Machine Learning Approaches Towards Tuning ALICE TRD Simulations

University of Cape Town

Author:
Nikhiel Ramraj

Student Number:
RMRNIK002

Supervisor: Assoc Prof T.Dietel

A dissertation submitted in fullfilment of the academic requirements for the degree of Master of Science

February 13, 2023

Declaration

I Nikhiel Ramraj, student number RMRNIK002 declare the following:

- The research reported within this thesis is a result of my own efforts
- All external sources of information have been acknowledged to the best of my knowledge. Where exact words or phrases from external sources have been included their content has been placed within quotations and acknowledged.

Abstract

In this work an exploration of the discrepancies existing between real and simulated data pertaining to the ALICE Transition Radiation Detector is carried out as a motivation to tune the necessary parameters in the ALICE Online-Offline simulation software (O^2). After such exploration a single parameter namely the Xe gas gain is subjected to modification. A machine learning approach is taken with the use of deep learning discrimination mechanisms namely artificial neural networks and convolutional neural networks to quantify the effect that our tuning has on the improvement of the simulation results and their conformation to the real data. The correspondence of the optimal values suggested by deep learning approaches is investigated with pulse height spectrometry. It is shown that the optimal parameters suggested by our deep learning models through inference of their performance metrics are not clear and in agreement with that suggested by naive pulse height inspections.

Acknowledgements

I firstly would like to convey my sincerest gratitude and appreciation to my supervisor Assoc Prof T.Dietel for all his guidance and assistance.

I thank my parents and my sister for their continuous support and love. The support from the elements of Magha is appreciated.

I wish to acknowledge the financial support of the ETDP SETA (Education, Training and Development Practices Sector Education and Training Authority).

This work is based on the research supported in part by the National Research Foundation of South Africa.

The support and resources of the University of Cape Town's High Performance Computing facility is also appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The atom which is considered the building block of matter is itself comprised of fundamental particles. The particle theory of matter attempts to ultimately provide a description of matter in terms of its particle constituents. Through much of the recent century the field of particle physics has advanced substantially, with significant strides being made. The fact that we are unable to simply observe these fundamental particles, but rather detect their presence through their interaction with other matter, has resulted in the construction of various apparatuses to achieve this. The most notable of these being the Large Hadron Collider (LHC), which is a 27 km long particle accelerator that facilitates particle collisions. The resulting fundamental particles from these collisions are detected as a consequence of their interaction with other materials. The LHC consists of several detectors, each with a mandate to advance their associated specific physics discipline. ALICE (A Large Ion Collider Experiment) is one such detector at the LHC with the aim of investigating charateristics of the Quark-Gluon-Plasma (QGP) [1]. This is a state of matter that exists at enormously high temperatures and densities when hadrons disintegrate into their gluon and quark components. The QGP state existed shortly after the Big Bang and is predicted by Quantum Chromodynamics (QCD) [2]. Among the various particle detection mechanisms employed at ALICE, we confine our study to that of the Transition Radiation Detector (TRD). Since the identification of the occurrence of particles is heavily reliant on detectors, their proper calibration and management of functionality is imperative.

1.1 Thesis structure and aim

The aim of this thesis is to firstly investigate the discrepancies that exist between real and simulated data generated from the ALICE Online-Offline software framework in an effort to identify any parameter(s) in the Monte Carlo mechanism that is poorly initialized. The clarification of these inherent differences serves as a motivation to tune the necessary parameter(s) in the simulation software that is responsible in order to generate data that conforms to real data to a greater degree. Secondly the aim is also to assess the suitability of deep learning discriminators as tuning mechanisms for Monte Carlo simulations in terms of their ability to suggest optimal parameters through examination of the behaviour of their performance metrics according to the parameter in question being tuned.

The organisational structure of this thesis is as follows. Chapter 2 provides a description of the transition radiation detector at ALICE in terms of its basic structure and functionality. Chapter 3 describes the mathematics of machine learning techniques utilized in this study. Chapter 4 describes the process followed in data acquisition and pre-processing and the necessary analysis strategies employed. Chapter 5 explains the results of our data exploration and tuning strategies through machine learning. Finally we summarise the pertinent results and their implications in the conclusion which is presented in Chapter 6.

Chapter 2

The ALICE Transition Radiation Detector

2.1 Transition radiation

Transition radiation refers to the emitted radiation from a charged particle that traverses the interface of two media with differing dielectric constants [3]. The prediction of the existence of such a phenomenon is attributed to Ginzburg and Frank [4], who also demonstrated the logarithmic increase in the transition radiation intensity with respect to rising particle energy. In the case of ultra-relativistic particles Garibian [5] showed the transition radiation of these particles was expected to lie in the x-ray domain. The cumulative energy loss of the traversing charged particle is proportional to its Lorentz factor γ and the probability of photon emission from a single particle is of the order of magnitude of the fine structure constant ($\alpha = 1/137$). The issue of low photon yield can be overcome by employing a multilayered dielectric radiator with a minimum spacing between each layer proportional to the wavelength of radiation [6]. For transition radiation in the x-ray domain this required minimal spacing is minute. Whilst transition radiation exists in both optical and x-ray regimes, the latter is more desireable owing to its capability in being utilized for detection and identification of ultra-relativistic particles.

2.2 Detector aim and structure

Transition Radiation Detectors (TRDs) as the name suggests function on the basis of transition radiation detailed above. The ALICE TRD was constructed for the improvement of electron identification and the enhancement of triggering capabilities [7, 8]. In addition the ALICE TRD provides assistance with track reconstruction and calibration. A longitudinal view of the TRD assembly in relation to some other ALICE detectors is depicted

in Figure 2.1. The TRD surrounds the TPC and ITS and is itself surrounded by the TOF and partially by the HMPID, PHOS and EMCAL. In totality the TRD has the capacity to host 540 read out chambers, however it comprises of 522 due to the non installation of some chambers in sectors 13 to 15 for the sake of material reduction in front of the PHOS detector [1]. The TRD comprises of 18 Supermodules, where an individual supermodule consists of an arrangement of 5 read out chambers stacked in the z direction and 6 layers in the r direction [1, 7, 8]. A visual representation of a supermodule is depicted in Figure 2.2 which also shows the global and local coordinate systems employed at ALICE. The global coordinate system is centered at the interaction point of the beams with the x axis pointing to the center of the LHC whilst also perpendicular to the beam. The y axis is directed upward and transverse to the beam and x axis. The z axis is parallel to the beam direction. The local coordinates is essentially a rotation of the global one such that the x axis is transverse to the read out chamber [1, 9].

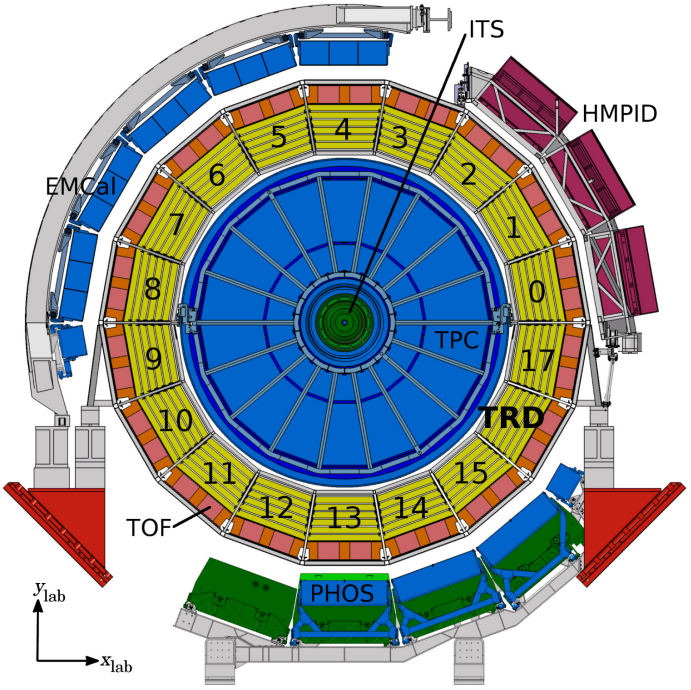


Figure 2.1: Cross sectional view of the TRD with its numbered sectors in relation to the other ALICE detectors [1] namely the Inner Tracking System (ITS), Time-Projection Chamber (TPC), Time-Of-Flight (TOF), ElectroMagnetic Calorimeter (EMCAL), Photon Spectrometer (PHOS) and the High-Momentum Particle Identification Detector (HMPID).

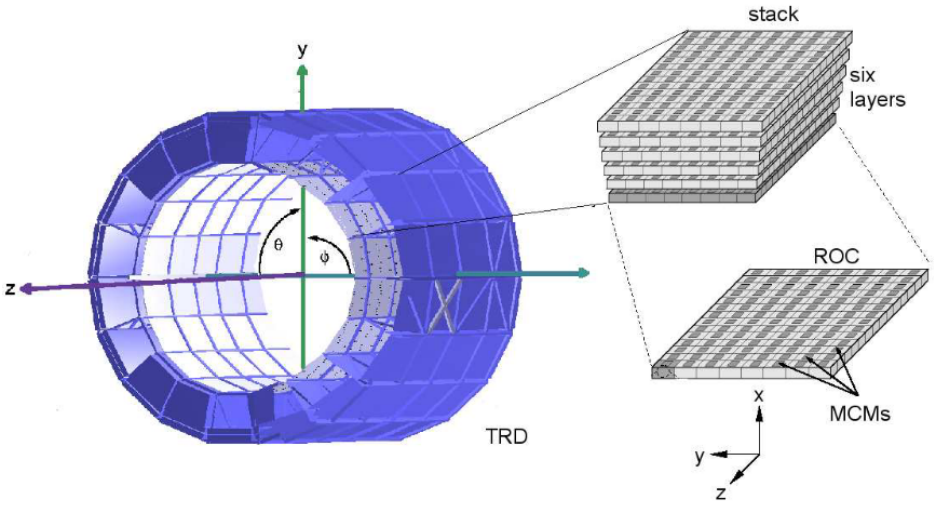


Figure 2.2: Structural arrangement of read out chambers in a single supermodule of the TRD. The global coordinate system is shown with respect to the overall TRD detector and the local coordinate system is depicted on the lower right with respect to a single chamber [9].

A single read out chamber comprises of 16 padrows, with each individual padrow consisting of 144 pad columns [7]. This leads to a total of 2304 pads for one read out chamber. There are 8 Read Out Boards (ROBs) located on a detector chamber. An individual ROB contains 16 Multi-Chip Modules (MCMs), which are connected to 18 pad columns within a particular pad row. These form part of the Front End Electronics (FEEs) which are responsible for the processing of generated TRD signals. An internal view of a read out chamber is depicted in Figure 2.3. The internal structure of each read out chamber can be described in terms of three major divisions which collectively facilitate the necessary signal generation and thereby allows for the detection of possible transition radiation and other aspects. These divisions include the radiator, drift region and the amplification region. The radiator is 4.8cm and constitutes the bottom of the chamber. The radiator achieves the necessary significant inhomogeneity as a medium to facilitate a high TR photon yield through inclusion of polypropylene fibre mats placed within rohacell foam sheets [1, 8, 10]. The drift region is 3 cm and is infused with a Xe/CO_2 gas mixture. The final division is the amplification region which contains anode and cathode wires. The anode wire plane is located between the cathode wires and the cathode pad plane which signifies the end of the amplification region.

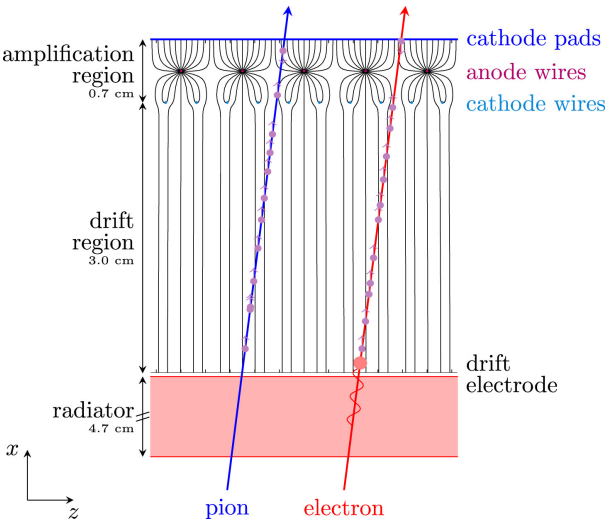


Figure 2.3: The internal subdivisions of a read out chamber in the xz plane with traversing particles [1].

2.3 TRD Signals

A traversing particle in the TRD that enters a read out chamber first encounters the radiator. The extreme inhomogeneity of the radiator promotes TR production for ultra-relativistic particles with the TR photons having $E \sim$ keV [7]. These photons are absorbed within the beginning of the drift region due to the infused Xe/CO_2 gas having an absorption length of 1 cm for keV energy scales. The traversing particle together with the TR photons uniformly ionise the gas in the drift region and thereby generate free electrons. These electrons drift to the nearest anode wires within the amplification region with the ions simultaneously migrating to the cathode wires. Within the vicinity of the anode wires in the amplification region the electric field drastically increases resulting in an "avalanche" that amplifies the signal. The ion movements induce a signal on the cathode plane which are then interpreted by the Front End Electronics (FEEs).

2.4 Pulse height spectrum

Average pulse height spectra is a measure of the average ADC (Analog to Digital Converter) count per timebin. This can be used in a quality control sense to investigate the overall behaviour of the TRD. An example of the average pulse height is depicted in Figure 2.4.

Pulse height spectrometry is a measurement of energy deposition in time arising from a particular process. The investigation and study of the traits exhibited by a pulse height spectrum yields pertinent properties regarding an experimental process and provides collective insight on the overall behaviour. In the context of the TRD, the pulse height spectrum

is a measure of the charge deposition per timebin resulting from a traversing particle within a read out chamber. The average pulse height is therefore a consideration of the average charge deposition for a particular timebin by all traversing particles. The pulse height spectrum can therefore be regarded as the average signal induced on the cathode pad planes of the TRD as a consequence of the interaction and passage of particles. An example of the average pulse height spectrum for the TRD is depicted in Figure 2.4 for pions and electrons with and without the inclusion of transition radiation. The three curves depict both similarities and differences in their temporal behaviour. All curves exhibit a large peak at early times (approximately 0.5 μs) and recede at times exceeding 2.5 μs which constitutes the "ion tail". With the exception of the pulse height for electrons with transition radiation, the remaining two curves both exhibit a plateau between approximately 0.7 μs and 2.4 μs which characterizes the drift region. The production of transition radiation therefore manifests as a peak in the average signal at approximately 2.5 μs. Overall it is evident that the average signal for electrons surpasses that of pions. This is attributed to a greater charge deposition by electrons caused by ionisation and transition radiation production.

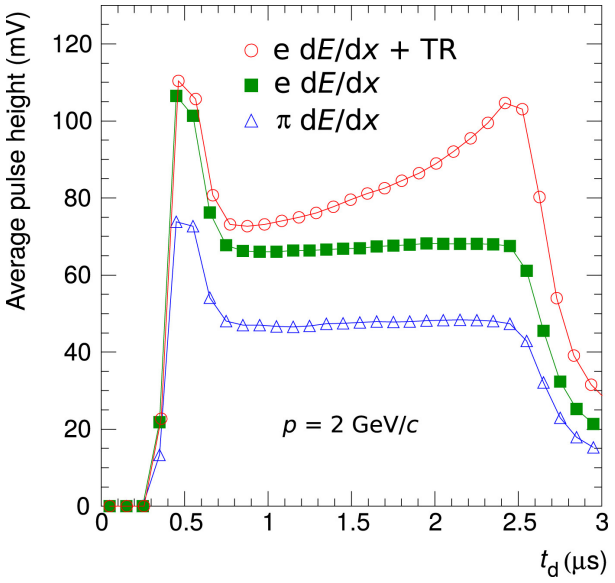


Figure 2.4: Pulse height spectrum showing the effect of transition radiation on the overall shape [1]

Figure 2.5 illustrates the temporal correspondence of the pulse height spectrum and a traversing particle in a read out chamber. The positions of the amplification peak and the "edge" of the pulse height coincide with that of the anode wires and the entrance of the drift chamber respectively. The main parameters that govern the shape of the pulse height spectrum are also depicted. In summary the measure of the difference in time between the amplification peak and the chamber entrance exhibits inverse proportionality with the drift velocity. The gain obeys a proportional relationship with the average pulse height whilst the width of the pedestal demonstrates proportionality with noise. The time offset is dependent on the amplification peak position.

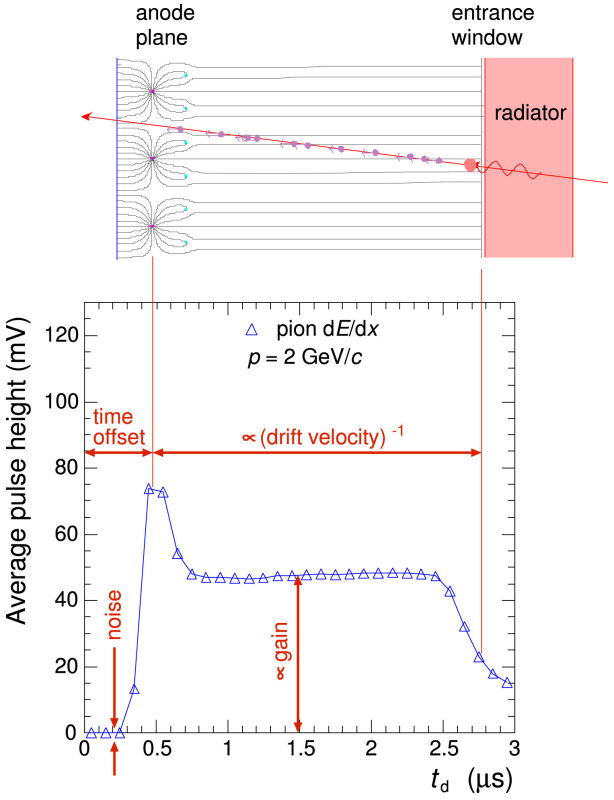


Figure 2.5: Calibration factors affecting the overall shape of the pulse height spectrum coinciding with a traversing pion in a read out chamber [1].

2.5 The necessity of simulations in high energy physics

A simulation imitates the behaviour of a real system through the use of an underlying model that can be used to describe the processes that occur. The process within a high energy physics experiment can be characterized by four aspects which are namely physics generators, detector simulations, calibration and finally reconstruction [11]. Within the context of this thesis, calibration and reconstruction are run as part of the simulation chain.

The major focus within this work is the detector simulation aspect which is elaborated further. Accurate simulation of particle dynamics and detectors are crucial in high energy physics [12]. Simulations are imperative for research and development in high energy physics since they allow for :

- Studies of detector effects and the correction of these effects
- examination of the behaviour of equipment
- investigations of results under varying experimental conditions
- investigations of how experimental results from an underlying theory would present themselves

In general the use of simulations allows for the anticipation of possible experimental results which provides valuable insight in the determination of the technical specifications required to conduct large scale experiments. This is imperative in the context of feasibility studies which quantify the possible successful extraction of significant scientific results.

To date, collider experiments generate enormous quantities of data. Future experimentation will result in the generation of even more data which would exacerbate the strain on software resources involved in the simulation and reconstruction of events [12]. Therefore the exploration of approaches towards improved simulation capability is imperative for future collision experiments in the LHC.

Simulations impart valuable information in the search for new phenomena since they provide insights on how the remnants or signatures of these phenomena may manifest in the detectors point of view.

2.6 ALICE Online-Offline software framework

Since the long shutdown of the LHC [13] various upgrades have ensued in order to prepare for the commencement of run 3. The ALICE Online-Offline software framework (hereon referred to as O^2) was developed to manage with the increased data generation and demands of run 3 collisions within the LHC [14]. The O^2 framework incorporates all computational functions necessary for a high energy physics experiment. Some of these functions include but are not limited to detector readout and calibration, data collection and reconstruction and physics analysis [15]. O^2 facilitates a stronger integration with offline and online computing and allows for detector functionality in continuous read out mode to be performed for the high latency gas detectors such as the TPC.

Chapter 3

Theory of machine learning

In this chapter a general overview of machine learning is presented together with explanations of relevant methods and techniques pertinent to the analysis in this thesis.

3.1 Artificial intelligence

Artificial intelligence refers to the ability of non-living entities such as computers and machines to exhibit human-like behaviour [16]. A well known technique to discern such ability is the Turing test [17] which was proposed by Alan Turing in 1950. The Turing test implies that if a machine is capable of convincing a knowledgeable person that it is a human then such machine is said to be "intelligent".

3.2 An introduction to machine learning

Machine learning refers to the automated improvement of computer systems as experience is gained [18]. The fact that this improvement is automated by the computer program in question without explicit programming implies that it "learns" from data. This introduces the notion of the learning problem which has been formulated in a generalized manner by Tom Mitchell as the following:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance in tasks T, as measured by P improves with experience E." [19]

The process of learning is facilitated by the following aspects [16] :

- The input data to the program
- A metric that computes a measure of the deviation between actual and ideal behaviour
- A feedback mechanism that is guided by the performance metric for the sake of improvement of results in future training.

Machine learning is now widely used in applications of natural language processing, anomaly detection, recommendation systems, computer vision and many others. The widespread use of machine learning can be attributed to increased computational strength and greater accessibility to data. The *No Free Lunch theorem* [20] asserts that there exists no single machine learning algorithm that exhibits complete superiority to all others over all possible data sets. The exceptionally good performance of a machine learning algorithm on a particular type of data set is compensated by its equally bad performance on disparate types of data. The data intensive nature of our current world and extensive use of machine learning requires continuous investigation into improved techniques and the suitability of these techniques on certain data types. This has established machine learning as an active research field concerned with assessing the relevance of certain data-generating distributions [21] in the real world and their corresponding optimal machine learning algorithms. Machine learning can be divided into two broad categories which we elaborate on below.

3.2.1 Supervised learning

In supervised learning the true responses are provided together with the corresponding feature variables. The machine learning model therefore is able to map the features to their associated responses.

3.2.2 Unsupervised learning

Unsupervised learning refers to the implementation of machine learning without the provision of the true responses. This results in the inability of the model in question to quantify the errors in predictions directly. Instead this must be done by consideration of clustering or the reduction in dimension of the data in order to infer response categories.

3.3 Classification performance metrics

In the context of classification problems the predicted target or response is of a categorical nature which may denote the distinct group or class associated with the predictors. Unlike in a regression scenario where the performance of a model can simply be quantified by some function of the distance between the predictions and truth, the performance in classification problems must be quantified by incorporating the misclassifications between the various response classes involved.

3.3.1 The confusion matrix

The confusion matrix is a graphical depiction of the correspondence of the ground truth classes and predicted classes from a model. For k distinct response classes there are k^2 elements in the confusion matrix. An illustration of this for a two class (binary response) case is shown in Figure 3.1.

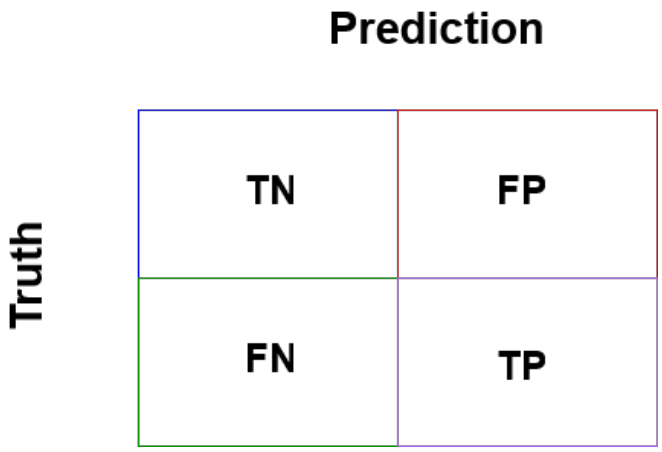


Figure 3.1: Confusion matrix of a binary classifier. TP, TN, FN, FP denote true positives, true negatives, false negatives and false positives respectively.

True negatives (TN) denote data values that are actually labelled as negative and also predicted to be negative by our model. False negatives (FN) denote data values that are actually not negative but still predicted to be so. True positives (TP) denote values labelled as positive which are correctly predicted by our model. False positives (FP) measure the incorrect prediction of data values to be positive whereas in actuality they are negative. The confusion matrix and our definitions of its elements can be used to derive important metrics to characterize the performance of classification models. We explain these below.

The accuracy of a classifier is expressed as

$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP} \tag{3.1}$$

The recall which is also termed the true positive rate or sensitivity is given by

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.2}$$

The specificity, also termed the true negative rate is given by

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.3}$$

The precision, also termed the positive prediction value is given by

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.4}$$

The F_1 score which is a harmonic mean of the precision and recall is expressed as

$$F_1 = 2\left[\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right]^{-1} = \frac{2TP}{2TP + FP + FN} \tag{3.5}$$

The false negative rate which is also termed the miss rate is given by

$$\text{Miss rate} = \frac{FN}{FN + TP} \tag{3.6}$$

3.3.2 The precision recall curve

The precision recall curve is used to assess the classification performance of a model under the conditions of class imbalances [22]. The curve depicts the trade off between the precision and recall of a model for varying probability thresholds. The worse performing model would have a precision equal and less than the proportion of the positive classes in the data. The ideal classifier would have a precision and recall of unity leading to the area under the curve (AUC) resolving to unity. Selecting the threshold that maximizes the AUC results in optimal classification. In this context both the precision and recall are dependent on the threshold selected. Practically the threshold which results in the precision and recall being closest to unity maximizes the AUC. This task can be expressed as

$$\arg\min_{\delta} D(\delta) = \sqrt{\left[1 - P(\delta)\right]^2 + \left[1 - R(\delta)\right]^2} \tag{3.7}$$

where $P(\delta)$ and $R(\delta)$ denote the precision and recall at threshold δ.

3.4 Artificial neural networks

Artifical neural networks are so called due to inspiration drawn from the structure of biological neurons. It is imperative to note amidst this inspiration that they do not simulate how the brain actually functions in terms of decision making.

3.4.1 The perceptron

The single layer perceptron is the most basic artificial neural network configuration and serves as a building block for more complex networks. An example of a perceptron is depicted in Figure 3.2. Its structure comprises of an input layer and a single computational neuron which serves as the output layer.

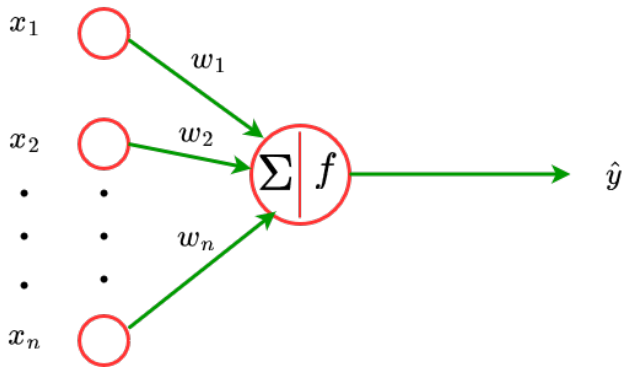


Figure 3.2: A single layer perceptron.

Using Figure 3.2 we now describe mathematically the general output of a perceptron. For n inputs $X = (x_1, x_2, ..., x_n)$ there are n corresponding weights $W = (w_1, w_2, ..., w_n)$. The output is then given by [23]

$$\hat{y} = f\left(\sum_{i=1}^{n} w_i x_i + b\right) = f\Big(WX + b\Big) \tag{3.8}$$

where f is the activation function and b is the bias.

3.4.2 The multi-layer perceptron

Multi-layer perceptrons are a combination of numerous perceptrons that are interconnected to form multiple layers. A general structure of a multi-layered network is depicted in Figure 3.3. The first and last layer of neurons denote the input and output layers respectively (represented as red neurons). The layers in between constitute the hidden layers of the network (represented as blue neurons) since we are unable to directly observe their outputs.

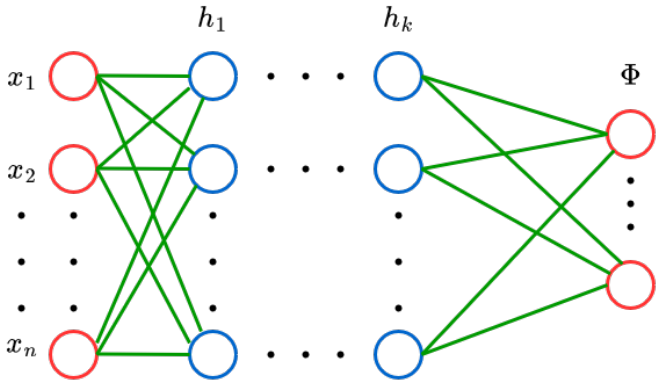


Figure 3.3: A generalized multi-layer perceptron.

For a generalized multi-layered perceptron with n inputs $X = (x_1, x_2, ..., x_n)$, k hidden layers and m neurons in the output layer Φ, the recursive equations governing the network are [23]

$$h_1 = f^{(1)}\left(W^{(1)}X + b_0\right) \tag{3.9}$$

$$h_{p+1} = f^{(p+1)}\left(W^{(p+1)}h_p + b_p\right) \ \forall p \in \{1, .., k-1\} \tag{3.10}$$

$$\Phi = f^{(k)}\left(W^{(k)}h_k + b_k\right) \tag{3.11}$$

$$\tag{3.12}$$

where h_1 denotes the first hidden layer and h_{p+1} denotes the $(p+1)^{\text{th}}$ hidden layer $\forall p \in \{1, ..., k-1\}$. The bias and activation function in the k^{th} layer is b_k and $f^{(k)}$ respectively, whilst the corresponding weights are $W^{(k)}$.

3.4.3 Activation functions

The choice of activation function affects the outcome of a neural network in terms of its capability to find relevant mappings and overall capture the unique trends in data for the

sake of prediction. The main functions used within this study are depicted in Figure 3.4. We briefly outline each function below.

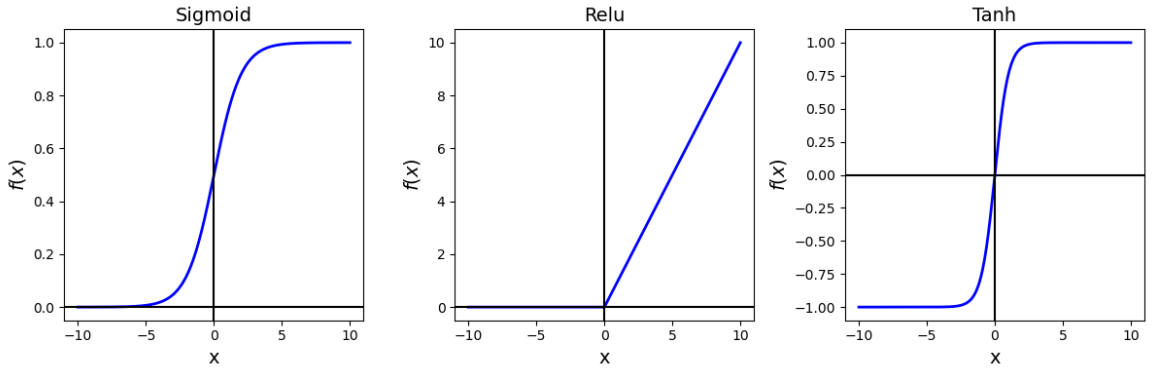


Figure 3.4: An illustration of some of the activation functions used within this thesis.

Sigmoid activation

The sigmoid activation is bounded between zero and unity and is therefore beneficial in producing probabilistic interpretations and developing loss functions arising from maximum likelihood models [23]. The sigmoid function is expressed as

$$f(x) = \frac{1}{1 + \exp(-x)}. \tag{3.13}$$

Rectified linear unit (ReLU)

The ReLU is a piece-wise function that facilitates easy optimization due to its similarity to linear functions [21]. Unlike the usual linear unit, ReLU outputs zero for half of its domain as observed in Figure 3.4. The ReLU function is expressed as

$$f(x) = \max(x, 0). \tag{3.14}$$

Tanh activation

The tanh activation exhibits a similar shape to the sigmoid, however it is distinctly different in its range which is between negative and positive unity. As a consequence it offers no probabilistic interpretability. The tanh function is expressed as

$$f(x) = \frac{\exp(2x) - 1}{\exp(2x) + 1}. \tag{3.15}$$

Output activations

The activation in the last layer within a deep learning architecture is determined by the type of problem being either classification or regression. In the case of classification the target values are simply the relevant classes and so the desired outcome would be for a model to output the probabilities of the associated classes. For a binary classifier, a sigmoid activation with a single neuron in the output layer is often used. In the case of multi-class classifications the softmax function is employed owing to its ability to compute probabilities for the respective classes [21].

In general, for a k-class classification scenario with classes ν_i for $i = 1, ..., k$ we define the vector $\boldsymbol{\nu} = (\nu_1, \nu_2, ..., \nu_k)$. The softmax function is then given by [23]

$$f(\boldsymbol{\nu})_i = \frac{\exp(\nu_i)}{\sum_j^k \exp(\nu_j)} \tag{3.16}$$

where each $f(\boldsymbol{\nu})_i$ in the transformed k-dimensional vector $f(\boldsymbol{\nu}) = (f(\boldsymbol{\nu})_1, f(\boldsymbol{\nu})_2, ..., f(\boldsymbol{\nu})_k)$ exists between zero and one and [24]

$$\sum_i^k f(\boldsymbol{\nu})_i = \sum_i^k \left(\frac{\exp(\nu_i)}{\sum_j^k \exp(\nu_j)} \right) = 1. \tag{3.17}$$

The implication of these conditions allow for the consideration of the softmax function as a probability distribution over k-classes [24].

3.5 Convolutional neural networks

These networks are so named due to the implementation of a convolution operation in their model architecture [21]. The one-dimensional convolution of an input x with a kernel ω can be expressed as

$$s(t) = (x * \omega)(t) \tag{3.18}$$

where the asterisk denotes the convolution operation and t is an index of the input x.

The input and kernel within a machine learning setting is often multidimensional in nature and represents an array of data and the associated parameters that are learnt through the learning algorithm respectively [21]. For a two-dimensional image I and kernel Ω their discrete convolution is given by

$$s(i,j) = (I * \Omega)(i,j) = \sum_m \sum_n I(m,n)\Omega(i-m, j-n) \tag{3.19}$$

where i, j are the indices of image I. Due to the commutative nature of convolution as a consequence of the relative "flip" of the kernel with respect to the input, the above expression can be equivalently expressed as

$$s(i,j) = (\Omega * I)(i,j) = \sum_m \sum_n I(i-m, j-n)\Omega(m,n). \tag{3.20}$$

In a practical sense, machine learning libraries and packages implement a "cross-correlation" function which essentially resembles the form of equation (3.20) without the implementation of a kernel flip. This is expressed as

$$s(i,j) = (\Omega * I)(i,j) = \sum_m \sum_n I(i+m, j+n)\Omega(m,n). \tag{3.21}$$

To demonstrate the notion of convolution consider an input A which is a (3×3) array

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.22}$$

and a (2×2) kernel given by

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}. \tag{3.23}$$

The convolution of these can be expressed as

$$A * B = \begin{bmatrix} \xi_1 & \xi_2 \\ \xi_3 & \xi_4 \end{bmatrix} \tag{3.24}$$

where

$$\xi_1 = a_{11}b_{11} + a_{12}b_{12} + a_{21}b_{21} + a_{22}b_{22} \tag{3.25}$$
$$\xi_2 = a_{12}b_{11} + a_{13}b_{12} + a_{22}b_{21} + a_{23}b_{22} \tag{3.26}$$
$$\xi_3 = a_{21}b_{11} + a_{22}b_{12} + a_{31}b_{21} + a_{32}b_{22} \tag{3.27}$$
$$\xi_4 = a_{22}b_{11} + a_{23}b_{12} + a_{32}b_{21} + a_{33}b_{22} \tag{3.28}$$
$$\tag{3.29}$$

3.5.1 Pooling layer

Pooling is generally enacted after convolution layers and facilitates the diminishment of the dimensionality of feature maps [25]. The pooling function computes a summary statistic of regions in the feature map. Such a function may include average pooling whereby the mean of a region is computed. Max pooling is also another popularly implemented function that selects the maximum value within regions of the feature map. Figure 3.5 depicts an example of pooling operations on a feature. The implementation of pooling operations lowers computational cost [26] and accomplishes translational invariance [27].

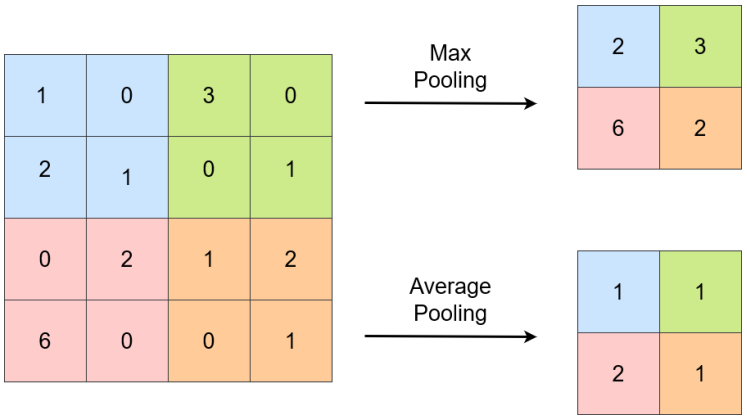


Figure 3.5: An illustration of max and average pooling for a (4×4) image input and a (2×2) kernel with stride of 2.

3.6 Loss functions

The loss function is essential in assisting a model gauge as to whether or not an improvement occurs at each iteration [28]. It assists in the provision of realization to the model on whether the existing weights in a particular iteration enable astute or poor performance on the training data. In the classification context entropy is employed as a loss function to networks. In the general scneario where there are n outcomes or target classes to be predicted the loss function is given as [28]

$$L = -\sum_{i}^{n} y_i \log_2(y_i) \tag{3.30}$$

which for the simple binary classification scenario reduces to

$$L = -[y \log(p) + (1 - y) \log(1 - p)]. \tag{3.31}$$

Equations (3.30) and (3.31) denote the categorical and binary crossentropy respectively.

3.7 Regularization

Measures that prevent or seek to curb the occurrence of overfitting can be considered as regularization mechanisms. The inclusion of such measures in model architectures is vital to constrain the model to learning actual trends in data rather than subtle noise which may impede the generalization capability of a model. We detail two such methods which are mainly implemented in this study.

3.7.1 Dropout layers

Dropout refers to the random relinquishing of neurons from a neural network within the hidden and visible layers [29]. This constitutes a temporary removal of the neurons and their associated connections to other neurons within a network. The goal of this is the prevention and restriction of a co-adaptation phenomena arising amongst the neurons within the network. This facilitates a robust behaviour of the neurons by constraining them to rely on the behaviour of the data rather than that of other neuron components [29].

3.7.2 Early stopping

In the context of deep learning models, these are implemented as callbacks which monitor the behaviour of training and validation loss and accuracy curves per epoch. A single epoch constitutes all of our training data being "seen" by our model once. At each epoch the

validation metrics are compared to that of the training and a decision is made on whether or not to progress with the next epoch. If the validation loss demonstrates no improvement over the training loss over a possibly specified number of epochs, the training of the model ceases. In this way the model is prevented from exacerbating any poor performance or learning any noise which would hinder its generalizability. This has significant benefits since it allows machine learning scientists to initially specify a large number of epochs without concern of overfitting arising from too many epochs. Early stopping mechanisms address the *a priori* indeterminacy of the number of epochs required to achieve maximum possible learning of actual data patterns by a model without leading to overfitting.

Chapter 4

Methodology

4.1 Data information and acquisition

The real data employed in this study consisted of ALICE run 3 data obtained from the AliGrid Monalisa repository [30]. The data is initially stored as compressed time frame files (`.ctf` extension) in the repository and has to be parsed to a compressed time frame reader workflow in order to obtain the associated digit and tracklet information. A detailed explanation of the procedure and workflow used to obtain such information is provided in section B.1 of appendix B. The end result of the implementation are two files namely `trdtracklets.root` and `trddigits.root` which encapsulate the necessary information and are used for further analysis. For the purposes of this study a single ctf file proved to contain sufficient data.

4.1.1 O^2 simulations

We simulate the effect of particle traversal through the TRD with a pion particle box gun simulation which was performed with the O^2 software framework. The parameters used in this simulation are stated in Table 4.1. The choice of a pion simulation was motivated by the sheer abundance of this particle in real TRD data which would hence enable a comparative study. An explanation of the simulation procedure and workflow is given in section B.2 of appendix B.

Table 4.1: O^2 simulation parameters used.

Parameter	**Value**
Pseudorapidity	$\|\eta\| < 0.84$
Detector	TRD
Events	50 000
pdg	211

The pdg value of 211 denotes the code for a pion particle within the Monte Carlo particle numbering scheme which was initially introduced by the Particle Data Group in 1988 [31]. The numbering scheme allocates a unique code to particles and provides an interface amongst event generators, detector simulators and physics analysis software in general [32].

Pseudorapidity is a defined concept that overcomes the limitations of rapidity and facilitates easier measurement for extremely energetic particles along the beam axis. Rapidity describes the direction of emission of a collision product in relation to the beam axis and is defined as [32]

$$\zeta = \frac{1}{2} \ln \left(\frac{E + p_z c}{E - p_z c} \right) \tag{4.1}$$

where E, p_z and c are the total energy, momentum component along the beam axis and the speed of light respectively. The benefit of such a defined quantity is the invariancy of its differences ($\Delta\zeta$) under Lorentz boosts along the z-axis. The computation of such a quantity is not practical owing to the inability to obtain the complete momentum. Hence the pseudorapidity η which exhibits dependence on only the angle θ between the beam axis and the emission trajectory given by [33]

$$\eta = -\ln \left[\tan \left(\frac{\theta}{2} \right) \right] \tag{4.2}$$

is used for practical computation and is equivalent to ζ for extremely relativistic particles.

4.1.2 TRD signal extraction

The `trddigits.root` and `trdtracklets.root` files from both real and simulated data was subjected to a local maximum code finder routine [34]. The purpose of this code routine is to loop over all pad columns and extract the local maximum of ADC sums for each event in the trigger record. TRD signals are certain to have traversed regions where a local maximum of ADC sum exists. The relevent pad columns corresponding to the occurrence of these maxima were extracted together with the adjacent receding and preceding pad columns. The neighbouring pad columns to the central maximum were arbitrarily labelled as A and B neighbours. Therefore a single extracted TRD signal constitutes a 3×30 array of ADC values. These arrays were flattened and concatenated to a single dataframe for easy storage and to facilitate easy feeding in machine learning algorithms.

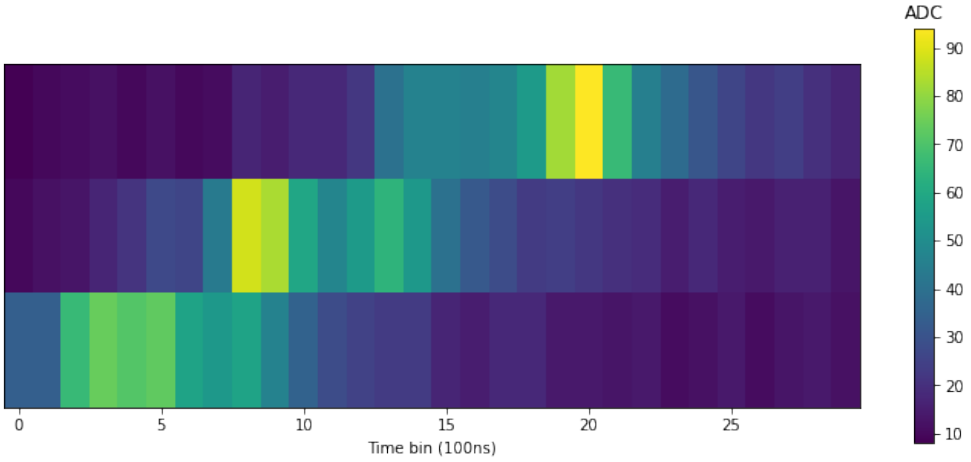
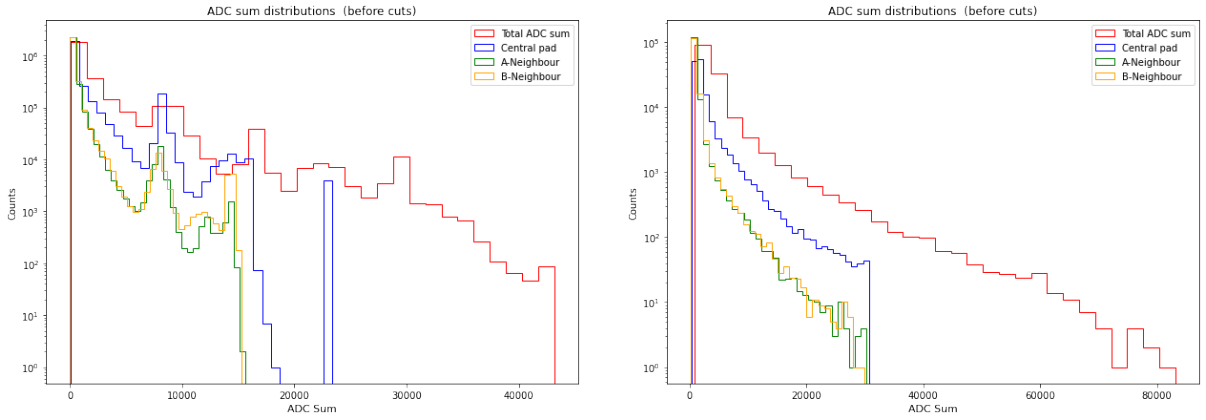


Figure 4.1: Example of an extracted TRD signal.

Algorithm verification

In order to assess the credibility of our signal extraction algorithm, we inspect the distributions of the ADC sums of each individual pad column across all 30 time bins in our signal window together with the total ADC sum of all pad columns. Ideally the ADC sum distribution of the central maximum pad column should supercede that of the adjacent A and B neighbour pads, whilst still remaining below the total ADC sum distribution. None of the individual distributions should exceed that of the total ADC sum distribution. Figure 4.2b demonstrates that this notion is obeyed by the simulated data. Examination of Figure 4.2a reveals that this is not the case with the real run data. The total ADC sum distribution is in fact surpassed by the central pad and neighbouring pad distributions at certain regions.

Figure 4.2b shows that the total ADC sum distribution of the simulated signals exists beyond a sum of 40000 and terminates at approximately just above a sum of 80000. This



(a) ALICE data.

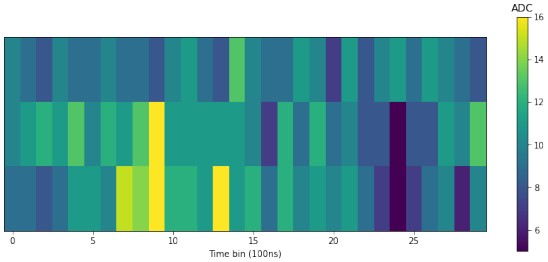
(b) Simulation data.

Figure 4.2: Distributions of ADC sums for extracted TRD signals.

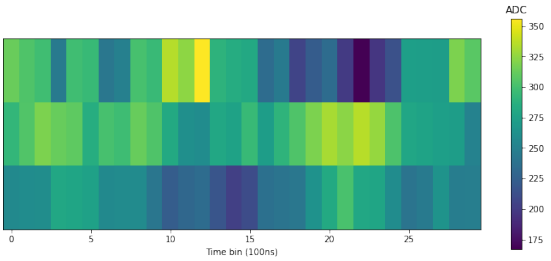
is in contrast to that of the real data whose total ADC sum distribution terminates at a sum approximately above 40000. In general the distribution of simulated signals exist at higher ADC sum values than that of real signals.

Invalid signals

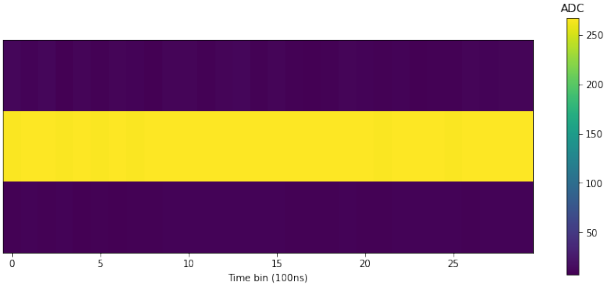
The inconsistencies observed in the ADC sum distributions signifies the prevalence of improper signals in our data. Examination of some of the individual signals in our data reveals a diverse set of abnormalities. The distinct categories of these abnormalities are illustrated in Figure 4.3. Figure 4.3a depicts general low noise fluctuations whereby low ADC values populate the entire signal window leading to a total ADC sum ≈ 900. Figure 4.3b depicts high noise fluctuations whereby the signal window is populated with large ADC values resulting in a total ADC sum $\approx 10^4$. Figure 4.3c depicts a "stripe" pattern whereby the central pad column contains a continuous stream of high ADC values with the neighbouring pad columns having a stream of considerably lower ADC values. Figure 4.4 illustrates other variants of the "stripe" abnormality and serves to demonstrate that the occurrence of such anomalies are not limited to just the central pad column in our 3×30 signal window.



(a) Low fluctuations



(b) High fluctuations



(c) Stripe abnormality

Figure 4.3: Improper signals

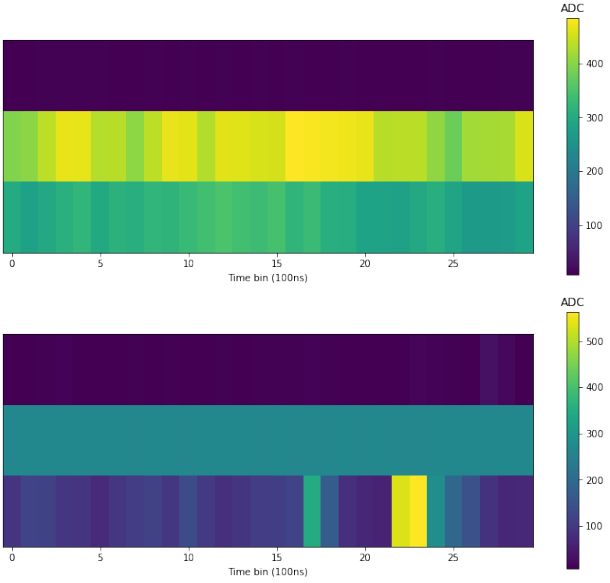


Figure 4.4: Variants of Stripe abnormalities.

4.2 Data preprocessing

The real and simulated data present obvious initial differences owing to the fact that impurities may be captured through our TRD signal extraction routine and as a consequence of detector fluctuations and possible readout issues with real data. The inherent initial differences present obstacles in using machine learning as a discrimination mechanism by weakening the validity of the results since a machine learning model would definitely discriminate well against distinctly differing data. The abnormalities in each data sample must then be filtered out so as to ensure a valid analysis of the pure signals of each data sample (real and simulated) which would imply meaningful results of any discrepancies identified. Both data samples are therefore subjected to noise filtration in order to ensure that the overall macroscopic behaviour of the data samples are similar.

The low and high noise fluctuations can be filtered by removing signals with a total ADC sum that exceeds a certain lower and upper bound respectively. The "stripe" noise can be filtered by inspecting if the average ADC of the central maximum pad column is very close to its maximum ADC value since this would constitute a continuous stream of high ADC values across the majority/entire 30 time bins. Since the "stripe" abnormalities also occur in the neighbouring pad columns, they are also subjected to the same inspection.

4.2.1 A strategy for stripe filtration

We now detail a strategy used to remove "stripe" abnormalities present in the data.

Let the ADC value in the i^{th} pad column at time bin t be σ_{it}. The ADC sum in the i^{th} pad column across all time bins can then be stated as

$$\sigma_i = \sigma_{i1} + \sigma_{i2} + ... + \sigma_{i30} = \sum_{t=1}^{30} \sigma_{it} \tag{4.3}$$

The total ADC sum of the signal is then expressed as

$$\sigma_T = \sum_{i=1}^{3} \sigma_i \tag{4.4}$$

where σ_2 denotes the ADC sum of the central maximum pad column and σ_1 and σ_3 denotes that of the neighbouring pad columns situated on either side of the central pad. The average ADC of the i^{th} pad column is given as

$$\overline{\sigma_i} = \frac{\sigma_i}{30}. \tag{4.5}$$

The maximum ADC value in the i^{th} pad column can be formally expressed as

$$\sigma_i^{\max} = \max\{\sigma_{it}\}_{t=1,\ldots,30}. \tag{4.6}$$

Now let the ratio of the average ADC to its corresponding maximum in the i^{th} pad column be R_{σ_i} which is expressed as

$$R_{\sigma_i} = \frac{\overline{\sigma_i}}{\sigma_i^{\max}} = \frac{1}{30\,\sigma_i^{\max}}\left(\sum_{t=1}^{30}\sigma_{it}\right). \tag{4.7}$$

Since the ADC values are all positive ($\sigma_{it} > 0$), the statistical average of ADC values in a particular pad column cannot exceed the maximum individual ADC value ($\overline{\sigma_i} \leq \sigma_i^{\max}$). Therefore R_{σ_i} is bounded between 0 and 1 and can be perceived as a percentage/representation of the proximity of the mean ADC to the maximum in a pad column. Abnormal signals which exhibit "stripe" like characteristics therefore have $R_{\sigma_i} \approx 1$. We filter out these abnormalities by enforcing that

$$R_{\sigma_i} < \epsilon \tag{4.8}$$

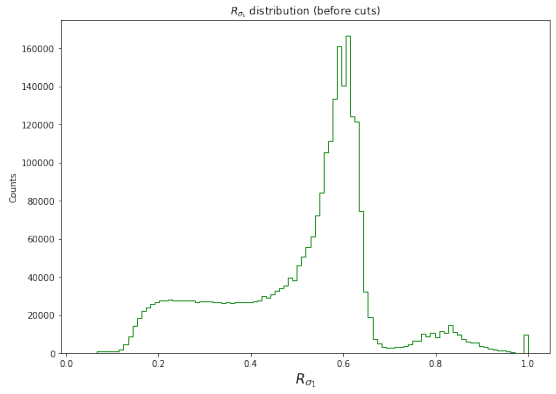
where ϵ is some tolerance threshold.

4.2.2 Quality requirements

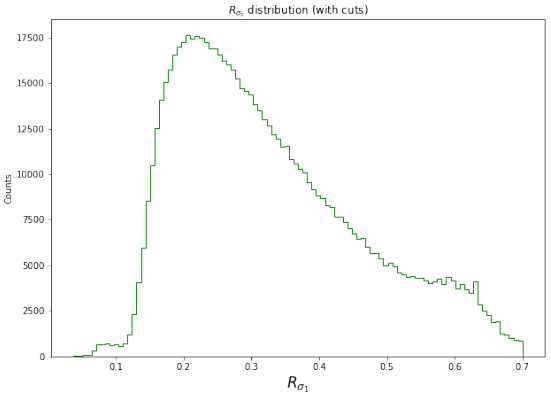
Based on the above definitions we impose quality requirements or "cuts" to facilitate the removal of the majority of signal abnormalities. The specifics of these are given in Table 4.2. Plots of the R_{σ_i} distributions before and after the implementation of cuts for real and simulated data are depicted in Figure 4.5 and Figure 4.6 respectively. The distributions of ADC sums after enforcing quality criteria is depicted in Figure 4.7.

Table 4.2: Quality criteria implemented

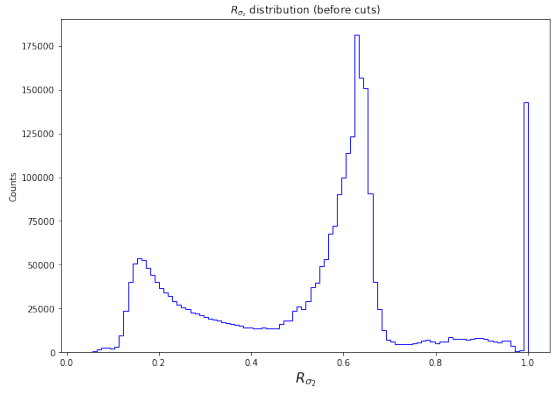
Criteria	**Explanation**
$1300 < \sigma_T < 30000$	Low and high noise removal
$R_{\sigma_1} < 0.7$	A-neighbour stripe filtration
$R_{\sigma_2} < 0.7$	Central pad stripe filtration
$R_{\sigma_3} < 0.7$	B-neighbour stripe filtration



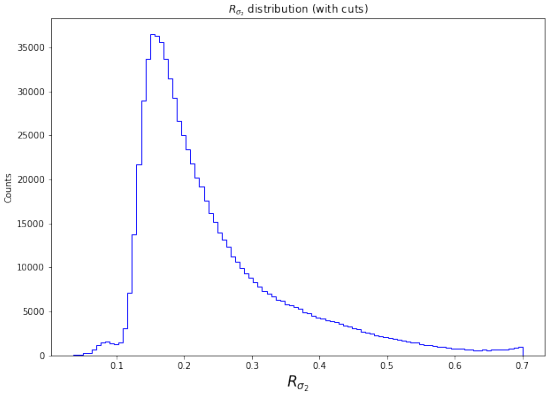
(a) R_{σ_1}



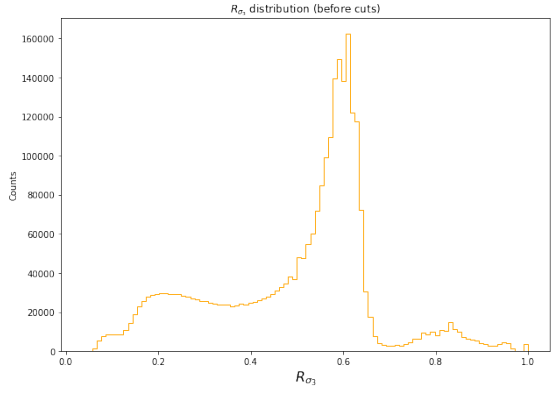
(b) R_{σ_1} after



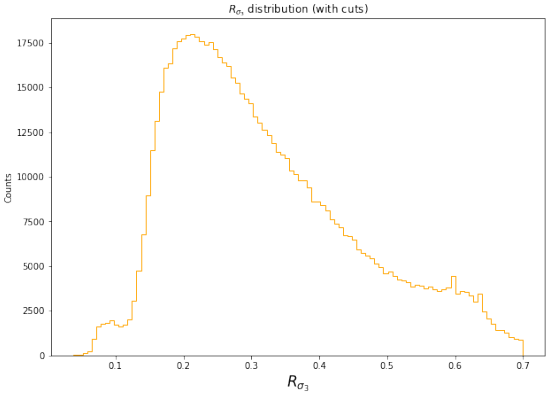
(c) R_{σ_2}



(d) R_{σ_2} after

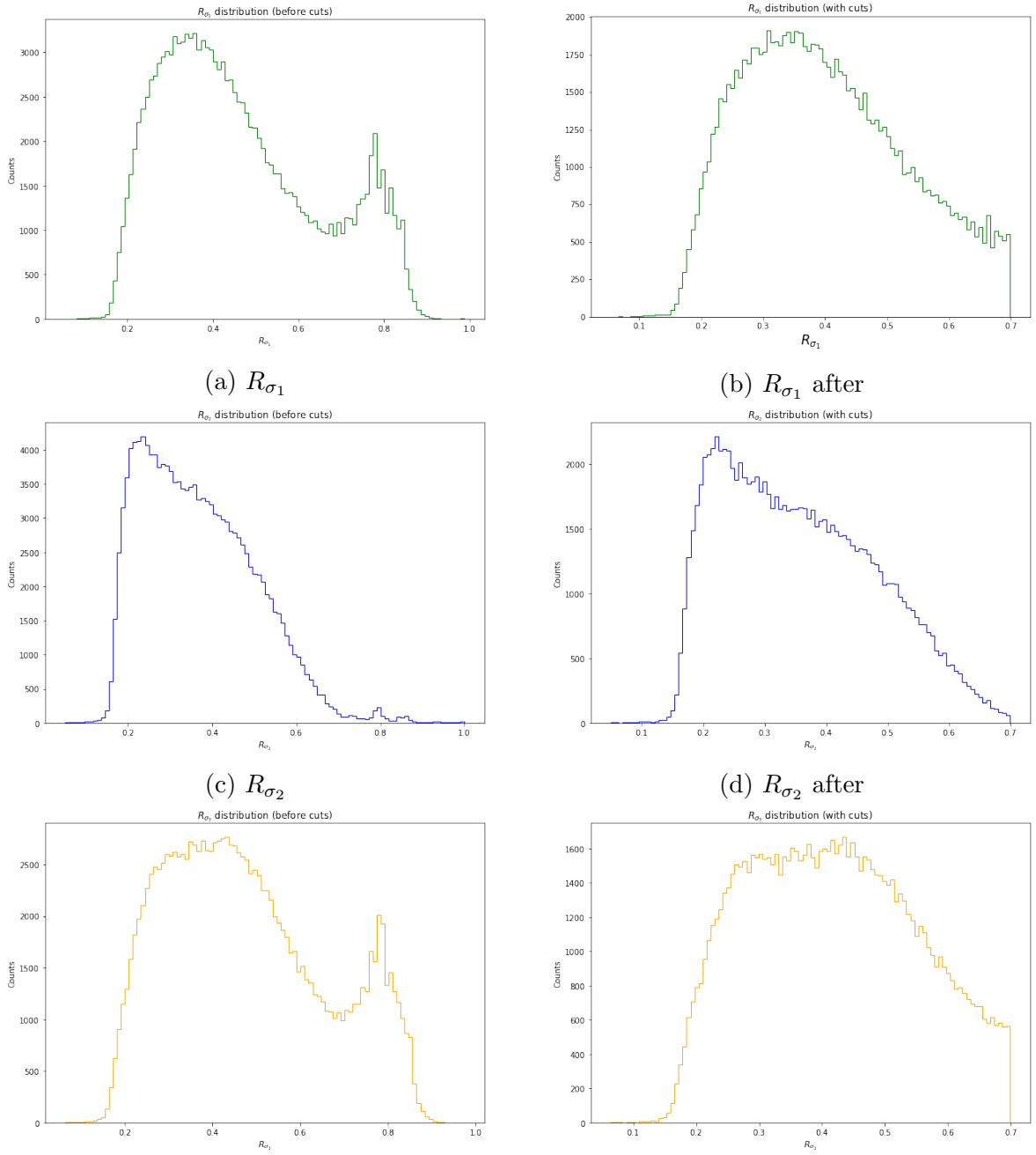


(e) R_{σ_3}



(f) R_{σ_3} after

Figure 4.5: R_{σ_i} distributions for ALICE data before and after applying quality criteria.

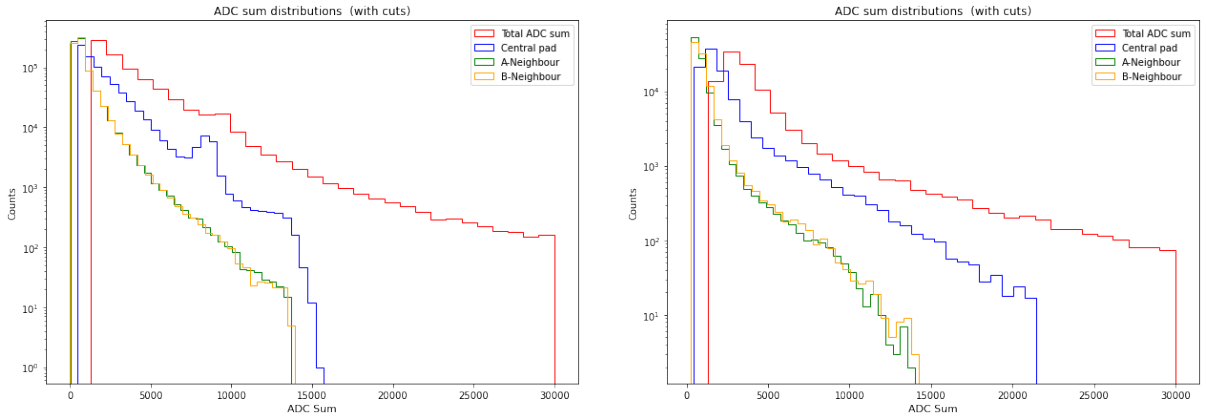


(a) R_{σ_1} (b) R_{σ_1} after

(c) R_{σ_2} (d) R_{σ_2} after

(e) R_{σ_3} (f) R_{σ_3} after

Figure 4.6: R_{σ_i} distributions for simulation data before and after applying quality criteria.



(a) ALICE data

(b) Simulation data

Figure 4.7: ADC sum distributions after applying quality criteria.

The implementation of quality requirements resolves the majority of inconsistencies initially observed and now ensure that the overall general behaviour of the real and simulated data are in reasonable agreement to allow for comparison.

4.3 Overall procedure

There are two major aspects in the overall process that is carried out in this thesis. The first entails the development of a strong classification mechanism through a machine learning approach and the second aspect deals with the evaluation of the effects of parameter tuning on simulation quality. This overall method is depicted in Figure 4.8. The total data batch consists of our entire conglomeration of real run data and simulated data. We perform a split on this data to create a training and testing set. Our machine learning models are trained with the train split data resulting in the formation of discriminators. Of the 10% testing data from the initial split we extract the real signals and combine them with new simulated data resulting from tuned simulations - this creates a new test set for each tuned parameter value which is sent to our discriminators for classification. The performance of the intial test set of the discriminator are compared to that of the metrics resulting from prediction on each of the tuned parameter test set data.

4.3.1 Parameter tuning strategy

There are numerous parameters within the O^2 software that can be modified, however due to time constraints we consider a single simulation parameter.

We first assess the effect of tuning an individual parameter above on the average pulse height spectrum. Thereafter we subject the tuned data to our discrimination mechanisms and compare any correspondence in the behaviour.

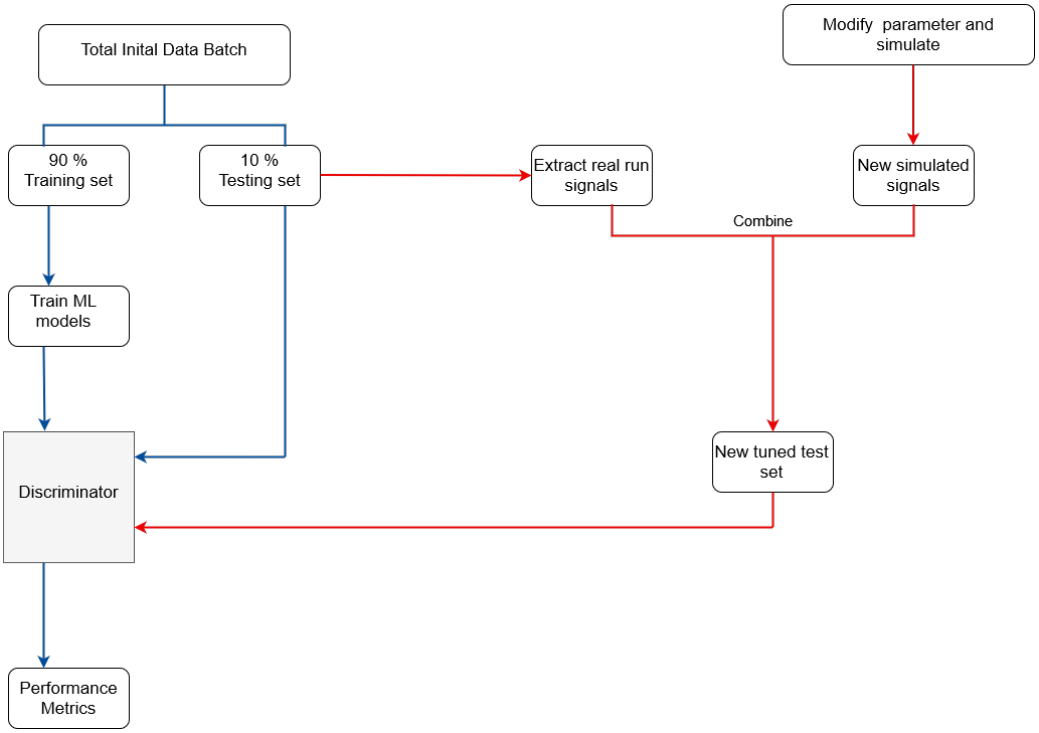


Figure 4.8: A flow diagram of the overall procedure followed.

4.3.2 Constancy of discriminator strength

In order to assess the effect of tuning parameter(s) on improvement of the simulation it is imperative to maintain a constant discriminator strength. As such after a parameter is tuned and new data generated based on the tuned parameter, we do not subject our discriminator to re-training on the new tuned data since this would invariably allow for the tuned data configurations to be gauged and de-sensitize the discriminator to the parameter that truly results in a better conformation of the simulated data to the real run data. This applies to all machine learning models that are employed as discriminators but particularly to deep learning methods due to their universal approximation nature.

Chapter 5

Results and analysis

In this chapter a detailed account of the results of our methodology implementation are presented. The overall properties of the signal types are first contrasted through an exploration of the data. The possible disparaties then serve as a motivation for the consideration of parameter tuning. The construction and usage of strong discriminator mechanisms are elaborated followed by a discussion of the simulation parameter tuning strategy employed.

5.1 Exploratory analysis

We first conduct an exploration of the characteristics and behaviour exhibited by the real and simulated signals and assess discrepancies and similarities that exist. Specific aspects of exploration and visualization are individually discussed in each subsection below.

5.1.1 ADC information

Descriptive statistics are computed for each individual signal array of each class (real and simulated) and visualized as a distribution. These are depicted in Figure 5.1 and Figure 5.2. Due to the class imbalance amongst real and simulated signals, the normalized counts/densities of each distribution are plotted to allow for comparisons without misrepresentations arising due to the different sample sizes. Five number summaries for these distributions are presented in Table 5.1. Both real and simulated data consist of a large proportion of signals with a lower average and cumulative ADC. Hence in both data sets, weaker signals are seen to be more prevalent to strong ones. There is a greater proportion of weaker signals in the real data which is also more right skewed than the simulated data. The real data has a lower median average ADC and cumulative ADC per signal in comparison to simulated data. The min and max cumulative ADC per signal for both real and simulated data conform to the upper and lower cuts imposed on the data during pre-processing.

In reference to the standard deviation per signal in Figure 5.1 it is observed that a higher proportion of simulated signals have low standard deviations. The median standard deviation per signal of real data surpasses that of simulated data. Overall it is seen that the real data consists of a greater proportion of signals which exhibit high variability in comparison to simulated data, however signals with much higher variability than what is observed in real data is present in simulated data in minuscule proportion. This is seen from the existence of the distribution beyond the termination of the real one.

The distribution of ADC values for the entire separate batch of real and simulated data shows that lower ADC values are more frequent as supported by the previous distributions of the cumulative and average ADC information. The simulated data exhibits a higher median ADC value. It is interesting to note the non existence of zero ADC values in simulated data since the minimum ADC value is 4. Both data sets consist of signals which have a maximum ADC value of 1023 which is expected.

The maximum ADC value per signal depicted in Figure 5.2 reveals that the real data consists of a larger proportion of signals with a higher maximum ADC value. Both real and simulated data distributions in Figure 5.2 decay with increasing maximum ADC value per signal, with the decay being greater for simulated data. A minor peak is observed around an ADC value of approximately 1000 for the real data with a sharp spike of the simulated data distribution at an ADC value of 1023. This suggests that the simulated data contain a higher proportion of signals where the maximum ADC value is the largest permitted ADC value (1023) in comparison to real data.

A possible reason for the deviations in ADC distributions could relate to the composition of particles in each data set category (real and simulated). The simulated data was generated using a pion particle boxgun simulation and as a consequence is exclusively populated with pions. This is not the case with the ALICE data which is non-homogenous in its particle composition and populated with pions as well as electrons.

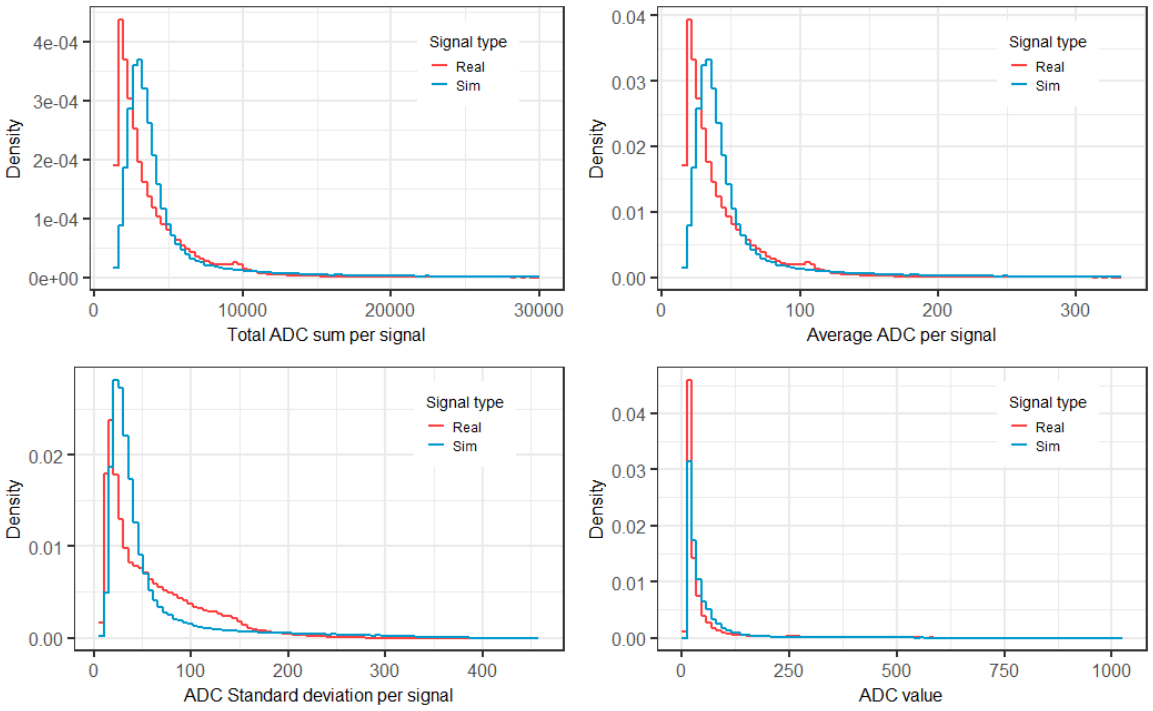


Figure 5.1: Comparison of descriptive statistical distributions of ADC information for real and simulated data.

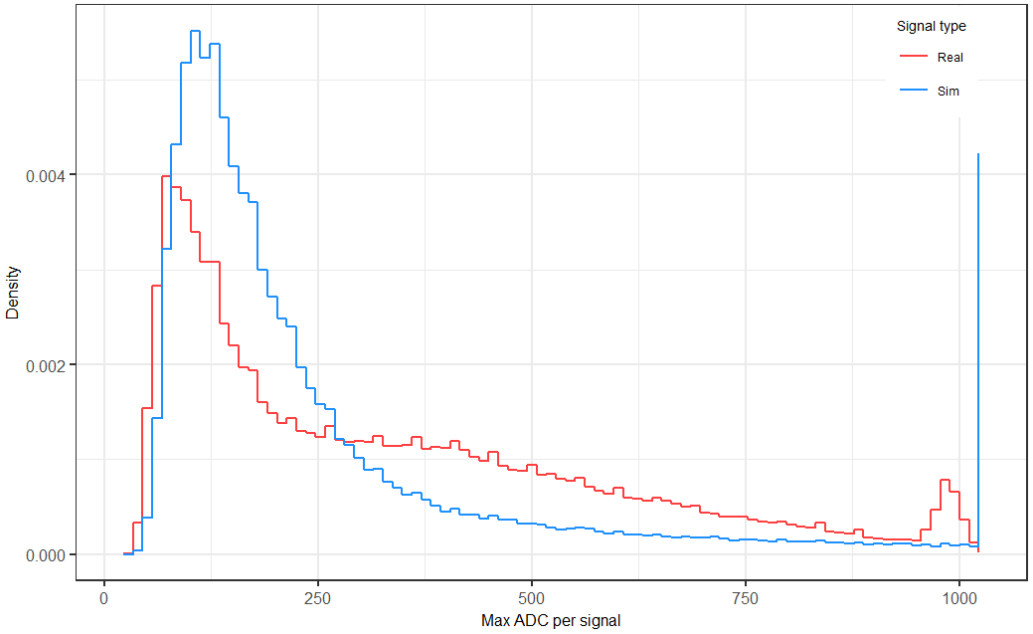


Figure 5.2: Comparison of distributions of the max ADC value per signal.

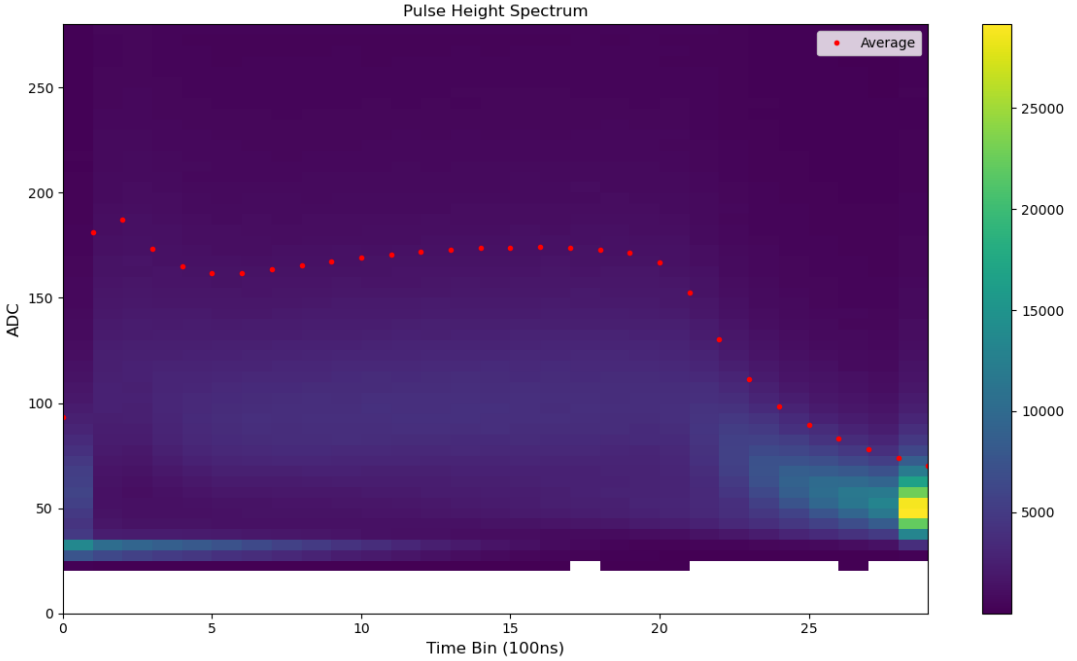
Table 5.1: Summary statistics of descriptive distributions

		Five-number summary				
Descriptive Distribution	**Data**	**Min**	**Q1**	**Q2**	**Q3**	**Max**
Cumulative ADC per signal	Real	1301	1893	2736	4550	29997
	Sim	1301	2598	3301	4533	29999
Avg ADC per signal	Real	14.56	21.03	30.40	50.56	333.30
	Sim	14.56	28.87	36.68	50.37	333.32
Standard deviation per signal	Real	3.16	19.26	42.03	83.69	307.89
	Sim	4.88	22.42	32.26	53.36	456.05
Maximum ADC per signal	Real	23	116	256	478	1023
	Sim	31	111	167	288	1023
ADC values	Real	0	11	16	34	1023
	Sim	4	14	24	51	1023

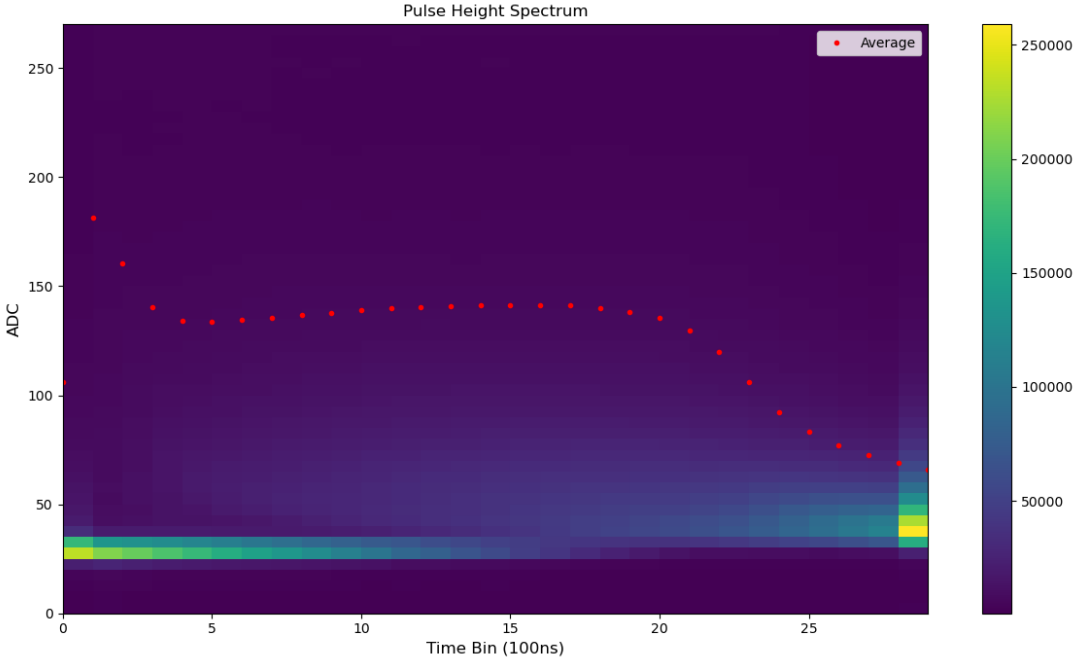
5.1.2 Pulse height spectra

Two dimensional pulse height histograms are presented as heatmaps for simulated and real data signals in Figure 5.3a and Figure 5.3b respectively. The two dimensional representation allows for a contrast of the interaction of the detector with all incoming signals and facilitates the easy observation of the frequency of signals with a particular ADC sum value at a certain time bin. It also allows us to gauge how the overall behaviour of all signals and how the saturation of certain ADC values at certain time bins skews the average pulse height into conforming to the familiar characteristic shape as seen in Figure 2.5.

Figure 5.3a is blank for ADC sums below approximately 25 throughout all time bins indicating that all simulated signals possess nonzero ADC sums (greater than 25) across all time bins. This is supported by the ADC value distribution Figure 5.1 and the associated summary statistics presented in Table 5.1 which affirm that the minimum ADC value in the simulated signal data is nonzero. The opposite is true for the real data with a considerable amount of entries of zero ADC sums across all time bins as depicted in Figure 5.3b.



(a) Simulation data



(b) Real data

Figure 5.3: Two dimensional pulse height spectra.

In both two dimensional histograms, horizontal streaks which eventually fade at higher time bins are present at an ADC sum value of approximately 30 to 40. These represent signals with low ADC values and hence low ADC sums in the first 15 time bins. The horizontal streaks are more pronounced in real data. For the simulated data it is clear how the saturation of signals with a particular ADC sum at a given time bin skews the average pulse height to conform to the observed shape. This is not as clearly visible in the case of real data.

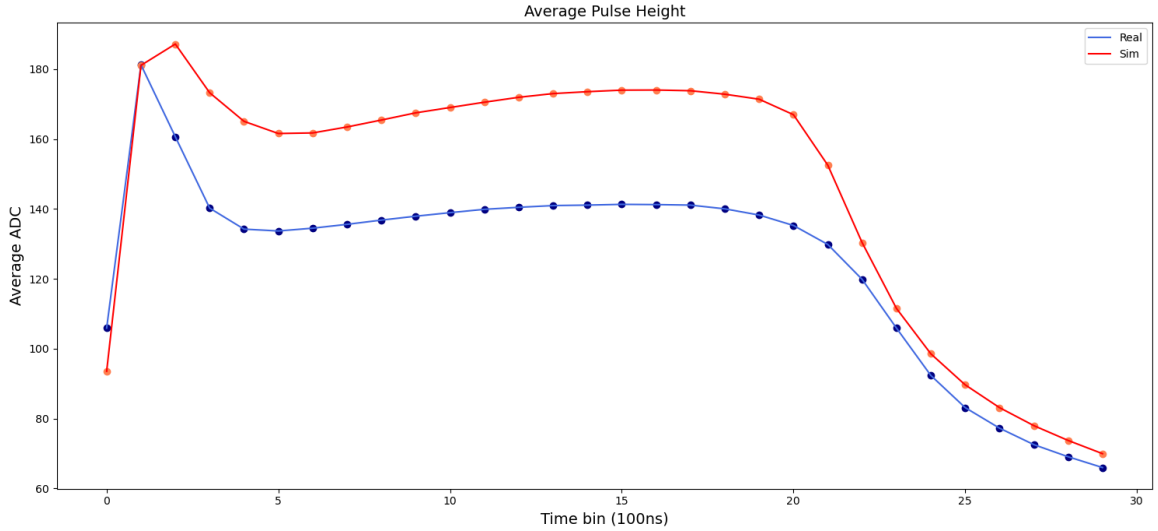
Naive comparison of the average pulse heights which are overlaid on the two dimensional histograms reveals some apparent differences. The average pulse height of the simulated signals are deficient of the expected amplification peak at early time bins as seen to be present for the real signals in Figure 5.3b. Both average pulse heights exhibit relatively flat behaviour in the drift region time bins as expected for pions which do not produce transition radiation. The ion tail however is observed to be slightly steeper in the simulated signals.

For a comprehensive comparison of the global traits of the real and simulated signals, the average pulse heights are superimposed on a single set of axes as shown in Figure 5.4a. The most glaring discrepancy amongst the two curves is the difference in height which is particularly pronounced in the drift region. Overall the simulated pulse height exceeds the real pulse height for all time bins except the first. Although the deficiency of a well pronounced peak in the simulated pulse height has been previously noted, consideration of the highest point on the curve as the peak reveals that the amplification peaks of the two curves are not in the same time alignment, with the simulated amplification peak occurring one time bin later than the real one. This may possibly point to a fault in the time response function within the simulation mechanism. Comparison of the curves at later time bins reveals that the ion tail of the simulated pulse height declines at a steeper rate in contrast to the real curve.

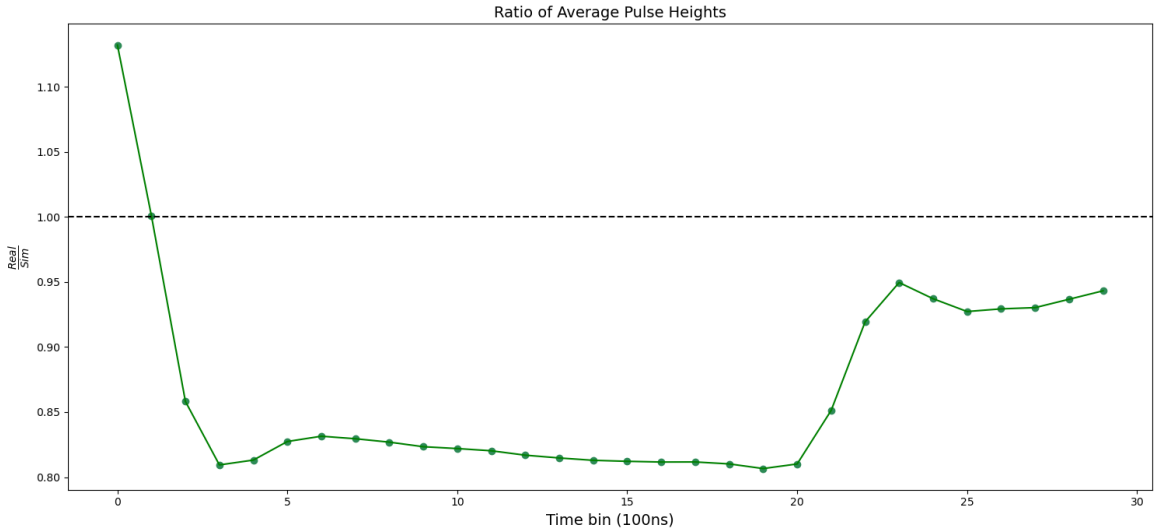
To accentuate the discrepancies between the curves, the ratio of the real and simulated average pulse heights are computed and presented in Figure 5.4b. The ratio plot is helpful in facilitating easier comparison of the ADC value dissimilarities of the two curves in each time bin. In the ideal case where the real and simulated average pulse heights are exactly identical their ratio resolves to unity which can be expressed as

$$\frac{\langle PH \rangle_{Real}}{\langle PH \rangle_{Sim}} = 1. \tag{5.1}$$

This is indicated by the horizontal line in Figure 5.4b.



(a)



(b)

Figure 5.4: a) Dual comparison of average Pulse height spectrum b) Ratio of Pulse height spectrum.

We examine the ratio curve in Figure 5.4b together with reference to the horizontal line at unity in order to assess the extent of deviation from an ideal scenario. Since the ratio plotted is $\frac{\text{Real}}{\text{Sim}}$, as the simulated pulse height rises in relation to the real one, the ratio becomes smaller and further below unity. Parallel trends exhibited by the ratio curve indicate similar gradients between the real and simulated pulse heights. Parallel trends closer to the ideal ratio line denote similarity amongst the average pulse heights. Non parallel trends further from the ideal ratio line at unity denote larger prominent dissimilarities amongst the pulse height curves.

The ratio curve initially begins above the the ideal ratio line which reflects the real average pulse height superior to the simulated one in the first time bin. The intersection of the ratio curve with the ideal ratio line denotes the intersection of both pulse heights observed in the second time bin in Figure 5.4a. The ratio curve then recedes below the ideal ratio line after the third time bin where the discrepancy between the pulse heights exacerbates. The ratio curve is observed to gradually decline from time bins 6 to 20 which is indicative of the non constant discrepancy in the ADC values throughout the drift region. This gradual decline is attributed to the gradual rise in the simulated average pulse height in relation to the real one within the drift region. The ratio curve is lowest at time bin 20 where the dissimilarity in the average pulse heights in Figure 5.4a is greatest. This corresponds to the entry point into the drift region of a read out chamber. The ratio curve then begins to approach the ideal ratio till time bin 23 where it eventually staggers, corresponding to the convergence of both pulse heights in the ion tail region as noted in Figure 5.4a. The non parallel nature of the ratio curve however informs of the subtle difference in steepness exhibited between the real and simulated ion tails.

Supermodules

For a deeper insight we conduct an exploration of the average pulse height for each individual supermodule in order to assess whether the discrepancies noted in the overall average pulse heights are uniformly present for all supermodules. The average pulse height spectra for each supermodule with the real and simulated data superimposed together is presented in Figure 5.5. Their corresponding ratio curves are presented in Figure 5.6. Collectively supermodules 15, 16 and 17 present the most dissimilarity in pulse heights. The greatest individual dissimilarity is seen in supermodule 16 where the real pulse height curve exhibits drastic fluctuations. The greatest similarity amongst the real and simulated data is seen in supermodule 7 whose corresponding ratio curve is also closest to the ideal ratio. The slight variations in the pulse height simulations amongst supermodules can be possibly attributed to an unequal number of extracted TRD signals belonging to each supermodule which may influence the average ADC value per time bin, thereby varying the overall average pulse height curve in a supermodule.

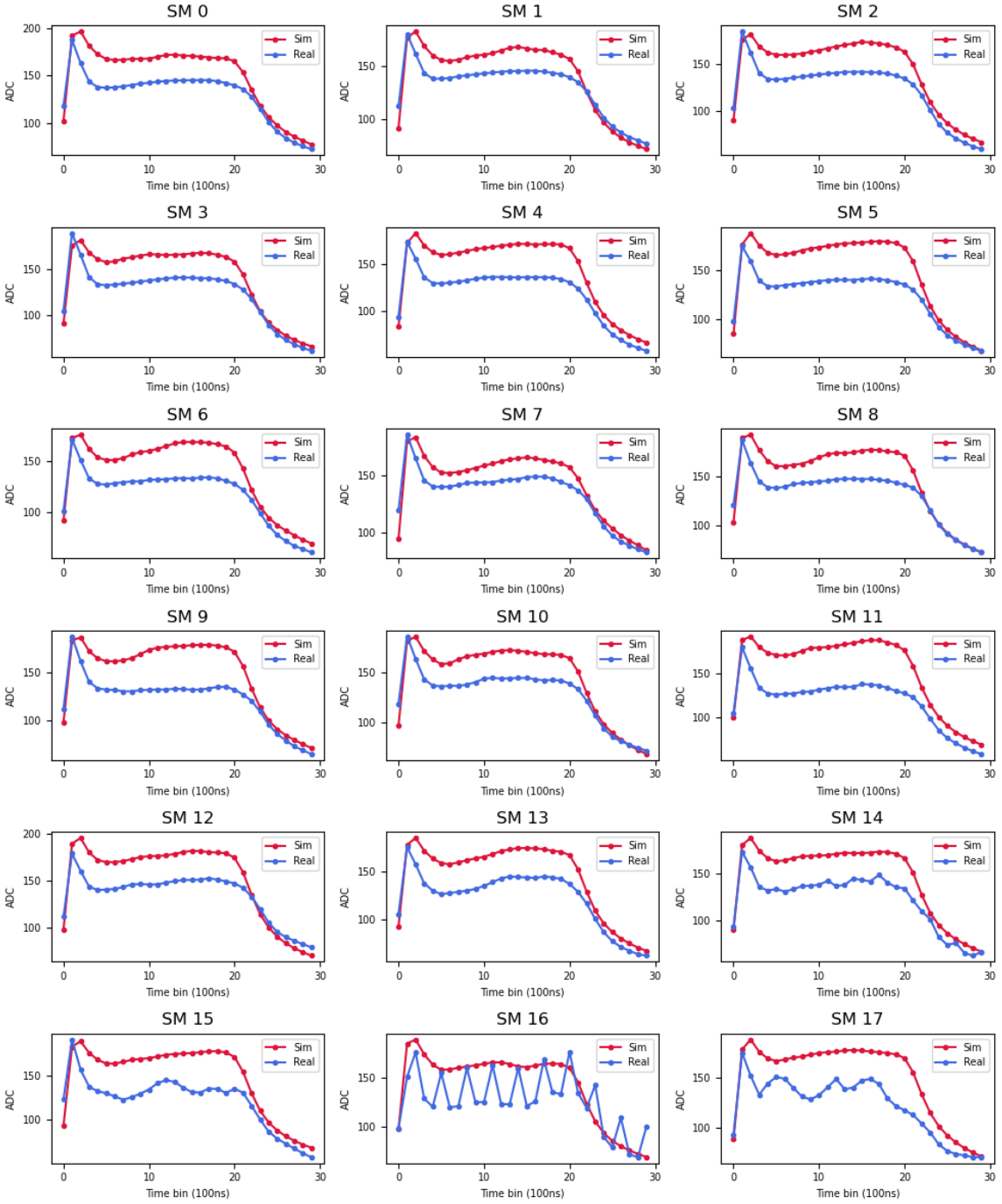


Figure 5.5: Dual comparison of average pulse height per supermodule.

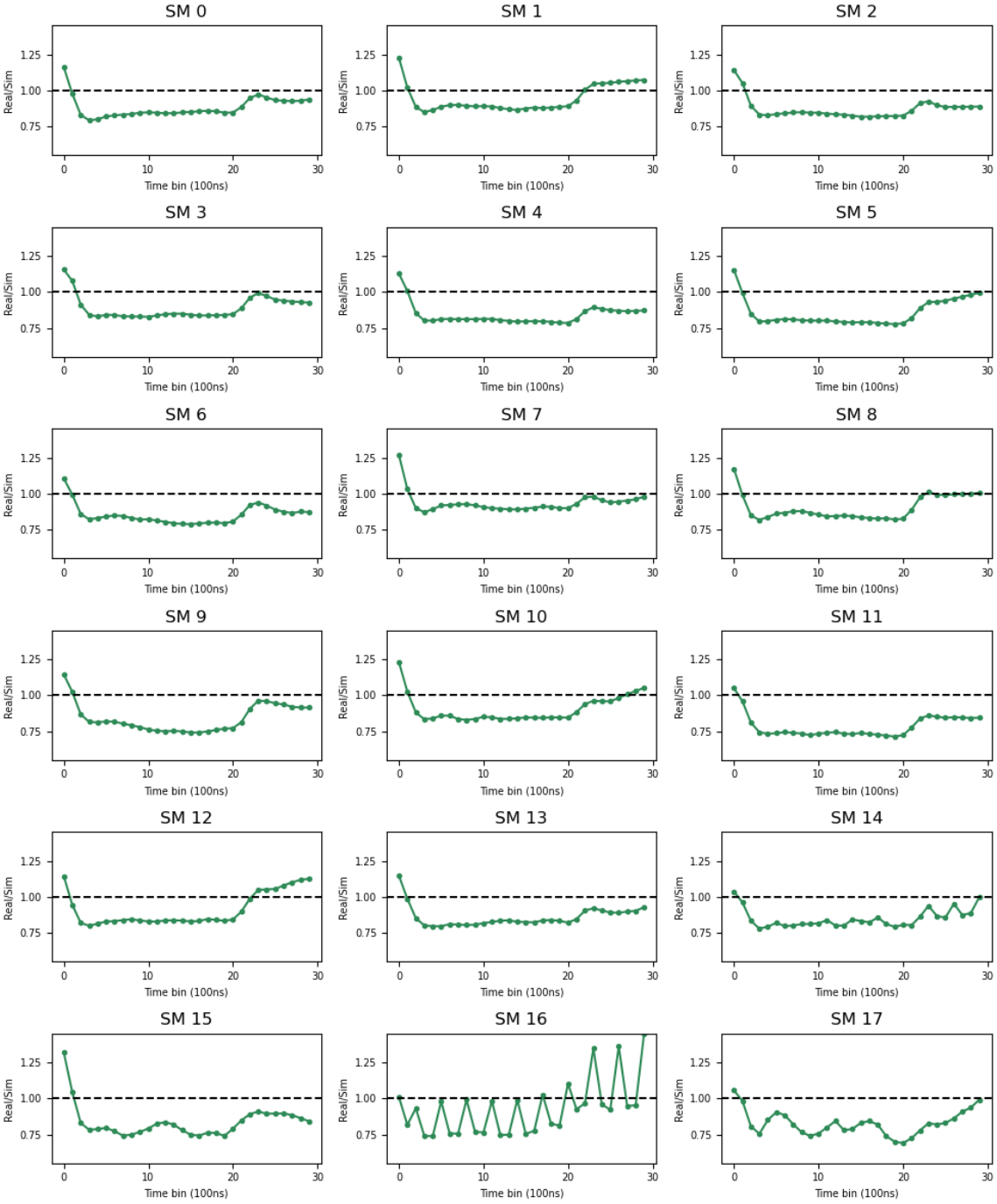


Figure 5.6: Ratio of average pulse heights for each supermodule.

5.2 Construction of strong discriminators

Robust machine learning models with strong classification capabilities are required as discriminators in order to ensure the significance of the effects of parameter tuning on simulation improvement. The quality of the tuning can be judged by the extent of improvement based on the strength of the discriminator used and the amount by which its discrimination strength diminishes. Hence a large depletion of a weak discriminator's strength as tuning is performed is not as significant in comparison to a slight depletion of a strong discriminator. In all model developments hereon, we assign simulated signals as the positive class (denoted by 1) and real signals as the negative class (denoted by 0).

5.2.1 Convolutional neural networks

We initially consider a simple network architecture and continue with a stacked approach of adding additional complexity in order to improve model performance if necessary. The initial architecture adopted is illustrated in Figure 5.7 and depicts a single convolutional layer leading to a pooling layer which in turn leads to a flattening layer which is fully connected to two output neurons with a softmax activation. A dropout layer is employed immediately after the flattening layer as a regularization mechanism.

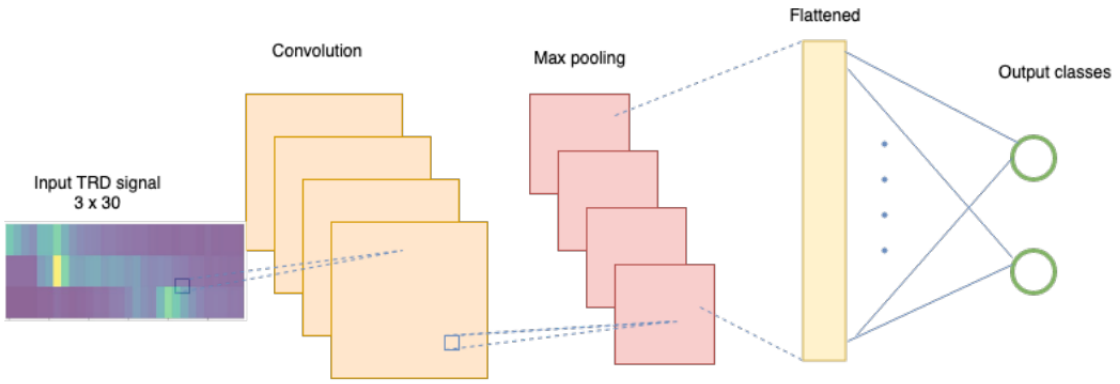


Figure 5.7: Initial general architecture adopted for the convolutional neural network.

Hyperparameter tuning

Hyperparameter optimization was performed by training CNN models based on the initial adopted architecture in Figure 5.7 over a parameter search space which is presented in Table 5.2 together with the optimal values. There are 180 unique combinations of the parameter values. As consequence 180 CNN models were trained for 12 epochs with a diverse combination of parameters for the adopted architecture. The purpose of training

for only 12 epochs is to ensure a swift search throughout all permutations of the parameter search space and assess which combinations allow a quicker convergence to higher accuracy within a few epochs. An illustration of the hyperparameter tuning results is provided in Figure 5.8.

Table 5.2: Hyperparameter tuning space for initial CNN architecture

Hyperparameter	**Values searched**	**Optimal values**
Convolution filter kernel size	2, 3	3
Activation	relu, tanh, sigmoid	relu
Number of convolution filters	10, 32, 64, 80, 120	120
Maxpooling kernel size	2, 3	3
Dropout rate	0.1, 0.2, 0.3	0.3

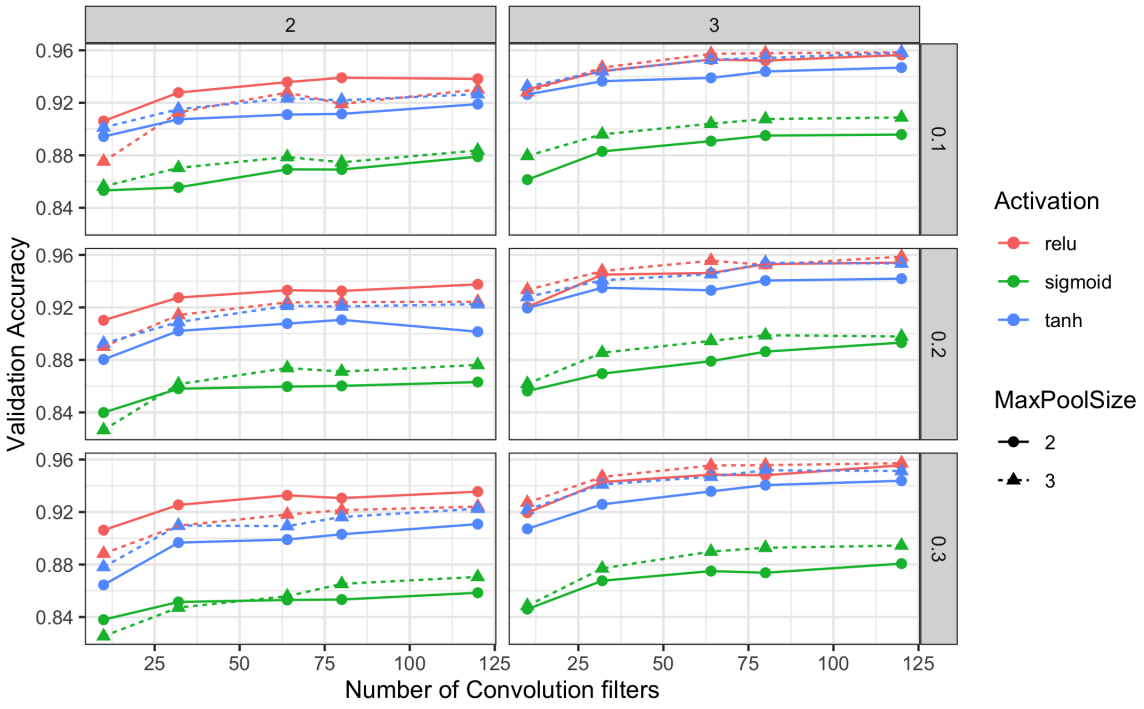


Figure 5.8: Initial CNN architecture hyperparameter tuning results.

In general it is observed that a higher number of convolution filters leads to higher validation accuracy. The relu activation outperforms the tanh and sigmoid functions. A kernel size of

3 for the pooling and filter size is seen to produce better performance. Using our optimal hyperparameters the CNN model architecture is retrained for 50 epochs with a batch size of 256. As a further regularization mechanism, early stopping is employed with a patience of 5 epochs. The retrained model achieved a validation accuracy of 95.86 %.

Additional complexity is now added to develop a new model architecture with better performance. A single modification is made to our initial adopted architecture through the addition of a hidden fully connected layer of neurons after the flattened layer. This new architecture is illustrated in Figure 5.9. We now focus on determining the optimum number of neurons in the hidden layer. The optimal parameters from the tuning of our initial architecture are used as a guidance and kept constant in our second tuning search with the exception of the number of convolution filters. The hyperparameter tuning space for our second adopted architecture is presented in Table 5.3 together with the optimal values. The proposed parameter tuning space yields 135 unique combinations resulting in the construction and evaluation of 135 models which are trained for 10 epochs with a batch size of 256. An illustration of the validation performance of each of these models is depicted in Figure 5.10.

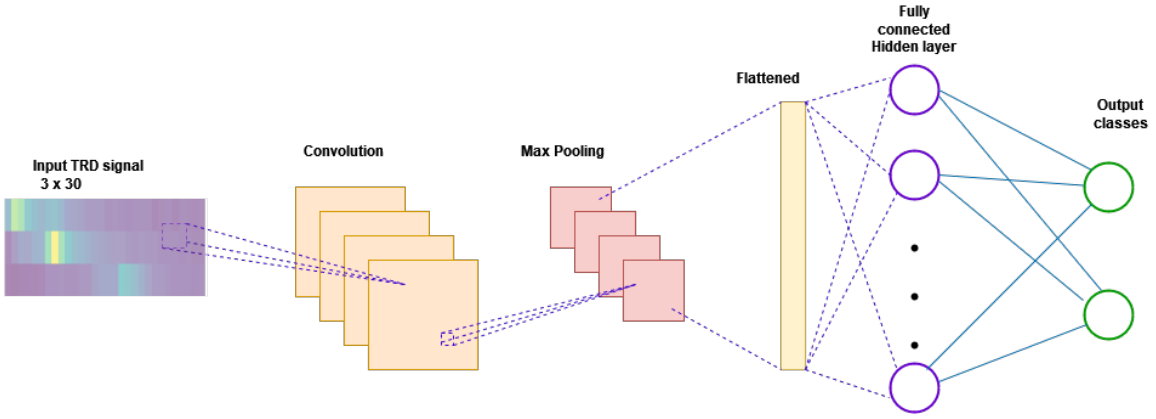


Figure 5.9: Second CNN architecture with additional single fully connected hidden layer.

Table 5.3: Hyperparameter tuning space for 2nd CNN architecture

Hyperparameter	**Values searched**	**Optimal values**
Number of convolution filters	64, 80, 120	120
Number of hidden neurons	6, 12, 24, 32, 64	64
Dropout 2	0.1, 0.2, 0.3	0.3
hidden activation	relu, tanh, sigmoid	relu

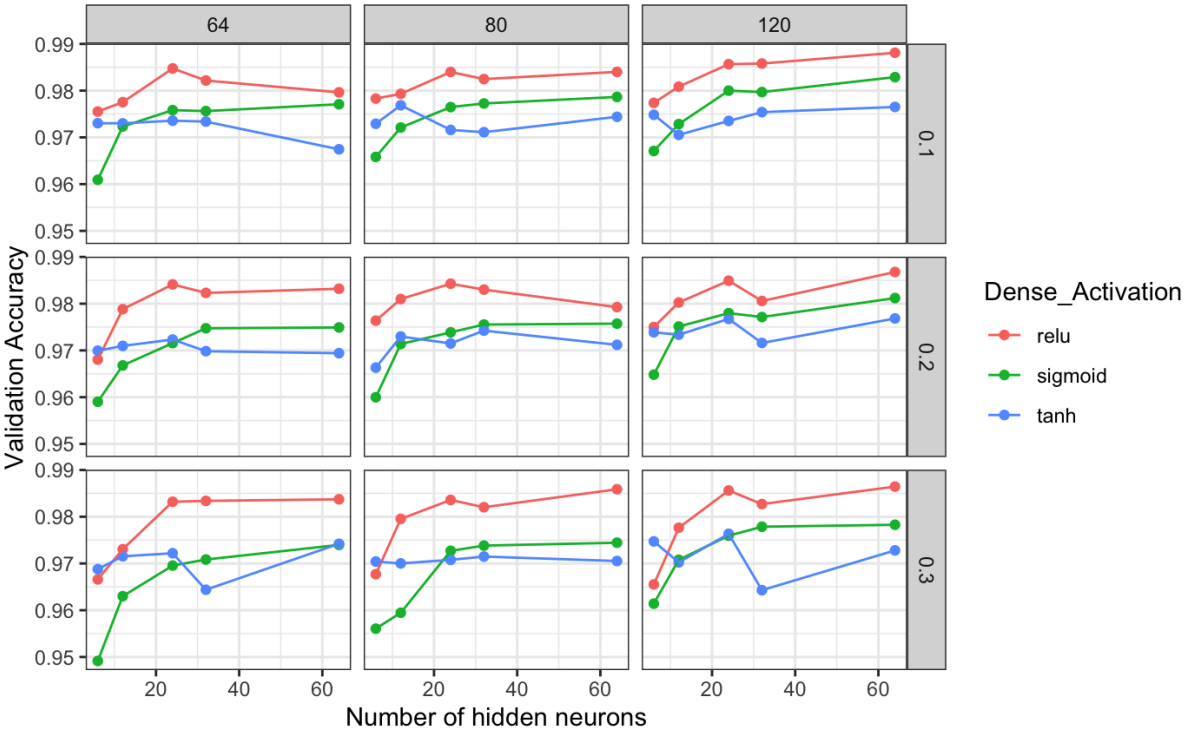


Figure 5.10: Second CNN architecture hyperparameter tuning results.

5.2.2 Artificial neural networks

We now employ the use of a fully connected artificial neural network as a comparative deep learning method to our CNNs. In this context the TRD signals which are in a 3×30 format are flattened to an array of 90 values which can then be fed into our ANN. A simple general architecture with an input layer, a single hidden layer and an output layer of two neurons with softmax output is considered. This is depicted in Figure 5.11. A dropout layer serves as an intermediary between our hidden layer and output to incorporate regularization.

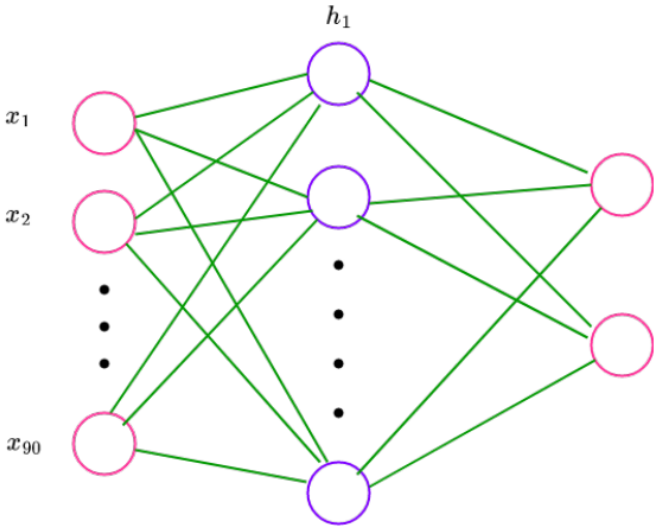


Figure 5.11: Initial general architecture adopted for the artificial neural network.

Hyperparameter tuning

The optimal parameters for our ANN architecture are determined by searching the parameter space within the predefined space considered which is presented in Table 5.4 together with the optimal values. A total of 99 artificial neural networks are trained for 10 epochs and assessed on their validation accuracy, the results of which are graphically depicted in Figure 5.12.

Table 5.4: Hyperparameter tuning space for simple ANN architecture

Hyperparameter	**Values searched**	**Optimal values**
Number of hidden neurons	10, 20, 30, 40, 50 60, 70, 80, 90, 100, 120	120
Dropout 2	0.1, 0.2, 0.3	0.1
hidden activation	relu, tanh, sigmoid	relu

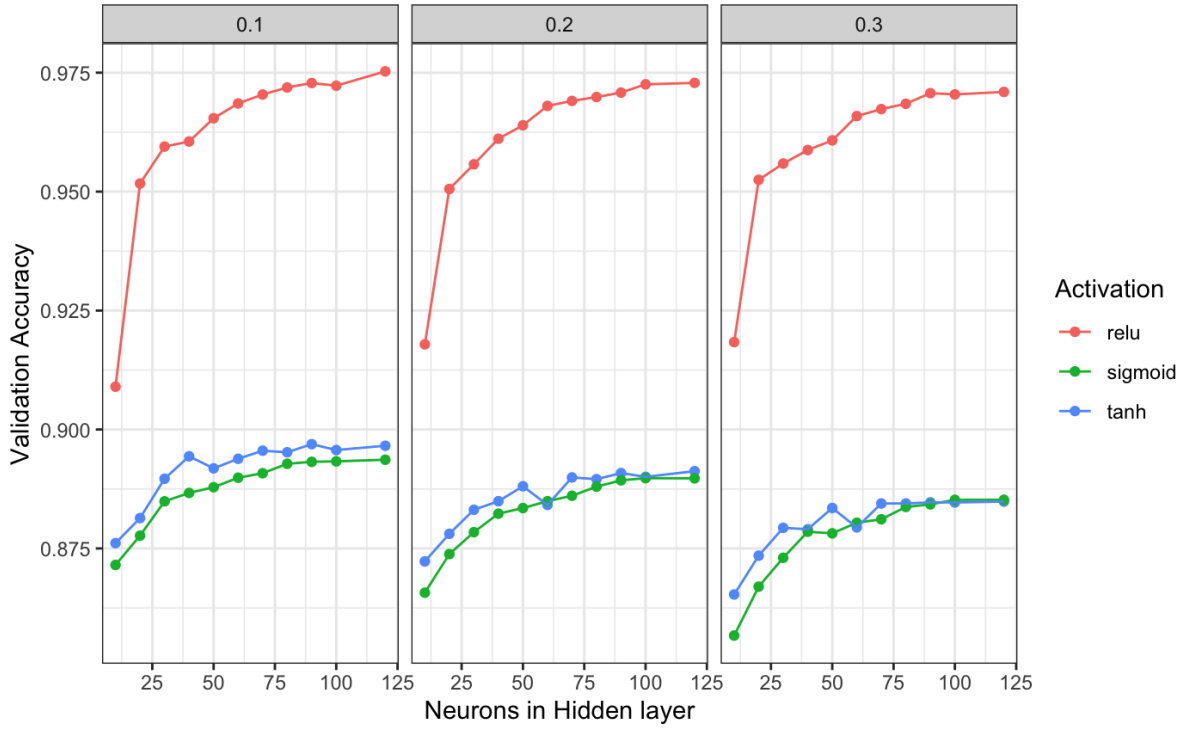
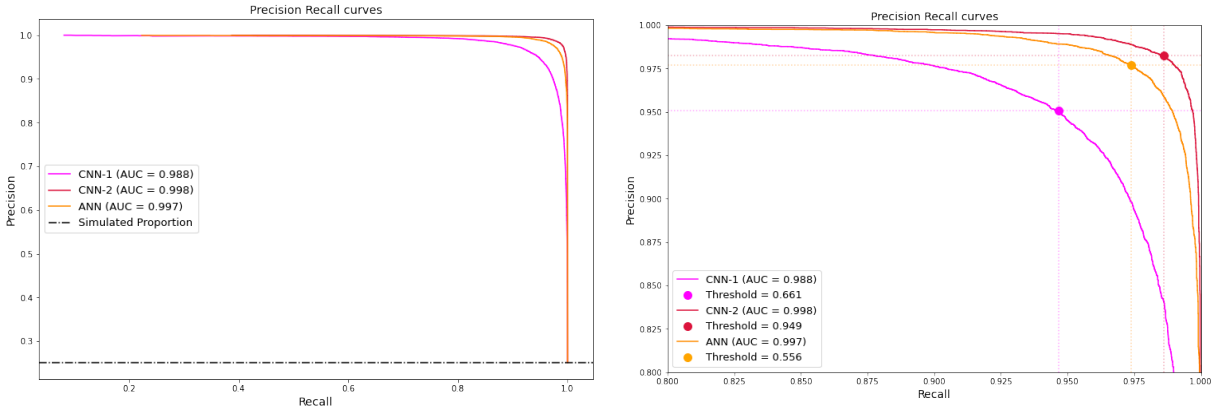


Figure 5.12: Initial ANN architecture hyperparameter tuning results.

5.2.3 Precision recall curve analysis

We generate the precision recall curve for each of our models on the validation set as seen in Figure 5.13. The optimal threshold was determined using equation 3.7. The validation set performance metrics of our models are presented in Table 5.5.



(a) (b)

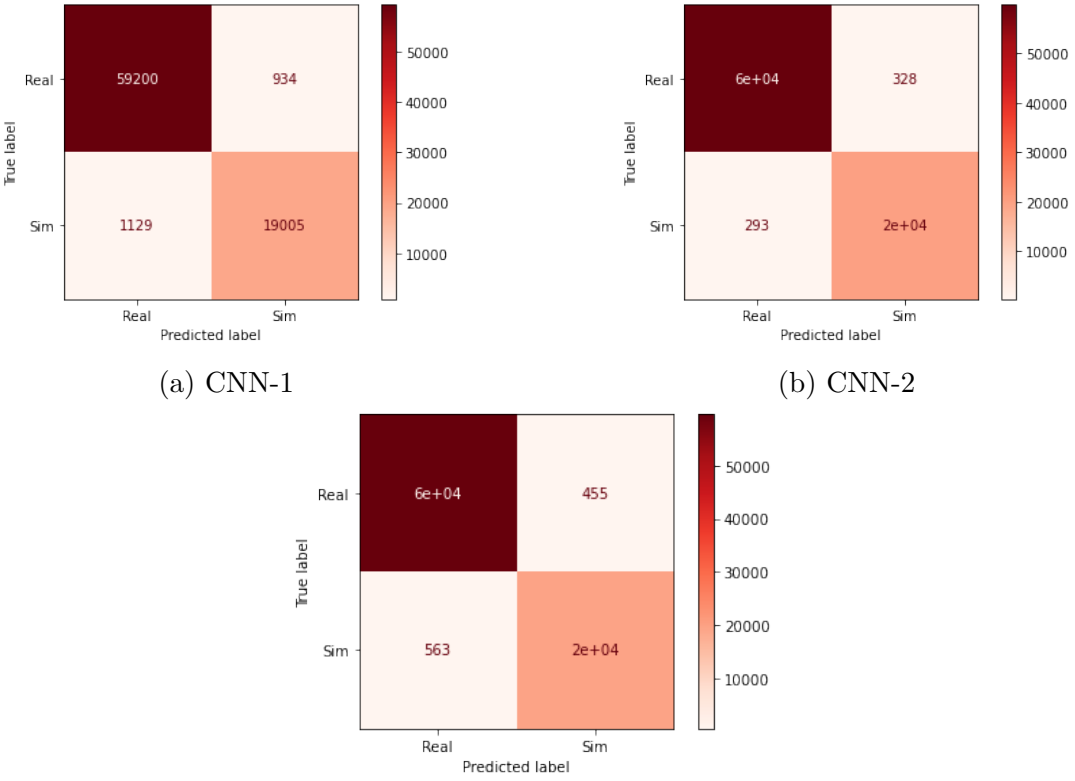
Figure 5.13: Precision recall curves for the validation data a) over entire domain b) zoomed in depicting threshold points.

Table 5.5: Performance metrics on the validation set for each model.

Performance Metric	**Model**		
	CNN-1	**CNN-2**	**ANN**
Threshold	0.661	0.949	0.556
Accuracy	0.9742	0.9920	0.9877
Precision	0.9506	0.9824	0.9771
Sensitivity	0.9464	0.9860	0.9737
F1 Score	0.9485	0.9842	0.9754
Specificity	0.9835	0.9941	0.9923
Miss Rate	0.0535	0.0139	0.0262

5.3 Initial test set predictions

The initial test set consisting of a conglomeration of real signals and untuned simulated signals are subjected to classification by the discrimination mechanisms that have been developed. The initial predictions of our machine learning models serve to demonstrate the reliability of our models on unseen data as well as highlight pertinent distinctions or patterns that are used to differentiate the signal types. The discrimination capability of our models on the initial test set also serves as a benchmark for their performance on the tuned simulation data. Confusion matrices of the initial test set predictions are shown in Figure 5.14 and the specific performance metrics of our deep learning models are presented in Table 5.6.



(a) CNN-1

(b) CNN-2

(c) ANN

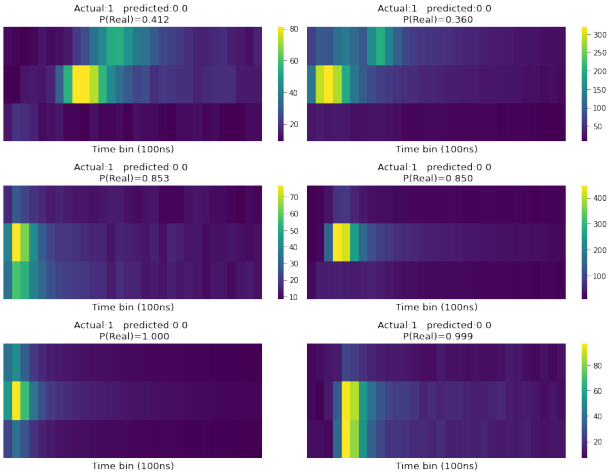
Figure 5.14: Confusion matrices of our deep learning models.

Table 5.6: Performance metrics on the initial test set for each model.

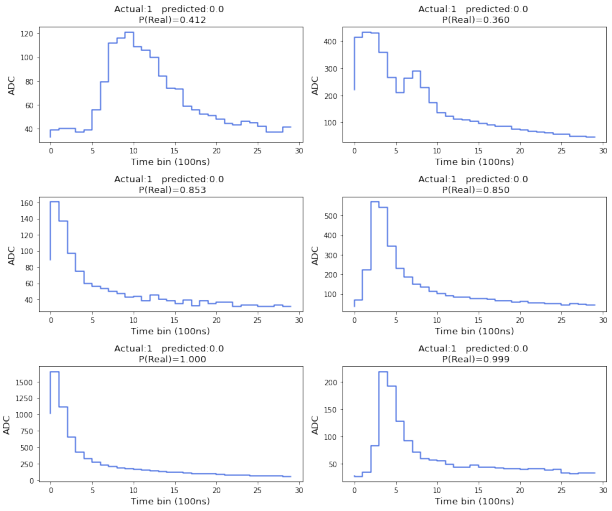
Performance Metric	**Model**		
	CNN-1	**CNN-2**	**ANN**
Threshold	0.661	0.949	0.556
Accuracy	0.9742	0.9922	0.9873
Precision	0.9531	0.9837	0.9772
Sensitivity	0.9439	0.9854	0.9720
F1 Score	0.9485	0.9845	0.9746
Specificity	0.9844	0.9945	0.9924
Miss Rate	0.0560	0.0145	0.0279

5.3.1 Examination of false negatives

The examination of signals that are false negatives is imperative since they represent simulated signals that deceived our models into perceiving them as real. The ability of such signals to "trick" our discriminators into falsely classifying them as real could imply that they possess certain traits of real data that allow them to mimic the appearance/behaviour of real signals to our models. Since the goal of this study is to manipulate the Monte Carlo simulation aspect of the data generation to conform to real data, the examination of any traits that allow simulated data to be characterized as real provides valuable insight. A sample of false negative signals arising from our CNN-1 model predictions is presented in Figure 5.15a with the corresponding pulse heights depicted in Figure 5.15b.



(a) False negative signals



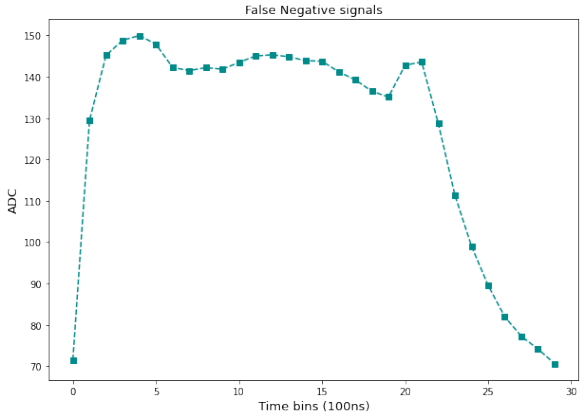
(b) Individual pulse heights

Figure 5.15: A sample of false negative signals from our CNN-1 model with corresponding individual pulse heights as step functions.

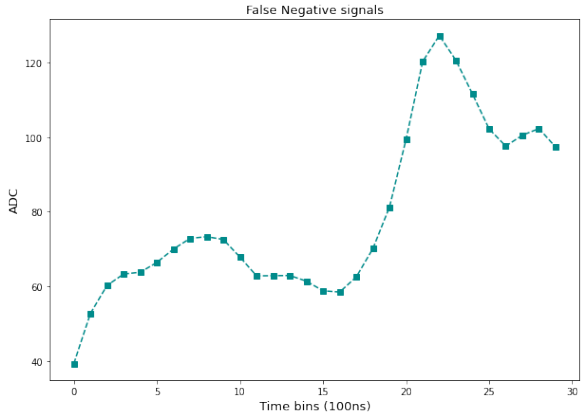
Pulse height spectra of false negatives

We perform a reconstruction of the average pulse height spectrum of the false negatives and assess their overall similarity to real data. This is illustrated in Figure 5.16. The study of the reconstructed pulse heights seeks to identify whether the simulated signals that are classified as real by our discriminators actually obey the similar overall temporal behaviour as seen by real run 3 signals. The reconstructed pulse height of the false negatives from CNN-1 exhibits the closest resemblance to real data in comparison to that of other models. The deficiency of the expected amplification peak for all reconstructed curves is still noted. The reconstructed pulse heights from the CNN-2 and ANN models depict peaks at later

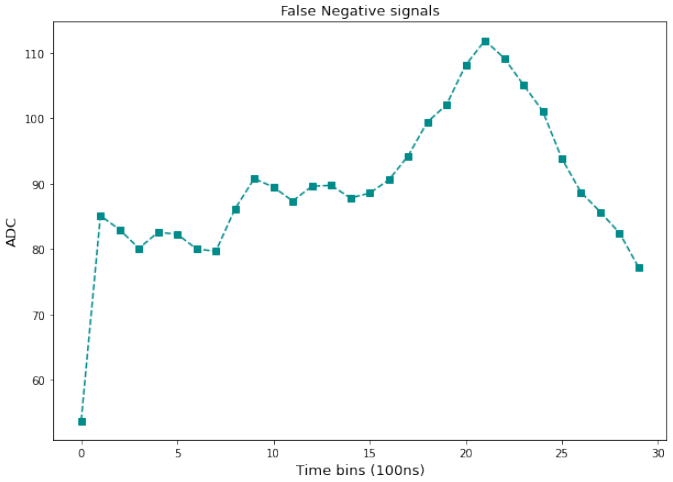
time bins and lower ADC sum values at earlier time bins with no proper formation of a plateau representing the drift region. From this it is deduced that the majority of simulated signals classified as real ones by the CNN-2 and ANN models contain an accumulation of high ADC values at later time bins.



(a) CNN-1



(b) CNN-2

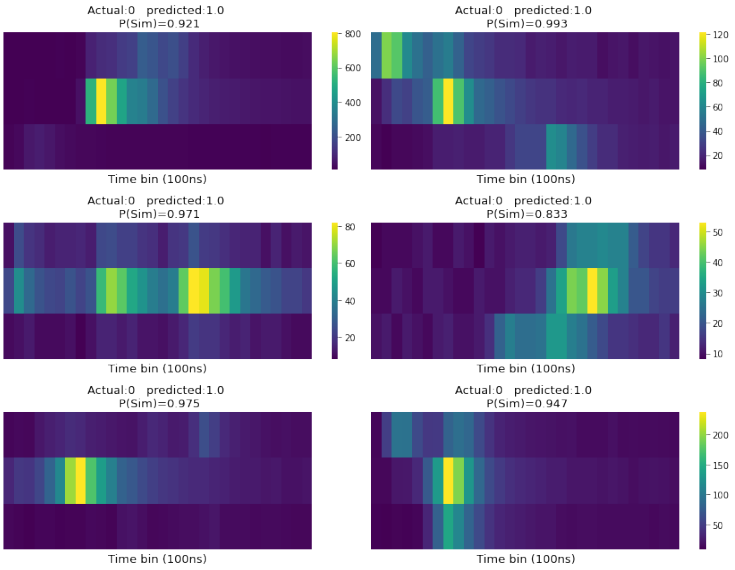


(c) ANN

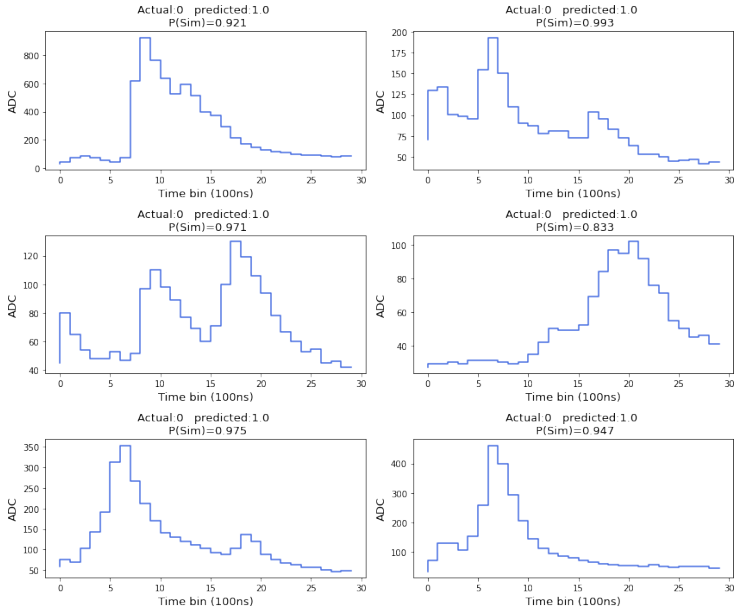
Figure 5.16: Pulse height spectrum of false negative signals.

5.3.2 False positives

These denote real signals that are incorrectly identified as being Monte Carlo generated. A sample of these signals together with corresponding pulse heights from our CNN-1 model is presented below in Figure 5.17.



(a) False positive signals

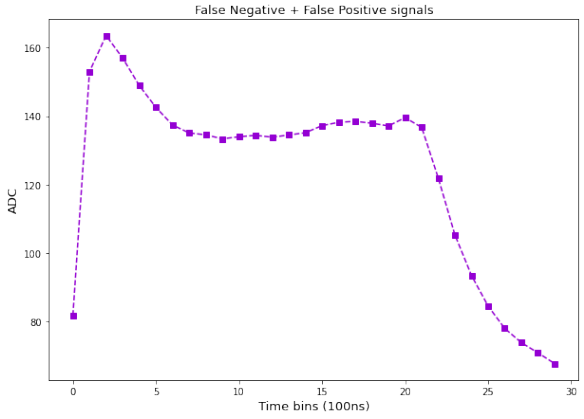


(b) Individual pulse heights

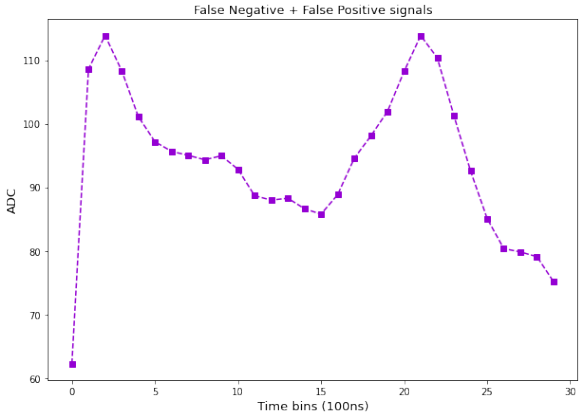
Figure 5.17: A sample of false positive signals from our CNN-1 model with corresponding individual pulse heights as step functions.

Pulse height spectra of combined false positives and false negatives

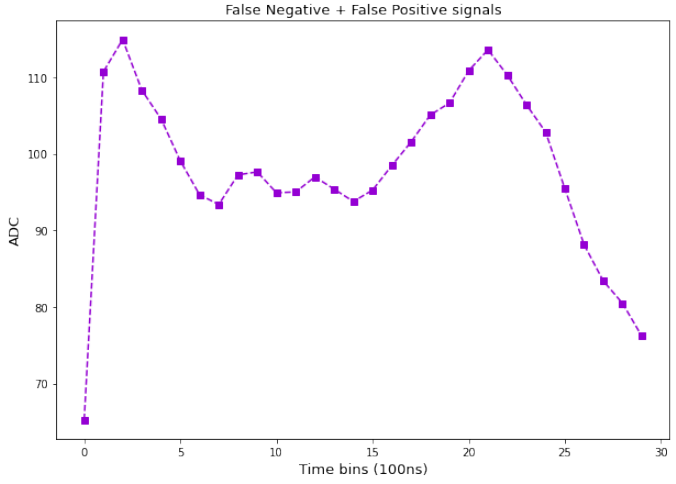
The average pulse height spectrum is now reconstructed from the entire collection of misclassified signals. This allows for the examination of any changes to the reconstructed pulse heights in Figure 5.16 which are brought about through the addition of false positives to our false negative signals. Figure 5.18 depicts these reconstructions and demonstrates that the addition of the false positives enhance the resemblance to real data to the extent that an amplification peak at early times is present. The peak at late times is still retained by CNN-2 and ANN resulting in a double peaked curve. The reconstruction from CNN-1 depicts the expected peak, drift plateau and ion tail characteristics and presents considerable similarity to the expected spectrum from real run data.



(a) CNN-1



(b) CNN-2



(c) ANN

Figure 5.18: Pulse height spectrum of combined false negative and false positive signals.

5.4 Tuning Monte Carlo simulations

5.4.1 Gas gains

The Xe gas gains are modified within the O^2 software and the necessary files are recompiled in order to enable the software to recognize the associated modification. The default gain is set to 4000. The gain is modified from 4000 till 400 in decreasing steps of 400 resulting in 10 tuned iterations inclusive of the default value. For each tuned gain value a 5000 event TRD simulation is performed to acquire data from which signals characterized by the associated gas gain value are extracted.

The average pulse height spectrum is generated for the signals obtained from each tuning iteration and compared with each other and real run data. This is depicted in Figure 5.19. The corresponding pulse height ratios is depicted in Figure 5.20. The modification of the gain constitutes a vertical shift which is observed in Figure 5.19. Increasing and decreasing the gain value leads to a vertical ascendance and descendance of the pulse heights respectively. This affirms that in general the gain proportionality relation asserted in Figure 2.5 is obeyed within O^2.

The optimal gain value can be considered to be the one that results in the simulated pulse height best resembling the real one in terms of proximity. From observation of Figures 5.19 and 5.20 the optimal gain value is 2800 since this results in the closest approach of the simulated pulse height to the real one and the pulse height ratio to the ideal ratio.

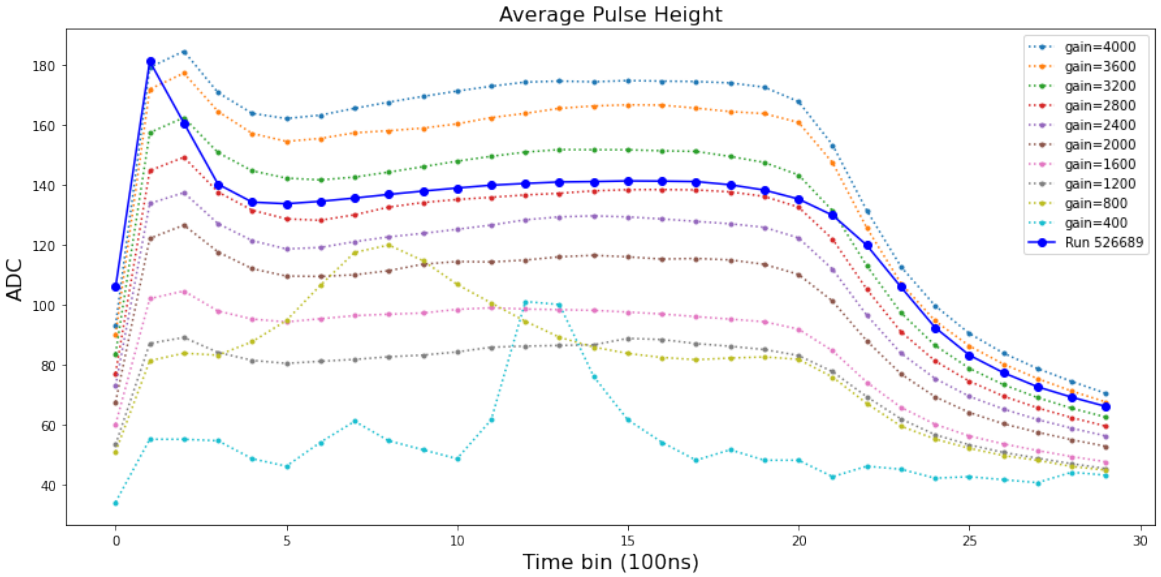


Figure 5.19: Pulse height spectrum of TRD simulated signals for modified Xe gas gain values.

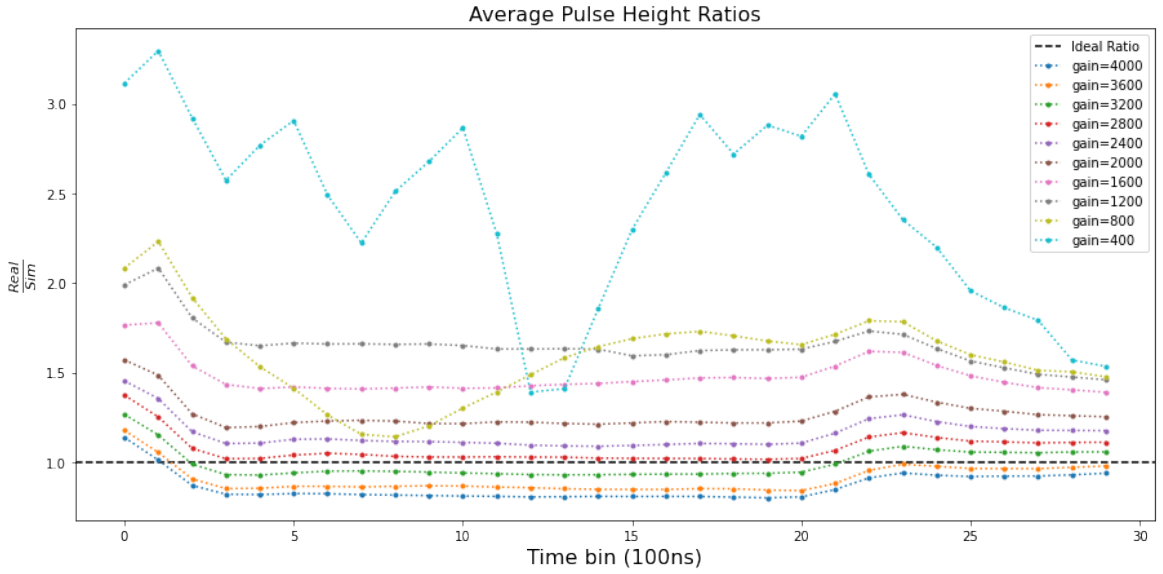


Figure 5.20: Pulse height ratio curves of of TRD simulated signals for modified Xe gas gain values.

5.4.2 Performance analysis of gain tuning

The performance metrics of our machine learning discriminators are examined for each tuned gain value in order to study the effect that our parameter tuning has on the predictive performance of our models. In the current context the inability to distinguish between real and simulated signals implies that the two signal classes are sufficiently similar in that our discriminator is deceived. An inference of deception or diminished performance on the part of our discriminators signifies such similarity of the real and simulated signals. The tuned gain value at which this deception occurs is suggestive of the optimal gain value. The dependence of our model performance metrics per tuned gain value is presented in Figure 5.21.

With reference to the accuracy curves, as the gain is increased from 400 all accuracy curves decrease. This decrease prevails till a gain value of 1600 which also constitutes a global minimum in the accuracy-gain relationship. For a gain $>$ 1600 the accuracy curves begin to increase again and taper towards their respective initial test set benchmark. The extent to which the accuracy diminishes at this minima is inversely dependent on the strength of the model. It is noted that for gas gains less than 1600 the accuracy begins to rise and eventually exceeds that of the initial test accuracy indicating that the signals arising from these gain values are easily distinguishable and distinctly different from the point of view of our discriminators. In the case of the accuracy-gain relation plot, the expectation is that the gain value that results in the occurrence of a minimum is suggestive of the optimal gain value.

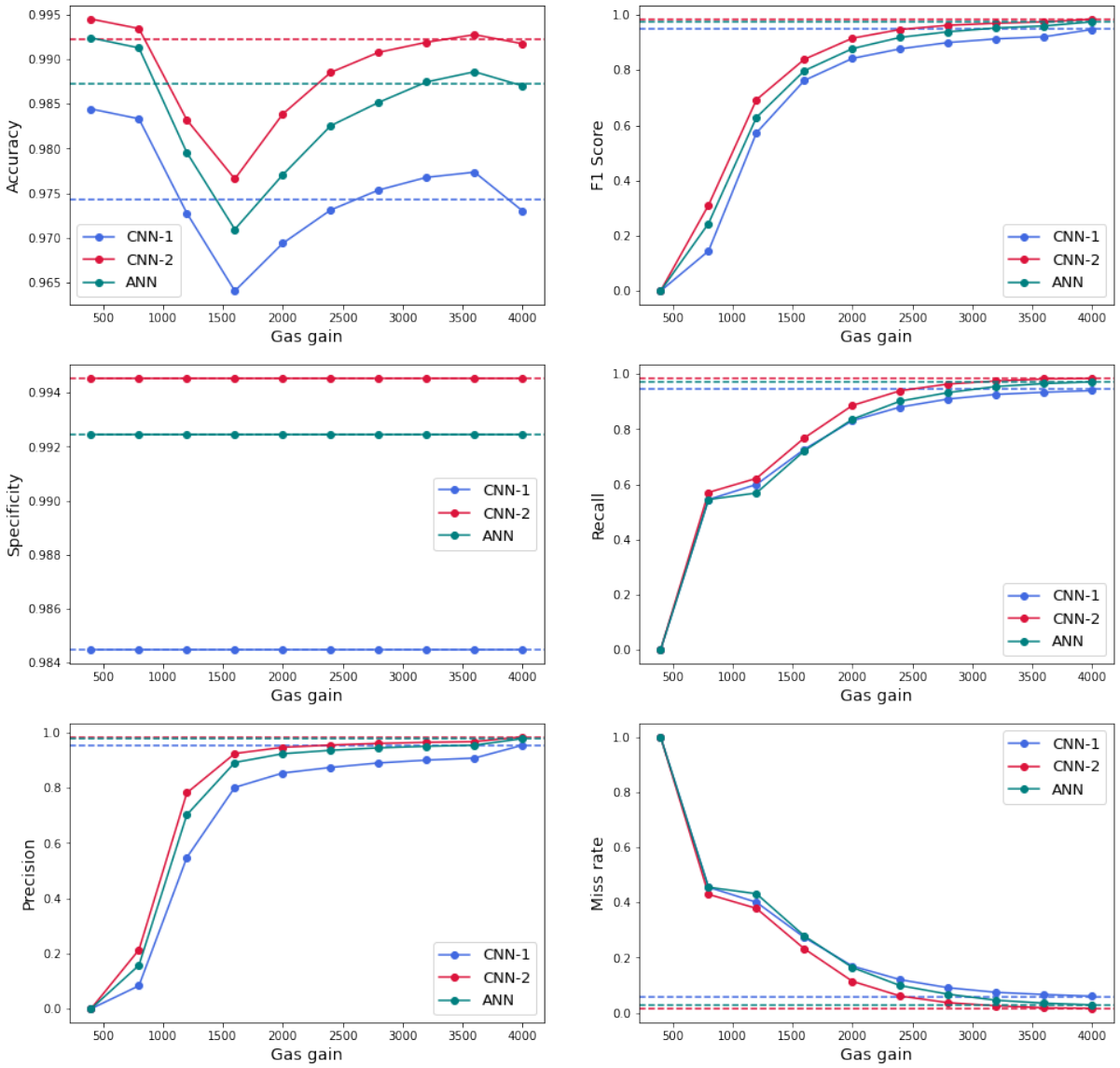


Figure 5.21: Performance metrics of our deep learning classifiers according to tuned gas gain value. The coloured dashed horizontal lines denote the respective initial test set performance of the models.

Out of all the performance metric and gain relations, the specificity of our discriminators demonstrates a constant relationship with the gain value. The specificity of all models remains unchanged throughout the entire range of our gain modification and maintains the initial test set specificity. Although a constant specificity is maintained for all models, the magnitude of the constant specificity characterizes the classification strength of our models. CNN-2 dominates followed by ANN and thereafter CNN-1. The fact that specificity is unaffected by the gain modification implies that our models are able to consistently discern the same fraction of real signals that are truly real.

The precision in this context denotes the fraction of all signals that are classified as simulated that are truly simulated. The precision-relation plot shows that the precision increases with an increase in the gain from 400 to 4000. The precision demonstrates a steep rise till a gain value of 1600 and thereafter gradually tapers to the initial test set precision value. The expectation would have been for the precision-gain relation to exhibit a minimum at a gain value of 2800. This is not the case and no such minima exists around a gain value of 1600 as suggested by the accuracy-gain relation plot. The F1 score relation follows the behaviour of the precision-gain relation.

The recall and miss rate (or false negative rate) are complementary, hence they are unique to the extent of a vertical flip. The recall in this context denotes the fraction of simulated signals that are actually identified as simulated by our discriminators. The miss rate hence represents the fraction of simulated signals that are incorrectly classified as real. The expectation with these curves is that the miss rate and recall should exhibit a maximum and minimum respectively at the gain value that results in the greatest conformation of the simulated signals to the real ones which would be suggestive of an optimal gain value. Ideally the suggestive optimal gain would be 2800 which would correspond to that observed in Figure 5.19. No distinct optimal gain value however, is suggested by the recall and miss rate curves. As the gain approaches the default value the curves approach their initial test set benchmarks.

Disagreement amongst deep learning and pulse height suggestions

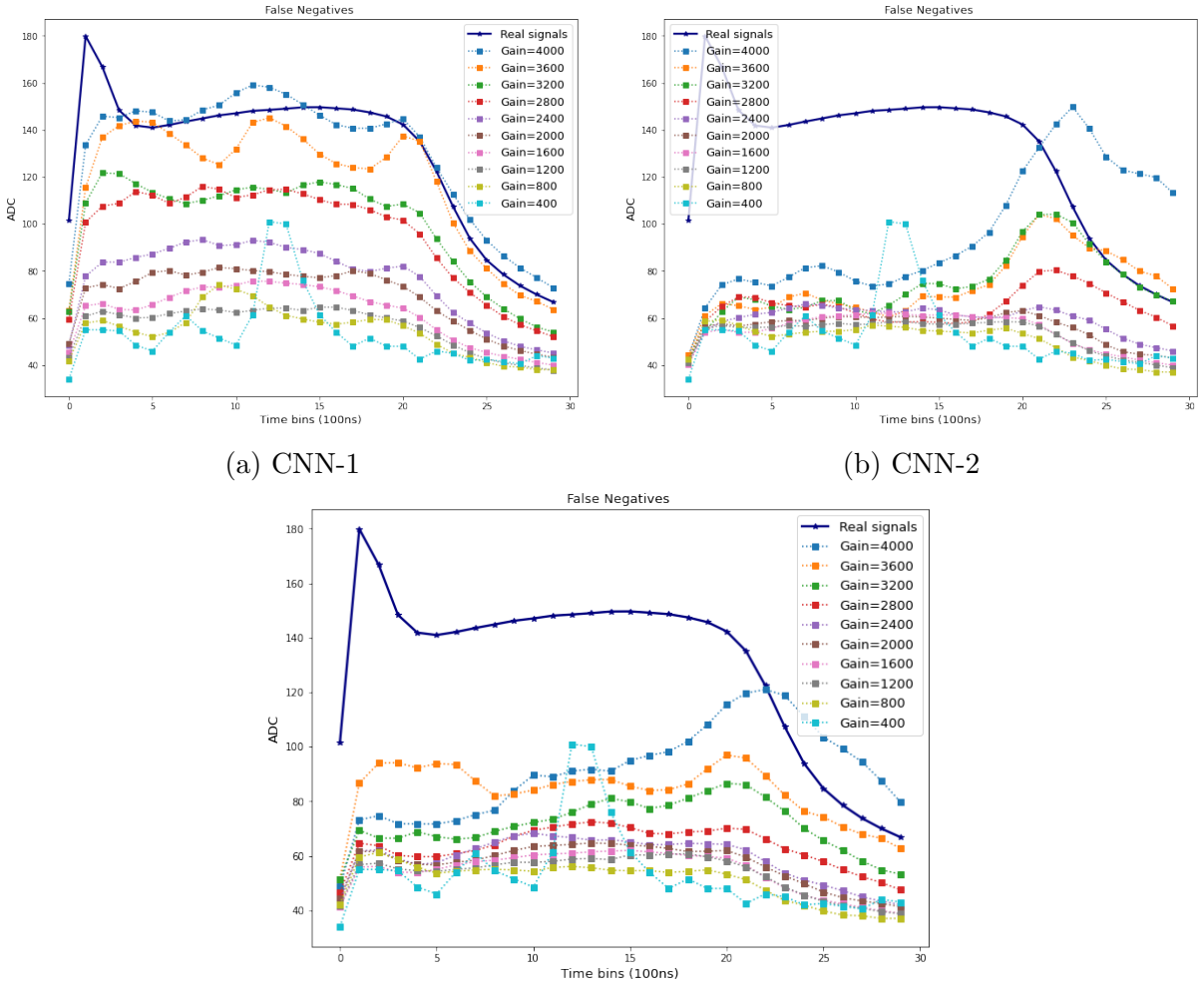
Pulse height inspection suggests that a gain value of 2800 is optimal since it results in the closest conformation of the simulated pulse height curve to that of real signals. The inspection of the deep learning performance metrics are not clearly suggestive of an optimal gain value. The metric that is most attentive to our gain tuning is the accuracy which dips at a gain value of 1600. Even if this minimum is considered the optimal value suggested by our deep learning models, it is still in disagreement with the observations of the pulse height fluctuations per gain. The observed disagreement could possibly be a consequence of the retention of supermodules 15, 16 and 17. It is possible that the presence of these problematic supermodules could have interfered with the overall effect our tuning procedure had on modifying simulated data to resemble real data.

5.4.3 Misclassified signal behaviour

The pulse height spectrum is reconstructed for misclassified signals arising from each of our discriminators for each tuned gain value. This is performed in order to ascertain any possible effect that the modification in gas gain has on the overall properties of misclassified signals. This also assists in determination of which gain value allows the misclassified signals to best resemble that of real run data.

False negative tuned signals

We examine the pulse height reconstructions of the false negative signals that arise from prediction of each of our discriminators on each tuned gain data which is depicted in Figure 5.22.



(a) CNN-1

(b) CNN-2

(c) ANN

Figure 5.22: Pulse height spectrum of false negative signals per tuned gas gain value.

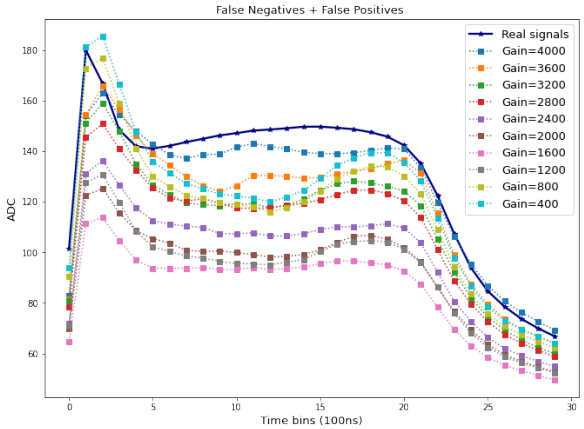
The false negatives signals resulting from CNN-1 demonstrate a better agreements with that of real data in comparison to the other discriminators. For our CNN-1 model the increase in gas gain from 400 till the default value of 4000 pulls the simulated pulse height of false negatives up to the expected curve. The agreement of false negative and real pulse height curves becomes the greatest for the default gas gain of 4000 according to CNN-1. The expected amplification peak remains deficient however.

The remaining model discriminators demonstrate poor agreement, with the simulated pulse heights of false negatives exhibiting a pronounced peak at later times that is observed to rise and exacerbate in response to an increase in the gas gain. Overall the reconstructions in Figure 5.22 serve to inform that the simulated signals that are believed to be "real" by our discriminators do not obey the macroscopic behaviour seen in the real pulse height even in the presence of gas gain tuning.

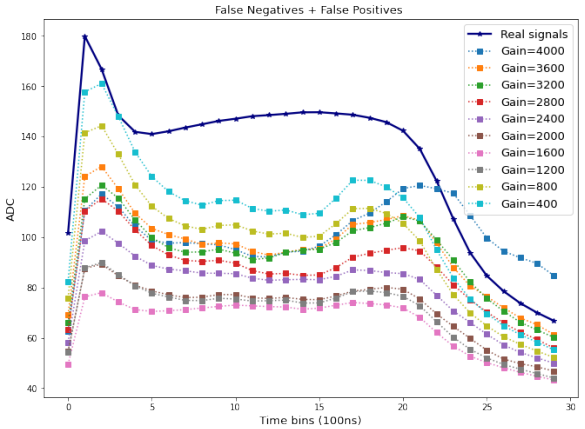
The expectation of the reconstructed pulse heights is that they would resemble the real pulse height behaviour for the optimal gain observed from pulse height inspection of the tuned gains. This is not the case for gas gains of 2800 which again highlights the disagreement amongst the observations of parameter optimality from our deep learning models and that of pulse height inspection. Even with the consideration of a gain of 1600 which promulgates a minimum for the accuracy-gain relation in Figure 5.21, no proper agreement is seen with the reconstructed pulse heights of false negative tuned signals.

Combined false negatives and false positive tuned signals

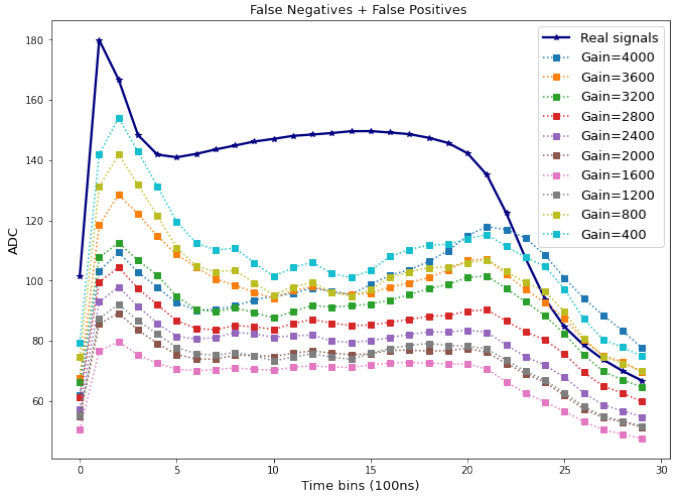
We now reconstruct the average pulse height of all misclassified signals for each modified gas gain, this is depicted in Figure 5.23. As previously observed with the intial test set pulse height reconstructions of all misclassifications, the addition of false positive signals improves the resemblance of simulated pulse height behaviour to that of real run data by the emergence of a peak at earlier time bins. In all reconstructions for all discriminators, a gas gain of 4000 is seen to exhibit the closest conformation to the real pulse height curve in comparison to other gains. The closest agreement amongst all discriminators is observed by CNN-1 for the default gain of 4000. This is inconsistent with the observations of pulse height inspection which proposes that a gain of 2800 should mitigate discrepancies amongst real and simulated data.



(a) CNN-1



(b) CNN-2



(c) ANN

Figure 5.23: Pulse height spectrum of combined false negative and false positive signals per tuned gas gain value.

Chapter 6

Conclusion

6.1 Comments on the use of deep learning discriminators as tuning mechanisms

It is possible that our deep learning models have learnt intricate features of our data which are not modified with the modification of our gas gain or does not adjust in the same way as our gas gain which may lead to clear distinctions between the signal classes irrespective of the optimal parameter suggested from pulse height inspection being trialed. It can also lead to the suggestion of the incorrect optimal tuned parameter value. The optimal gain value suggested from our deep learning discriminators is inferred from examination of the extrema exhibited by the performance metrics per tuned gain value and is seen to be inconsistent with the observations of pulse height spectra. Overall the deep learning discriminators are receptive to changes in the data as a consequence of parameter tuning, however not all performance metrics clearly depict this and in the correct expected manner. In order for this approach to be more reliable, more extensive effort is necessary to decipher what a deep learning model learns and to restrict such models to only learning significant details.

6.2 Limitations and future considerations

This study explored the tuning of a single parameter namely the Xe gas gain within the O^2 Monte Carlo simulation software, future studies could incorporate simultaneous tuning of multiple parameters and assess their simultaneous effect on simulation performance. A smaller increment of parameter changes should also be performed to obtain a more precise relation between simulation quality and the associated parameters which are modified. The machine learning approaches used within this thesis are of a supervised nature, an unsupervised approach could also be explored in optimizing the simulation mechanism. A

reinforcement learning strategy that rewards and punishes a model and hence exercises control in what is learnt could be also be explored.

Appendix A

Additional figures

A.1 Pulse Heights

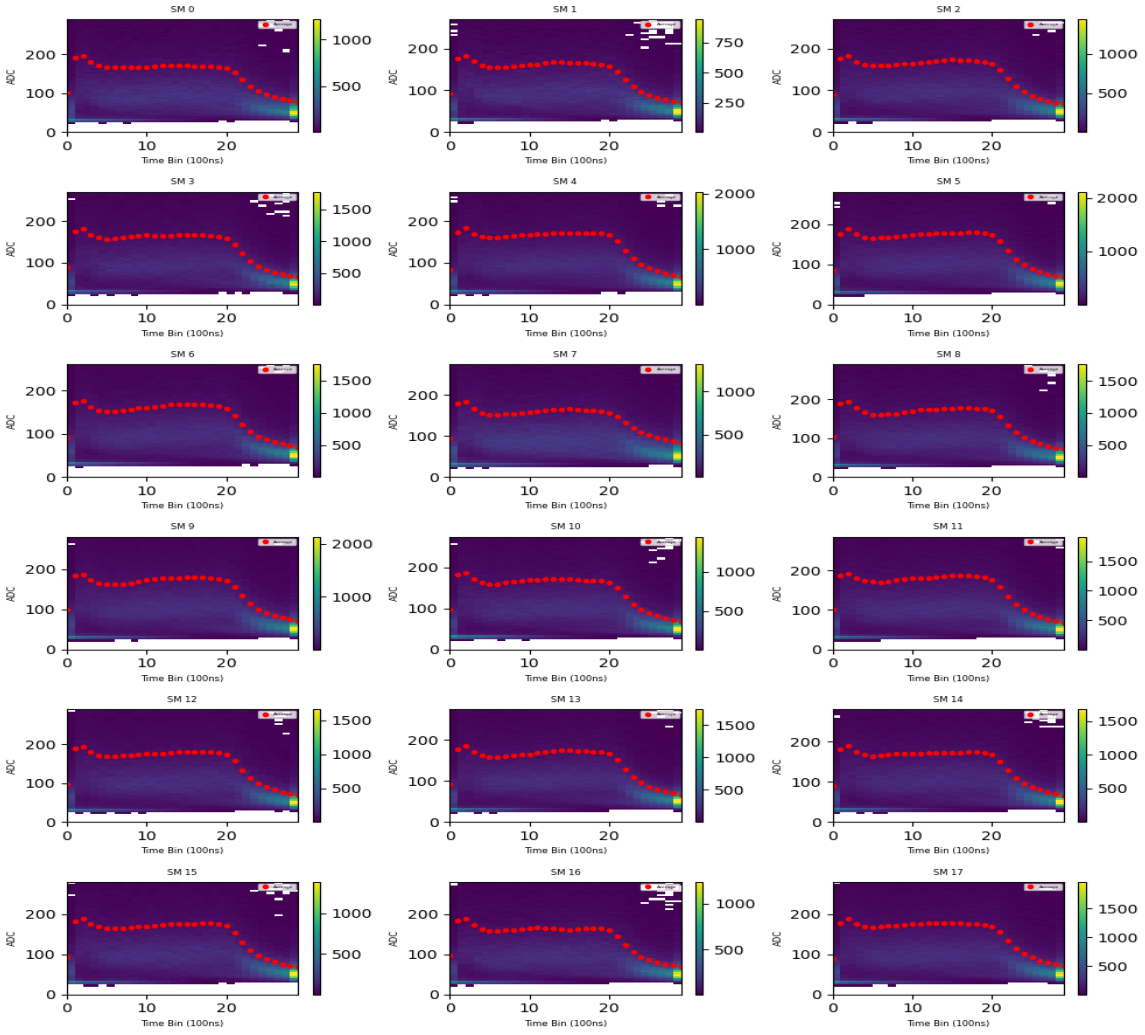


Figure A.1: Two dimensional pulse height spectra per supermodule for simulated data.

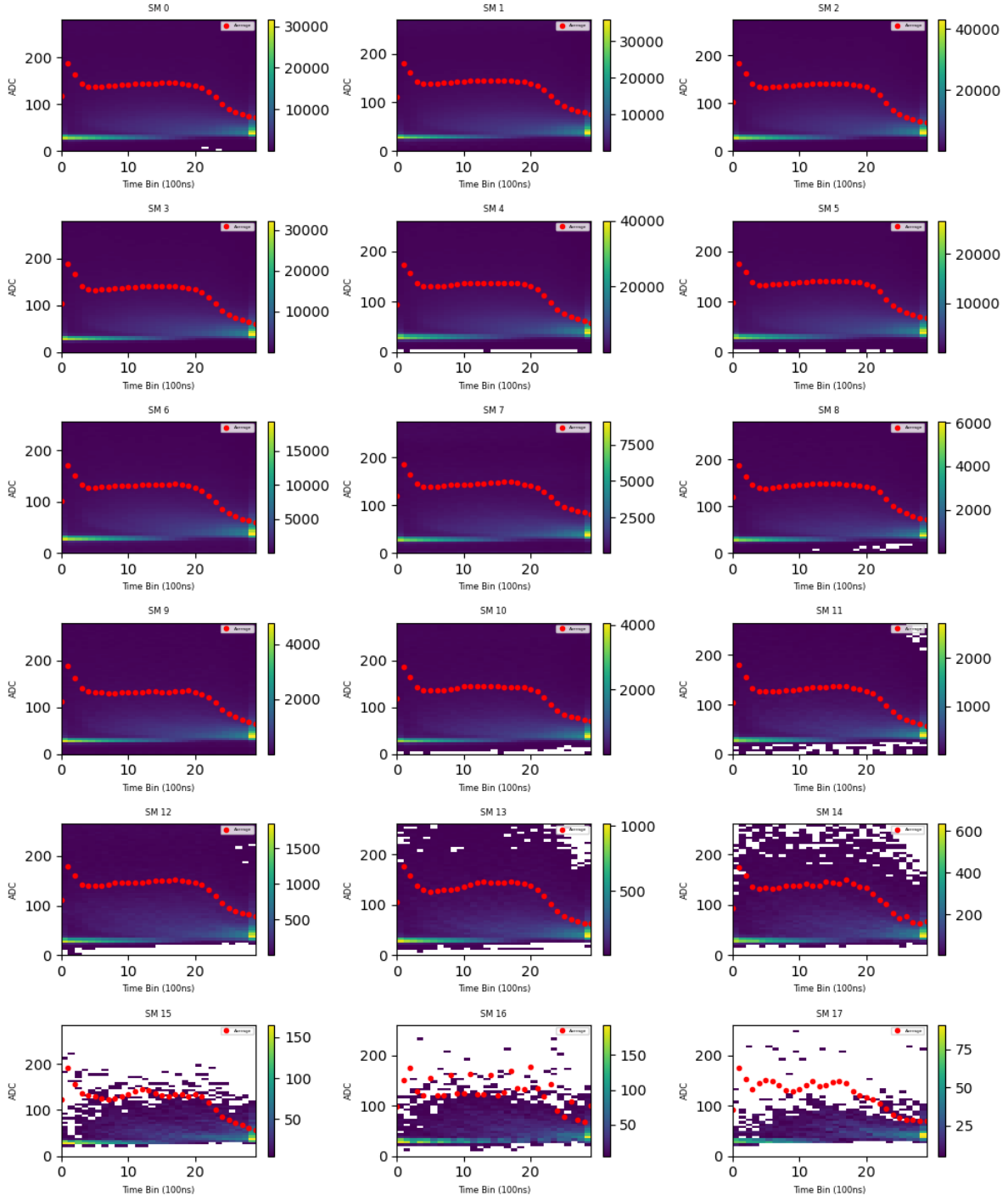
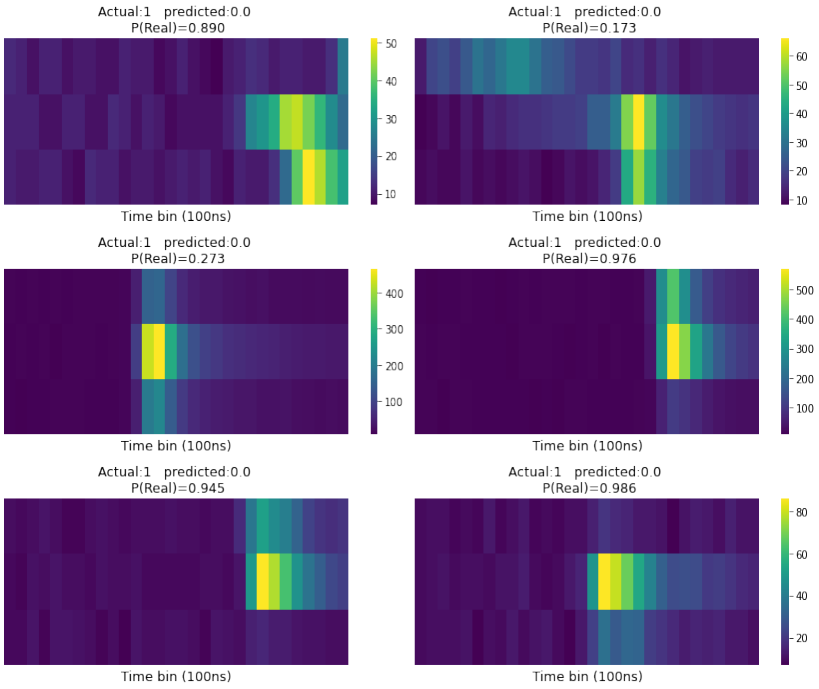
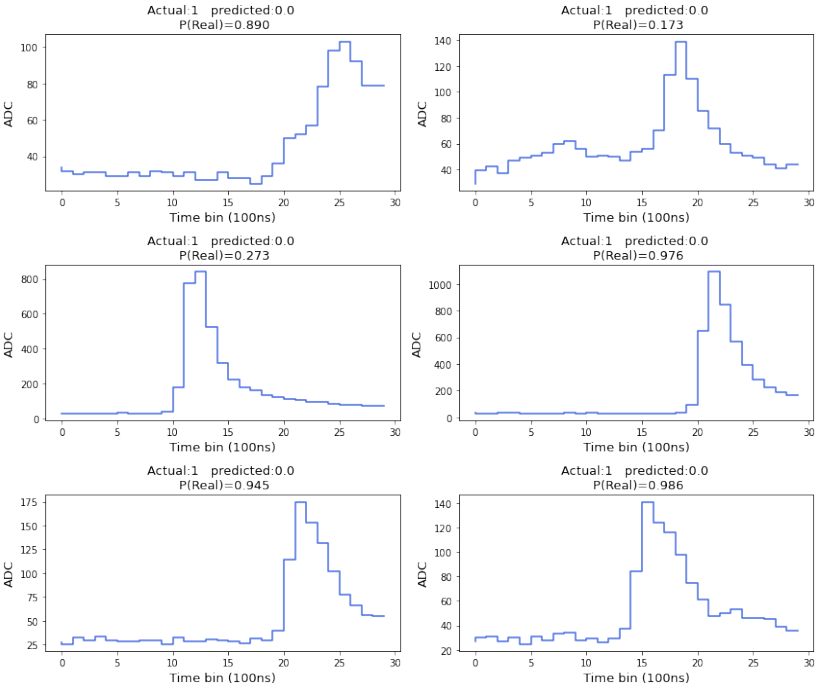


Figure A.2: Two dimensional pulse height spectra per supermodule for real data.

A.2 False negative signal samples

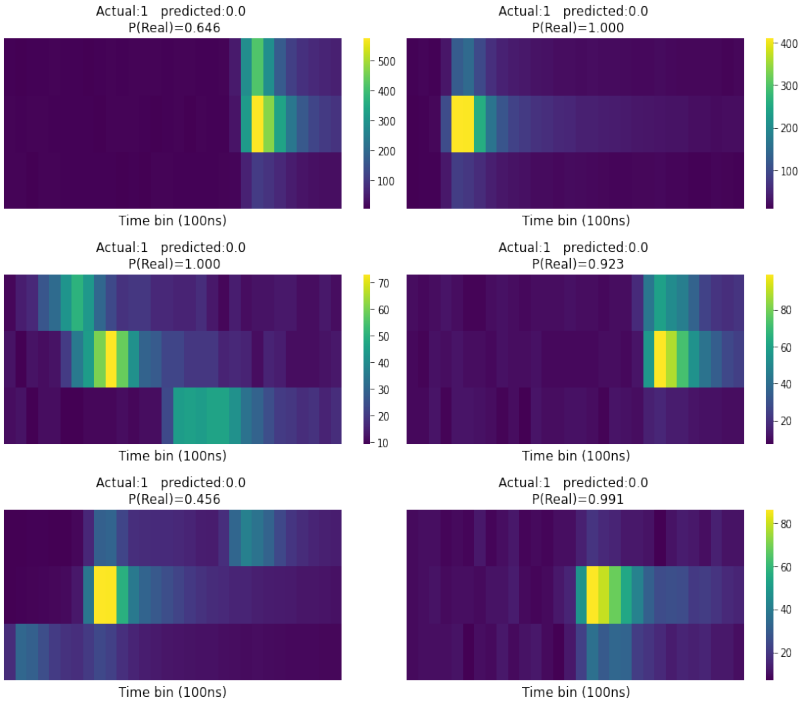


(a) False Negative Signals

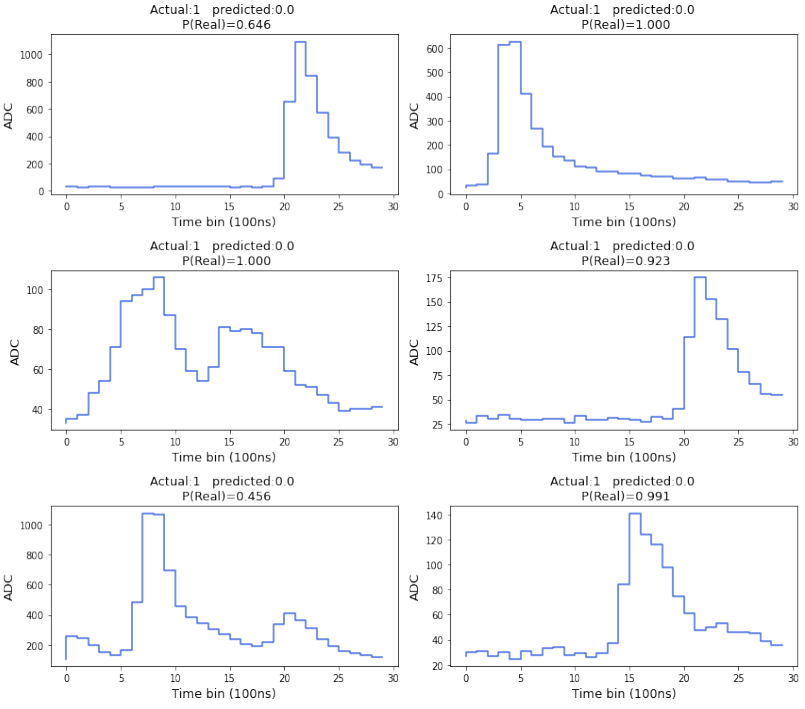


(b) Individual Pulse heights

Figure A.3: A sample of false negative signals from our CNN-2 model with corresponding individual pulse heights as step functions.



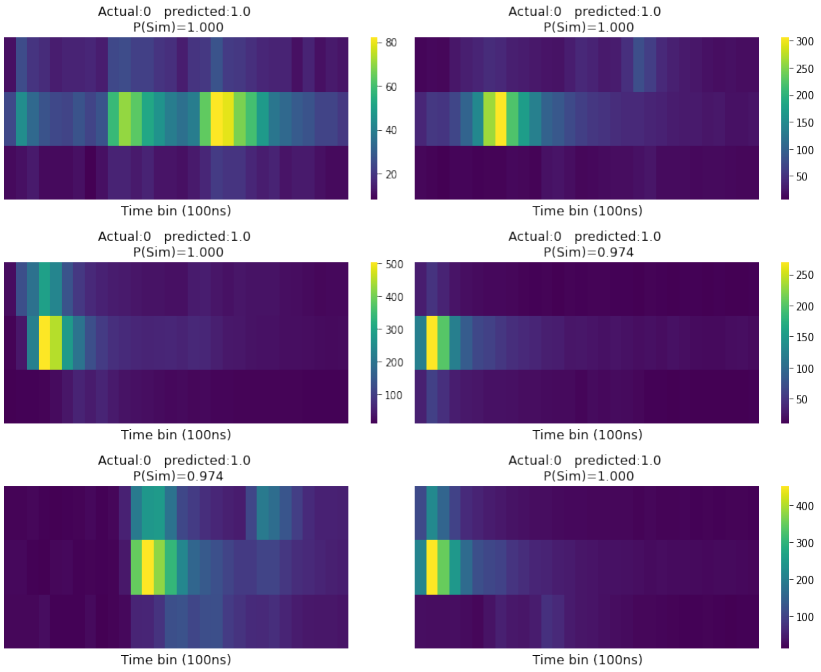
(a) False Negative Signals



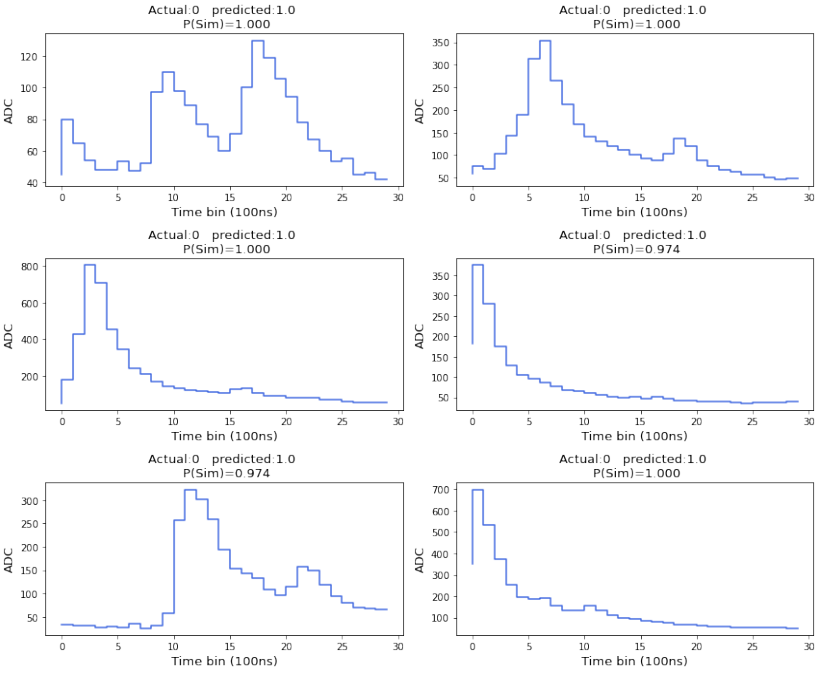
(b) Individual Pulse heights

Figure A.4: A sample of false negative signals from our ANN model with corresponding individual pulse heights as step functions.

A.3 False positive signal samples

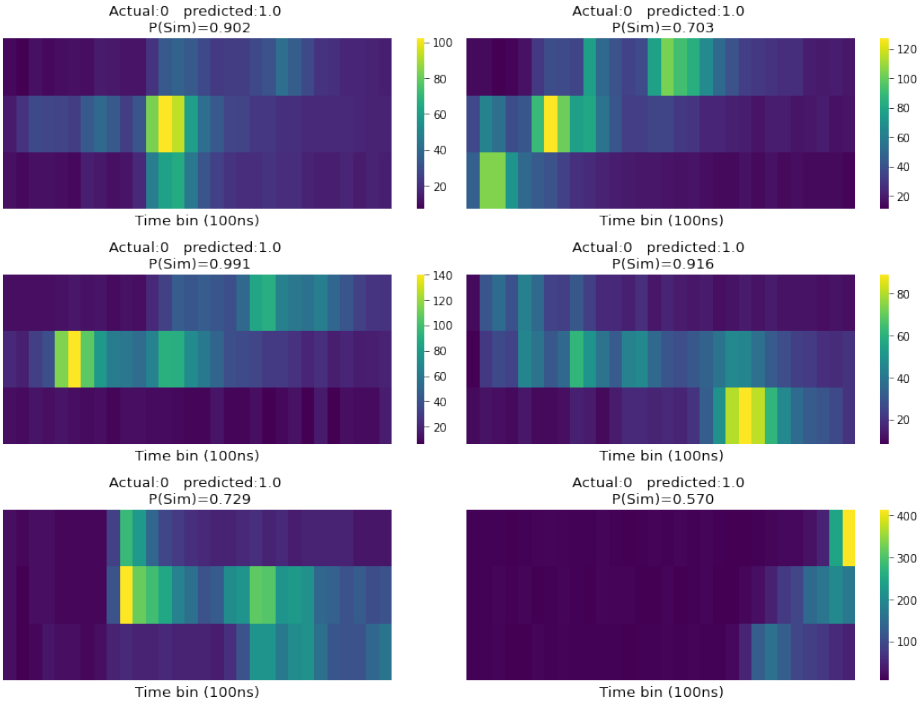


(a) False Positive Signals

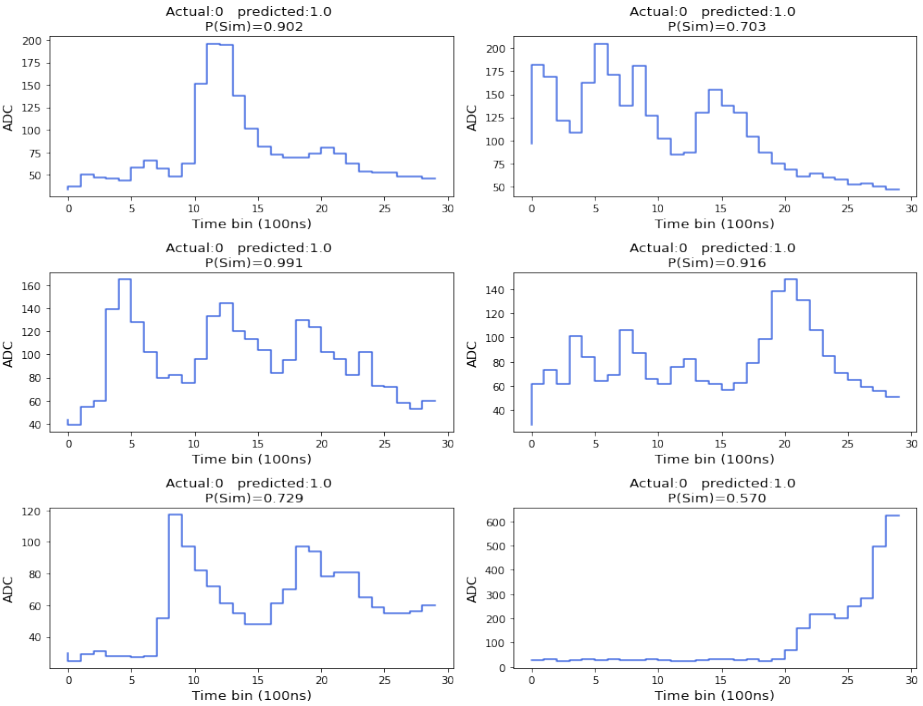


(b) Individual Pulse heights

Figure A.5: A sample of false positive signals from our CNN-2 model with corresponding individual pulse heights as step functions.



(a) False Positive Signals



(b) Individual Pulse heights

Figure A.6: A sample of false positive signals from our ANN model with corresponding individual pulse heights as step functions.

Appendix B

ALICE data acquisition

B.1 Real run 3 data

We detail the process of obtaining real run data and Monte Carlo simulation data. For the purposes of this study data from LHC22o run 526689 was utilized.

The files were transferred by secure copy protocol to the author's Cernbox. The path of the these files were first obtained and stored in a `.txt` file through the following command:

```
find `pwd` -name 'o2_ctf_*.root' > files.txt
```

A bash workflow script `convert.sh` was then created to incorporate the various workflows that the ctf files must be subjected to. The contents of this script are given below.

```
#!/bin/sh
eval $(alienv printenv VO_ALICE@O2::nightly-20220728-1)

o2-ctf-reader-workflow --onlyDet TRD --ctf-input files.txt \
| o2-trd-digittracklet-writer \
--run --batch
```

The execution of the above workflow produces a `trdtracklets.root` and `trddigits.root` file.

B.2 Monte Carlo simulation data

Here we detail the process of obtaining simulated data with the ALICE O^2 software.

The O^2 software was first activated by executing:

```
alienv enter O2Physics/latest-master-o2
```

Once in the O^2 environment, the actual simulation with the consideration of all parameters stated in Table 4.1 was done by execution of the following:

```
o2-sim -m PIPE MAG TRD -n 50000 -g boxgen \
--configKeyValues 'BoxGun.pdg=211;BoxGun.eta[0]=-0.84;BoxGun.eta[1]=0.84'
```

The O^2 simulation produces numerous `.root` files each containing associated simulation results. Of particular interest is the `o2sim_HitsTRD.root` file which is subjected to a digitizer workflow in order to obtain the necessary TRD digit and tracklet information pertaining to the simulation. The digitization of the detector hits was performed by executing:

```
o2-sim-digitizer-workflow -b
```

which then produces the `trddigits.root` and `trdtracklets.root` files.

Bibliography

[1] "The alice transition radiation detector: Construction, operation, and performance," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 881, pp. 88–127, 2018.

[2] R. Pasechnik and M. Šumbera, "Phenomenological review on quark–gluon plasma: Concepts vs. observations," *Universe*, vol. 3, no. 1, 2017.

[3] H. Wiedemann, *Theory of Synchrotron Radiation*, pp. 857–894. Cham: Springer International Publishing, 2015.

[4] V. L. Ginzburg and I. M. Frank, "Radiation of a uniformly moving electron due to its transition from one medium into another," *J. Phys. (USSR)*, vol. 9, pp. 353–362, 1945.

[5] G. Garibian, "Contribution to the theory of transition radiation," *Sov. Phys. JETP*, vol. 6, no. 6, p. 1079, 1958.

[6] B. Dolgoshein, "Transition radiation detectors," *Nucl. Instrum. Meth. A*, vol. 326, pp. 434–469, 1993.

[7] T. A. Collaboration and K. Aamodt, "The alice experiment at the cern lhc," *Journal of Instrumentation*, vol. 3, p. S08002, aug 2008.

[8] P. Cortese, "ALICE transition-radiation detector: Technical Design Report," 2001.

[9] E. S. Wulff, *Position Resolution and Zero Suppression of the ALICE TRD*. PhD thesis, Diplomarbeit, Westfälische Wilhelms-Universität Münster, 2009.

[10] H. Klingenmeyer, "Tracklet-based particle identification with the alice trd for lhc run 3," Master's thesis, 2017.

[11] Elvira, V. Daniel, "Impact of detector simulation in particle physics collider experiments - highlights," *EPJ Web Conf.*, vol. 214, p. 02019, 2019.

[12] V. Daniel Elvira, "Impact of detector simulation in particle physics collider experiments," *Physics Reports*, vol. 695, pp. 1–54, 2017. Impact of Detector Simulation in Particle Physics Collider Experiments.

[13] M. Bernardini and K. Foraz, "Long Shutdown 2 @ LHC," 2015.

[14] Eulisse, Giulio, Konopka, Piotr, Krzewicki, Mikolaj, Richter, Matthias, Rohr, David, and Wenzel, Sandro, "Evolution of the alice software framework for run 3," *EPJ Web Conf.*, vol. 214, p. 05010, 2019.

[15] P. Buncic, M. Krzewicki, and P. Vande Vyvre, "Technical Design Report for the Upgrade of the Online-Offline Computing System," tech. rep., 2015.

[16] A. V. Joshi, *Introduction to AI and ML*, pp. 3–7. Cham: Springer International Publishing, 2020.

[17] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, pp. 433–460, 10 1950.

[18] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[19] T. M. Mitchell and T. M. Mitchell, *Machine learning*, vol. 1. McGraw-hill New York, 1997.

[20] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[22] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, pp. 1–21, 03 2015.

[23] C. C. Aggarwal, *An Introduction to Neural Networks*, pp. 1–52. Cham: Springer International Publishing, 2018.

[24] U. Michelucci, *Feedforward Neural Networks*, pp. 83–136. Berkeley, CA: Apress, 2018.

[25] R. Yamashita, M. Nishio, and R. Do, "Gian, and k. togashi,"," *Convolutional neural networks: An overview and application in radiology," Insights Imag*, vol. 9, no. 4, pp. 611–629, 2018.

[26] P. Li, R. Jing, and X. Shi, "Apple disease recognition based on convolutional neural networks with modified softmax," *Frontiers in Plant Science*, vol. 13, 2022.

[27] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[28] J. Moolayil, *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. 01 2019.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] "Monalisa repository for alice." https://alimonitor.cern.ch/. Accessed 10 June 2022.

[31] G. Yost, R. Barnett, I. Hinchliffe, G. Lynch, A. Rittenberg, R. Ross, M. Suzuki, T. Trippe, C. Wohl, B. Armstrong, G. Wagman, F. Porter, L. Montanet, M. Aguilar-Benitez, J. Hernandez, G. Conforto, R. Crawford, K. Schubert, M. Roos, N. Törnqvist, G. Höhler, K. Hagiwara, S. Kawabata, D. Manley, K. Olive, K. Hayes, R. Schindler, B. Cabrera, R. Shrock, R. Eichler, L. Roper, and W. Trower, "Review of particle properties," *Physics Letters B*, vol. 204, p. 1, 1988.

[32] R. L. Workman and Others, "Review of Particle Physics," *PTEP*, vol. 2022, p. 083C01, 2022.

[33] V. Khachatryan, A. M. Sirunyan, A. Tumasyan, W. Adam, T. Bergauer, M. Dragicevic, J. Erö, C. Fabjan, M. Friedl, R. Frühwirth, *et al.*, "Transverse-momentum and pseudorapidity distributions of charged hadrons in p p collisions at s= 7 tev," *Physical Review Letters*, vol. 105, no. 2, p. 022002, 2010.

[34] Ramraj.N, "Source code." https://github.com/NikR108/MSc-ALICE.git.