UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

Computer Science Honours
Final Paper
2015

Title: Online Particle Identifier Number Generator Algorithms created with Artificial Neural Networks for Particle Identification.

Author: Ryan Wong

Project Abbreviation: ALICE RAD

Supervisor: Assoc. Prof. Patrick Marais

Category	***Min***	***Max***	***Chosen***
Requirement Analysis and Design	0	20	0
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	20
System Development and Implementation	0	15	5
Results, Findings and Conclusion	10	20	20
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		10
Adherence to Project Proposal and Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	0
Total marks		**80**	**80**

Online Particle Identifier Number Generator Algorithms created with Artificial Neural Networks for Particle Identification

Ryan Wong
University of Cape Town
Computer Science
Cape Town, South Africa
ryan.wong@alumni.uct.ac.za

ABSTRACT

The ALICE Transition Radiation Detector (TRD) at CERN is planned for a major upgrade in 2017/18. One of the main tasks of the ALICE TRD is the identification of particles, mainly electrons and pions. The upgrade will increase interaction rate of particles, which means new particle identification algorithms need to be developed. The particle identification algorithms are divided into online and offline algorithms. The online algorithms are used to determine the most effective data to be stored by the TRD while the offline algorithms use the data to determine the classification of the particles.

This paper focuses on the development of the online algorithm to create an 8-bit value known as a Particle Identifier (PID) number. The PID number is used to represent a particle's tracklet obtained from the TRD. A simple approach to create these PID numbers is with the summation of the tracklet data. The summation approach has limitations in the distribution of PID numbers and pion efficiencies. This paper identifies an alternative approach with the use of Artificial Neural Networks (ANNs) as a more effective algorithm. The neural networks are trained with the backpropagation algorithm to create an effective network to determine PID numbers for tracklets. Various neural networks are tested with inputs from unprocessed data and preprocessed variables. Results show that the ANNs produce an improved pion efficiencies and PID distributions than the summation approach when tested with simulated data created with AliRoot. The results obtained show that use of ANNs has the potential to be a viable approach for online algorithms.

Keywords

Artificial Neural Network, Feature Extraction, Particle Classification, Backpropagation, AliRoot

1. INTRODUCTION

CERN, European Organization for Nuclear Research, is the largest particle physics laboratory in the world. The fundamental structure of the universe is studied at CERN. Most of the experiments involve the operation of the Large Hadron Collider (LHC) [1]. The LHC is a powerful particle accelerator which allows two high energy particle beams to travel at close to the speed of light before they are made to collide [7]. The collider was built to allow physicists to test the predictions of theories which involve particle physics, high energy physics and other unsolved questions of physics. Seven detectors have been constructed at the LHC which range between general purpose particle detectors to very specialised detectors. A Large Ion Collider Experiment (ALICE) is a heavy-ion detector in the LHC ring which contains three main cylindrical sub-detectors: the Inner Tracking System (ITS), Time Projection Chamber (TPC) and Transition Radiation Detector (TRD). The purpose of ALICE is to observe the physics of strongly interacting matter and the quark-gluon plasma at extreme values of energy density.

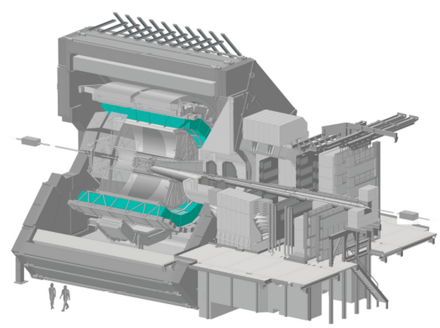

Figure 1: An illustration of ALICE. The highlighted region represents the Transition Radiation Detector (TRD).

This paper focuses on the Transition Radiation Detector illustrated in Figure 1, which is the outermost of the three main sub-detectors. The TRD is used to identify electrons and pions which are detected from the emission of transition radiation [3]. In 2017/18 the ALICE detector is scheduled for an upgrade which increases the interaction rate of the particles. After the ALICE upgrade, new algorithms need to be implemented for particle identification. These algorithms are divided into online and offline algorithms. The online algorithms are used to reduce the data volume that will be transferred to permanent storage while the offline algorithms use the data, calculated by the online algorithms, for the classification of particles.

The online algorithms are affected by the upgrade as the amount of data stored by the electronics in the TRD need to be significantly reduced. These algorithms need to be efficient due to the increase in the interaction rate but still provide enough information for the offline algorithms to produce comparable particle identification capabilities as with the raw TRD data. The aim of this paper involves the identification of effective and efficient online algorithms. In this paper the limitations of a summation approach to create an online algorithm is determined and explores the use of neural networks to improve on the summation approach while ensuring the algorithm remains efficient. The efficiency of the algorithm is ensured by limiting the complexity of the neural network through tests with different input types. The hypothesis of this paper is that the use of Artificial Neural Networks for an online algorithm produces better pion efficiencies and PID distributions than the summation approach. An additional question of this paper is, what topology of a neural network has the greatest potential for an online algorithm.

The paper is organised as follows. Section 2 discusses the background and related work by highlighting the TRD data format, the AliRoot framework and features of the particles that need to be identified. The concepts of Artificial Neural Networks are introduced and current applications of neural networks in particle identification are explained. Section 3 then describes the framework built to create the software for experiments with Artificial Neural Networks, followed by Section 4 which gives an overview of the methods used to create the online algorithms with the framework. The results from the tests with the summation and Artificial Neural Network approach are analysed in Section 5 with Section 6 comparing the approaches. Section 7 rounds off the paper with conclusions and future work.

2. BACKGROUND AND RELATED WORK

In this section, details of the Transition Radiation Detector data format and the features of the particles are explained. The concepts of Artificial Neural Networks and their applications in particle identification are also described.

2.1 Transition Radiation Detector Data Format

The TRD is composed of 18 super-modules which consist of 5 stacks. Each of these stacks consists of 6 chambers, which corresponds to 540 chambers in total. The chambers measure the charge that is deposited on the readout pads, which are organised into 144 columns which have either 12 or 16 rows, depending on the type of chamber [20]. For each event, every pad is sampled up to 30 times. The data that is sampled is the an Analogue-to-Digital (ADC) signal. For this work, we used 27 of these samples known as time-bins. The breakdown of the TRD layout is presented in Figure 2.

Particles which go through the stack in the TRD leave behind a track. A particle can cross up to six chambers, leaving a short track segment or tracklet in each of them. A tracklet consists of four pad-columns which best represents the particle. The number of ADC values stored in the MCMs, to represent the particle's tracklet, is $27 \times 4 = 108$, where each ADC value is 10-bits.

Currently the TRD has two modes which are used by the readout electronics: tracklet mode and raw readout mode. The raw readout mode has no practical limitations but requires signal handshaking which has high overhead. The tracklet mode avoids the handshaking latency but is limited to the read-out of four 32-bit words for each MCM [13]. The tracklet mode can perform approximately 120 clock cycles.

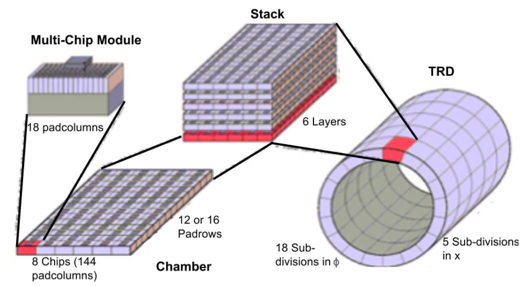

Figure 2: An illustration of the breakdown of the TRD.

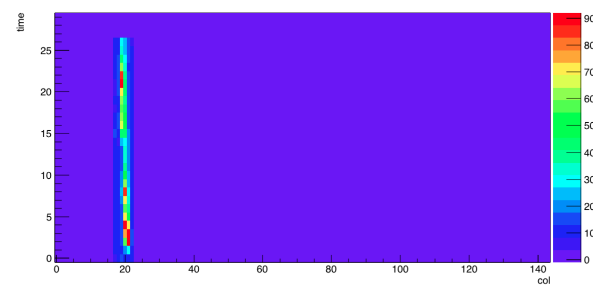

Figure 3: A 2D histogram representing the ADC values recorded from single pad-row in a chamber (144 pad-columns with 27 time-bins). The histogram contains one tracklet which is situated around the 20th pad-column.

The proposed approach for the TRD upgrade is to eliminate the raw readout mode and use 8-bits from the tracklet mode to store details about the tracklet [10]. The remaining bits are used to store other information about the tracklet such as the position, angle and the momentum. The 8-bit value is called the Particle Identifer (PID) number. The PID number is planned to represent the tracklet information and replace the 108 ADC values stored in the MCMs. The PID numbers for the particles are then used by the offline algorithms for particle identification instead of the raw tracklet ADC information. The purpose is to reduce the volume of the data stored and time it takes to calculate the data to be stored, to allow the TRD to process more particles from the increased interaction rate [4].

2.2 AliRoot Framework

The official software framework used within ALICE is AliRoot [9]. AliRoot is written in C++ and is based on Root, which is a large scale Object-Oriented data analysis framework. It is used for data simulations, event reconstruction, detector calibration, detector alignment, visualization and data analysis. The generation of particles with AliRoot are achieved with Monte Carlo methods to simulate the results produced by the physical detectors at CERN. Monte Carlo methods are a broad class of computation algorithms that rely on repeated random sampling to obtain numerical results and are often used in physical and mathematical problems [14]. Further details on how the particles are simulated are explained in the official AliRoot documentations [8].

2.3 Artificial Neural Networks

An Artificial Neural Network (ANN) is a processing algorithm which uses statistical learning that is inspired by the biological nervous system [21]. ANNs have various applications in pattern recognition and data classification through learning processes. ANNs are generally used to identify trends or patterns that are too complex to be noticed by humans or other computer techniques.

The main components of ANNs are neurons and synapses. A neural network is composed of a number of neurons which are small processors that are limited to simple calculations. The connection between neurons are the synapses. The synapse each have their own weight. The intensity of the weights indicate the affect a neuron has on a network. ANNs are usually represented as directed graphs where the neurons represent vertexes and synapses are edges with direction. Neural networks have three types of layers: the input layer, hidden layer and output layer. The neurons in the input layer is used to insert the data into the network which then undergoes a forward pass through the synapses to the hidden layers and finally to the output layer where the results are extracted from the output neurons.

There are three major learning paradigms: supervised learning, unsupervised learning and reinforcement learning [5]. The focus of this paper investigates a supervised learning technique which uses training data to create a function that produces desired outputs. The training data consists of the inputs and target outputs. The learning algorithms use the training data a produce an inferred function. One of the well-known algorithms for training neural networks is backpropagation [15]. The backpropagation algorithm attempts to find a function that best maps a set of inputs to its target output. The aim of the backpropgation algorithm is to minimise the error when training. The algorithm for backpropagation is highlighted in Algorithm 1 [23]. The backpropagation is a slow learning process which updates the network with one training sample at a time. The first step in learning involves a forward pass of the network with the training sample as input. The output is then compared to the target output and the difference calculated is the error. The next step involves a back pass which computes the new weights (W_{new}) for the network with Equation 1 based on the error, learning rate (LR), output (0) and momentum (M). The learning rate affects the speed at which the ANN adjusts its weights while the momentum is used to prevent a convergence to local minimum or non-optimal solution.

$$w_{new} = w_{old} + LR \cdot O \cdot error + M \cdot \Delta w \tag{1}$$

The process is repeated with the training data until a stopping criteria is satisfied.

Algorithm 1 Backpropagation Algorithm

initialize *network* with random weights
do
  for item in training data
    Forward pass: through *network* $\rightarrow$ output
    error = target - output
    Back pass: compute Δw for all weights
    update *network* weights
until stopping critera satisfied
return *network*

2.4 Data Features

The particle's tracklet data is used to distinguish between electrons and pions. The tracklet data is simplified to distinguish these differences by combining the four pad-columns for each time bin into one summed ADC value. Equation 2 shows the calculation used to combine the four pad-columns for 27 time bins.

$$sum(tb)_n = \sum_{i=0}^{4} tb_{n,i} \quad n = \{0, 1, 2, ..., 26\} \tag{2}$$

Figure 4 shows the average summed ADC value across the 27 time bins for electrons and pion. Several features can be distinguished. One of the major differences is the total sum of all the ADC values across all the time bins are greater for electrons than pions. Additional differences observed are that electrons create two ADC peaks and pions create one ADC peak. Both electrons and pions have a peak near the first few time bins but the electrons contain an additional peak around the last few time bins.

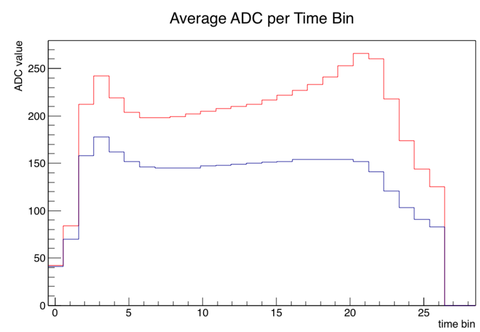

Figure 4: Histogram representing the average ADC signal for each time bin. Electrons are represented in red and pions in blue. The average result is obtained from 2000 particles

The different data features of electrons and pions are obtained from the average result from two thousand particles. Individually many of the particles do not have these distinct features. An electron or pion tracklet individually produces completely different results to the average case and in some cases pions may appear to have electron properties and vice versa. The results vary due to different energy losses and photon emissions which cannot be accurately tracked by the TRD. Further details about the causes of the varying tracklet results is discussed by Wilk [24]. It is important to identify the relevance of the features to distinguish between electrons and pions.

2.5 Particle Identification with ALICE TRD

Multiple approaches have been explored for particle identification with ALICE TRD. Some of the classical methods of particle identification include Truncated Mean [2], Cluster Counting [19] and Likelihood on Total Deposited Charge [6].

Recently two approaches have been experimented with Artificial Neural Networks which saw improved results in particle identification compared to the classical methods. The first approach uses time slices extracted from the raw ADC signals [24] and the second approach uses preprocessed variables [11].

2.5.1 Time Slices Approach

The simplest approach with ANNs use the ADC values from the tracklets and divides them into groups called time slices. The application of these networks saw improved results compared to the Likelihood on Total Deposited Charge method. The network composed of inputs for six time slices was trained and tested with 2002 test input data from four ALICE TRD prototype chambers [12].
A network implemented by AliRoot makes use of eight time slices as inputs and two hidden layers, with fifteen and seven hidden neurons respectively. The number of output neurons depend on the number of classifications. In the case of identifying electrons and pions, two output neurons are used.

2.5.2 Preprocessed Variables Approach

The second approach uses preprocessed variables as inputs to the neural network. The preprocessed variables are determined by the data features observed for each type of particle. Five features have been identified by Kim et al. to distinguish the difference between electrons and pions [11]. These features include the number of clusters above a high threshold, the number of time bins above a low threshold, the time bin with the largest amount of charge, the deposited charge of the second largest cluster and the integrated charge below a low threshold. The topology for the ANN uses one hidden layer with one hidden node and one output. The topology was found to be very successful in separating electrons from pions in simple Monte Carlo simulations [17]. However, the approach had not been evaluated with test beam data or AliRoot simulation.

2.5.3 Pion Efficiency

The accuracy of the algorithms for electron identification are measured with the pion efficiency. The pion efficiency is the percentage of pions misidentified as electrons for a given electron efficiency. A typical electron efficiency of 90% is used. The pion efficiency is found by identifying the area in which 90% of the electrons are most likely to be situated then calculating the fraction of pions situated in the area. Figure 5 highlights the area with 90% of electrons where the pion efficiency is calculated by determining the fraction of pions in this area.

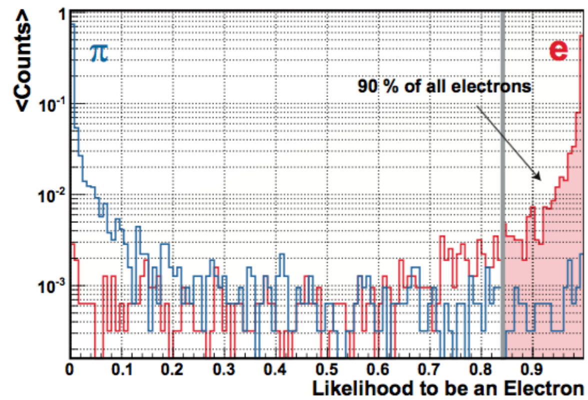

Figure 5: Histogram representing the frequency of particles with the likelihood to be an electron.

3. FRAMEWORK

The goal of this work is to create software that experiments with potential avenues for a PID generator algorithm. The software is developed in C++ to allow for compatibility with AliRoot. The software is developed with clear arguments, methods and naming conventions for easy modifications and transferability. The backpropagation algorithm for training the neural networks can be viewed as a blackbox and extensive knowledge in neural networks is not required when using the software. The results from training record the pion efficiencies, PID distribution and errors of the networks from trained and untrained data.

3.1 Backpropagation

The backpropagation algorithm is applied to adjust the weights of the neural network. The ANN goes through 10 000 iterations of the backpropagation algorithm with the training data as input. The number of iterations, learning rate and momentum can be adjusted but for the experiments conducted the variables are set to 10 000, 0.5 and 0.7 respectively. These numbers are chosen to ensure the neural network is given enough time to learn and find a general trend on the performance of a network when compared to other networks. In all cases the inputs are normalised such that the inputs are between 0 and 1. The inputs are normalized to improve the performance of the learning process [16]. Due to the normalisation, all the weights are randomly initialised according to a Gaussian distribution. The Gaussian distribution has standard deviation of 1 with the mean of 0. The neurons each used the sigmoid function as their activation function which is represented in Equation 3. The sigmoid function is used as it is considered more biologically realistic than other activation functions [18].

$$\phi(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The conventional validation method is used by training the ANN on 80% of the tracklets and the remaining 20% are used to determine the accuracy of the neural network on the unseen data. The time it takes to train the ANN depends on the amount of generations, size of the neural network and the size of the training sample, for approximately 4400 tracklets take between 1 and 2 hours to train for 10 000 generations.

3.2 Data Extraction from AliRoot

In this paper, analysis and tests are performed on particles created from simulations with the AliRoot framework. A total of 2000 particles were simulated. The results from the simulations assumed that the particles' tracklets did not overlap.
The process of extracting the tracklet information involve several steps which require both the AliRoot's built-in functions and external functions. The tracklet is retrieved by using AliRoot's built-in functions to return a pad-column, which gives the general area of the tracklet. The pad-column is used to obtain ten surrounding pad-columns. The track-

let is obtained from these ten pad-columns by choosing the four pad-columns with the highest consecutive total summed ADC values. In the ALICE TRD, the method to obtain the tracklet differs as the ADC values goes through a preprocessor to find the appropriate pad-columns for the tracklet.
Out of the 2000 particles generated approximately 5500 tracklets were extracted. The online PID generation algorithm treats each of the tracklets as the particle it is associated with. Therefore the online PID generation algorithm assumes that each particle only creates one tracklet. The offline particle identification algorithms use the collective PID numbers associated with a particle to determine the classification of the particle. For the remainder of this paper, the particle's tracklets are treated independently of each other. Therefore each particle is assumed to have one tracklet and each tracklet is assigned their associated particle type.

4. IMPLEMENTATION

The aim of this work is to create an efficient algorithm that minimises the pion efficiency at a fixed electron efficiency of 90%. The pion efficiency is be minimised to reduce the number of particles misidentified by the offline particle identification algorithms. The online algorithm to create PID numbers needs to be efficient as it must be completed in the tracklet-mode to remove signal handshaking. Therefore the number of operations for the online algorithm is limited to around 100 arithmetic operations due to the limitations of the number of clock cycles in the tracklet-mode.

In the follow section, two approaches are discussed to create the online PID generator algorithm. The first approach focuses on the summation of ADC values and the second involves the use of Artificial Neural Networks.

4.1 Summation Approach

The simplest approach to create an algorithm that calculates PID numbers for tracklets is a summation approach. The summation approach uses the total sum of the ADC values in a tracklet. The summation of the ADC values is one of the major features that makes electrons different to pions. As discussed in Section 2.4, electrons tend to produce higher ADC values across all time bins than pions. This data feature is used to produce a 8-bit PID number for particles by taking summation of the ADC values across all time bins (sum_p) and convert it to a number between 0 and 255 (8-bits). In order to reduce the summation value to a number between 0 and 255, the maximum (sum_{max}) and minimum (sum_{min}) possible summation of ADC values for an electron and pion tracklet is required. The PID number for a tracklet is be calculated with Equation 4.

$$PID = \frac{sum_p - sum_{min}}{sum_{max} - sum_{min}} \times 255 \tag{4}$$

4.2 Artificial Neural Network Approach

As discussed in Section 2.3, ANNs have applications in data classification through learning processes. The neural networks adapt to inputs by tuning numeric weights based on experiences. The idea of the ANN is to create a network of weighted sums, which produces a PID number from given inputs. The PID number is calculated by completing a forward pass of the neural network with the tracklet data as input.

Several neural network topologies are considered for the construction of a PID generator. These topologies are divided into two sets: a networks with one output and networks with two outputs. The two sets indicate different formats for the PID numbers.
The network with **one output** treats the output signal near 1 as most likely to be electrons and 0 as pions. The output is converted into a PID number by multiplying the output calculated by 255. Therefore, electron PID numbers are expected to be around 255 and pion PID numbers are expected to be around 0. This approach used all 8-bits to create a single PID number.
The network with **two outputs** creates two 4-bit PID numbers. Each 4-bit PID number represents the type of particle, either electron (PID_e) or pion (PID_p). The aim is for electrons to produce signals of around 1 for PID_e and around 0 for PID_p and the pions to produce signals of around 0 for PID_e and around 1 for PID_p. In both cases the signal is converted to a 4-bit number by multiplying the output by 15.

Both of the topologies are tested with a set of four different input types. The input types are a mixture of preprocessed variables and unprocessed tracklet ADC information. The various inputs are used to test the effectiveness and extent of simplification of the raw data and the possibility of using the features discussed in Section 2.4 as inputs into the neural network.

Due to limitations of the tracklet mode, the complexity of the neural networks are limited to around 100 arithmetic operations. Therefore there are a maximum number of hidden nodes for each network topology. The calculation shown in Equation 5 is used to determine the maximum number of hidden nodes (H) for the different number of inputs (I) and outputs (O).

$$100 = \underbrace{(I)}_{\text{input}} + \underbrace{[(I \cdot H) + (H \cdot 2)]}_{\text{input to hidden}} + \underbrace{[(H \cdot O) + (O \cdot 2)]}_{\text{hidden to output}} \tag{5}$$

For simplicity reasons, the calculations in the neurons are assumed to have two operations and the processing of the tracklet data before being stored in the input neurons are not considered. Two operations are selected for calculations in the neuron as the neuron's main calculation is the activation function. The activation function used in this work is the sigmoid function but it is possible to use a much simpler function which is limited to two operations. The number of hidden layers is limited to one as one hidden layer is sufficient for the majority of complex problems [22].

The next components of the paper discusses the four input types used to create PID generators with ANNs. The number of inputs tested are 27, 9, 5 and 8. Table 1 shows the maximum number of hidden nodes calculated by Equation 5 for each input.

Table 1: Maximum Number of Hidden Nodes for Each Input Type

Outputs \ Inputs	27	9	5	8
1	2	7	11	8
2	2	6	10	7

27 Inputs: Time Bins.

The simplest neural network approach uses 27 time bin ADC values from the tracklet information obtained from Equation 2 as inputs. The network is used to determine the effectiveness of inputs with tracklet data that has undergone little processing. Due to the large number of inputs, the complexity of the network is limited to two hidden nodes.

9 Inputs: Time Slices.

The second method reduces the amount of the input data but allows for a more complex network compared to the 27 inputs method. The increase in the complexity of the neural network creates an increase in weights from the hidden to output layer and thus find an effective PID generator algorithm. The nine inputs are achieved by converting the 27 time bins into 9 time slices by taking the sum of three consecutive time bins. These nine time slices are then used as inputs into the neural network. Nine time slices are chosen as the time slices are equally calculated from three time bins.

5 Inputs: Preprocessed Variables.

The third method uses the features discussed in Section 2.4 as inputs to a neural network. The features are based on the average tracklet results, which are observed in Figure 4. This approach tests the effectiveness of creating PID values based on inputs created by preprocessed variables. The calculations to determine the values of the preprocessed variables only requires simple counting methods and summations which is not as computationally expensive as the preprocessed variable discussed in Section 2.5.2.
Five of the main features used as inputs include:

- The total sum of all charges across all time bins ADC_{sum}.
- The number of peaks created above a threshold T_p.
- The number of time bins above a low threshold T_l.
- The number of time bins above a high threshold T_h.
- The time bin with the highest ADC value $high_{tb}$.

8 Inputs: Preprocessed Variables and Time Slices.

This method combines the inputs of preprocessed variables and unprocessed variables. The inputs are the five features used in the five input method and three time slices. Three time slices are used to keep a balance between the preprocessed and unprocessed variables while ensuring the time slices each have an equal share of time bins. Each time slices is calculated from the sum of nine consecutive time bin ADC values. The three time slices provide a distinction between the electrons and pions by dividing the time bins between the first peak, the constant incline for electrons and the second peak for electrons observed in Figure 4.

5. RESULTS

The results are based on the simulated particles created with AliRoot discussed in Section 3.2. The results recorded for the two approaches are the pion efficiencies and the PID frequency distribution for electrons and pions.

5.1 Summation Approach

The AliRoot simulations from the summation approach produces a pion efficiency of 57.7% at an electron efficiency of 90%. The pion efficiency is very high and it may thus be difficult for particle identification algorithms to determine the classification of the particles. A histogram of the PID frequency distribution is represented in Figure 6. The general trend is for the frequency of the electrons and pions to form a peak in which tapers off. The electrons tend to have a greater PID number than pions as expected.

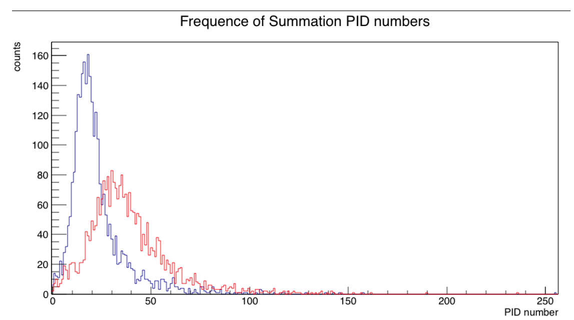

Figure 6: Histogram representing distribution of electron (red) and pions (blue) according to their frequencies of each PID number.

The problem with the summation approach is that the PID numbers tend to be between 0 and 100. Therefore the summation approach does not utilise the full range of the PID distribution. The two peaks in frequency for electrons and pions are close together which make it very difficult to distinguish the between electrons and pions as the frequency of PID numbers have significant overlap as shown in Figure 6.

5.2 Artificial Neural Network Approach

The results from experiments with neural network are based on the 20% of particles that are untrained by the ANN. In addition to the pion efficiencies, the error for the neural network is calculated with the root mean square (RMS) which is then divided by the number of outputs of the network. The equation for the error calculation is shown in Equation 6. The error of the network determines if the neural network is suitable as a PID generator algorithm. A low error of a network indicates that the network has the potential to be used as a PID generator algorithm.

$$error = \sqrt{\frac{\sum_{n=0}^{particles}(T_i - O_i)^2}{particles}} \div outputs \tag{6}$$

27 Inputs: Time Bins.

As shown in Figure 7 the errors for these networks are very high. The limitations of the number of hidden nodes

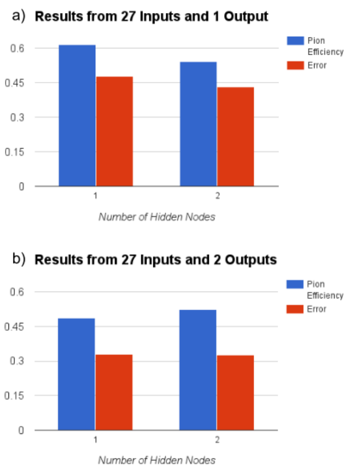

Figure 7: Results from 27 inputs indicating the pion efficiencies (blue) and errors (red) for each hidden node used. (a) shows the results from one output and (b) shows the results from two outputs.

causes an problem in reducing the error of the ANN. The pion efficiency calculated fluctuated significantly with multiple tests, which indicates that the number of weights in the neural network is too low. This problem occurred for both one and two output topologies.

In order to improve the pion efficiency a lower number of inputs is required which allows the complexity of the neural network to be increased and thus identify a better solution for a PID generator algorithm.

9 Inputs: Time Slices.

In both nine input approaches the error is considerably higher for the neural network with one hidden node. The high error is due to the same problem of the 27 input approach where there is not enough weights in the network to determine the features to classify the particles. The errors on the remaining networks as shown in Figure 8 depicts a lower and generally constant value for the different number of hidden nodes, which can be improved by increasing the number of generations to train the ANN. The pion efficiency generally decreases as the number of hidden nodes increases but starts to increase when six hidden nodes are reached for the one output approach. The network with two outputs have pion efficiencies which has a similar trend to the one output method except the decrease stops at three hidden nodes.

5 Inputs: Preprocessed Variables.

The tests conducted with the five input approach used the following parameters for the threshold:

- T_p is set to 70% of the maximum average ADC value for electrons from the training sample.
- T_l is set to average ADC value for the pions from the training sample.
- T_h is set to average ADC value for the electrons from the training sample.

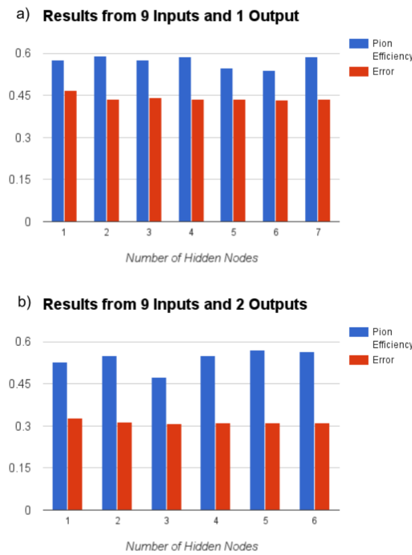

Figure 8: Results from 9 inputs indicating the pion efficiencies (blue) and errors (red) for each hidden node used. (a) shows the results from one output and (b) shows the results from two outputs.

These parameters are chosen to use features from both electrons and pions. In both five input methods the error is high for the initial low number of hidden nodes, but the error slowly decreases to a generally constant error. Figure 9 shows that the pion efficiency generally decreases and then increases at ten hidden nodes for the one output approach. The two output approach has a similar effect with a decrease in pion efficiency until nine hidden nodes.

8 Inputs: Preprocessed Variables and Time Slices.

The experiments conducted for the eight input method used the same parameters for the preprocessed variables as the five input method. As shown in Figure 10 the eight input method has similar errors to the five input approach. The two output method has a pion efficiency at its lowest at two hidden nodes and increases as the number of hidden nodes increases but at seven hidden nodes it decreases again and indicates a possibility of a more effective network at higher hidden number of nodes.

The one output method has its lowest pion efficiency at five hidden nodes, while the pion efficiency increases as the number of hidden nodes increases.

6. DISCUSSIONS

The approaches to create the online PID generators for particle identification are analysed and compared based on their pion efficiencies. These pion efficiencies are calculated from the tests conducted with simulated data from AliRoot. Other factors based on the distribution of the PID numbers for electrons and pions are analysed. The factors include the extent in which the PID numbers are distributed across the range of PID numbers and the frequency of each of the PID number that occurs. These factors are used to determine

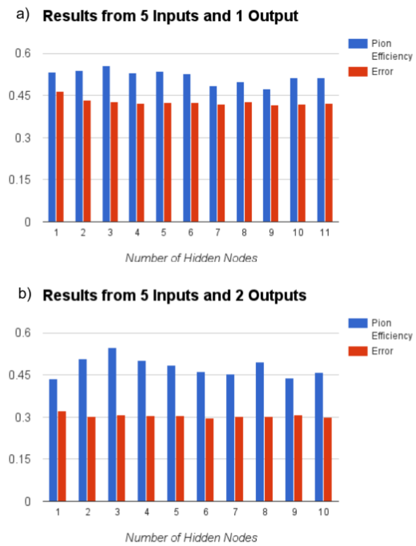

Figure 9: Results from 5 inputs indicating the pion efficiencies (blue) and errors (red) for each hidden node used. (a) shows the results from one output and (b) shows the results from two outputs.

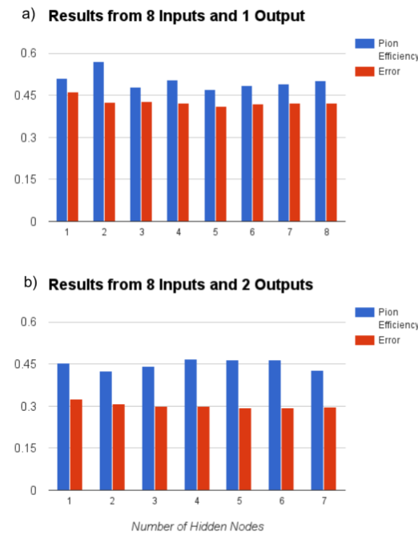

Figure 10: Results from 8 inputs indicating the pion efficiencies (blue) and errors (red) for each hidden node used. (a) shows the results from one output and (b) shows the results from two outputs.

the limitations and potential drawbacks of the PID generator algorithm.

6.1 Summation vs ANN Approach

The summation approach is a simple approach which performs summation followed by a conversion into PID numbers. The approach with neural networks is more complex approach which uses a network of weighted sum, where the weights are based on features observed from training. Both of these approaches have pros and cons. The summation approach is computationally cheap but produces a pion efficiency of 57.7% with AliRoot simulation data which may be difficult for particle identification algorithms to correctly identify the particles. The neural network approach is more computationally expensive but produces a pion efficiency of between 40% and 55% depending on the structure of the network. One of the key features of the neural network approach is that the PID frequency distribution for electrons and pions created peaks on either ends of the PID number range. The frequency distribution is very different to the summation approach as the summation approach creates two peaks that overlap significantly. The frequency distribution of the PID numbers for the summation approach also has an problem in which the PID numbers tend to be between 0 and 100 and therefore not utilise the range of the PID numbers completely unlike the ANN approach. The frequency distribution of both approaches are shown in Figure 11. The improvement of the frequency distribution for the ANN approach is important as the particle identification algorithms can use the feature for easier classification as high PID numbers indicate one type of particle and low PID numbers indicate the other type of particle.

The pion efficiency for the ANN tends to increase due to a problem with misclassification. Therefore in some cases the neural network incorrectly classified particles and formed small peaks in frequency for particles on the opposite end of the target. This problem is potentially due to insufficient weights between the hidden and output layer for the neural network to identify the features of the tracklet. Further details about the misclassification is discussed in the next section.

6.2 Comparison with ANN Methods

The ANN approaches are analysed based on the best pion efficiencies obtained from the different types of inputs. The comparisons are completed on one and two output approaches. The aim is to identify the neural networks that have the most potential for a PID generator algorithm.

6.2.1 Comparisons with One Output

The pion efficiencies for the one output approach performed generally poorly for the networks that used the time bins or time slices as input. The approaches that used preprocessed variables as inputs performed slightly better as shown in Table 2. The preprocessed variables provided the network with additional information which may have increased the speed for the neural network to identify the features. From Table 2 an observation of the five input network showed that it needed a higher number of hidden nodes compared to the eight input network. This is potentially due to the effect of including the three time slices in the eight input method and the information provided by these inputs made up for the additional hidden nodes required for the five input network. This indicates that the time bin information is an important feature in the classification of particles and needs to be used in conjunction with the preprocessed variables.

The neural networks with low numbers of hidden nodes have a limited PID distribution. In many of the networks with a low number of hidden nodes, the PID numbers are

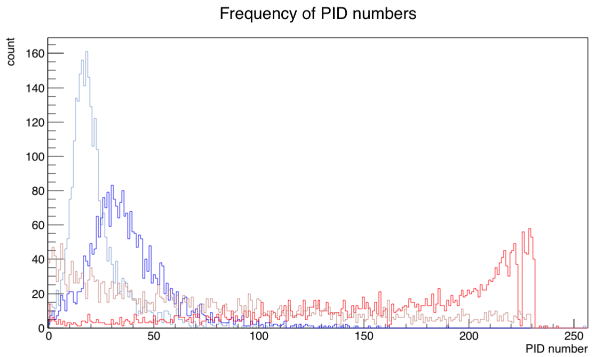

Figure 11: PID frequency distribution of the summation approach (blue) and an example ANN approach (red). The light blue represents the pions and dark blue represents electrons. The dark red represents the electrons and light red represents the pions.

Table 2: Best ANN approaches with one output

Inputs	Hidden No	Pion Efficiency
27	2	54.28%
9	6	53.98%
5	9	47.32%
8	5	46.15%

limited to be between a small range, a sample from nine inputs with two hidden nodes and one output shown in Figure 12 has a limitation of PID numbers between 0 and around 220. The cause of the limited range is due to the number of weights from the hidden to output layer being insufficient to fill the PID number's range. The limitations of the range slowly decreases as the number of hidden nodes increases. There is also a limitation in the how many hidden nodes can be used as increasing the hidden nodes to too high numbers may cause the network to over-fit the training data and therefore only learn the data that it is given. This will cause the pion efficiency to be worse for the untrained data. This problem occurred for the nine input method with two outputs as can be seen in Figure 8 where the pion efficiencies slowly increase as the number of hidden nodes increases. Another problem that occurs that is mentioned in Section 6.1 is that some tracklets are completely misclassificated, therefore small peaks in the frequency distribution are created on the opposite ends of the target PID number. This problem arises when there are too few weights assigned to the network and thereby the network misclassifies the particles. A sample of this problem is displayed in Figure 12 at PID number 0 where many electrons (red) are misclassified as pions (blue).

6.2.2 *Comparisons with Two Outputs*

Table 3 shows the pion efficiencies with two output networks have a similar trend to the one output networks. Networks that use preprocessed variables produce better results than networks with the time bin ADC value inputs. The same observations of the limitations of the one output approach applies to the two output approach even though a smaller distribution of PID numbers are used, since the PID numbers are limited to two 4-bit numbers. These limitations

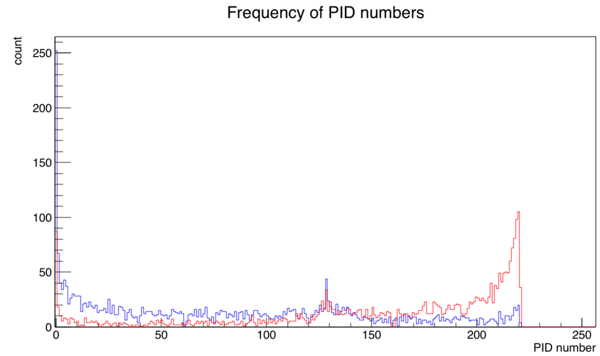

Figure 12: PID frequency distribution of the neural network with nine inputs, two hidden nodes and one output. The electrons are represented in red and pions in blue.

of limited range of the PID distribution and PID misclassification occur with both approaches as the output from the neural network is converted from a signal between 0 and 1 to a PID number with the same method of multiplication with the highest possible PID number (either 15 or 255).

Table 3: Best ANN approaches with two outputs

Inputs	Hidden No	π_{eff}
27	1	48.73%
9	3	47.43%
5	9	43.95%
8	2	42.48%

One of the issues with the two output method is that the two 4-bit PID numbers are correlated. The network produces PID_e and PID_p that are inverses of each other. Therefore the two PID numbers are correlated such that $PID_e = 15 - PID_p$ which causes the diagonal line shown in Figure 13.

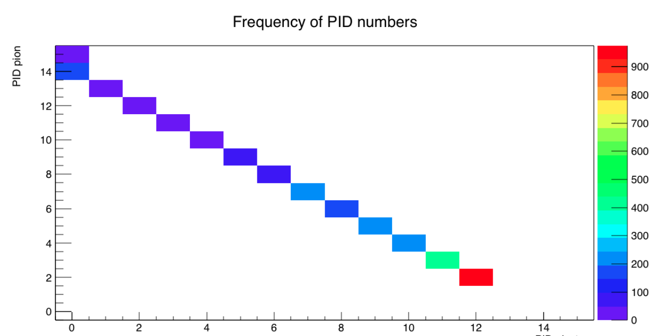

Figure 13: PID frequency distribution with two outputs. The histogram shows the general correlation between the two 4-bit PID numbers

The correlation is due to the low number of hidden nodes used which means that the outputs are based on these few hidden nodes. A example description of this problem is shown in Figure 14. In the case where one hidden node is used, the number of weights from the hidden to the output layer is two. Therefore these outputs are each based on one weight from the one hidden node. The correlation occurs where both outputs just depend on this hidden node and the weight that links them. A method of reducing the correlation is by increasing the number of hidden nodes which will increase the number weights that are connected from the hidden to the output layer. Therefore the output neurons will become more dependent on their independent weights and not on the hidden neurons.

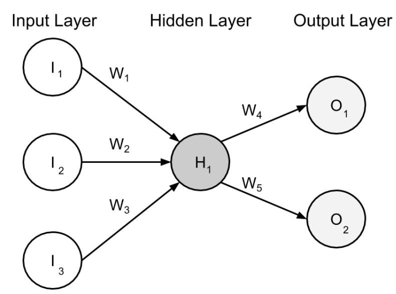

Figure 14: Neural network with one hidden node with two outputs.

6.2.3 Comparison between One and Two Outputs

The results of the AliRoot simulations from the two output networks produces lower pion efficiencies than the one output networks for all types of inputs tested as shown in Figure 15. The two output networks produces better results due to the need to determine two PID numbers. The one 4-bit PID needs to bring the particle closer to its classification while the other 4-bit PID number pushes the particle away from misclassification.
The network that produces the lowest pion efficiency is the eight input networks which combines inputs from preprocessed variables and time slices. This shows that simple preprocessed variables improve the PID generator algorithm but the time slices are still required to further improve the PID generation algorithm.

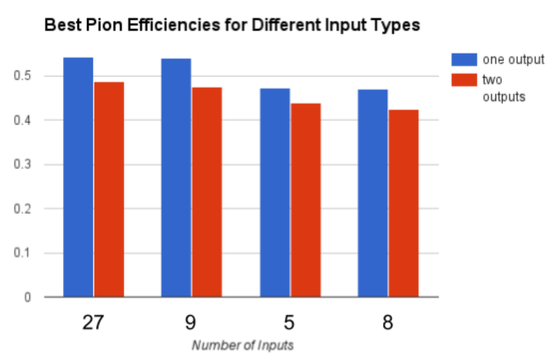

Figure 15: Pion efficiencies of the best networks for each input type. The blue indicates the one output approach and the red indicates the two output approach.

In both cases the number of hidden nodes for the neural network need to be chosen appropriately to ensure that the pion efficiencies are minimised, the PID frequency distribution uses the full range of PID numbers and reduces the number of particles misidentified.

6.3 Overall Comparison

The results from the neural network showed improved results compared to the summation approach. These results are based on the 2000 particles simulated with AliRoot. The limitation of the sample size of the particles for the network may affect the effectiveness of the training. The affect of a single tracklet potentially has a major affect on the weightings of the neural network. Since 80% of the particles are used to train the network and 5500 tracklets are produced from the 2000 particles, this means that around 4400 tracklets are used for training. Therefore a single tracklet has 0.0227% of affect on the output neural network. This significant especially if the tracklet has the properties of an electron if it is a pion or vice versa. Therefore depending on the required accuracy for the neural network the amount of tracklets needed can range from hundreds of thousands to millions.
Other limitations include the learning parameters chosen. Therefore in the experiments conducted the learning rate was set to 0.5 with a momentum of 0.7. These parameters chosen can be changed to produce better results by choosing a slower learning rate and momentum and an increased number of generations. The time taken to train the data may take weeks if a large sample of particles and generations are used. The appropriate learning rate and momentum needs to be chosen according to experimental tests with various parameters.

7. CONCLUSIONS

The aim of this paper is to determine if Artificial Neural Networks can be used as an effective online PID generator algorithm as an alternative to the summation approach. The algorithms were tested on two thousand simulated particles from AliRoot. The summation approach was found to have drawbacks in its limitation in the PID frequency distribution and pion efficiencies. The PID frequency distribution of electrons and pions tend to be between 0 and 100 and the frequencies for PID numbers overlapped significantly in this area. The pion efficiency for the summation approach was 57.7% which is very high and may thus be very difficult for the offline particle identification algorithms to classify the particles according to the PID numbers.
The ANN approach reduced these drawbacks of the summation approach by training the network with the backpropagation algorithm to determine an effective neural network to be used as a PID generator algorithm. Two types of PID numbers were tested with two different neural network approaches. The first approach created one output which was converted into a single 8-bit PID number. The second approach created two outputs which was converted to two 4-bit PID numbers. Each of the output approaches were tested with four different input types of either preprocessed variables, time bin information or both. The preprocessed variables were based on the data features observed from the average time bin ADC values for pions and electrons.
The two output neural network approach produced better pion efficiency results when compared to the one output neural network. Overall the neural network that produced the

best results was the network with eight inputs and two outputs. The eight inputs used both preprocessed variables and time slices. This showed that preprocessed variables improve the effectiveness of the online algorithms but the time bin information still provide important information that can improve the online PID generator algorithms.

Overall the hypothesis that the PID generator algorithms using ANNs produces better results than the summation method is not rejected and a network with preprocessed variables and time slices information as inputs and two outputs showed to have the most potential for an online PID generator algorithm.

Although the results were successful, we observed that the limitation of two thousand particles may have a major affect on the effectiveness of the neural network in practical applications and an increased number of particles used for training is required at a reduced the learning rate and momentum.

7.1 Future Work

There are two major areas for potential future work. The first area is with regards to the training of the neural network and the second area to explore is the structure of the neural network used as an online algorithm. As discussed in the Section 6.3, the limitations in the number of particles used as training data causes an issue in determine the accuracy of the neural network on the physical detectors. The network needs to be trained on a larger number of particles at a slower learning rate and momentum to ensure that the backpropagation algorithm is able to find the global minimum for the error. Other concepts such as regressors can be applied to improve the learning process of the ANNs. The structure of the neural network can also be improved upon by reducing the computational complexity. The complexity of the network can be reduced by removing the normalisation of the inputs, simplifying the activation function and instead of the network producing signals between 0 and 1, the network can directly output the PID numbers.

8. REFERENCES

[1] K. Aamodt, A. A. Quintana, R. Achenbach, S. Acounis, D. Adamová, C. Adler, M. Aggarwal, F. Agnese, G. A. Rinella, Z. Ahammed, et al. The alice experiment at the cern lhc. *Journal of Instrumentation*, 3(08):S08002, 2008.

[2] A. Andronic, C. Blume, P. Braun-Munzinger, H. Daues, A. Devismes, C. Finck, R. Schulze, S. Sedykh, R. Simon, H. Stelzer, et al. Prototype tests for the alice trd. *Nuclear Science, IEEE Transactions on*, 48(4):1259–1264, 2001.

[3] V. Angelov. Design and performance of the alice trd front-end electronics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 563(2):317–320, 2006.

[4] P. Braun-Munzinger, L. Musa, and P. V. Vyvre. Upgrade strategy for the alice central barrel. 2012.

[5] J. G. Carbonell. *Machine learning: paradigms and methods*. Elsevier North-Holland, Inc., 1990.

[6] M. L. Cherry, D. Müller, and T. A. Prince. The efficient identification of relativistic particles by transition radiation. *Nuclear Instruments and Methods*, 115(1):141–150, 1974.

[7] A. Collaboration, K. Aamodt, et al. The alice experiment at the cern lhc. *Jinst*, 3(420):S08002, 2008.

[8] A. Collaboration et al. Aliroot, alice offline simulation, reconstruction and analysis framework.

[9] A. Collaboration et al. Technical design report for the alice computing, cern/lhcc 2005-18; aliroot, alice offline simulation, reconstruction and analysis framework. *aliceinfo. cern. ch/Offline/S.*

[10] A. collaboration et al. Technical design report for the upgrade of the alice readout and trigger system. Technical report, ALICE-TDR-015 November 4, 2013.

[11] B. K. et al. Electron/hadron separation: Neural networks as a tool. *Presentation at the ALICE Physics Week in Prague*, 2008.

[12] T. Gunji. Study of electron identification capability of the alice transition radiation detector. *Master thesis, Department of Physics, Graduate School of Science, University of Tokyo*, 2003.

[13] M. Gutfleisch. *Local Signal Processing of the ALICE Transition Radiation Detector at LHC (CERN)*. PhD thesis, Heidelberg U., 2006.

[14] J. Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

[15] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.

[16] T. Jayalakshmi and A. Santhakumaran. Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1):1793–8201, 2011.

[17] Y. Kwon. Physics-motivated neural networks approach to electron/hadron separation. *Presentation at the ALICE Analysis Meeting at GSI*, 2008.

[18] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[19] T. Ludlam, E. Platner, V. Polychronakos, M. Deutschmann, W. Struczinski, C. Fabjan, W. Willis, I. Gavrilenko, S. Maiburov, A. Shmeleva, et al. Particle identification by electron cluster detection of transition radiation photons. *Nuclear Instruments and Methods*, 180(2):413–418, 1981.

[20] T. Mahmoud, A. T. Collaboration, et al. The alice transition radiation detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 502(1):127–132, 2003.

[21] T. M. Mitchell. Artificial neural networks. *Machine learning*, pages 81–127, 1997.

[22] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[23] S. Russell, P. Norvig, and A. Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 1995.

[24] A. Wilk. *Particle identification using artificial neural networks with the ALICE transition radiation detector*. PhD thesis, Münster (Westfalen), Univ., Diss., 2010, 2010.