Bachelor Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Axion-Like-Particle Search Using Machine Learning for the Signal Sensitivity Optimization with Run-2 LHC Data Recorded by the ATLAS Experiment

Ondřej Matoušek

Supervisor: Doc. Dr. André Sopczak
May 2023

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Matoušek Ondřej**	Personal ID number:	**498865**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Axion-Like-Particle Search Using Machine Learning for the Signal Sensitivity Optimization with Run-2 LHC Data Recorded by the ATLAS Experiment

Bachelor’s thesis title in Czech:

Pátrání po axionu podobných částicích s využitím strojového učení pro optimalizaci citlivosti k signálu s daty experimentu ATLAS z LHC Run-2

Guidelines:

The neutral Standard Model Higgs boson was discovered in 2012 at CERN, and the search for further particles of extended models continues. In particular, the search for an Axion-Like-Particle (ALP). An ALP can be produced with a signature of two photons. Using machine learning technology, this analysis addresses the separation of ALP production from unwanted background reactions. There are three analysis levels: generator level, full ATLAS detector simulation, and real recorded data. In this project, the data from the full ATLAS detector simulation shall be used and the performance of the machine learning algorithms be optimized in the search for the ALPs, separating.
Instructions:
1) Study the provided root framework and data storage in ntuple format.
2) Study the provided object (photons, electrons, muons, taus, jets) information in the ntuples.
3) Get familiar with the features for the machine learning application.
4) Design and implement a classifier to separate signal and background events based on simulated data.
5) Optimize the separation of signal and background.
6) Order the features according to the separation power.
7) Determine the significance of separating signal and background as a function of the ALP mass.
8) Compare the obtained significance expectation with previous results.

Bibliography / sources:

[1] https://home.cern
[2] https://atlas.cern
[3] Heather Gray, Bruno Mansoulié, The Higgs boson: the hunt, the discovery, the study and some future perspectives, https://atlas.cern/updates/feature/higgs-boson
[4] Guest et al., Deep Learning and Its Application to LHC Physics
article in Annu. Rev. Nucl. Part. Sci. 2018. 68:1–22, https://arxiv.org/pdf/1806.11484.pdf
[5] M. Andrews et al., End-to-End Event Classification of High-Energy Physics Data
http://www.sergeigleyzer.com/wp-content/uploads/2017/12/end-end-event.pdf
[6] Hussain Kitawaga, Optimization of diphoton acoplanarity for an Axion-Like Particle in Light-by-Light scattering with the ATLAS detector at CERN, CERN-STUDENTS-Note-2020-029, https://cds.cern.ch/record/2742416

Name and workplace of bachelor’s thesis supervisor:

doc. Dr. André Sopczak High Energy Physics, IEAP CTU in Prague

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **17.10.2022** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

doc. Dr. André Sopczak Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Acknowledgements

I am extremely grateful to my supervisor, Doc. Dr. André Sopczak, for his invaluable mentorship and unwavering support throughout this project. Thanks also to my cat, for never-ending support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

in Prague, 21. May 2023

Abstract

The neutral Standard Model Higgs boson was discovered in 2012 at CERN, and the search for further particles of extended models continues. In particular, the search for an Axion-Like-Particle (ALP). Using machine learning technologies, this analysis addresses the separation of ALP production from unwanted background reactions. In this project, the Run-2 data from the ATLAS detector are used and the efficiency as well as the significance of the machine learning algorithm is optimized as a function of theoretical ALP mass.

Keywords: machine learning, binary classification,Axion-Like-Particles, neural networks, CERN, ATLAS,

Supervisor: Doc. Dr. André Sopczak

Abstrakt

V roce 2012 byl v CERNu objeven neutrální Higgsův boson standardního modelu a v současnosti pokračuje pátrání po dalších časticích rozšířeného modelu. Konkrétně se hledá částice podobná axionu (Axion-Like-Particle, ALP). Tato analýza se s využitím technologií strojového učení zabývá oddělením produkce ALP od nežádoucího šumu pozadí. V tomto projektu jsou použita data Run-2 z detektoru ATLAS a je optimalizována účinnost a významnost algoritmu strojového učení v závislosti na teoretické hmotnosti ALP.

Klíčová slova: strojové učení, binární klasifikace, Axionu-podobné částice, neuronové sítě, CERN, ATLAS

Překlad názvu: Pátrání po axionu podobných částicích s využitím strojového učení pro optimalizaci citlivosti k signálu s daty experimentu ATLAS z LHC Run-2

Contents

Figures

Tables

Chapter 1

Introduction

Axion-Like-Particles (ALP) are yet unobserved particles which should be able to explain inconsistencies in the theory of quantum chromodynamics. They should be produced by proton-proton collisions with the signature of diphoton production. The process during which the ALPs could be observed is known as light-by-light scattering $\gamma\gamma \rightarrow \gamma\gamma$, in which an ALP decays into two photons. Dissociation of protons might accompany the collision. It might accompany both protons entering the collision, only one of them, or none. These situations are measured during the collision of two proton beams in the Large Hadron Collider, which is accompanied by the emission of two photons, which are then detected by the ATLAS detector as $pp \rightarrow pp\gamma\gamma$ [1] (further details in [2]). Data from the ATLAS Forward Proton detectors are used, so that it is possible to get more information about the momentum and angles of the emergning protons. Using binary classification machine learning algorithms, state-of-the-art automatic selection of significant photons will be optimized, so that they can be distinguished from background noise.

Chapter 2

Topic introduction

2.1 Standard Model

The Standard Model of particle physics (Figure 2.1) is a generally accepted theory describing the electromagnetic, weak and strong interactions. The theory does not include a description of gravity nor relations between gravity and the other three fundamental forces. The Standard Model also classifies elementary particles and antiparticles, which are divided into several groups. The basic division is in terms of the spin of a particle. Particles with integer spin are called bosons, whereas particles with half-integer spins are called fermions. The fermions are then divided into quarks (antiquarks) and leptons (antileptons), where the quarks participate in strong interaction and leptons participate in the weak interaction. The bosons are also called force carriers, representing corresponding interactions. The bosons are then divided into gauge bosons (vector represented), in which we find gluons, photons, and W, as well as Z bosons. These four bosons have all spin 1. The last elementary boson is the Higgs boson, which has zero spin [3].

2.2 CERN - LHC

The European Organization for Nuclear Research, better known as CERN (Conseil Européen pour la Recherche Nucléaire), is an international organi-

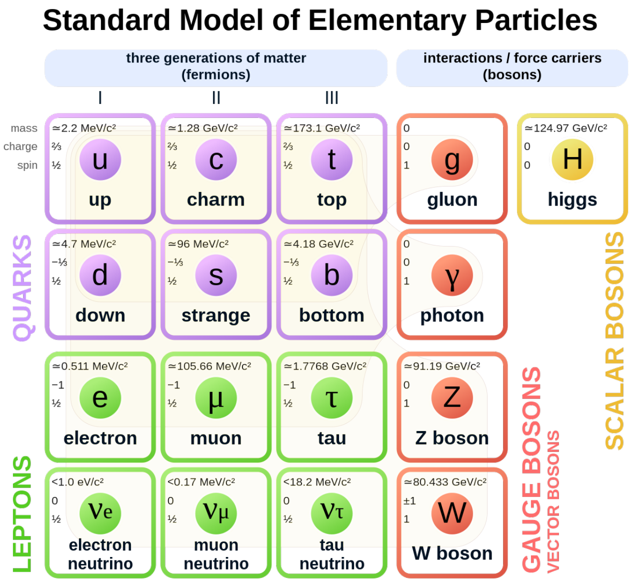

Figure 2.1: Standard Model table [4].

zation with the largest particle physics laboratory and the biggest particle accelerator (Large Hadron Collider) in the world [5]. The highest significance of CERN for this thesis is the mentioned LHC, providing data for high-energy physics experiments for several hundreds of universities and laboratories. The first experiments with collisions in the LHC began in 2008, which was followed by the discovery of Higgs Boson in 2012 [6].

2.3 ATLAS

ATLAS (Figure 2.2) is one of the seven particle detectors in the LHC and was one of the two detectors which successfully found Higgs Boson in 2012. ATLAS is used for measuring different properties of elementary particles. Examples of these measurements are momenta, energies, masses, charges, and spins of individual particles. The interaction point of two proton beams in the LHC is surrounded by four different sub-detectors that are dedicated to observing different features of particles, a large magnet system and a Forward Proton detector [7]. These systems are:

- Inner detector
- Liquid argon calorimeter
- Hadronic calorimeter
- Muon spectrometer

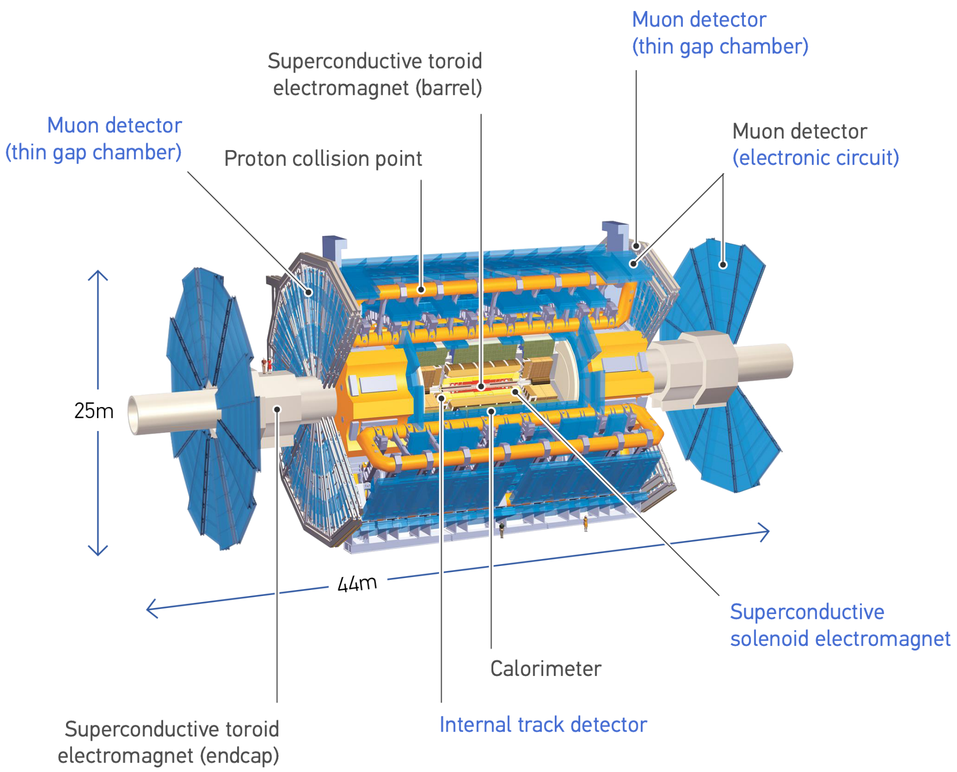

Figure 2.2: ATLAS detector description [8].

2.3.1 Inner detector

The main purpose of the inner detector is to measure momenta of charged particles near the interaction point and to unveil information about the types of these particles. The magnetic field in the inner detector makes particles curve as they travel. The measured curvature of the particle track determines the charge and momentum of a particle, and the starting point of the curve is used to determine the type of a particle [7].

2.3.2 Calorimeters

In the ATLAS detector, there are two main Calorimeter systems - electromagnetic and hadronic. Both systems work as sampling calorimeters, i.e the energy of a particle is absorbed in a high-density metal, from which the original particle energy is calculated [7].

Liquid argon calorimeter

It is designed to measure the energy of electrons, photons, and hadrons. It consists of multiple layers of either tungsten, copper, or lead with a unique honeycomb structure. When particles interact with the metal layers, they are converted into lower energy particle showers. The honeycomb structure is filled with liquid argon, which is ionized by low-energy particles. This produces an electric current that is measured [7].

Hadronic calorimeter

The second Calorimeter surrounds the Liquid argon Calorimeter and measures the energy of hadronic particles, which do not deposit their whole energy in the first Calorimeter. It consists of two parts. First, there are steel layers, which when hit by particles, create a shower of new particles. After this happens, plastic scintillators surrounding the steel structure produce photons from these particles, which are converted into electric current, from which we can directly measure the original energy of a particle [7].

2.3.3 Muon spectrometer

It is the largest one of the four sub-detectors, surrounding the initial measuring devices. It has approximately one million readout channels and if infividual detectors were placed side by side, the total area would be 12 000 m^2. The size is critical for precise measurements of the momentum of the muons. Its radius at its furthest part from the proton beams is 11m [7].

2.3.4 Magnet system

The magnet system is responsible for the curvature of the trajectory of charged particles, which allows us to measure their properties (momentum, charge, spin...) [7]. The curvature is caused by Lorentz force [9], which is described as $\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$, where $\vec{F}$ is the force acting on charged particles, $\vec{E}$ is the strength of the electric field in which the particles move, q is the electric charge of the particles, $\vec{v}$ is the speed of the particles and $\vec{B}$ is the magnetic field in which the particles move , as shown in Fig. 2.3.

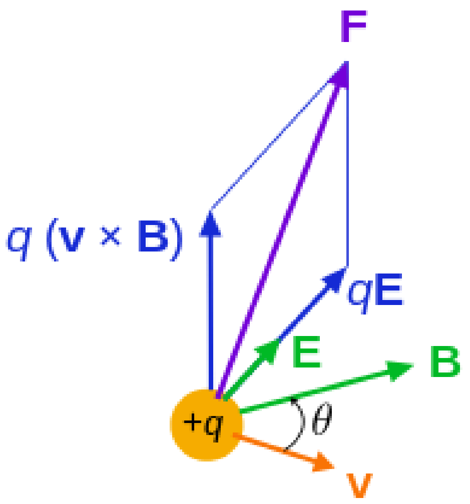

Figure 2.3: Lorentz force vectorised depiction, where θ is the angle between $\vec{B}$ and $\vec{v}$ [9].

2.3.5 AFP

A complementary detector to ATLAS is the ATLAS Forward Proton (AFP) detector [10], which allows us to measure the momentum and emission angle of forward protons. It is shown in Fig. 2.4.

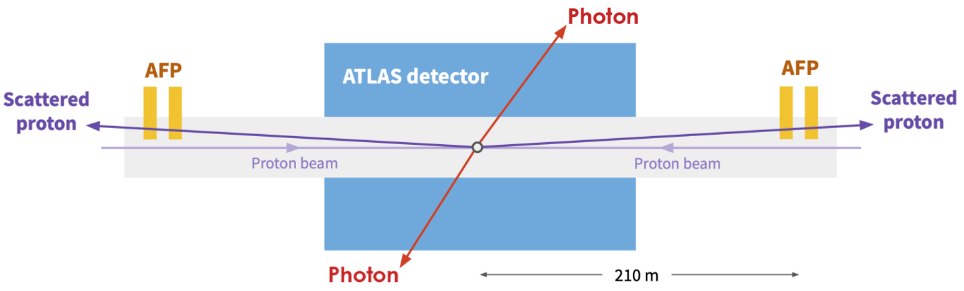

Figure 2.4: ATLAS main detector and AFP detectors.

2.4 Quantum chromodynamics

Quantum chromodynamics is the physical theory of strong interaction between quarks and gluons. The name of this discipline comes from the attribute that is given to quarks - their colour. In analogy to the theory of quantum electrodynamics, we can give to the colour of a quark the same significance as we give to the electric charge in quantum electrodynamics [11].

2.4.1 CP-violation

The CP-symmetry is a combination of C-symmetry and P-symmetry (Figure 2.5). Charge conjugation symmetry (or C-symmetry) describes the symmetry of physical laws during charge conjugation. Charge conjugation is a transformation, which describes the commutation of particles and corresponding antiparticles. This means that it changes the sign of all quantum charges. P-symmetry (parity-symmetry) describes the mirroring or inversion of spatial coordinates of particles. This says that equations describing particles should be invariant to mirroring transformations. The CP-violation is a term describing the violation of conservation laws, which states that the weak force does not obey the CP-symmetry. This means, that when describing a process during which the particles and their corresponding antiparticles are interchanged, this process is not physically equivalent to a mirror image of the same process without the charge conjugation (the exchange of particles and antiparticles). In quantum chromodynamics, we talk about the lack of CP-violation. Mathematically there is no reason why strong force interactions should lack CP-violation. However, no data have been yet seen confirming CP-violation in this field of study [12].

2.5 Axion-like-Particles

Axions are hypothetical particles whose existence was proposed to explain the lack of CP-violation in quantum chromodynamics. The originally proposed axions were very light. Axion-Like-Particles are hypothetical particles that resemble axions but are much more massive. They could be produced by a two-photon collision and then decay again into two photons. Detectors will meassure the interaction described as $\gamma\gamma \rightarrow ALP \rightarrow \gamma\gamma$ indirectly as $pp \rightarrow pp\gamma\gamma$. The interaction $\gamma\gamma \rightarrow \gamma\gamma$ is known as light-by-light scattering. During this interaction, there are 3 different situations in which the collision of two protons can result. They are all shown as Feynman diagrams in Fig. 2.6. In the single and double dissociative processes, one or two of the protons will dissociate. In the exclusive process, the protons stay intact [1].

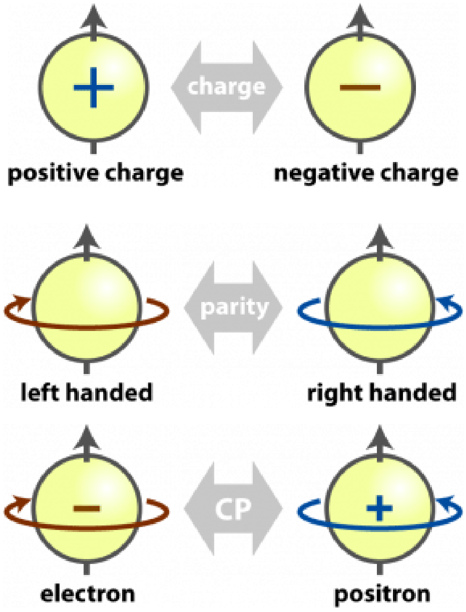

Figure 2.5: C,P and CP symmetries [13].

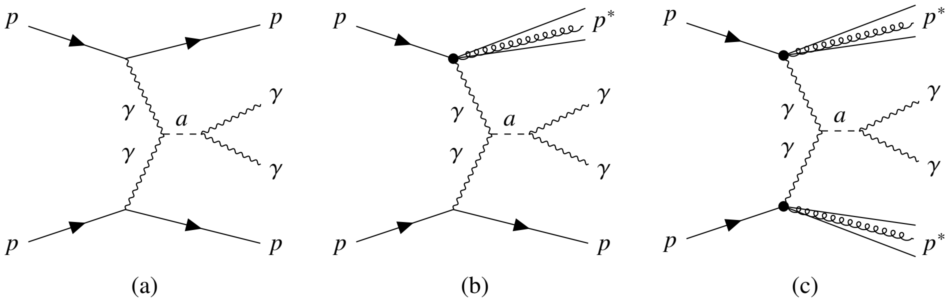

Figure 2.6: Feynman diagrams of exclusive, single dissociative and double dissociative processes [1].

2.6 State of the art

Currently, the format of data used in the analysis is ntuples. It is a tabular form of data, where each significant event recorded consists of rows of data with a fixed length [14]. The selection process consists of two parts. First is the preselection, where the amount of data to process is narrowed down to a smaller amount with a series of cuts. Then comes the main selection process, where the signal is distinguished from the background noise. This whole process is currently operating without machine learning methods. The preselection consists of several cuts. These cuts are to be replaced as an optimization technique by a machine learning algorithm.

2.6.1 Preselection

The preselection requires at least two photon candidates to take part in the interaction. For these photons, there are several equations that have to be satisfied in order for an event to pass this threshold. First, we have to cut out every event for which the invariant mass of both photons is not in the interval of [150,1600] GeV. This range is used, because it directly correlates to the high-efficiency range of the AFP detectors. It would be pointless to make the interval wider, because not many events outside of current range could be observed. Second is the equation limiting the pseudorapidity. The pseudorapidity is used to describe the angle of a particle with respect to the particle beam. If $\eta = 0$, the particle is traveling in the direction perpendicular to the beam. If the particle is close to the beam, η will be large [15]. The equation defining the η cut states that

$$|\eta| < 2.37. \tag{2.1}$$

Another equation describes the transversal momentum of a charged particle. The cut limits the momentum p_T in the following manner:

$$p_T > 40 \text{ GeV}. \tag{2.2}$$

The fifth selection criterion is the acoplanarity cut defined as

$$A_\phi^{\gamma\gamma} := 1 - \frac{|\Delta\phi_{\gamma\gamma}|}{\pi} < 0.01. \tag{2.3}$$

This is a property of two photons as denoted by the index $\gamma\gamma$. Here $\Delta\phi_{\gamma\gamma}$ is the difference of azimuthal angle between two photons, which ranges $(-\pi, \pi]$. This condition requires the two photons to be back to back to each other or at least very close to it. The acoplanarity distribution of the data before the acoplanarity cut is shown in Fig. 2.7.

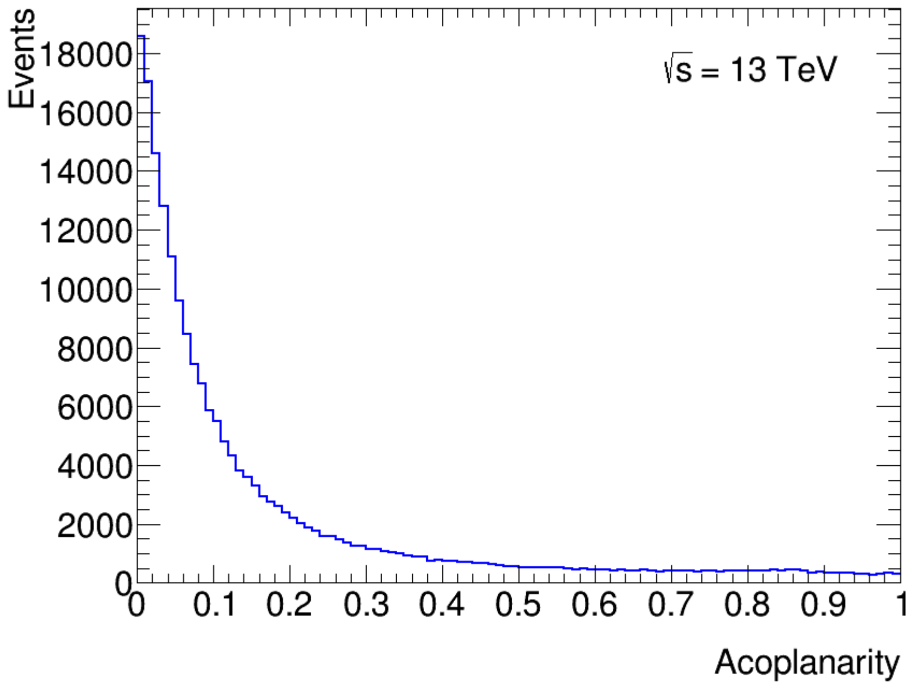

Figure 2.7: Diphoton acoplanarity distribution of the Run-2 data before the acoplanarity cut.

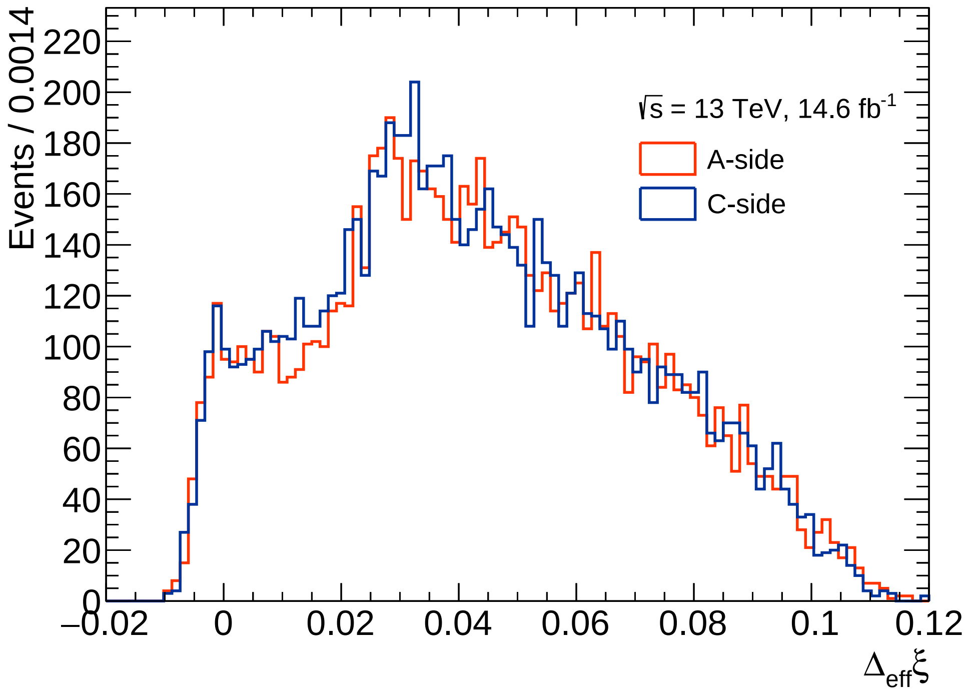

Figure 2.8: $\Delta_{\text{eff}}\xi^{\pm}$ distributions of the Run-2 data after the acoplanarity cut [16].

2.6.2 Main selection

The main event selection uses another cut to achieve the result and to choose a suitable diphoton candidate. The cut is the $\xi^{\pm}_{\gamma\gamma}$ cut. $\xi^{\pm}_{\gamma\gamma}$ is the proton energy loss fraction. This works thanks to the conservation of momentum between the two protons and two photons. We use ± to differentiate between the two virtual photons that are radiated from the protons. If we define the z-axis as the direction of one of the proton beams in the accelerator, then the + direction is the direction of this proton, and - is the opposite direction, in which the second proton moves. The two AFP detectors, which are on each side of the main ATLAS detector are denoted each with a letter - A and C, corresponding to the + and − directions. The positive direction of the defined z-axis is on the A side. The $\xi^{\pm}_{\gamma\gamma}$ is defined as

$$\xi^{\pm}_{\gamma\gamma} := \frac{m_{\gamma\gamma}}{\sqrt{s}} e^{\pm y_{\gamma\gamma}}, \begin{cases} \xi^{+}_{\gamma\gamma} = \xi^{A}_{\gamma\gamma} \\ \xi^{-}_{\gamma\gamma} = \xi^{C}_{\gamma\gamma} \end{cases}, \tag{2.4}$$

where $y_{\gamma\gamma}$ is the rapidity of the diphoton system, $\sqrt{s}$ is the center of mass of the beam and $m_{\gamma\gamma}$ is the energy of two photons. We want the $\xi^{\pm}_{\gamma\gamma}$ of two photons recorded in the main ATLAS detector to be ideally the same as the energy loss fraction of protons measured in the AFP detectors for both of the protons in their corresponding directions. The cut is defined as

$$|\Delta\xi^{\pm}| < 0.004 + 0.1\xi^{\pm}_{\gamma\gamma}, \begin{cases} \Delta\xi^{+} = \Delta\xi^{A} := \xi^{+}_{AFP} - \xi^{+}_{\gamma\gamma} \\ \Delta\xi^{-} = \Delta\xi^{C} := \xi^{-}_{AFP} - \xi^{-}_{\gamma\gamma} \end{cases}, \tag{2.5}$$

which can then be rewritten as

$$\Delta_{\text{eff}}\xi^{\pm} := |\Delta\xi^{\pm}| - 0.1\xi^{\pm}_{\gamma\gamma} < 0.004 + 0.1\xi^{\pm}_{\gamma\gamma}, \tag{2.6}$$

for the threshold to be a constant and a linear term $0.1\xi^{\pm}_{\gamma\gamma}$. The $\Delta_{\text{eff}}\xi^{\pm}$ data distribution after the acoplanarity cut is shown in Fig. 2.8. The proton energy loss fractions for the central detector and AFP detectors have to be in a specific interval to fit the selection. The intervals are defined as

$$\xi^{\pm}_{\gamma\gamma} \in [0.031/1.1, 0.084/0.9], \tag{2.7}$$

$$\xi^{\pm}_{AFP} \in [0.035, 0.08]. \tag{2.8}$$

After these two minor interval cuts, the main cut of the $\Delta_{\text{eff}}\xi^{\pm}$ is applied, which concludes the event selection.

Chapter 3

Neural networks

3.1 Theoretical background

3.1.1 The significance of Neural networks

In an experiment, we measure a number of physical quantities, which may be correlated with each other and with the nature of the physical phenomenon which we wish to identify. The aim is to calculate a single output quantity which is otpimally correlated with the physical phenomenon of interest.

3.1.2 The perceptron

Neural networks serve as a powerful tool from within the discipline of machine learning. They help us find complex patterns in enormous amounts of data. They allow the automatization of numerous tasks. The main idea behind a neural network is its most basic component - the perceptron (Figure 3.1). The single cell perceptron can be mathematically described as $\mathbf{f}(\vec{x}) = \vec{x} \cdot \vec{w}^T$. A set of numbers referred to as a vector $\vec{x}$ is defined, which contains measured pieces of information, referred to as input. A function $\mathbf{f}$ maps an input vector $\vec{x}$ to the output. It uses a set of pre-trained weights $\vec{w}$ to determine the output value. It is a very simple binary classifier through which the more advanced machine learning algorithms operate [17].

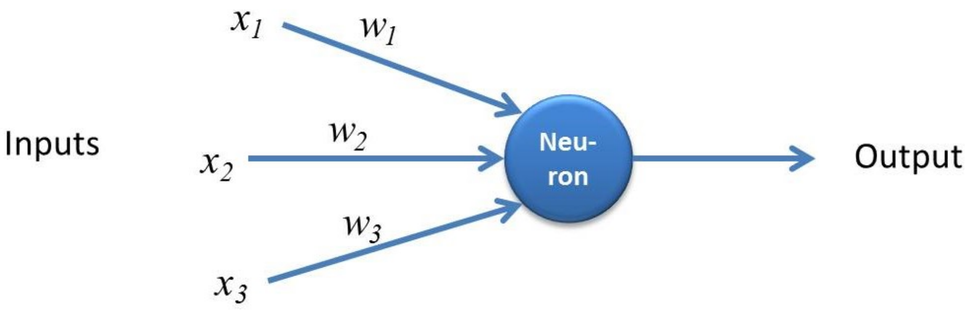

Figure 3.1: Single perceptron unit [18].

3.1.3 Multilayer perceptron

The architecture later used in this thesis is called a multilayer perceptron (MLP). The main advancement from a single-cell perceptron is its capability to approximate any vector function. These abilities come from the architecture of these so-called neural networks. The MLP consists of an input layer, one or more hidden layers, and an output layer (Figure 3.2). These layers consist of multiple perceptrons stacked successively. The output of the perceptrons forming one layer will go directly into the following layer. These are called fully connected linear layers and serve as the building blocks of neural networks [19].

3.1.4 The training process

Neural networks work on a feed-forward principle. As the input data is fed into the first layer of a neural network, the network will perform transformations and output a decision vector. This vector will be passed to a loss function. The loss function is a mathematical function that evaluates the error of the output. It quantifies the difference between the predicted value and the true value, which is usually described in the training process by a label. The idea is, the lower the value of a loss function, the better the neural network performance. Then, the back-propagation process comes into play. For each output y there are gradients $\frac{\partial y}{\partial \vec{w}}$ computed with respect to each weight in the neural network. The process of computing the gradients is made simpler using the chain rule which makes computing gradients in deep neural networks possible. Let the gradient of output vector $\vec{y}$ with respect to the weights of the input layer be $\frac{\partial \vec{y}}{\partial \vec{w}}$. It would be hard to compute this gradient through the hidden layer. It is possible to use the chain rule to compute it in much simpler terms. The chain rule for a neural network with one hidden layer

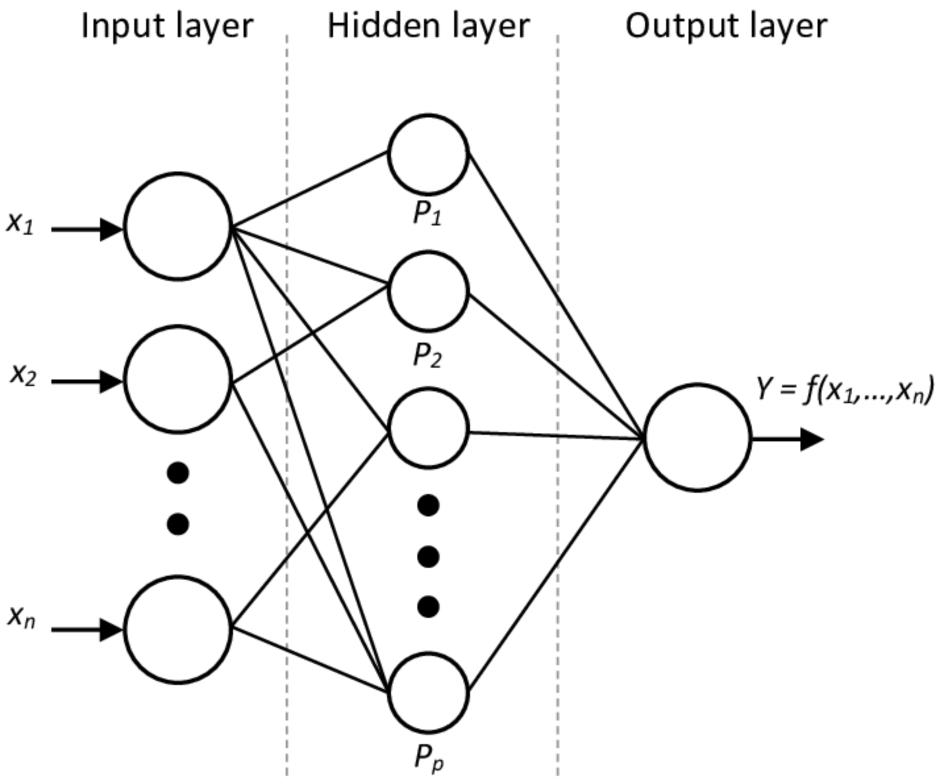

Figure 3.2: Multilayer perceptron with a hidden layer [20].

would look as follows $\frac{\partial \vec{y}}{\partial \vec{w}} = \frac{\partial \vec{y}}{\partial \vec{h}} = \frac{\partial \vec{h}}{\partial \vec{w}}$. The important change in the weights which is called a gradient descent is the following formula:

$$\vec{w}_{new} = \vec{w}_{old} - \alpha \cdot \frac{\partial \vec{y}}{\partial \vec{w}}. \tag{3.1}$$

The weights are adjusted by adding the old weights to the gradient multiplied by a negative learning rate alpha. The purpose of this is clear due to the attributes of a gradient of a function. The gradient of a function gives back the direction of the highest growth of the function output. Because we want the value of the loss function to be as low as possible, we want to go in the opposite direction. The multiplicand α gives the network the possibility to update its weights in a much better resolution. Usually, the α parameter is set to much lower values than 1 [21].

3.1.5 Activation function

The activation functions are added to the MLP as a way to include nonlinear transformation into the networks. These non-linearities enable the network to model complex nonlinear patterns, which the linear-only networks could not. Some improve the approximation and expressiveness capabilities, whilst others focus more on the generalization and overfitting problems. The logistic sigmoid and hyperbolic tangent are among the most used activation functions. They are defined as $sigmoid = \frac{1}{1+e^{-x}}$ and $tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. While effective, the use of these functions is deprecated, due to their computational demands, saturation of very large or very small values, and the problem with vanishing gradient. The vanishing gradient appears when the computed gradient of the output of a network is too small and the learning steps do not affect the weights in any meaningful way. The activation functions which partly solve these problems are called the rectified linear units (ReLU). ReLU is defined as

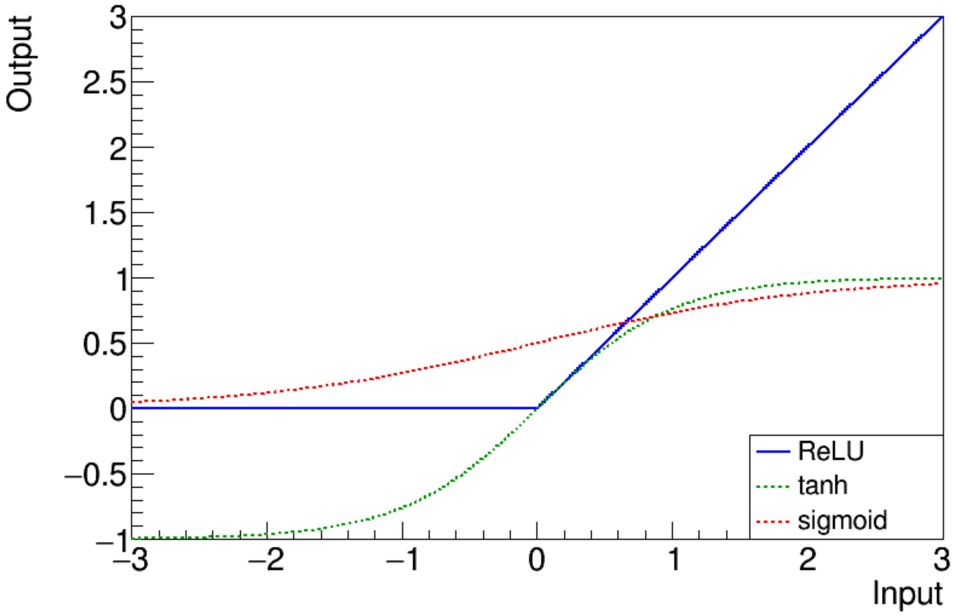

Figure 3.3: ReLU, tanh and sigmoid activation functions.

$ReLU = max(0, x)$, which makes the gradient computing very easy because for every $x \in (-\infty, 0)$, the gradient is equal to 0, and for every $x \in (0, \infty)$ the gradient is equal to 1 [22]. Figure 3.3 shows examples of the described activation functions.

3.1.6 Binary classification

Neural networks can be used to map a set of input features (input vector $\vec{x}$) to a binary output. The training is done by using sets of events that are known to be signal events, and another set containing events denoted as background events. The neural network is trained to give a probability of a set of features being a signal. Then, a threshold is applied on the probability, which labels the input features as signal or background. This may now be applied to a given event whose provenance is not know a priori. The threshold function is very simple. If the output is higher than the threshold, it is classified as a signal. If the output is lower than the threshold, the event is classified as a background. In this case, the sets of experimental events that are classified as signal and background still contain fractions of background and signal, respectively. The influence of these events is negligible. However, these fractions may be calculated and corrected for.

Chapter 4

Implementation of the binary classifier

4.1 Implemented code

The whole implemented classifier and all other Python supportive scripts used are available on GitLab[1].

4.2 ROOT Framework

ROOT is an object-oriented programming language developed by CERN for use in particle physics [23].In terms of this thesis, the root framework is important for the purpose of importing and visualizing data that are used as input features to the network. Ntuples mentioned earlier are stored in the format of .root files. It is better in terms of clarity of the code to load data from the root files into a script, which then extracts any important data branches and after manipulating and converting them into the correct format saves them into a CSV file. Later, this file is loaded by the neural network for training or testing purposes.

[1] https://gitlab.cern.ch/omatouse/alp-search-using-ml

Input features
Acoplanarity
Pseudorapidity of the leading photon
Pseudorapidity of the subleading photon
Transversal momentum of the leading photon
Transversal momentum of the subleading photon

Table 4.1: Input features used in the neural network analysis.

4.3 Data used in training

The data used in neural network training and testing can be divided into two categories - signal and background. First, we have the data we consider as a signal. These simulated signal events had been produced using the SuperChic 4.02 Monte Carlo generator [24]. For each process type from Fig. 2.6 there have been several theoretical masses proposed and for each of these masses, an ntuple has been generated. In total, there are 73 .root signal files with 581216 total events. Real measured Run-2 data have been selected as background events. There is a possibility that somewhere in the measured data there could be an event that we would normally classify as a signal. This would reduce the efficiency of the selection process slightly, but this is assumed to be a negligible effect. A signal would show up as a narrow peak in the invariant mass distribution. In total, there are 201313 background events. Because the mass of simulated signal data is constrained to be in the interval [150,1600] GeV, it is better to remove any recorded data from outside this range. If this cut is applied, the amount of background data is reduced to 68380 events. The same variables as those used in the previous cut-based selection are chosen as the input features:

4.3.1 Distributions of input features

Because there are so many signal files, it would not make any sense to compare all of their distributions. As a representative sample of the distributions of input features, the simulated single dissociative ALP ntuple with mass 700 GeV has been selected. Both the signal and background distributions were normalized so that if we integrate the whole distribution it adds up to one:

$$\int\limits_{-\infty}^{\infty} f(x)dx = 1, \tag{4.1}$$

where f(x) is the probability density of a distribution. To achieve this, all distributions are divided by their corresponding integrals. Unlike other types of normalization such as the Z-score normalization, this will not scale units of the features in any manner. Only the number of events displayed on the y-axis is scaled by a scalar factor. The normalization process is important because the usage of a sole signal ntuple has approximately 40 times fewer events than the background ntuple. Without it we would not be able to compare the distributions well. Figure 4.1 shows that the signal ntuple has a smaller range of acoplanarity with the most number of events being close to zero. This is also the reason why the acoplanarity cut is used in the previous cut-based analysis to filter the events to match simulated ALP samples.

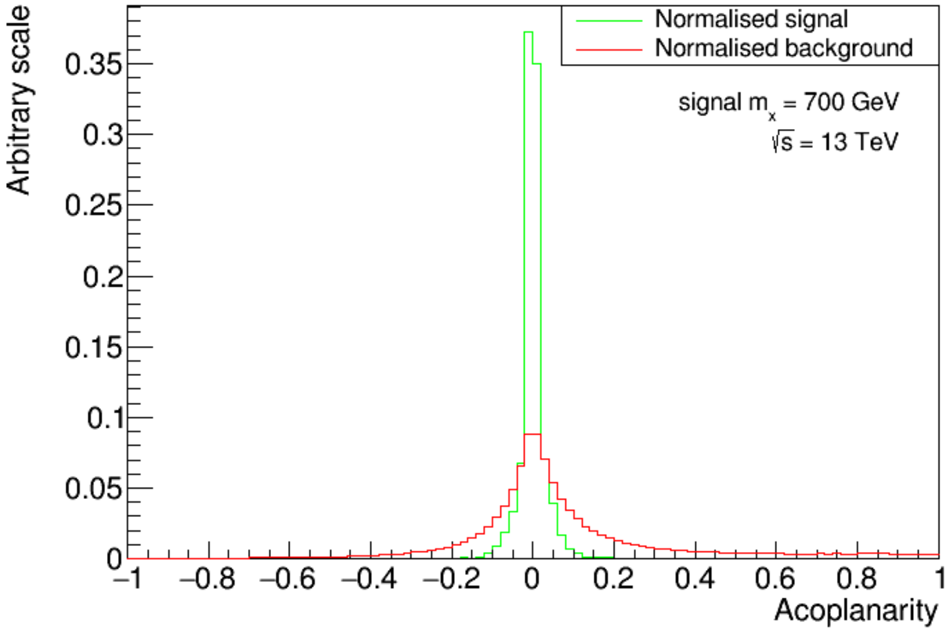

Figure 4.1: Acoplanarity distribution, where background is Run-2 recorded LHC data. The signal is a single dissociative ALP with a mass of 700 GeV. Both signal and background are normalised to a unit area.

For both leading and subleading photon transverse momenta in Figs. 4.2 and 4.3 it is important to note that the signal and background distributions are very different. Unlike in the pseudorapidity distributions in Figs. 4.4 and 4.5, where they share the approximate shape.

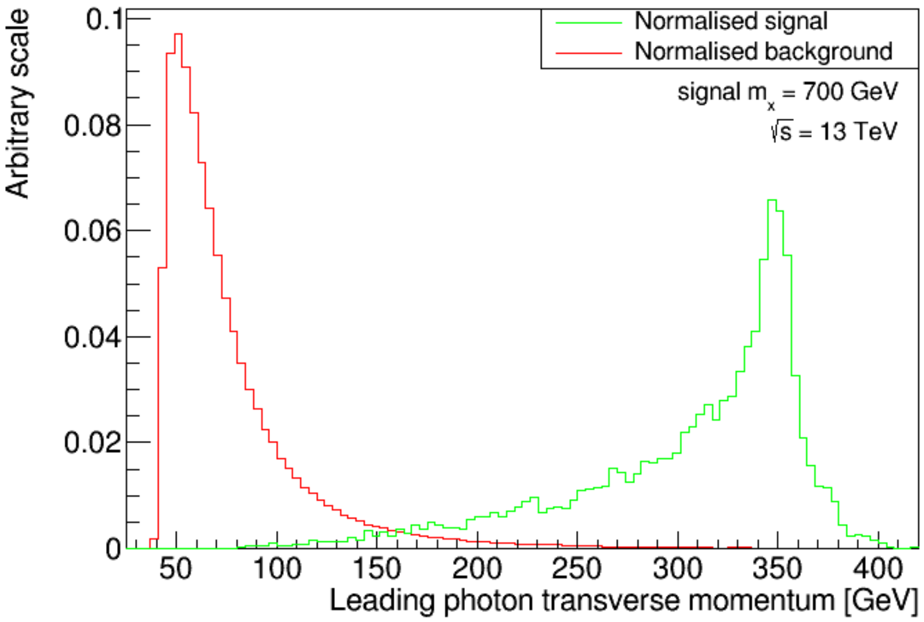

Figure 4.2: Leading photon transverse momentum distribution, where background is Run-2 recorded LHC data. The signal is a single dissociative ALP with a mass of 700 GeV. Both signal and background are normalised to a unit area.

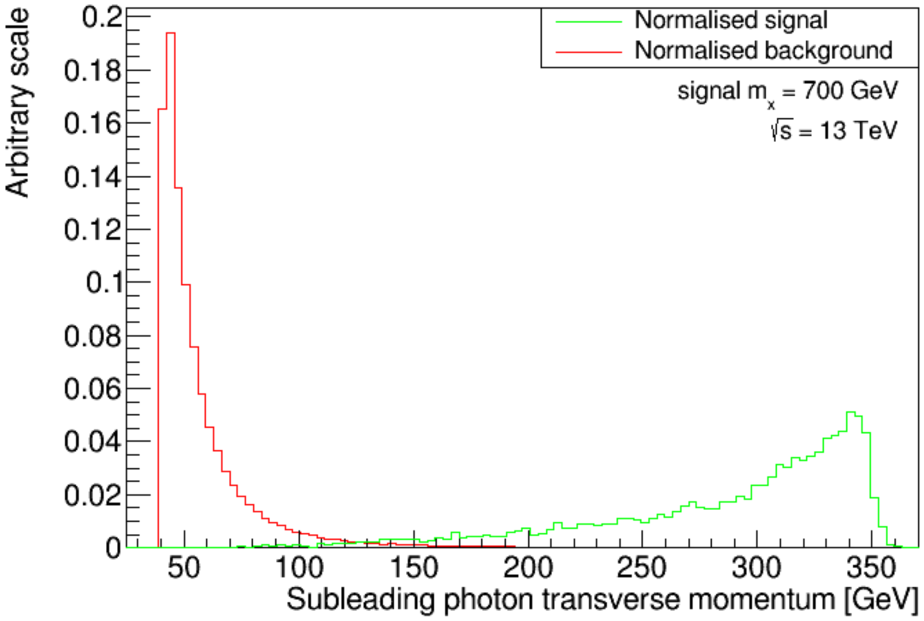

Figure 4.3: Subleading photon transverse momentum distribution, where background is Run-2 recorded LHC data. The signal is a single dissociative ALP with a mass of 700 GeV. Both signal and background are normalised to a unit area.

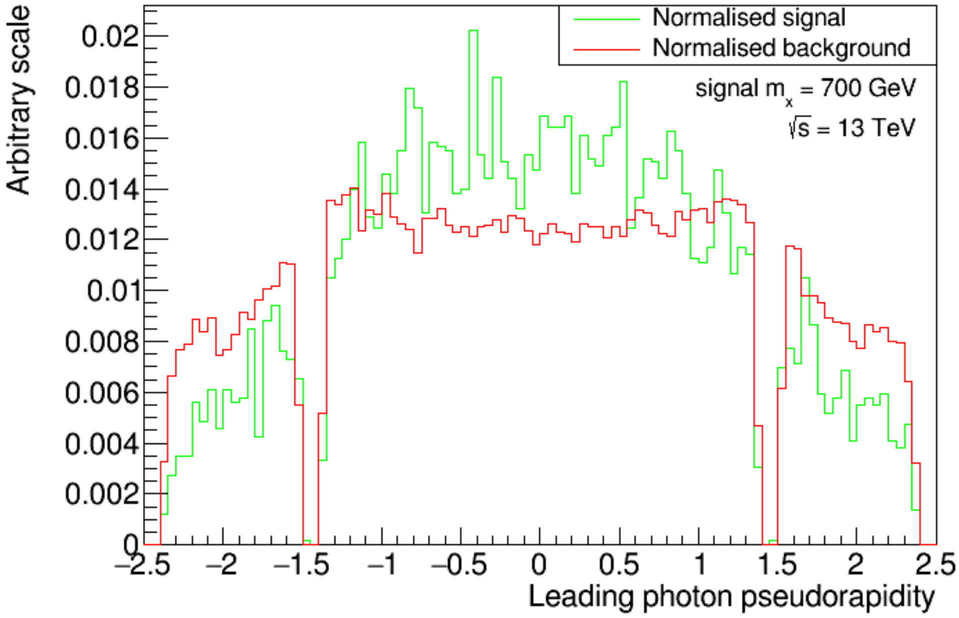

Figure 4.4: Leading photon pseudorapidity distribution, where background is Run-2 recorded LHC data. The signal is a single dissociative ALP with a mass of 700 GeV. Both signal and background are normalised to a unit area.

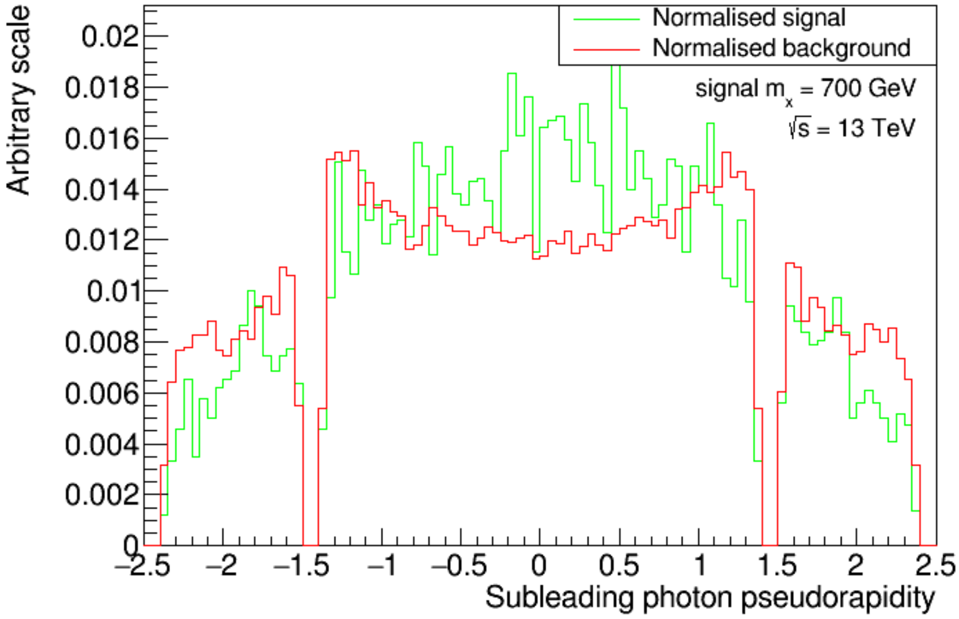

Figure 4.5: Subleading photon pseudorapidity distribution, where background is Run-2 recorded LHC data. The signal is a single dissociative ALP with a mass of 700 GeV. Both signal and background are normalised to a unit area.

Layers of the NN model
Linear layer with 6 input and 16 output features
Batch norm
ReLU
Linear layer with 16 input and 4 output features
Batch norm
ReLU
Linear layer with 4 input and 2 output feature
Softmax layer

Table 4.2: Layers of the neural network model used.

4.4 Neural network model

The neural network model was designed in the standard Pytorch framework using its machine learning functions. [25]. Previous studies use deep learning for event classification in High-Energy Physics [26]. With respect to the low number of input features in Table 4.1, it might be better to start with a light model with fewer layers. The basic structure of the model is therefore very similar to a multi-layer perceptron. The difference is that in between each layer is a batch normalization layer and after that comes a ReLU layer to give the network a nonlinear transformation. When designing the model, the input features had to be taken into account. The model has only three fully connected layers, in order to prevent overfitting. The layers of the network are stacked in the following manner, defined in Table 4.2. The last layer called Softmax is defined as

$$Softmax(x_i) = \frac{\exp x_i}{\sum_j \exp x_j}, \tag{4.2}$$

and it transforms the output of the last linear layer into probabilities. The output tensor has to sum up to 1 and both of the values lie within the interval $[0, 1]$. The following equation has to be valid: $1 - first\ output = second\ output$. In addition to the model, few other functions from the Pytorch framework have been used. First, the Dataset and DataLoader classes are used to import data into the model. As an optimizer, Adam was chosen with default hyperparameters [27]. As a loss function, the cross entropy loss was chosen, because of its wide use in binary classification problems [28]. The weights were randomly generated using a Pytorch module called $xavier\ uniform$ which generates random data with uniform distribution. The data are normalized using the Z-score normalization [29]. It is described by the following equation

$$normalised\ data = \frac{data - E}{\sigma}, \tag{4.3}$$

where E is the expected value and σ is the standard deviation calculated from the data using the numpy library. There are several times more signal events than there are background events and this class imbalance has to be taken care of. The solution could be to simply get more data, but in this case, it is not possible, so the correct solution would be to add weights to the loss function. The weights are directly integrated into the cross entropy loss Pytorch function. Weights are calculated by the following formula

$$weights = \frac{1}{occurance}, \tag{4.4}$$

which is then normalized by dividing the whole weights tensor (weights tensor $\in \mathrm{R}^{1\times2}$) by the sum of its components.

4.4.1 Training and validation

The data used to train and test the network are created by randomly mixing the background and signal events together. Then a portion of the data is taken as the training part, and the rest is used for testing the network. In this case, 70% of the mixed data is used for training, and 30% is reserved for later use. In other words, the training process, which consists of training and validating the network uses 70% of the total data, and the 30% remaining data is saved for later analysis of the network.

During training, the data are divided again into an actual training part, which consists of 80% of the data, and into a smaller portion of 20% of the data which is used after each epoch to calculate a validation loss, which is used to prevent overfitting. This validation is used only for training, hence 20%:80% division of data is needed. For training the following hyperparameters are used:

- Lr = 0.001
- β = (0.9, 0.999)
- Eps = 1e-8
- Weight decay = 0
- Batch size = 64
- Epochs = 19
- Threshold = 0.5

Validation in training as well as in later analysis uses batch size equal to unity. The data is randomly shuffled. The development of loss function value during training of the model is shown in Fig. 4.6. The confusion matrix after training is shown in Fig. 4.7.

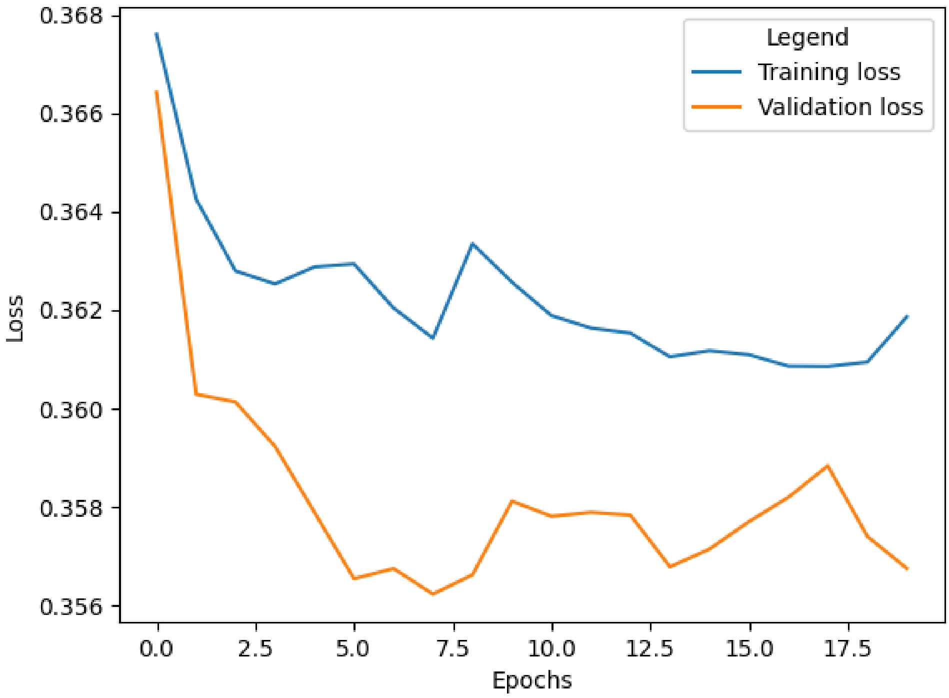

Figure 4.6: Loss during training of the neural network, with 56% of the original 581216 signal and 68380 background events being used for training and 14% is used for validation.

4.4.2 ROC curve

The receiver operator characteristic (ROC) is used to compare the classifying capabilities of different models. To understand the comparison few terms have to be defined. During the validation of a data sample, the validation outcome can be divided into 4 categories. True positive (TP) means that the data sample has been predicted as a signal and in reality is a signal. False positive (FP) means that the data sample was classified as a signal but actually it is background. True negative (TN) and false negative (FN) are analogous to TP and FP except the predicted values are opposite. The true positive rate (TPR) states the probability of classifying an event as a signal correctly. It is calculated as

$$TPR = \frac{TP}{FN + TP}. \tag{4.5}$$

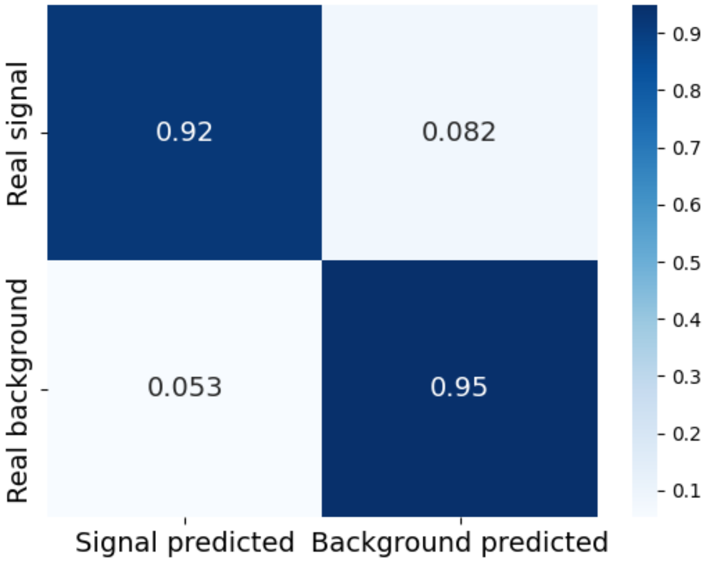

Figure 4.7: Confusion matrix with unoptimized threshold 0.5, where 56% of the original 581216 signal and 68380 background events have been used for training.

A false positive rate (FPR) states the proportion of background instances falsely classified as signal events. It is calculated as

$$FPR = \frac{FP}{FP + TN}. \tag{4.6}$$

To create a ROC curve for the model, the 30% of data that has been split from the original data mix for validation will be used. The threshold is moved in the interval [0,1] with a step of 0.01. For each threshold, FPR and TPR are calculated. The FPR and TPR are then plotted on the x and y-axis. ROC of the trained model is shown in Fig. 4.8.

ROC optimization

Using the ROC curve for optimization of the capabilities of a classifier is a standard step in the model creation process. Two methods were proposed. First, Youden's J statistic is used. The principle behind this is to calculate the J value for each threshold, which is done as follows

$$J = sensitivity + specificity - 1 = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1 \tag{4.7}$$

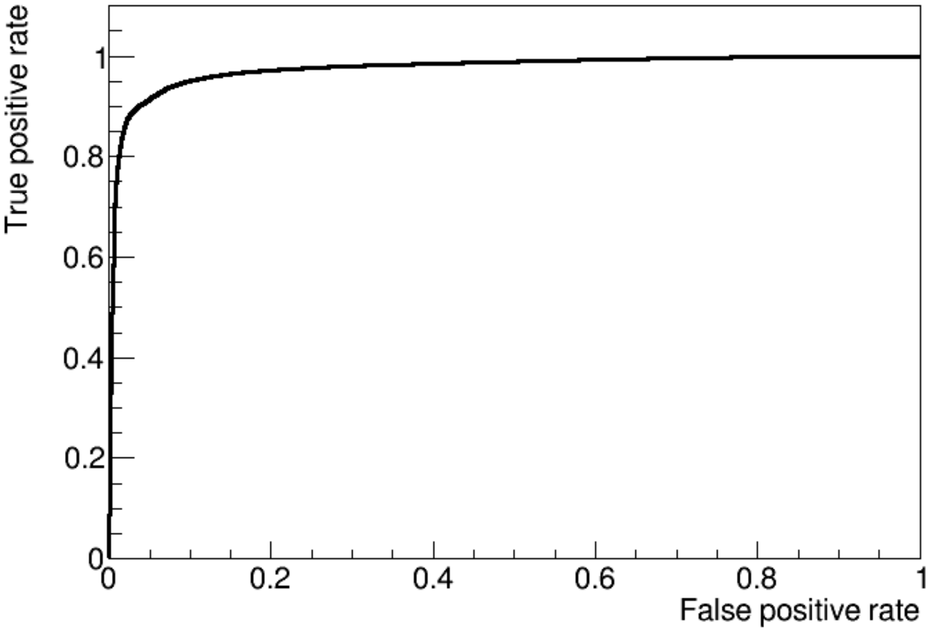

Figure 4.8: ROC curve of a binary classification model, where 56% of the original 581216 signal and 68380 background events have been used for training.

and then selecting the threshold for which the corresponding J value is the highest. It is the implementation of a formula $argmax(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$. This method, however, often fails on a dataset that has an imbalanced number of class events, such as the one used in this thesis [30]. Therefore, another metric has to be used to obtain a better result. F_1 scores is a name of an ROC metric that can be used, even when the dataset has an imbalance in the number of events between both classes for a robust threshold selection. The idea is to calculate the F_1 scores which range from 0 to 1, where 0 is the least accurate classifier and 1 is the most accurate classifier [31]. The F_1 scores are calculated as follows

$$F_1 = \frac{2TP}{2TP + FP + FN}. \tag{4.8}$$

Optimization comparison

Youden's J statistics states that the optimal threshold for a given model is $thr = 0.44$. F_1 scores state the optimal threshold as $thr = 0.0045$. The resulting thresholds are compared using confusion matrices with comparison to a threshold of $thr = 0.5$, as shown in Fig. 4.7. It is clear that Youden's J statistic does not help to optimize the classifying capabilities of the model. The sole difference between $thr = 0.5$ (Figure 4.7) and $thr = 0.44$ (Figure 4.9)

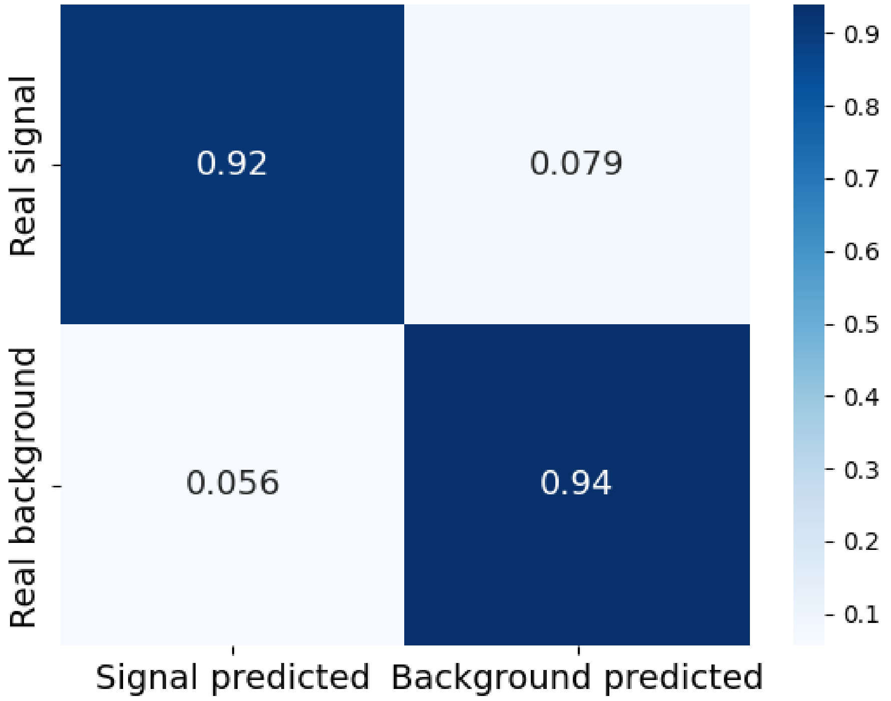

Figure 4.9: Youden's J statistic confusion matrix, where 56% of the original 581216 signal and 68380 background events have been used for training.

is the worsening of the background prediction by 0.01. On the other hand, F_1 scores (confusion matrix in Fig. 4.10) made quite a leap in the classification of signal events. On the other hand, its capabilities have a downside of drastically worse background prediction. From this ROC analysis, the optimal threshold should depend on the situation for which the model would be used. If the goal is to have the most true positive hits, then F_1 scores should be used. As a generally balanced classifier, the threshold of $thr = 0.5$ is sufficient as shown in Fig. 4.7.

4.4.3 Feature importance

Shapley values

Shapley values were first introduced by Lloyd Shapley as a solution to an open question in game theory, i.e how to correctly evaluate the importance of

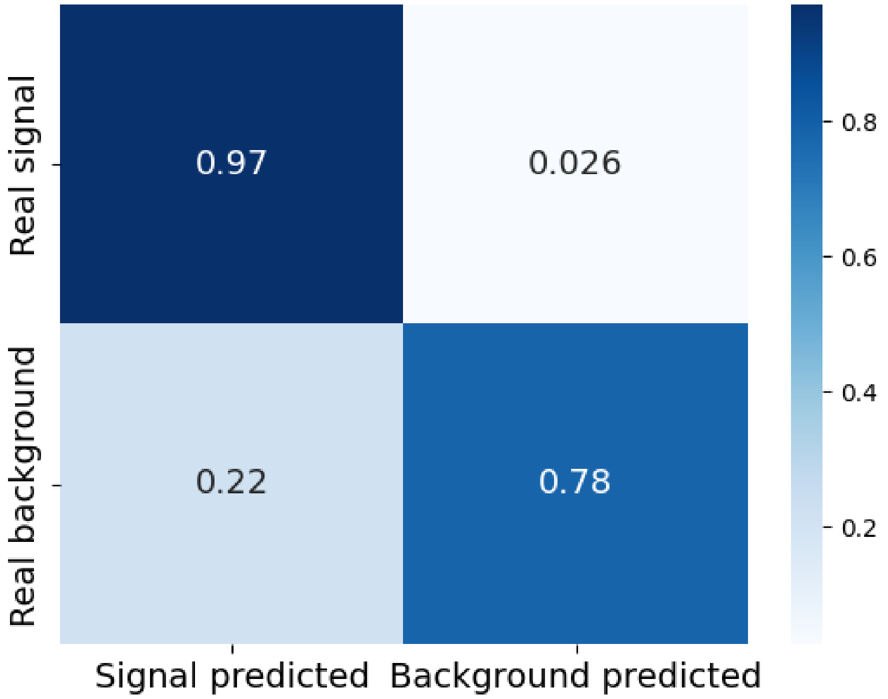

Figure 4.10: F_1 scores confusion matrix, where 56% of the original 581216 signal and 68380 background events have been used for training.

individual players in a team during a cooperative game. The game could be any kind of cooperative game where a coalition of players tries to get as big an overall gain as possible. The gain in question is any kind of gain, which is game-specific. The main idea is that each player in a coalition must have a personal contribution that must be expressable by a certain quantity. This quantity can be obtained for all of the players in a coalition and can distribute the overall gain of a coalition in-between the players, which depends on the amount of personal contribution to the whole team [32].

The use of Shapley values in sorting feature importance

To understand the full capabilities of a neural network model, it is important to analyze large quantities of different data. It is desired to achieve as complete analysis as possible. The correct approach includes finding the importance of individual input features, and sorting them from the most important to the least important. If a feature has very small or no effect on the network prediction, it can be eliminated from the whole training process. There are many ways to obtain feature importance. The Shapley values are very often used and are also known to be a robust way to compute the individual importances. Instead of their classical significance as a metric used to determine the influence of individual players in a cooperative game, in machine learning they are used to find the approximate influence of each input feature on the final model prediction. To use the Shapley values correctly, a shap module is used with its Deepexplainer class to predict the feature importance [33]. The input to Deepexplainer is a sample of background and signal data. The Deepexplainer was used to calculate Shapley values twice. First with the signal sample being low mass single dissociative 300 GeV ntuple, and second time with a representative sample from the higher mass events being a single dissociative 1600 GeV ntuple. The Deepexplainer was used each time with 1000 background and 1000 signal data events and the process of estimating Shapley values was repeated 50 times. A mean shap value for each feature was plotted with the corresponding standard deviation as an error bar graph. In the higher mass plot (Figure 4.11), it is clear that the transverse momenta are the most important features. This could be explained by elementary physics, because the more mass a particle has, the more momentum it generates. This is the biggest difference between the lower and upper mass particles that the network sees. For lower mass plot (Figure 4.12), an important feature that also follows the theoretical background is the Acoplanarity. It has a significant role in the decision process, because it correlates with the mass of ALP [34]. This is also why the Acoplanarity cut was so important in the original selection. Overall in the lower mass plot, the transverse momenta decisions are distributed between other features.

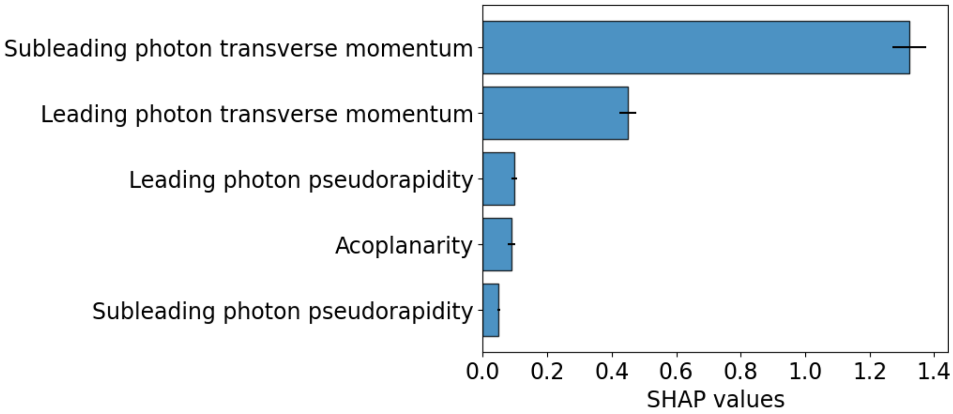

Figure 4.11: Shapley values for 1600 GeV single dissociative events, where 56% of the original 581216 signal and 68380 background events have been used for training.

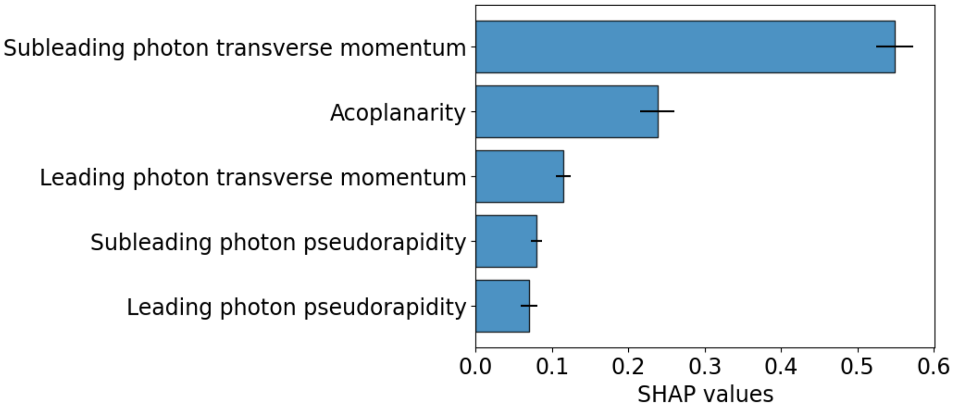

Figure 4.12: Shapley values for 300 GeV single dissociative events, where 56% of the original 581216 signal and 68380 background events have been used for training.

Chapter 5

Data analysis using a neural network

5.1 Goal of the analysis

The main goal of this project is to increase the efficiencies and the corresponding significances of the neural network approach in comparison with the previous cut-based analysis. Out of the three event types shown in Fig. 2.6, the main focus is on the single dissociative events. The reason for this choice is quite simple. The comparison is made with respect to efficiency plots in [1]. The efficiencies in the paper [1] are only shown for single dissociative and exclusive events, hence the double dissociative efficiency comparison is not included. The exclusive event type already has a nearly 100% efficiency rate when applying the cut-based analysis, which means that it would be to no purpose to optimize the network for this event type.

5.2 Threshold selection

5.2.1 Bisection method

The bisection method is usually used as an algorithmic approach for finding the roots of nonlinear equations. The goal is to find such an x value that would satisfy $f(x) = 0$, where $f(x)$ is continuous in the relevant x range. The algorithm behind this method executes the following steps. First, two different values x_1 and x_2 have to be found, for which $f(x_1) > 0$ and $f(x_2) < 0$. This is done, because if some two points fulfill these conditions, we know that the root must be somewhere in between them, because the function has to cross the x-axis. Whether $x_1 < x_2$ or $x_2 < x_1$ is dependent on the function this algorithm is evaluating. When this is done, the interval $[x_1, x_2]$ (or $[x_2, x_1]$) is cut in the middle and a new point is created $x_3 = \frac{x_1+x_2}{2}$. Now $f(x_3)$ is calculated. Based on the value of $f(x_3)$, we can remove one duplicate point with the same sign (+ or −). This removes half of the interval $[x_1, x_2]$ (or $[x_2, x_1]$). These steps are repeated until x_n is found in acceptable numerical proximity of the correct solution of the equation $f(x) = 0$ [35]. This process is visualised in Fig. 5.1.

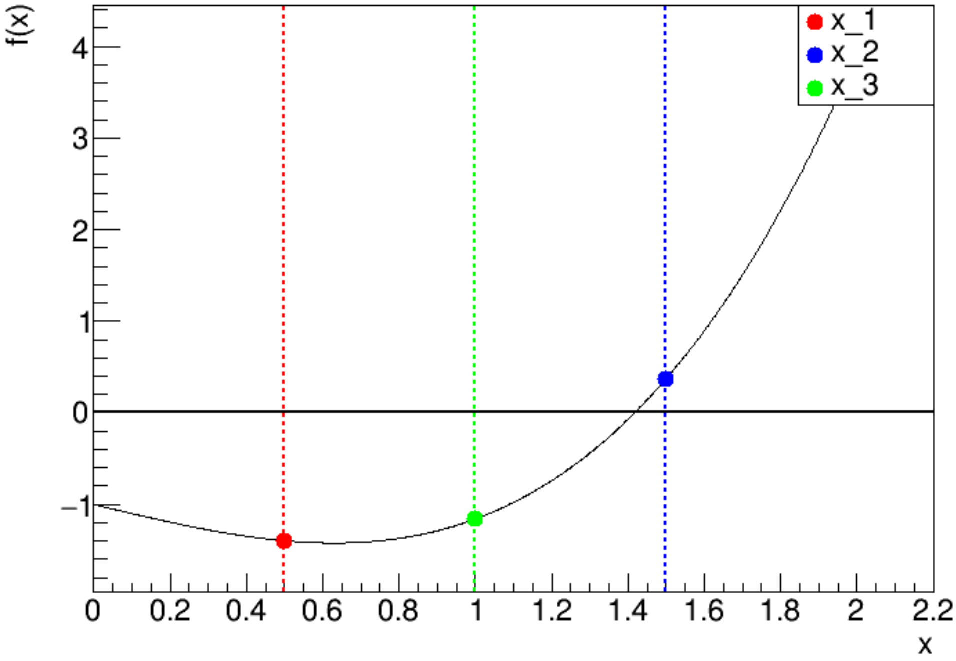

Figure 5.1: Visualisation of the first step of the bisection method.

5.2.2 Use of bisection for threshold search

For a fair comparison, a threshold must be selected with respect to the results of the previous analysis. When passing the recorded Run-2 data through the cut-based analysis, 441 events have been selected as potential ALP candidates [1]. In the threshold selection, all of the recorded LHC data that passed through the $\Delta\xi$ are used, instead of the portion used for testing in feature importance analysis. The threshold selection is based on the bisection method. First Run-2 data pass through the network and probabilities for each event being an ALP candidate are calculated. A modified bisection algorithm can now be used. The function $f(x)$ on which this method is applied takes threshold value as an input x and outputs the number of events the network classifies as ALP candidates. The interval, in which threshold search is waged, is defined as [0,1], from where the two extreme points are chosen as $x_1 = 0$ and $x_2 = 1$. For both of these points, the number of events passed through is calculated. Now the bisection step is applied, and the number of past events through the network is computed for a point $x_3 = \frac{x_1+x_2}{2}$. The next step in a standard bisection method would be to check whether $f(x_3) > 0$ or $f(x_3) < 0$. This is the step that is modified, because, instead of comparing the function value with 0, it is compared with 441. The threshold found by this method converged to $thr = 0.14$. The dependency of the number of events passed through the network on the height of the threshold is shown in Fig. 5.2.

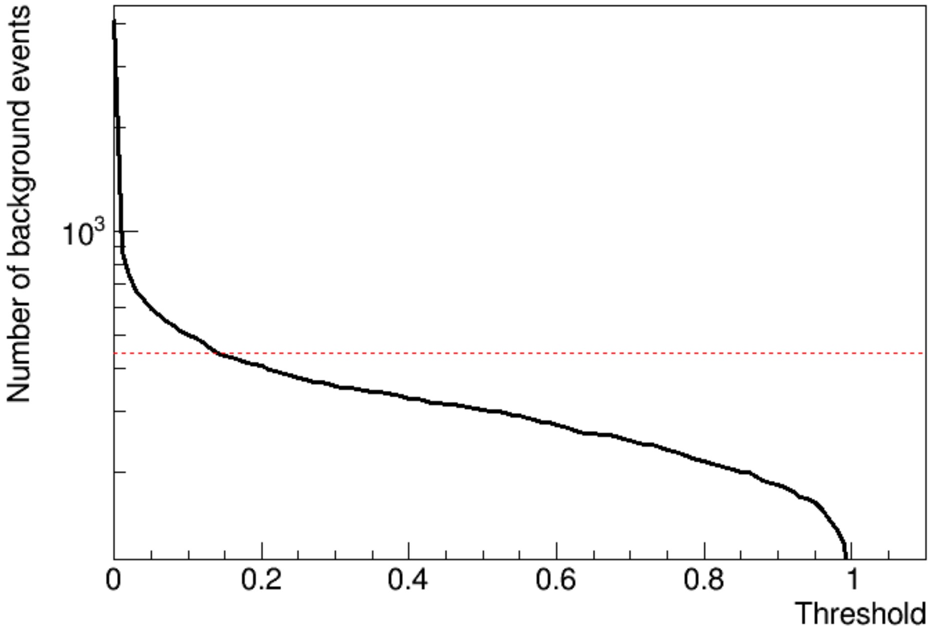

Figure 5.2: Number of selected events as a function of the neural network threshold, where all 4086 background events that passed through the $\Delta\xi$ cut have been used. The model had used 56% of the original 581216 signal and 68380 background events for training. The horizontal line at 441 represents the number of background events in the cut-based analysis.

5.3 Efficiency and significance analysis

Because the $\Delta\xi$ cut from selection process 2.6 has to be applied to the data regardless of the preselection method, it is applied before the efficiency and significance analysis takes place. This drastically decreases the number of events that go through the network preselection. The number of background data (Run-2 recorded data) is reduced to 4086. The number of signal events entering the efficiency analysis after $\Delta\xi$ cut is listed in Table 5.1.

Mass (GeV)	200	300	400	500	600	650	700	800	900	1000	1200	1400	1600
Events	1612	2416	3176	3699	4092	4300	4444	4784	4742	4895	4672	4081	3298

Table 5.1: Number of events after $\Delta\xi$ cut for each single dissociative signal event mass.

To be able to determine the efficiency of a network, the total number of events that passed through the network is divided by the original number of generated events. For all the masses, this is achieved by firstly calculating the original number of events using the cross-section σ (a measure of the probability that an event happens during a collision of two particles) and the integrated luminosity L (a measure of the number of collisions during the data-taking period) by multiplying them with each other such as $N_0 = \sigma \cdot L$, where N_0 is the original number of events. All events also have their weights. Both of these information are directly accessed from the ntuples. They are used to calculate the total number of weighted events for the efficiency analysis by adding all the weights of those events, which were selected by the network together. The efficiency for a single mass type ntuple is calculated as $\varepsilon = \frac{sumw}{N_0}$, where $sumw$ is the sum of weights from the selected events. The comparison in Fig. 5.3 is made with a graph that uses recalculated values that match the efficiency plot from [1]. The cross-section for each mass is shown in Table 5.2.

Mass (GeV)	200	300	400	500	600	650	700	800	900	1000	1200	1400	1600
σ (fb)	4.1470	2.5640	1.7330	1.2400	0.9165	0.7974	0.6946	0.5399	0.4242	0.3400	0.2227	0.1512	0.1054

Table 5.2: Cross-section, σ, for each single dissociative event mass.

Both efficiency curves peak at around 1000 GeV and decline towards the lower and higher masses. This can be explained by looking at Fig. 5.4. The proton

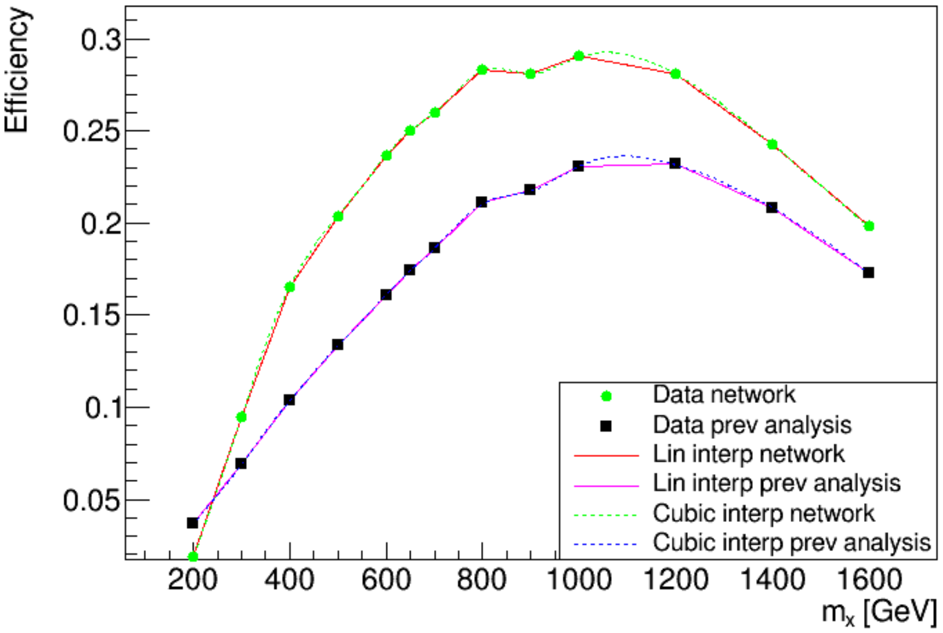

Figure 5.3: Efficiency comparison with single dissociative events, where 56% of the original 581216 signal and 68380 background events have been used for training of the neural network.

energy loss fractions of A(C)-side AFP detectors are denoted by $\xi^{+}_{\gamma\gamma}(\xi^{-}_{\gamma\gamma})$, and lie on the x and y-axis. The two yellow bands show the high-efficiency (acceptance) ranges of the two AFP detectors. This means that everything outside of the yellow area is undetectable. The blue hyperbolae with mass descriptions show where the theoretical ALPs with their corresponding masses would lie on the graph. Because most of the blue hyperbolae are with almost all their length outside of the yellow area (high acceptance), we do not detect many of the ALP candidates that would lie outside of the high acceptance area. This is not the case for 1000 GeV hyperbola, where the whole hyperbola lies within the yellow bands. For this specific mass, there is much higher detection sensitivity, and thus the efficiency is higher as well.

The resulting efficiency curves in Fig. 5.3 show that the network has better efficiency everywhere except at 200 GeV. The significance can be calculated as

$$Significance = \frac{S_{ev}}{B_{ev}} = \frac{S_{ev}}{441}, \tag{5.1}$$

where S_{ev} is a number of events selected from the theoretical ALP generated files, and B_{ev} is a number of events selected from the background file. The background file consists of the Run-2 measured events. Because of the nature of the comparison, $B = 441$ is kept constant for all signal masses. The significance in Fig. 5.5 peaks at 400 GeV, but otherwise follows the trend of the cut-based analysis. For 200 GeV the significance obtained by the network is again lower than the one from the previous analysis.

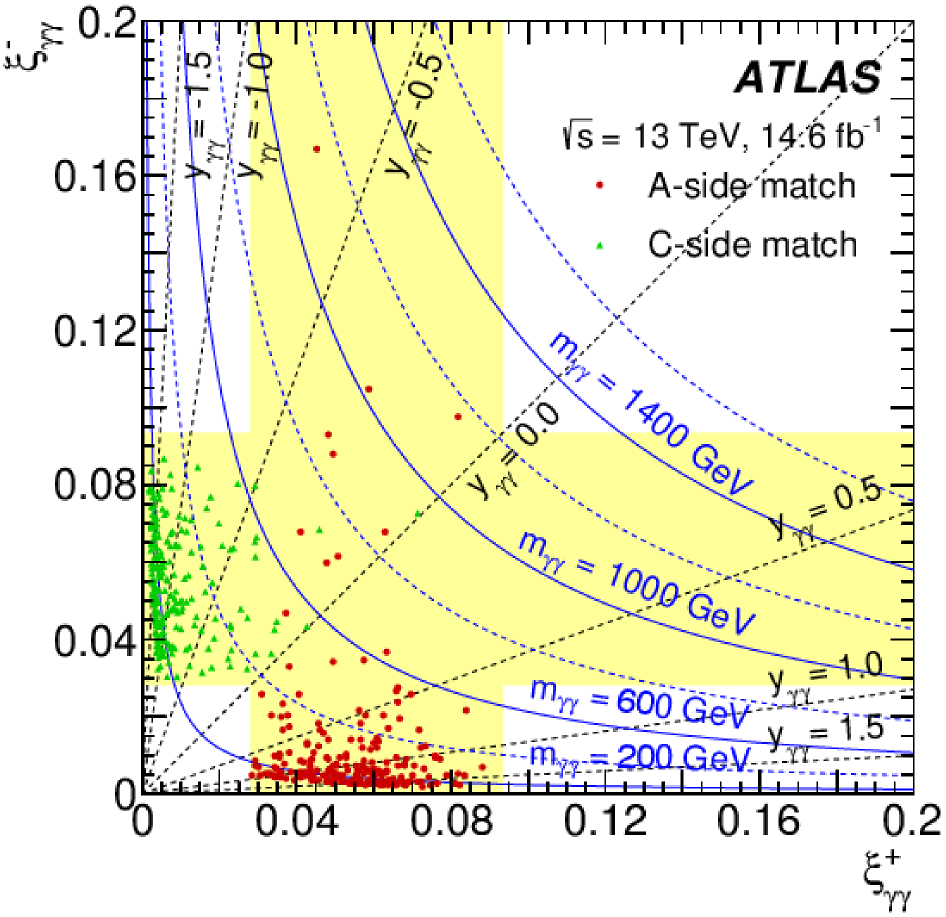

Figure 5.4: ($\xi^+_{\gamma\gamma}$,$\xi^-_{\gamma\gamma}$) distribution of the selected data candidates after the full event selection in $m_{\gamma\gamma} \in$ [150,1600] GeV with $m_{\gamma\gamma}$ contours (blue) and $y_{\gamma\gamma}$ contours (black). The range of $\xi_{\gamma\gamma}$ in which forward-proton matching is possible, $[0.035 - \xi_{th},\ 0.08 + \xi_{th}]$, is indicated by the yellow rectangle for each side. Events passing the matching requirement on the A(C)-side are represented by the red dots (green triangles). No event passed the matching requirement for both the A-side and C-side [1].

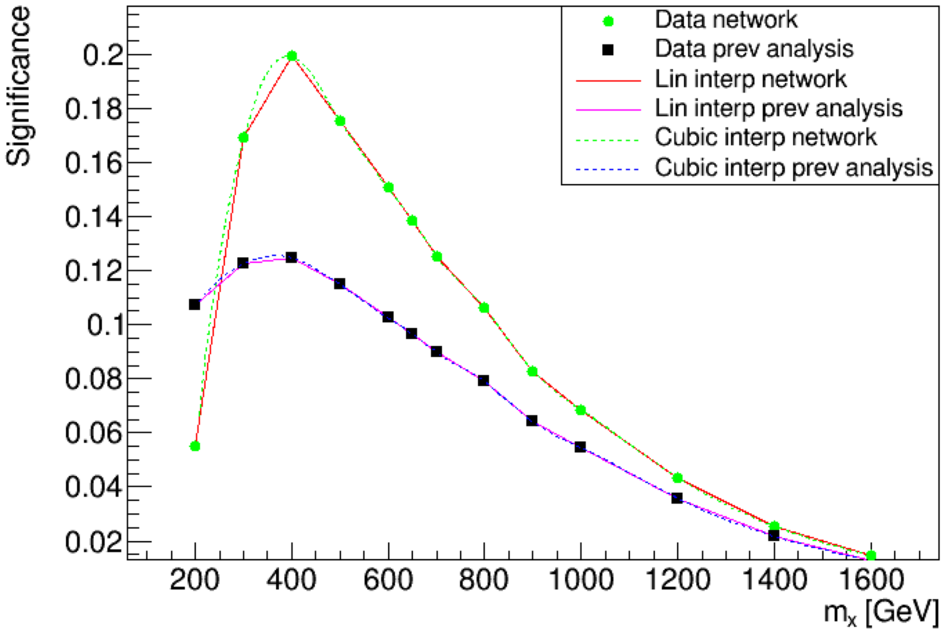

Figure 5.5: Significance comparison with single dissociative events, where 56% of the original 581216 signal and 68380 background events have been used for training. The network is trained on all masses.

5.4 Model optimization

For a signal mass at 200 GeV, the model performs worse than the previous cut-based analysis. One solution to this issue could be to optimize the decision threshold of the neural network output. This however would make the direct comparison of the efficiency with the previous cut-based analysis impossible. A second solution of optimization of the efficiency and significance at 200 GeV exists. First, to prove this statement, the network is trained on 200 GeV data samples only. The significances are compared in Fig. 5.6 and it is clear that there is room for improvement in Fig. 5.3. The goal is to add more low-mass data samples to the training. This is accomplished by giving more weight to the 200 GeV signal events to the training process. Two tests were made and their results are shown as significance and efficiency comparison plots. Firstly, 4 times the original weight is placed before the training on the 200 GeV data samples. The results are shown in Figs. 5.7 and 5.8. For the second test, 8 times more weight is placed before the training on the 200 GeV data samples. The results are shown in Figs. 5.9 and 5.10. When adding 4 times the original amount of weight of the 200 GeV data into the training process, the separation significance has already drastically improved, but is still under the significance of previous cut-based analysis. When adding 8 times the original amount of weight of the 200 GeV data into the training process, the model completely outperforms the previous analysis. The efficiency is shown in Fig. 5.9, but the difference is better displayed in Fig. 5.10.

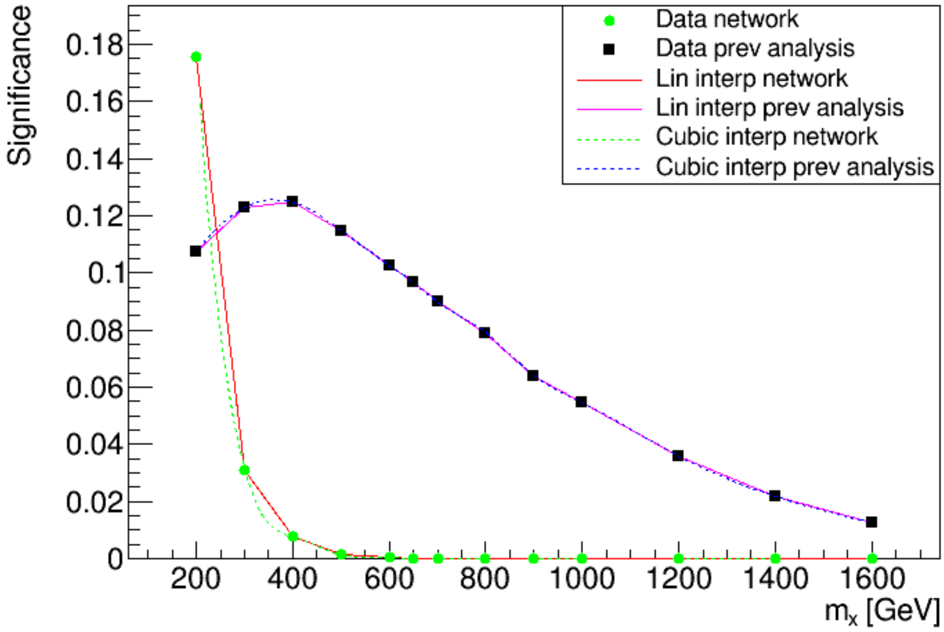

Figure 5.6: Significance comparison between previous cut-based analysis and a neural network model trained on 200 GeV samples only [1]. In total 31761 signal and 68380 background events have been used for training.

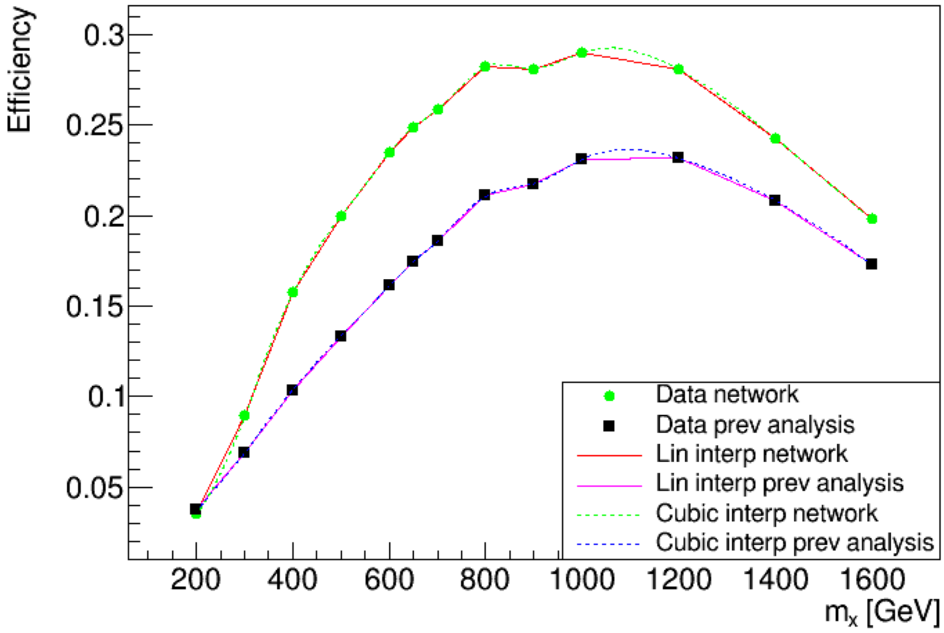

Figure 5.7: Signal efficiency comparison between previous cut-based analysis and a neural network model using augmented training [1]. The training is augmented by giving the 200 GeV signal samples 4 times more weight to improve the efficiency at this mass.

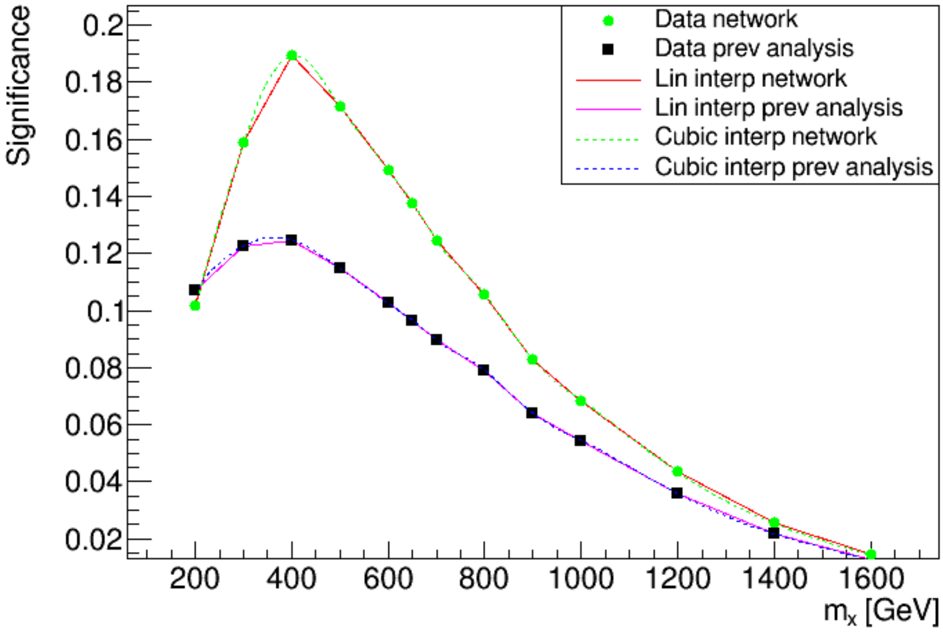

Figure 5.8: Significance comparison between previous cut-based analysis and a neural network model using augmented training [1]. The training is augmented by giving the 200 GeV signal samples 4 times more weight to improve the efficiency at this mass.

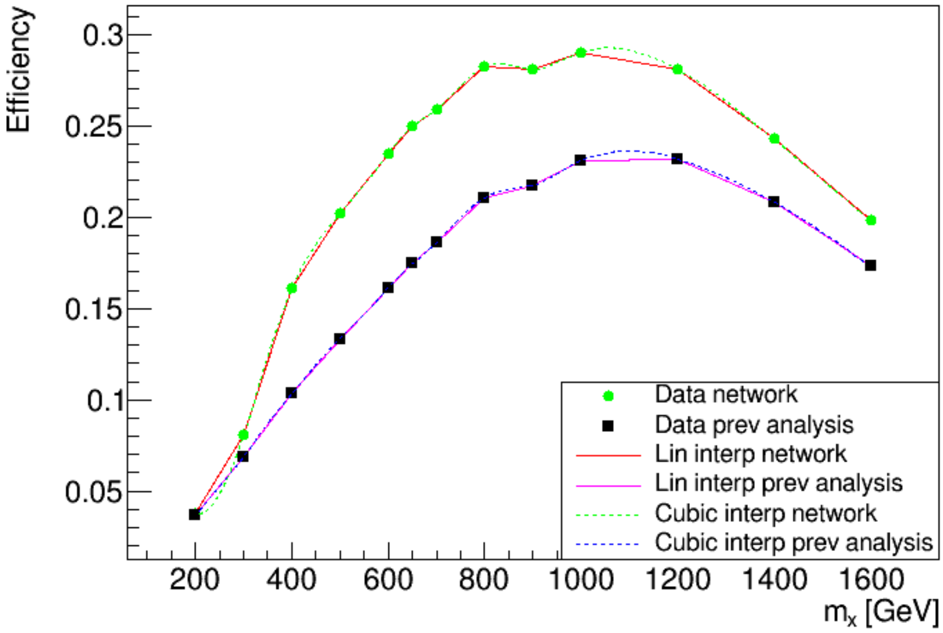

Figure 5.9: Signal efficiency comparison between previous cut-based analysis and a neural network model using augmented training [1]. The training is augmented by giving the 200 GeV signal samples 8 times more weight to improve the efficiency at this mass.

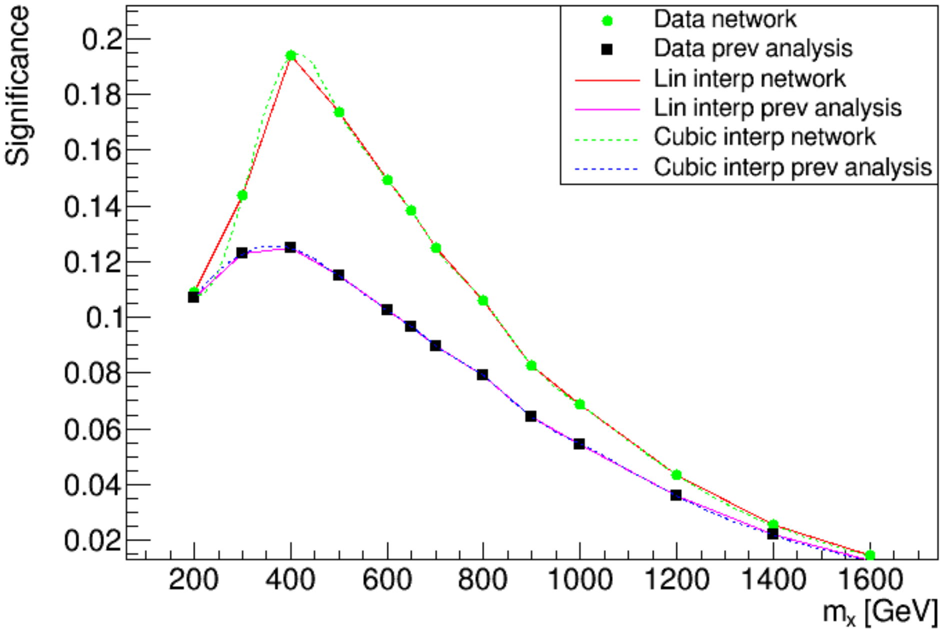

Figure 5.10: Significance comparison between previous cut-based analysis and a neural network model using augmented training [1]. The training is augmented by giving the 200 GeV signal samples 8 times more weight to improve the efficiency at this mass.

Chapter 6

Conclusions

The neural network model has replaced a crucial step of preselecting data and demonstrated better efficiency and significance than the previous cut-based analysis. This improvement in performance is largely due to the ability of neural networks to identify complex patterns in the data that may not be apparent through traditional methods. The neural network model has been trained on both simulated and recorded data. The model has shown great potential in improving the search for Axion-Like-Particles. One of the limitations of the first trained model was its inability to classify data on lower masses, particularly at 200 GeV. A solution for this issue could be optimizing the decision threshold using F_1 scores, which could result in a significant improvement in efficiency, and possibly the significance. To directly compare both the cut-based and the neural network preselection methods, however, this has not been done, and the comparison is based on the same number of background events. In practice, it was addressed by training the network with higher weights of the 200 GeV signal. This improved the comparison at 200 GeV significantly and at the same time did not worsen the efficiencies of other masses.

Future studies might benefit from larger datasets and higher number of signal samples as well as higher number of input features. The augmentation of the number of input features would potentially allow for a neural network to follow more complex hidden patterns, resulting in higher efficiencies and significances.

Bibliography

[1] ATLAS Collaboration. Search for an axion-like particle with forward proton scattering in association with photon pairs at ATLAS. In *JHEP*, 2023. Available from: https://arxiv.org/abs/2304.10953.

[2] A. Sopczak et al. Search for an Axion-Like Particle in Light-by-Light scattering using the ATLAS central detector and the ATLAS Forward Proton detector. 2020. Available from: https://cds.cern.ch/record/2714416.

[3] CERN. The Standard Model of particle physics. https://home.cern/science/physics/standard-model, 2014. [Online; accessed 20-May-2023].

[4] Wikipedia. Standard Model — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Standard%20Model&oldid=1134367106, 2023. [Online; accessed 09-February-2023].

[5] CERN. CERN. https://home.cern/, 2018. [Online; accessed 20-May-2023].

[6] CERN. The history of CERN. https://www.home.cern/about/who-we-are/our-history, 2018. [Online; accessed 20-May-2023].

[7] CERN. ATLAS Fact sheet. https://atlas.cern/Resources/Fact-sheets, 2021. [Online; accessed 20-May-2023].

[8] CERN. ATLAS Scheme. https://atlas.cern/, 2018. [Online; accessed 09-February-2023].

[9] Wikipedia. Lorentz force — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Lorentz%20force&oldid=1137122954, 2023. [Online; accessed 09-February-2023].

[10] J. Chwastowski. ATLAS Forward proton detector. http://atlas-project-lumi-fphys.web.cern.ch/, 2019. [Online; accessed 20-May-2023].

[11] Ch. Sutton. Quantum chromodynamics. https://www.britannica.com/science/quantum-chromodynamics, 2023. [Online; accessed 20-May-2023].

[12] Ch. Sutton. CP violation. https://www.britannica.com/science/CP-violation, 2010. [Online; accessed 20-May-2023].

[13] Brian Koberlein. QCD symmetries. https://briankoberlein.com/blog/left-hand-of-darkness/, 2015. [Online; accessed 09-February-2023].

[14] CERN. Ntuples vs TTrees. https://www.slac.stanford.edu/exp/glast/wb/test/pages/rootPages/ntuplesVsTTrees.htm, 2015. [Online; accessed 09-February-2023].

[15] CERN. Pseudo-rapidity. https://atlas.cern/glossary/pseudo-rapidity. [Online; accessed 20-May-2023].

[16] Gen Tateno. Search for resonances in light-by-light scattering in 14.6 fb^{-1} of pp collisions at $\sqrt{s} = 13$ TeV. 2023. Available from: https://cds.cern.ch/record/2849362.

[17] F. Rosenblatt. The Perceptron, a perceiving and recognizing automaton. Cornell Aeronautical Laboratory, 1957.

[18] THE GENIUS BLOG. What is a perceptron. https://kindsonthegenius.com/blog/what-is-perceptron-how-the-perceptron-works/, 2018. [Online; accessed 20-May-2023].

[19] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Technische Universität Wien, 1989.

[20] Hadley Brooks and Nick Tucker. Electrospinning Predictions using Artificial Neural Networks. In *Polymer*, 2014. Available from: https://doi.org/10.1016/j.polymer.2014.12.046.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.

[22] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. In *Neurocomputing*, 2022. Available from: https://arxiv.org/abs/2109.14545.

[23] ROOT. ROOT. https://root.cern/. [Online; accessed 20-May-2023].

[24] L. A. Harland-Lang, M. Tasevsky, V. A. Khoze, and M. G. Ryskin. A new approach to modelling elastic and inelastic photon-initiated production at the LHC: SuperChic 4. In *EPJ*, 2020. Available from: https://doi.org/10.1140%2Fepjc%2Fs10052-020-08455-0.

[25] Pytorch. Pytorch. https://pytorch.org/. [Online; accessed 20-May-2023].

[26] M Andrews, M Paulini, Sergei Gleyzer, and B Poczos. End-to-End Event Classification of High-Energy Physics Data. In *Journal of Physics: Conference Series*, 2018. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/1085/4/042022/pdf.

[27] Pytorch. Adam optimiser. https://pytorch.org/docs/stable/generated/torch.optim.Adam.html. [Online; accessed 24-May-2023].

[28] Pytorch. Cross-entropy loss. https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html. [Online; accessed 20-May-2023].

[29] S. Gopal Krishna Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. In *IARJSET*, 2015. Available from: https://arxiv.org/abs/1503.06462.

[30] W. J. Youden. Index for Rating Diagnostic Tests. In *Cancer*, 1950. Available from: https://acsjournals.onlinelibrary.wiley.com/doi/epdf/10.1002/1097-0142%281950%293%3A1%3C32%3A%3AAID-CNCR2820030106%3E3.0.CO%3B2-3.

[31] Tom Fawcett. An introduction to ROC analysis. Elsevier Science Inc., 2006.

[32] Lloyd S Shapley. A Value for n-Person Games. Princeton University Press, 1953.

[33] SHAP. Shap Deep explainer. https://shap-lrjball.readthedocs.io/en/latest/generated/shap.DeepExplainer.html, 2018. [Online; accessed 20-May-2023].

[34] Hussain Kitagawa. Optimization of diphoton acoplanarity for an Axion-Like Particle in Light-by-Light scattering with the ATLAS detector at CERN. 2020. Available from: https://cds.cern.ch/record/2742416.

[35] Alpaslan Ersöz and Mehmet Kurban. Bisection Method and Algorithm for Solving The Electrical Circuits. In *IJRASET*, 2013. Available from: https://www.researchgate.net/publication/259849595_Bisection_Method_and_Algorithm_for_Solving_The_Electrical_Circuits.