was based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as the confusion matrix [4]. Table 2 shows the confusion matrix of the BDT algorithm.

Although a confusion matrix provides information needed to determine how well a model performs, summarizing the information with a single number would it make it more convenient to compare the performance of different models. This can be done by using other performance metrics such as accuracy and error rate, which are defined as:

$$\text{Accuracy}_{BDT} = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = 98\%.$$

$$\text{Error Rate}_{BDT} = \frac{\text{number of wrong predictions}}{\text{total number of predictions}} = 2\%.$$

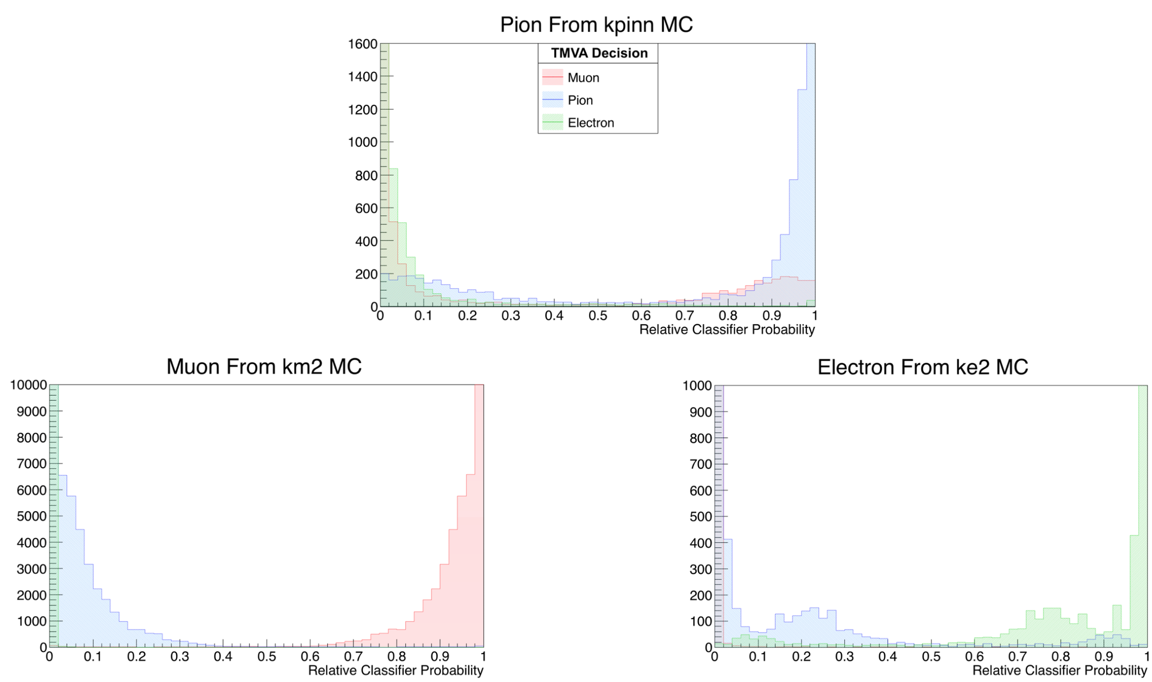


Figure 4: Distribution of relative class probabilities for Muons, Pions, and Electrons produced by the BDT algorithm.

As mentioned in section 3, the output of the BDT classifier is a relative probabilities for every class type. The BDT classifier determines the class type for an event by comparing the relative probabilities of the three classes and assigning the event with the class that has the highest probability among the others. The distribution of the relative probabilities for the three classes are shown in Figure 4.

### 4.2 Raw Data Performance

The performance of the BDT algorithm was tested by running the algorithm on raw and unfiltered data. The BDT algorithm classified each event and the Missing Mass Squared plots were created for each of the filters:

- K$\mu$2: The dataset was mostly classified as muons by the BDT algorithm as seen in Figure 5.
- K$\mu$3: The dataset was mostly classified as muons by the BDT algorithm as seen in Figure 6.
- K2$\pi$: The dataset was mostly classified as pions by the BDT algorithm as seen in Figure 7.
- Ke3: The dataset was mostly classified as electrons by the BDT algorithm as seen in Figure 8.
- $\pi\nu\nu$: Figure 9 combines the 4 filters above. The red peak on this histogram is K$\mu$2 data, the red bump to the right is the K$\mu$3 data, the blue peak to the right is the K2$\pi$ data, and lastly, the green bump to the far right is the Ke3 data.

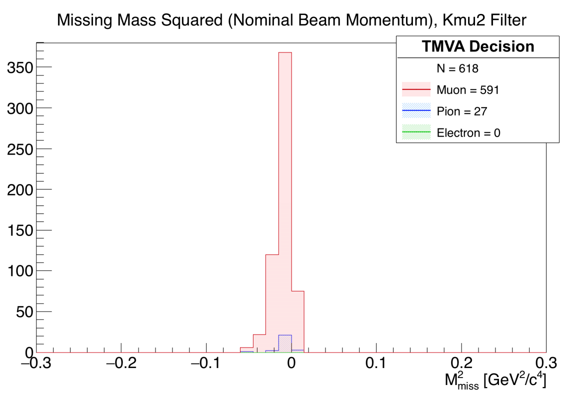


Figure 5: Distribution of the Missing Mass Squared for K$\mu$2.

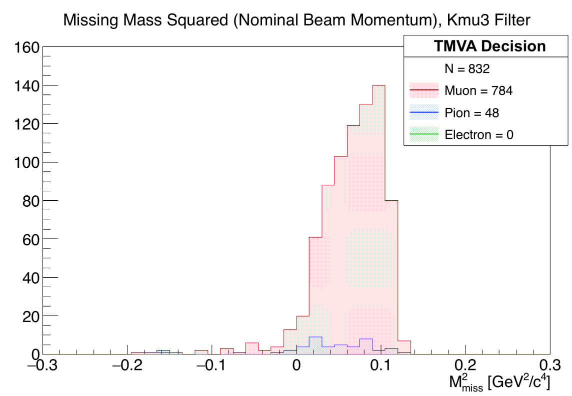


Figure 6: Distribution of the Missing Mass Squared for K$\mu$3.

Figure 10 shows the single-track momentum distribution for the $\pi\nu\nu$ data. The BDT classifier classified the data into muons, which is the red distribution, pions, which is the peak distribution, and electron, which is the green distribution to the left.

# 5 Conclusion

The NA62 experiment is primarily a search for an extremely rare kaon decay $K^+ \to \pi^+\nu\bar{\nu}$ that occurs once in a 10 billion transitions. This measurement requires a strong suppression of the background. In particular, the $K^+ \to \mu^+\nu_\mu$ decay that represents one of the major sources of background for the measurement[3]. Therefore, a particle identification algorithm is a crucial requirement for the L2 trigger.

A particle identification algorithm has been developed exploiting the BDT technique that combines information from multiple detectors. The free parameters of the algorithm have been tuned in order to achieve optimal performance. The BDT achieved 96% pion accuracy, 99% electron accuracy, and 99% muon accuracy.

In order for the BDT algorithm to be integrated into the L2 trigger framework, the algorithm needs to

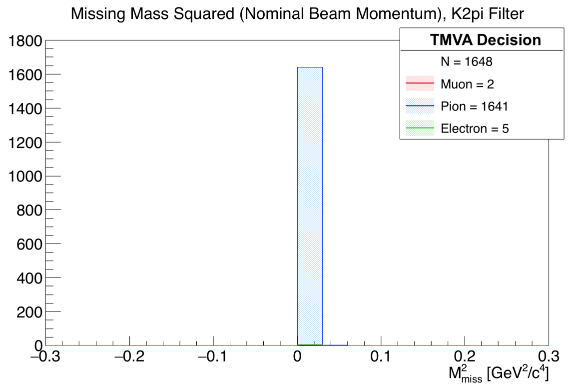


Figure 7: Distribution of the Missing Mass Squared for K2$\pi$.

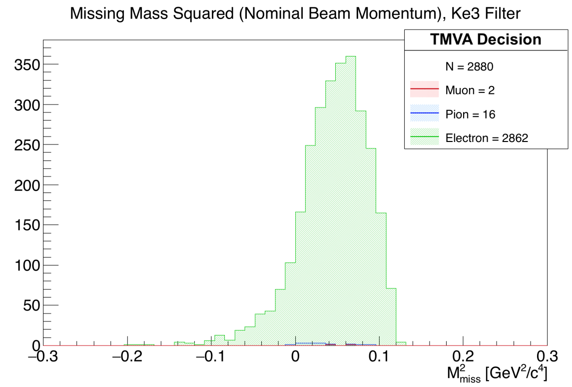


Figure 8: Distribution of the Missing Mass Squared for Ke3.

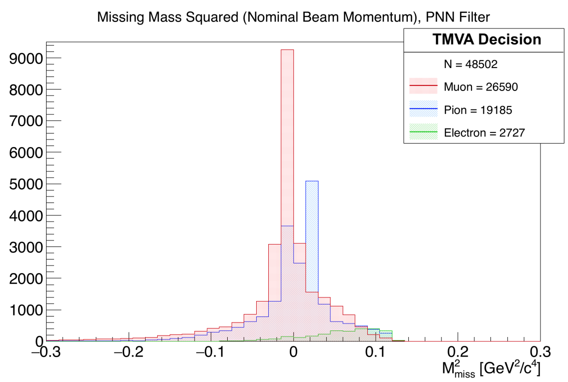


Figure 9: Distribution of the Missing Mass Squared for K2$\pi$.

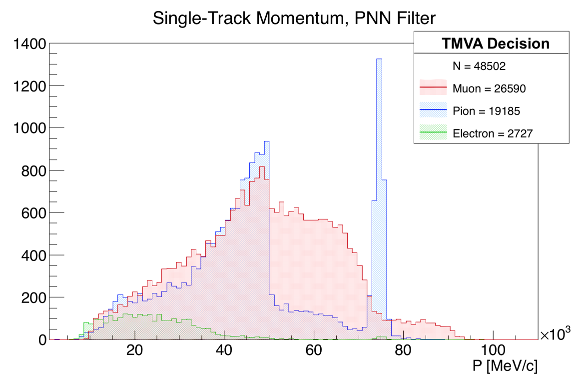


Figure 10: Distribution of the Missing Mass Squared for Ke3.

be optimized and modified to classify events online and in real time. Additionally, we have yet to know what information will be available for the L2 trigger, therefore, some of the discriminating variables might need to be modified or replaced. Computing the RICH Likelihood discriminating variable is especially time intensive and might not be available for the L2 trigger. This means that it might not be used as a discriminating variable in the BDT model. Thus, the BDT model needs to be retrained and reevaluated before integrating it into the L2 framework.

## References

[1] M. Boretto. *The Data Acquisition System of the Na62 Experiment at Cern.* EPJ Web of Conferences, 214:01037, 01 2019.

[2] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, M. Backes, T. Carli, O. Cohen, A. Christov, D. Dannheim, K. Danielowski, S. Henrot-Versille, M. Jachowski, K. Kraszewski, A. Krasznahorkay Jr., M. Kruk, Y. Mahalalel, R. Ospanov, X. Prudent, A. Robert, D. Schouten, F. Tegenfeldt, A. Voigt, K. Voss, M. Wolter, and A. Zemla. *TMVA - Toolkit for Multivariate Data Analysis*, 2007.

[3] A. Riccardo. *Particle Identification with Calorimeters for the Measurement of the Rare Decay* $K^+ \to \pi^+ \nu \bar{\nu}$ *at NA62.* PhD thesis, Mainz U., 2019.

[4] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Pearson, 2006.