
Dark matter signals at the LHC with deep learning



UPPSALA
UNIVERSITET

INDEPENDENT PROJECT IN ENGINEERING PHYSICS

MAT-VET-F-23012

Authors:

ANDERSSON Max,
GLÖCKNER Edward,
LÖFKVIST Carl

Supervised by:

ENBERG Rikard,
MORETTI Stefano,
SINGH Prashant,
WALTARI Harri

Teknisk-naturvetenskapliga fakulteten
Uppsala Universitet, Utgivningsort Uppsala
May 24, 2023

Abstract

While holding a firm position in popular culture and science fiction, Dark Matter (DM) is nonetheless a highly relevant topic at the forefront of modern particle physics. We study the applicability of characterizing DM particle candidates SUSY neutralino and sneutrino using Deep Learning (DL) methods. We focus on the monojet and mono-Z signatures and the emergence of missing transverse energy \cancel{E}_T as the result of the undetectable DM candidates. Based on kinematic distributions of outgoing particles as input, a DM candidate classifier is built for each signature, along with a DM mass regressor. The DM candidate classifier obtained near perfect accuracy of 0.995 for the monojet, and 0.978 for mono-z signature. The monojet and mono-Z mass regressors achieved a Mean Absolute Percentage Error (MAPE) of 17.9 % and 8.5 % respectively. Furthermore, we discuss both the shortcomings and simplification that our choice of model implied, as well as an interpretation of the results. Finally, we debate the prospects of DL in the discovery of new physics and its use in experiments.

Contents

1	Introduction	2
1.1	The Standard Model	3
1.2	Dark Matter	5
1.3	Beyond/Extended Standard Model	5
1.3.1	Supersymmetry	5
1.3.2	Minimal Supersymmetric Standard Model	6
1.3.3	Weakly Interactive Massive Particle	6
1.4	Large Hadron Collider	6
1.4.1	Proton-proton collision	7
1.4.2	Monojet signature	7
1.4.3	Mono-Z signature	7
1.4.4	Kinematic distributions	8
1.4.4.1	Pseudo-rapidity	8
1.4.4.2	Transverse Energy and Momentum	8
1.5	Event generation	9
1.5.1	Models	9
1.5.2	Parton and hadronization level	10
1.6	Machine Learning Architecture	10
1.6.1	Gaussian Process	10
1.6.2	Deep Neural Network	10
1.6.3	Convolutional Neural Network	11
2	Method	14
2.1	Model	14
2.2	Datasets	14
2.3	Data generation	16
2.4	Deep learning implementation	16
2.4.1	Implementation of classifier	17
2.4.2	Implementation of regressor	18
2.4.2.1	Monojet regressor	20
2.4.2.2	Mono-Z regressor	21
3	Results and Discussion	23
3.1	Kinematic Distributions	23
3.2	CNN classifier	25
3.2.1	Monojet	26
3.2.2	Mono-Z	28
3.3	CNN regressor	29
3.3.1	Monojet	30
3.3.2	Mono-Z	31
3.4	Sustainability aspect	32
4	Conclusions	33
5	Acknowledgements	33

6	Populärvetenskaplig sammanfattning	34
A	Appendix	37
A.1	Machine learning models	37
A.1.1	Classification CNN	37
A.1.2	Regressor CNN	37
A.1.2.1	Monojet regressor	37
A.1.2.2	Mono-Z regressor	38
A.2	Kinematic distributions	38

Word List

- HEP = High Energy Physics
- LHC = Large Hadron Collider
- $WIMP$ = Weakly-Interacting Massive Particle
- DM = Dark Matter
- SM = Standard Model
- $MSSM$ = Minimal Supersymmetric Standard Model
- BSM = Beyond Standard Model
- $SUSY$ = Super Symmetry
- ML = Machine Learning
- DL = Deep Learning
- DNN = Deep Neural Network
- CNN = Convolutional Neural Network

1 Introduction

Since DM does not scatter light nor interact with ordinary matter, it is essentially invisible. It is therefore no easy task to gain knowledge about it. One way is to extend the Standard Model, SM, into various theories. The SM is the fundamental insight into how our universe is built. It contains elementary particles such as quarks and leptons that interact with the four fundamental forces, the strong force, the weak force, the electromagnetic force, and the gravitational force. Each fundamental force has its corresponding boson. In 2012, the ATLAS and CMS experiments discovered the Higgs boson at the Large Hadron Collider, LHC, at CERN [2, 8]. This marked the completion of the SM as scientists were missing the last component to describe how all of the particles gain their mass. Even though the SM can describe most of our universe with great precision, astronomical phenomena that seem to contradict it has been observed.

To investigate DM, one has to look beyond the SM, BSM. The most popular DM research of today revolves around finding a weakly interacting massive particle, WIMP. WIMP's are a hypothetical group of particles that interact via the gravitational force, but weakly with the other, and thus makes it a candidate for the dark matter particle. The Supersymmetric model paradigm, SUSY, is a supersymmetric hypothetical extension of the SM. In this report, the Minimal Supersymmetric SM, MSSM, is of interest as it introduces the minimal amount of supersymmetric particles to allow for the existence of WIMP's. Their presence in collision events at the LHC can be implicitly detected and analyzed as “missing” transverse energy and momenta [18].

Although producing petabytes of creative commons licensed data, the LHC is yet to produce a single byte of insight into the characteristics of dark matter. Therefore to obtain data upon which a Deep Learning, DL, model can be trained, one can resort to simulation aids. **MadGraph** is one such aid that provides tools for simulating SM and BSM events using Monte Carlo methods. It provides a remarkable ability to adjust many different parameters, as well as tools for analyzing results along with the complementary tool **MadAnalysis**.

This report aims to investigate how DL can be part of the discovery of new physics. From different signatures, monojet, and mono-Z, together with various models with different parameters, there will be an adjustment for the missing transverse energy and momenta, invariant mass, etc. These adjustments can be analyzed and learned by a Convolutional Neural Network, CNN. The end goal is to train models to predict upon which model paradigm and signature the event was generated, as well as the mass of the dark matter produced.

An introduction to the project and the underlying theory is presented in section 1. Section 2 presents the generation of training and validation data and how the DL models were implemented and trained. In section 3 the results are presented and discussed. Finally, in section 4, conclusions are drawn.

1.1 The Standard Model

Hundreds of years of studies on how the universe is built have come down to a few fundamental particles that interact through three of four known fundamental forces. The elementary particles can be divided into two categories, fermions, and bosons. A fermion is a matter particle that obeys the rules of Fermi-Dirac statistics, Pauli's exclusion principle, which states that two fermions can't occupy the same quantum state in the same atom [17]. The fermions are therefore the building blocks of matter. Bosons are force particles, namely force carriers, and make it possible for fermions to interact with one another. Bosons don't have to obey Pauli's exclusion principle which means that they indeed can occupy the same quantum state within the same atom [21].

Figure 1 sorts the elementary particles into fermions and bosons. The fermions are graded from the first to the third generation going from left to right. The first generation is the lightest and what all matter is built with. The higher generations of fermions are less stable and are likely to decay into a more stable state i.e. into a lower generation. From these elementary particles, other particles can be built e.g. a photon.

Without the bosons, fermions would never react and never build any matter in the universe. Each boson corresponds to three different fundamental forces, electromagnetic force, strong force, and weak force. The electromagnetic force is a collection of photon bosons that interacts with every particle that has an electric charge, i.e. all the elementary particles except the neutrinos. The strong force, that the gluon boson is associated with, is the strongest fundamental force and interacts with quarks and every particle that contains quarks. This force holds the quarks together. The weak force, associated with the W and the Z bosons, is the reason for the decay of particles and is the main reason for fusion reactions or that heavy particles decay into a more stable state. The weak force is the only force that interacts with all of the elementary particles and is the only force that neutrinos can sense. The Higgs boson gives all the fermions mass. The only fundamental force that the SM doesn't take into the act is the gravitational force. This force has a very small influence on its strength on sub-atomic particles that it can be neglected [33]. The relative strength and the range of the forces are presented in table 1.

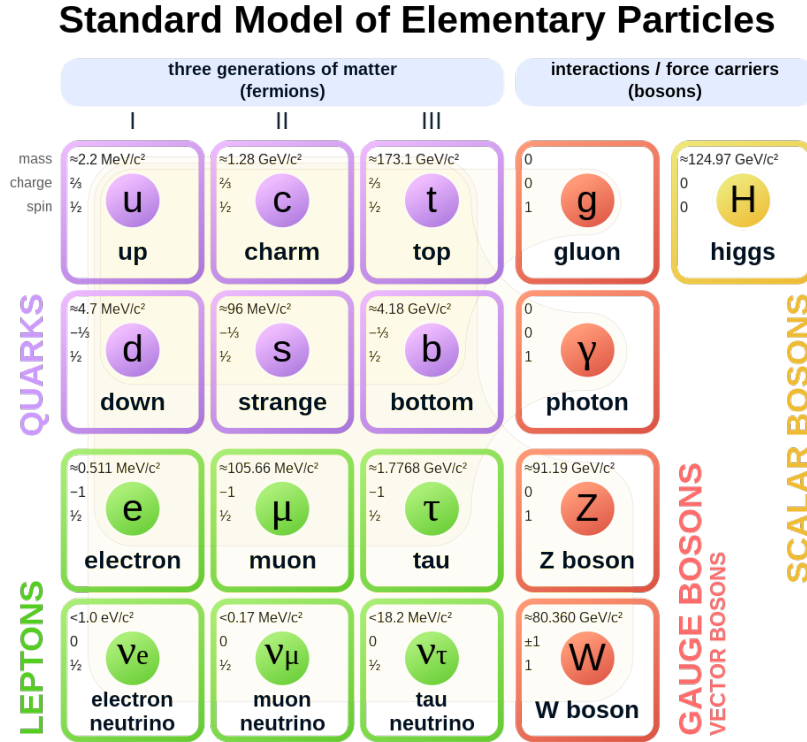


Figure 1: All of the Standard Model elementary particles. Fermions in purple and green. Bosons in red. Scalar Boson Higgs in yellow [25].

Table 1: Fundamental forces with their relative strength based on the strong force felt by proton-proton pair and their range [34].

Fundamental Force	Relative Strength	Range
Gravity	10^{-38}	∞
Weak Force	10^{-13}	$< 10^{-18}\text{m}$
Electromagnetic Force	10^{-2}	∞
Strong Force	1	$< 10^{-15}\text{m}$

The Standard Model equation can predict almost any experiment and phenomenon with very high numerical precision. The compressed lagrangian of the Standard Model can be written as,

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\bar{\psi} \not{D} \psi + h.c. + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. + |D_\mu \phi|^2 - V(\phi) \quad (1)$$

where e.g. the second term describes the interaction between the particles and matter particles. For the interested, it can be read further in [11, 37]. The extended equation is the most important equation of physics as it predicts the behavior of the subatomic world. There are a few situations where the SM lagrangian doesn't work.

As mentioned in section 1 it can't describe the behavior of astronomical phenomena such as the spiral galaxies rotational speed and cosmic radiation.

1.2 Dark Matter

Dark matter, contrary to regular matter, does not interact with the electromagnetic force, making it very difficult to detect. The Swiss-American astronomer Fritz Zwicky is often credited as the dark matter pioneer for his studies conducted in 1933. While studying the redshift of various galaxies from data published by Edwin Hubble and Milton Humason in 1931, Zwicky made notice of the large difference in apparent velocities of the Coma cluster. Although Hubble and Humason already had noted this difference, it was Zwicky's application of the virial theorem that led to his breakthrough. Using the theorem Zwicky found that, under his assumptions, the cluster should exhibit a velocity dispersion decay as $\frac{1}{r}$ when the radius exceeds the border of the cluster. To his disbelief, Zwicky observed little to no decay. It was as if there existed more mass beyond the boundary of the regular mass. From this, he concluded not only the presence of an invisible kind of matter but the fact that it substantially outweighs regular matter.

Since these observations, several experiments have been conducted to prove the existence of a kind of matter that does not interact the traditional way. Although still unfruitful, promising research and discoveries have been done at CERN using the LHC. Dark matter itself would escape unnoticed by the LHC, but scientists say that its existence could implicitly be detected as "missing" energy and momenta that escaped the detectors as dark matter. [7]

The existence of dark matter cannot be explained using the Standard Model (SM) paradigm. Research on models that extend the SM, so-called beyond the SM (BSM) is today at the forefront of modern particle physics research. Many of these propose new paradigms and particles as candidates for the dark matter particle[12].

1.3 Beyond/Extended Standard Model

To allow physics to explain dark matter, the theory has to go Beyond the Standard Model, BSM. Different models are hypothetically developed to test in theories and this case, in colliders. Models beyond the Standard Model can be e.g. Minimal Supersymmetric Standard Model (MSSM), Next-To-Minimal Supersymmetric Standard Model (NMSSM), or string theory. In this report, MSSM is used.

1.3.1 Supersymmetry

Symmetry in terms of physics is where transformations don't affect laws of nature, i.e. it is invariant. The universe is symmetric through translations, rotations, and other fundamental symmetries such as charge, parity, and time. There are violations of some of the symmetries such as CP-violation.

The supersymmetric principle, SUSY, extends the number of symmetries that the universe obeys. This principle is an idea of translational symmetry from one space-time to another. This means the fermions and the bosons have their supersymmetric

partners, which doubles the number of elementary particles. Fermions' supersymmetric partners would have characteristics of a regular boson, and vice versa. The only difference between them would be the spin, as fermions have spin $1/2$ and bosons have either spin 0 or 1. The mass of its partners is still unknown as there is no evidence that they exist, but is expected to be near the electroweak scale [13, 18].

The supersymmetry principle and its superpartners solve hypothetical theories of the unification of the fundamental forces at the beginning of the universe. The electroweak force is a unified force of the electromagnetic force and the weak force. At higher temperatures, each of the coupling constants of the fundamental forces changes and is expected to unite at a certain temperature. This theory is satisfied theoretically if the superpartners exist. Other measurements like the rotational speed of spiral galaxies would be satisfied with the supersymmetric elementary particles.

1.3.2 Minimal Supersymmetric Standard Model

The Minimal Supersymmetric Standard Model, MSSM, is where the SM is extended with its supersymmetric partners and is only applicable during some assumptions to involve dark matter. The assumptions that the MSSM has to obey are the R-parity conservation and that a Higgs doublet is added. The R-parity means that if the lightest supersymmetric partner, LSP, exists after the collision, it has to be stable. This means that it doesn't decay. If the model doesn't obey the R-parity symmetry then it won't be any special rules that determine the decay of the supersymmetric particles of the masses of a small GeV to a few TeV. This means that there wouldn't be any natural dark matter candidates that could fit the theories in section 1.3.1 [13, 18].

1.3.3 Weakly Interactive Massive Particle

The most anticipative LSP is the lightest neutralino. Neutralino is a superposition of the supersymmetric particles photino, zino, and neutral higgsino and the result of this is that the fields form four mass eigenstates that weakly interact with ordinary matter. The lightest neutralino mass is predicted to be around 10 GeV to approximate one TeV. This kind of particle is also a WIMP and is an attractive dark matter candidate in any of its forms [18].

Other particles that serve as dark matter candidates are sneutrino and anti-sneutrino. Sneutrino, with spin 0, is the superpartner of the neutrino in the MSSM. It is a promising dark matter candidate because of its electrical neutral quality which also interacts weakly with other particles.

1.4 Large Hadron Collider

By colliding particles at speeds near the speed of light, heavier particles such as protons can be "split" into their sub-components. This process can reveal information about said particles in a way that is unachievable at lower velocities and energies. The Large Hadron Collider (LHC) is the largest particle accelerator in the world

currently. It accelerates particles using electric fields and uses magnetic fields to bend the particles around the 27-kilometer ring. When the speed of the particles has reached their final velocities they finally collide with each other. [16].

1.4.1 Proton-proton collision

The colliding protons are accelerated to velocities near that of the speed of light, which renders the classical equation of kinetic energy $E = mv^2/2$ inept. One has therefore resort to special relativity with Einstein's equation,

$$E = (\gamma - 1)mc^2,$$

$$\gamma = \frac{1}{\sqrt{1 - \beta^2}},$$

$$\beta = \frac{v}{c},$$

where v is the speed at which the photons travel and m is the rest mass of the photon. With rest mass of $1.672 \cdot 10^{-27}$ kg and velocity $v = 0.999999991c$ m/s, each proton will have a kinetic energy of 7 TeV, also called one beam [23]. In this report, two protons with the center-of-mass energy of $\sqrt{s} = 13$ TeV will be considered.

Protons consist of two up quarks and one down quark and are bounded together by gluons, i.e. the strong force [6]. To annihilate the protons, quarks, and gluons have to interact. To achieve the final dark matter candidates of one collision, the quarks and gluon have to interact through some virtual particles, a mediator. This is usually denoted as a heavier unstable particle that has to decay into SM particles or as dark matter candidates. The mediator can e.g. be a decaying Z boson. From the same process, different signatures can be observed [31].

1.4.2 Monojet signature

When the protons get close to each other, an energetic gluon is exchanged between them which results in a quark separation. When the quark is separated more and more, the gluon field is stretched and will eventually break into a quark and an anti-quark. This process is repeated over and over until it reaches a detector. The result of this is a *jet* full of particles which can be reconstructed using software tools mentioned in section 1.5 [32].

Jets can also be produced from initial-state radiation where the gluons are emitted from the protons. This happens before the collision which leads to the same signature plus the missing transverse energy [4]. This type of jet is produced during the proton-proton collision in this project.

1.4.3 Mono-Z signature

In the process of a mono-Z signature, no jets are produced. Mono-Z signature is observed when a Z boson is created during the annihilation together with dark

matter candidates. The Z boson will decay into SM particles. This will result in the signature plus the missing transverse energy. For the dark matter candidates, there can be the same assumptions as for the monojet signature [26].

1.4.4 Kinematic distributions

The LHC has several detectors and subdetectors to gain information about a proton-proton event. All detector parts provide their piece of the puzzle which is reconstructing an event based on information. By applying a magnetic field across the outgoing particles, their momentum is revealed as the shape of their trajectory, the straighter the higher the momentum. The detectors also allow for accurate measurements of the velocity of the particle, which in combination with its momentum reveals its mass. Finally, detectors have calorimeters which can measure the energy of the particle through absorption [15].

1.4.4.1 Pseudo-rapidity

Pseudo-rapidity, often denoted η , is a common measurement in colliding beam experiments to express the correlation between the angle with respect to the axis of collision and the number of outgoing particles. The x-axis often displays a symmetric interval centered around 0, which is the perpendicular direction of the beams. [28]

1.4.4.2 Transverse Energy and Momentum

As the protons are moving parallel to the beams prior to colliding, they exhibit no transverse momentum. By conservation, transverse momenta should remain zeros after the collision. Therefore, any non-zero results in transverse momenta are the result of particles that have gone undetected by the detectors. By measuring energy and momentum, two highly correlated kinematic quantities, in the transverse plane, one can gain indirect information of the undetected particles.

Missing transverse energy, often denoted \cancel{E}_T , is an important measurement in collision experiments, especially dark matter focused ones. \cancel{E}_T emerges as the result of particles that leave the detector undetected. The LHC has specific detectors that can detect muons, which interact weakly with matter, whereas particles such as the neutrino and DM are largely invisible. Their kinematics therefore emerge in the quantity \cancel{E}_T .

However, to train a DL model, further information is needed about the outgoing particles of interest. The total transverse energy, denoted E_T , is the sum of the transverse momentum \mathbf{p}_T of all outgoing particles. In terms of the information conveyed to a DL model, \cancel{E}_T and E_T are thus interchangeable being each others complement. The quantity M_T , called the transverse mass, is a useful quantity in the case of the mono-Z because two particles are created, namely the positive and negative lepton. In this two particle system, M_T is defined as

$$M_T^2 = (E_{T,l+} + E_{T,l-})^2 - (p_{T,l+} + p_{T,l-})^2, \quad (2)$$

1.5 Event generation

As previously mentioned in sections 1.2 and 1.3, the search for a dark matter candidate is in the realms beyond what the SM provides. To conduct consistent experiments at the very limit of known physics at a large scale, simulation tools are used. Monte Carlo methods have proven useful in capturing the quantum mechanical variability that causes the differing final multi-particle states. By simulating a large number of events under some predefined condition, one can perform accurate calculations. High accuracy measurements are essential when measuring \cancel{E}_T and distinguishing the possible weak interaction of WIMP's from the SM background.[20]

The simulation aid of choice for this study is the popular framework **MadGraph**. **MadGraph** provides a set of tools that enables the generation of hard events, calculation of cross sections, and a plethora of tools for analyzing and manipulating events. The software can be extended to enable random event generation, parton showering and hadronization using the **Pythia** program. **Delphes** extension allow for detector simulation of isolated leptons, MET calculations, and collections of jets, all of which are of interest in this project. [35] [1]. Developed at CERN, **ROOT** provides tools for efficiently analyzing large amounts of data, which is useful when simulating a large number of events. [29] [22]

In order to make sense of the large amounts of event data produced by **MadGraph**, a tool for analysis is required. **MadAnalysis** is a tool that allows for efficient and sophisticated physics analyses on parton levels on Les Houches Event Files (**.lhe**), such as those produced by **MadGraph** [24]. In the case of this project, the arrays containing data of kinematic distributions are of interest.

1.5.1 Models

One great advantage of MC simulation tools is the ability to alter the features of particles to test theoretical BSM's. One such discussed in 1.3.2 is the MSSM. **MadGraph** allows for efficient implementation of models and event simulation using its included particles and paradigms. In the case of this project, multiparticle final states are calculated from proton-proton collisions using the MSSM. In this project, we study two dark matter candidates; the neutralino and the sneutrino. We therefore simulate events where a proton-proton collision generates two neutralinos and a jet, as well as where the collision generates a sneutrino and anti-sneutrino pair along with a jet. With the use of **MadAnalysis**, the output of the events is used to train a machine learning model (see section 1.6).

The theoretical models for MSSM with mono-z signature are similar to the monojets. One model represents proton-proton collision which generates two neutralinos with a Z boson. The other model generates one sneutrino and one anti-sneutrino together with the Z boson. For both models, the Z boson decays into a positively charged lepton and one negatively charged lepton.

1.5.2 Parton and hadronization level

The generation of events can be simulated through different levels. Parton and hadronization levels represent two different approaches to the generation. After the high-energy collisions when partons interact with each other, the proton containing partons separates. From known experiments, partons alone such as quarks and gluons will reform into color-neutral hadrons due to the strong force. This can be done by hadronization shower and detection from `Pythia` and `Delphes` to construct more realistic events. Since the parton and hadronization levels are different from each other, they will produce slightly different kinematic distributions.

1.6 Machine Learning Architecture

Machine learning has become widely popular in the recent decade, where deep learning is leading the evolution of artificial intelligence. Deep learning has been incorporated in various applications and disciplines, and more recently in searches for new physics at the LHC [5]. In this section the architecture of the machine learning algorithms implemented are explained, as well as the necessary background in order to understand these models.

1.6.1 Gaussian Process

Benchmark etc

1.6.2 Deep Neural Network

In this report various sorts of convolutional neural networks (CNN's) are implemented to establish whether deep supervised learning techniques can be utilized to disentangle DM signals at the LHC or not. The CNN architecture is a subset of the broader deep learning technique, most commonly applied to analyze structured data, e.g. images. CNN and most other relevant machine learning algorithms can further be narrowed down to the subfield of Deep Neural Networks (DNN) [19].

Most Neural Networks (NN) are made up of many connected neurons which receive inputs, or activations, from other neurons. The input neurons of the neural network are activated by perceived information of the real-world environment, while other neurons get activated by weighted connections to the previously activated neurons. Sets of neurons at the same level of activation are denoted as a layer, where the input layer consists of the input neurons. The following layers are referred to as hidden layers, and the final layer is called the output layer which produces the prediction or output of the network [30].

Each layer of a DNN typically has a set of parameters, which in the case of supervised learning, are learned by training the model through minimizing a loss function. Backpropagation is the key to supervised learning, where the gradients of the loss function to the network parameters are calculated. The parameters are then updated using an optimization algorithm such as gradient descent in order to minimize the training error [30]. A visualization of the architecture of a DNN is depicted in figure (2).

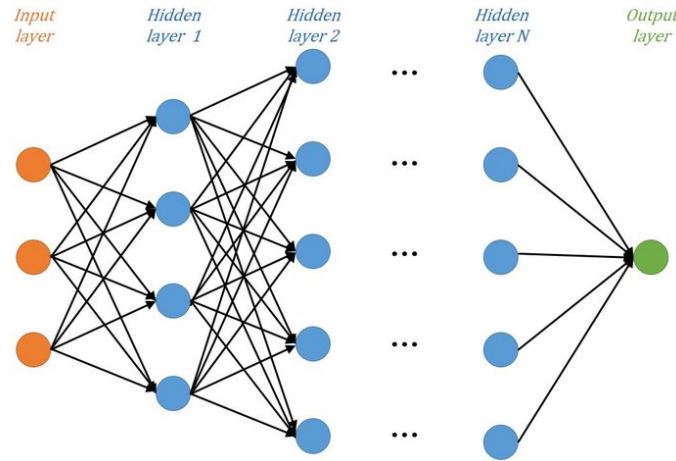


Figure 2: A visualization of a deep neural network, consisting of an input layer, multiple hidden layers and an output layer. [9].

1.6.3 Convolutional Neural Network

Convolutional neural networks are a subset of DNN:s, used for processing structured data such as images. CNN:s utilize three different types of layers, namely: convolutional layers, pooling layers, and fully connected or dense layers. The convolutional and pooling layers are used to extract features of the data, and the fully connected layer maps the extracted features into the final output [38].

The input of a CNN is represented as a tensor most commonly with a dimension of two, but the dimension can vary. In the convolutional layer a so-called kernel, a small array of numbers, is applied across the layers' input. The output of this operation is referred to as a feature map, and this process is repeated using multiple different kernels to form a number of feature maps, each representing a different characteristic of the input tensor. Two important hyperparameters of such an operation are size and number of kernels [38]. A visualization of the operation can be seen in figure 3.

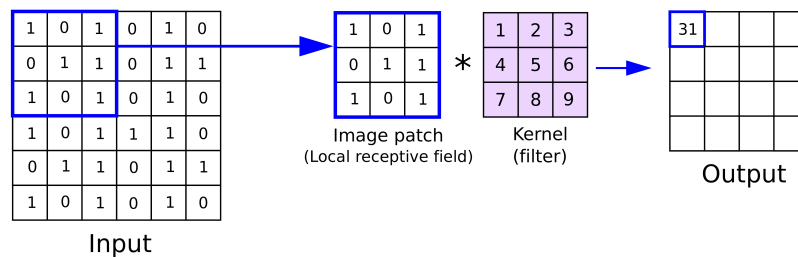


Figure 3: Visualization of a convolution [36].

The pooling layer works as a downsampling operation, which purpose is to reduce the dimensionality of the feature maps produced by the convolutional layers. Some important hyperparameters of the pooling layer are filter size, stride, and padding. There are many variations of pooling operations, and the two most popular ones are

max pooling and average pooling. Max pooling works by extracting a patch from the input feature map, and outputs the maximum value of each patch. Average pooling, on the other hand, works by outputting the average of each patch [38]. These operations are visualized in figure 4.

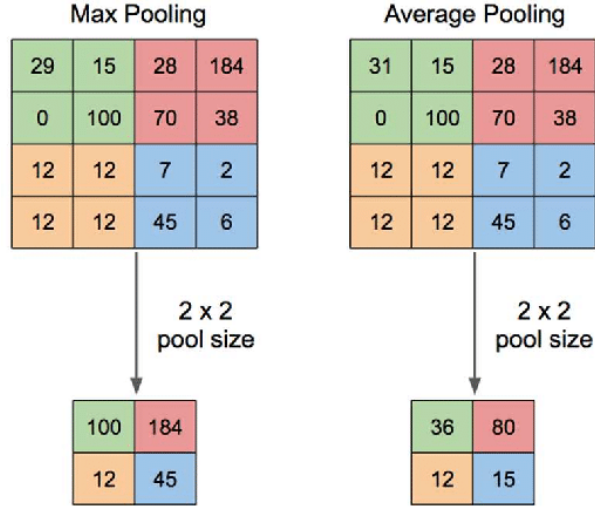


Figure 4: Visualization of max pooling and average pooling [39].

The output feature map of the final convolutional or pooling layer is then flattened, meaning transformed into a 1D array. This is then connected to one or multiple dense layers which weights are learnable through minimizing a loss function through back-propagation and optimization [38]. An example of the full architecture of a CNN model is illustrated in figure 5.

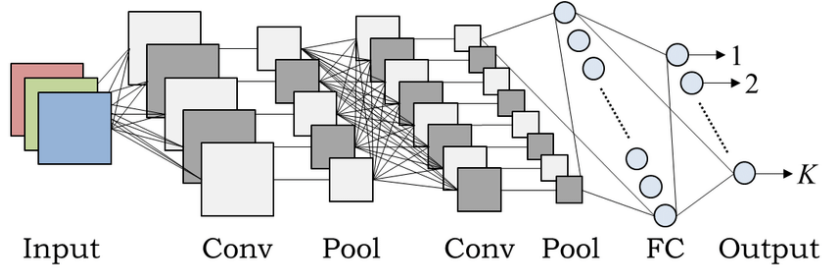


Figure 5: Example of a full CNN architecture. *Conv* references a convolutional layer, *Pool* references a pooling layer and *FC* represents a fully connected or dense layer[14].

Table 2 lists the parameters and hyperparameters of a CNN model.

	Parameters	Hyperparameters
Convolutional layer	Kernels	Kernel size, number of kernels, stride, padding, activation function
Pooling layer	N/A	Pooling method, filter size, stride, padding
Dense layer	Weights	Number of weights, activation function
Others	N/A	Model architecture, optimizer, learning rate, loss function, mini-batch size, epochs, regularization, weight initialization, dataset splitting, etc.

Table 2: Parameters and hyperparameters of a CNN model[38].

2 Method

This section goes over the methodology used in this project. The section covers the structure of the generated data sets, as well as the confinement that has been applied to the data. Finally, the section features a description of the implementation of the DM candidate classifiers and mass regressors using the Python programming language and Tensorflow.

2.1 Model

MSSM is used in this project in order to simulate proton-proton collisions to produce different dark matter candidates. Since this project is limited to solely varying one parameter, MSSM is therefore a suitable model which is simplified to a level with only a few parameters. Most of the parameters are fixed such as couplings and the Higgs mass which is set to its experimental value. The only flexible parameter in the model is set to the mass in order to generate different kinematic distributions with the span of $m_{DM} \in [150, 1000]$.

The tools of simulation applied in this report were built on a foundation of experience from real experiments at the LHC. In order to make the simulation results more realistic, a trigger level for the missing transverse momentum is applied. The trigger levels of the \cancel{E}_T in this report are set at 150 GeV for the lower boundary and 500 GeV for the upper (see figure 10), meaning that other energies are rejected. This is justified by the use of these very thresholds by the ATLAS project [3] at CERN.

2.2 Datasets

In order to train and validate the models, four sufficiently large and statistically sound datasets were constructed from four different categories, namely

1. monojet signature with neutralino pair, spin-1/2, as DM candidates,
2. monojet signature with sneutrino and anti-sneutrino pair, spin-0, as DM candidates,
3. mono-z signature with neutralino pair as DM candidates and
4. mono-z signature with sneutrino and anti-sneutrino pair as DM candidates.

The category correlates to a dataset of one-dimensional arrays of 30 bins, for categories 1 and 2, and 40 bins for categories 3 and 4, containing data of the kinematic distributions specified below. In addition to that, a separate dataset is generated by **MadAnalysis** containing histograms generated using those very arrays. These sets were not used to train a DL model in this project. Each dataset consists of 1,500 data points, each one initialized using a different mass of the DM candidate (sneutrino/neutralino) on the interval 150 - 1000 GeV. Each histogram is of size 646 by 420 pixels. The kinematics distribution of interest depends on the signature being observed. The monojet signature datasets thus contain

1. E_T ,

2. η of the jet,
3. mass of the DM candidate and
4. a class.

The mono-z signature datasets contains

1. η of the positively charged lepton,
2. p_T of the positively charged lepton,
3. E_T ,
4. M_T ,
5. mass of the DM candidate and
6. a class.

For the categorical classification model, the class will act as a label, whereas the mass of the DM particle will be the target when training the regression model.

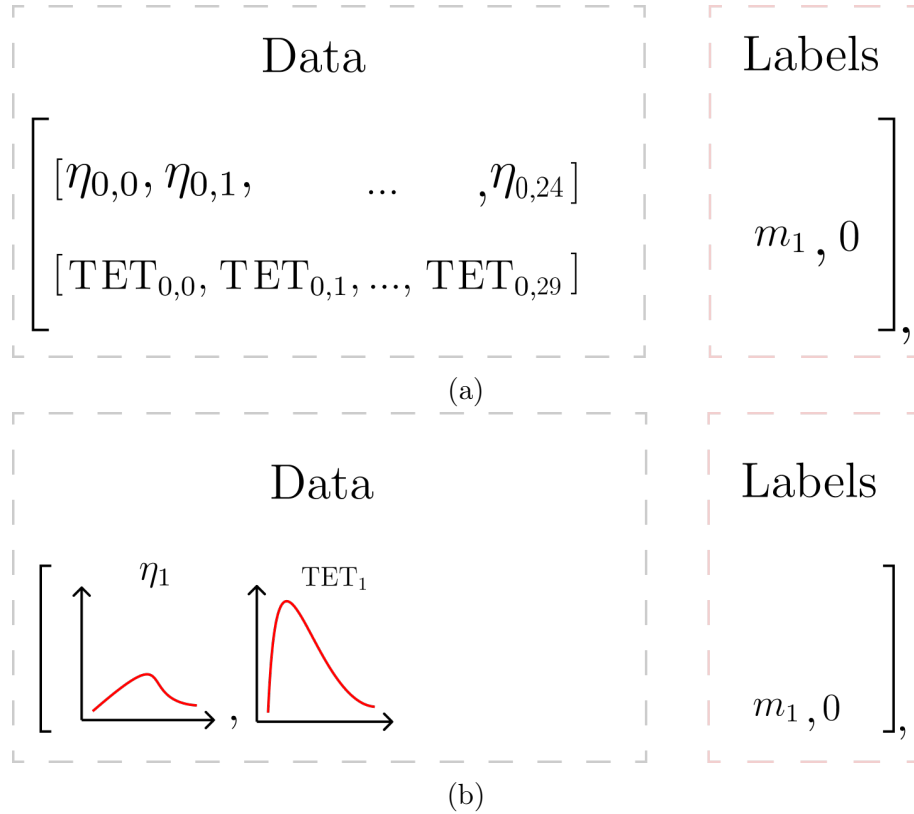


Figure 6: An example of a data point using mass m_1 , for the model indexed 0, for **a)** the array dataset, and **b)** the histogram dataset.

The datasets were then split into a training set, a validation set, and a testing set with a ratio of 0.64 : 0.16 : 0.20. The training and validation set was used to train and tune the CNN models, and the testing set was used as a final evaluation of the model performance.

2.3 Data generation

A data point is generated by **MadGraph** and **MadAnalysis** (see section 1.5) through a series of commands, each one responsible for initializing the event with the correct preconditions for that specific event. Most important of which determines the in- and outgoing particles of the collision. Listed in figure 3 are the commands that correlate to each category of the event.

The particles that reside to the left of the "greater than" symbol are ingoing particles, whereas the right ones are outgoing ones. A comma represents a sub-process taking place, which in this case is the decay of outgoing particles. Each point in a dataset is generated by averaging the output from 10,000 collisions.

Category	MadGraph generation command
Monojet neutralino	p p > n1 n1 j
Monojet sneutrino	p p > sve sve~ j
Mono-z neutralino	p p > n1 n1 z, z > l+ l-
Mono-z sneutrino	p p > sve sve~ z, z > l+ l-

Table 3: Process category and generating command in *MadGraph*. Figure 4 features a symbol explanation

Symbol	Particle
p	Proton
n1	Neutralino
j	Jet
sve (sve~)	Sneutrino (antisneutrino)
z	Z boson
l(±)	Lepton (pos./neg. charge)

Table 4: Explanation of symbols in *MadGraph*

To greatly reduce computational time, data generation was performed on UPPMAX (*Uppsala Multidisciplinary Center for Advanced Computational Science*), Uppsala University's resource of high-performance computers.

2.4 Deep learning implementation

The DM candidate classifier and the mass regressor were implemented in the Python programming language. The models were constructed using Tensorflow (TF), an open-source machine learning platform, together with the Keras API. Figure 7 displays a broad overview of how the full deep learning architecture could look in production, where the histograms together with the predicted signature by the classifier get passed as input into the regressor model. However, note that all four

models are trained and tested independently from one another in order to avoid errors propagating through the network.

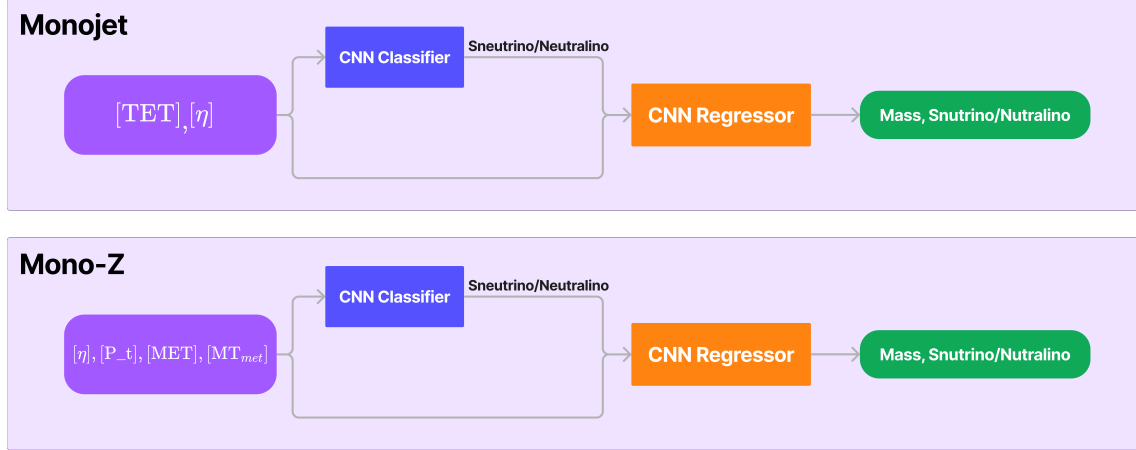


Figure 7: Layout of how the full deep learning architecture would look like in production.

Section 2.4.1 and 2.4.2 elaborates further on the implementation of each model. In order to confine this project in terms of size and time, only models using the one dimensional arrays as inputs are trained.

2.4.1 Implementation of classifier

Two CNN's were built to classify **a)** monojet signatures with sneutrino or neutralino as DM candidate, and **b)** mono-z signatures with sneutrino or neutralino. For the monojet classifier, the two arrays of the kinematic distributions of size $(30, 1)$ are concatenated to a $(2, 30, 1)$ tensor, and similarly the four arrays in the mono-z classifier are concatenated to a $(4, 40, 1)$ tensor. The classifier CNN consists of two **Conv1D** layers with a rectified linear unit (ReLU) activation function each followed by a **MaxPooling1D** layer for downsampling. A **BatchNormalization** layer is included for regularization. The output is then flattened and passed through two **Dense** layers with ReLU activation function, followed by a final **Dense** layer with softmax activation for classification. The output is the predicted DM candidate. Table 5 displays the architecture of the CNN model together with the layer hyperparameters, and table 6 displays the hyperparameters of the compilation and training of the model. The total number of parameters for the two classification models can be found in table 7, and each layer with the input shape and the output shape can be found in the appendix, see 16.

Layer	Hyperparameters	Activation function
Conv1D	Filters: 32, Kernel size: 3, Padding: Valid	ReLU
MaxPooling1D	Pool size: 2	N/A
Conv1D	Filters: 32, Kernel size: 3, Padding: Valid	ReLU
BatchNormalization	N/A	N/A
MaxPooling1D	Pool size: 2	N/A
Flatten	N/A	N/A
Dense	Units: 32	ReLU
Dense	Units: 8	ReLU
Dense	Units: 2	Softmax

Table 5: Architecture of the classifier CNN model and hyperparameters of each layer.

Compilation and training	
Optimizer	adam
Learning rate	0.000027
Loss function	Categorical Crossentropy
Metric	Accuracy, Recall, AUC
Training size	1916
Validation size	479
Testing size	600
Batch Size	32

Table 6: Hyperparameters of the compilation and training of the classifier CNN models. AUC stands for *Area under the ROC Curve*.

Total parameters of architecture	
Monojet classifier	9914
Mono-Z classifier	12154

Table 7: Total parameters for respectively CNN classifier model.

The models were trained on a maximum of 1000 epochs with early stopping used as a regularization technique using patience of 2.

2.4.2 Implementation of regressor

Two CNN regressors, one for each signature, were built in order to predict the mass of the DM candidate. For the monojet regressor, the model takes the same (2, 30, 1) tensor as input and passes it through convolutional and pooling layers

in order to extract features and reduce spatial dimensions, and similarly using a $(4, 40, 1)$ tensor for the mono-Z regressor. Once the input has been passed through a flattened layer, an encoded signature of the DM candidate (sneutrino/neutralino) is concatenated with the array. The array is then passed through several dense and batch normalization layers before outputting a predicted mass of the DM particle.

The idea is that as visualized in figure 7 the classifier and regressor for one signature can be serialized together. This works by the classifier first predicting the DM candidate using the input arrays, and that information together with the arrays allows the regressor to predict the mass of the DM candidate. The two models could thus work co-dependently as one deep learning unit.

Since the mono-Z data with four kinematic distributions is more complex compared to monojet data, the architecture of the two regressor CNN's were built differently in order to capture and learn the features and relationships specific to each dataset. Figure 8 illustrates the architecture of the regressor CNN models.

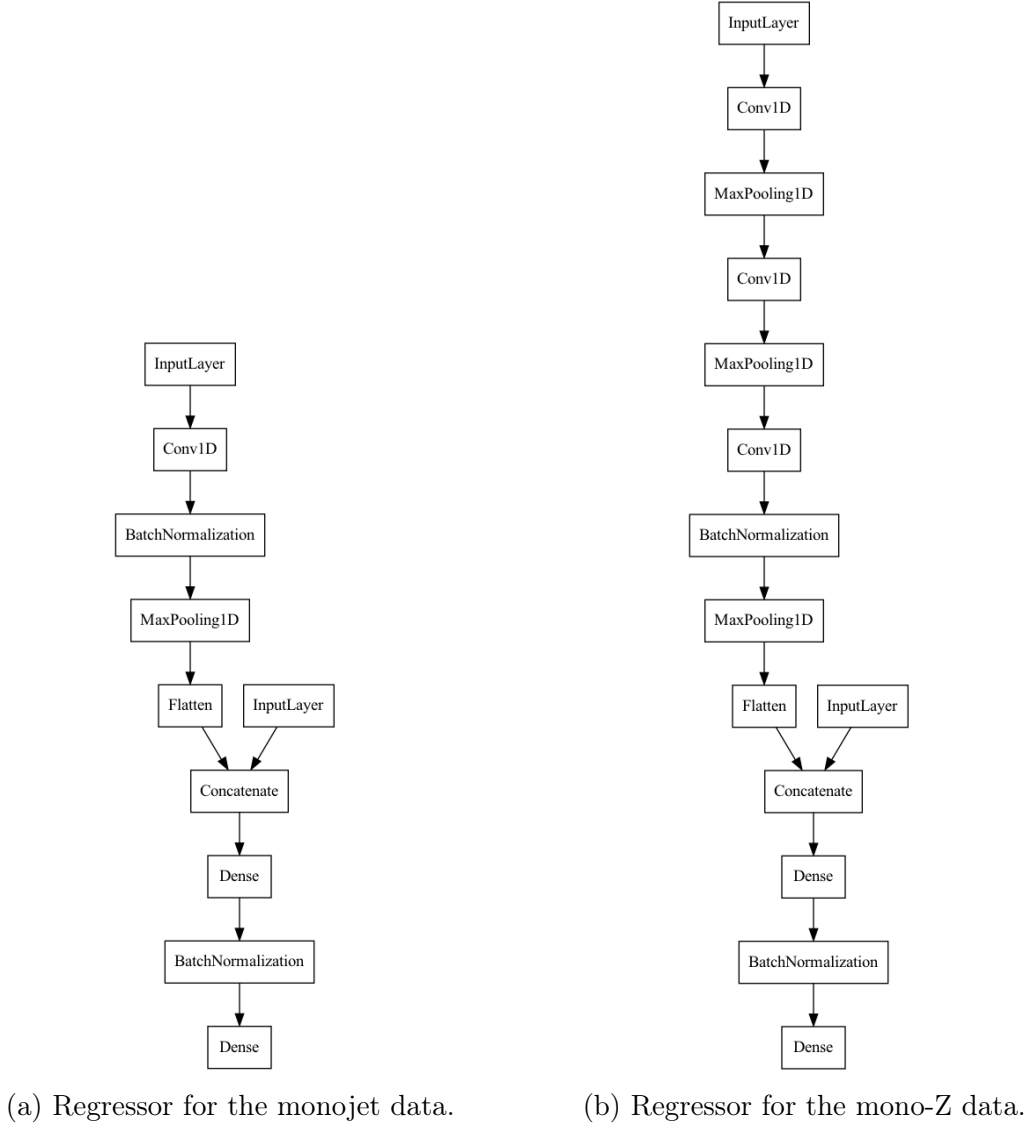


Figure 8: Architecture of the regressor CNN models.

2.4.2.1 Monojet regressor

The regressor CNN for the monojet data consists of one **Conv1D** layer with **ReLU** activation followed by a **BatchNormalization** and **MaxPooling1D** layer. The output is then flattened, concatenated with the encoded DM candidate data and passed through one **Dense** layer with **ReLU** activation followed by a final **Dense** layer with linear activation, giving the predicted mass of the DM candidate. Table 8 displays the architecture of the CNN model together with layer hyperparameters, and table 9 displays the hyperparameters of the compilation and training of the model. The number of parameters for the monojet regressor model can be found in table 10, and each layer with the input shape and the output shape can be found in the appendix, see 17.

Layer	Hyperparameters	Activation function
Conv1D	Filters: 10, Kernel size: 3, Padding: Valid	ReLU
BatchNormalization	N/A	N/A
MaxPooling1D	Pool size: 3	N/A
Flatten	N/A	N/A
Concatenated	N/A	N/A
Dense	Units: 16	ReLU
BatchNormalization	N/A	N/A
Dense	Units: 1	Linear

Table 8: Architecture of the regressor CNN model for the monojet data, and the hyperparameters of each layer.

Compilation and training	
Optimizer	adam
Learning rate	0.00007
Loss function	Mean squared error
Metric	MAE, MAPE, RMSE
Training size	1916
Validation size	479
Testing size	600
Batch Size	32

Table 9: Hyperparameters of the compilation and training of the regressor CNN for the monojet data. MAE stands for *mean absolute error*, MAPE stands for *mean absolute percentage error* and RMSE stands for *root mean squared error*.

Total parameters of architecture	
Monojet regressor	1663

Table 10: Total parameters for the monojet regressor.

Similarly as for the classification models the regressors were also trained using early stopping as a regularization technique. However, due to the inherent complexity of the regression task, predicting masses at a continuous spectrum, more training was required to learn the patterns of the data. To account for this, the max number of epochs was chosen to be 2000 using patience of 80.

2.4.2.2 Mono-Z regressor

The regressor CNN for the mono-Z data consists of a total of three **Conv1D** and **MaxPooling1D** layers, together with a **BatchNormalization** layer. The reason for using more convolutional layers compared to the monojet data is due to the higher complexity and dimension of the input data. The output is then flattened and concatenated with the encoded DM candidate data, before passing through two **Dense** layers with the same activation functions as for the monojet case, together with another **BatchNormalization** layer. Table 11 displays the architecture of the CNN model together with layer hyperparameters, and table 12 displays the hyperparameters of the compilation and training of the model. The total number of parameters for the mono-Z regressor can be found in table 13, and each layer with the input shape and the output shape can be found in the appendix, see 18.

Layer	Hyperparameters	Activation function
Conv1D	Filters: 32, Kernel size: 3, Padding: Valid	ReLU
MaxPooling1D	Pool size: 2	N/A
Conv1D	Filters: 12, Kernel size: 3, Padding: Valid	ReLU
MaxPooling1D	Pool size: 2	N/A
Conv1D	Filters: 8, Kernel size: 3, Padding: Valid	ReLU
BatchNormalization	N/A	N/A
MaxPooling1D	Pool size: 2	N/A
Flatten	N/A	N/A
Concatenated	N/A	N/A
Dense	Units: 16	ReLU
BatchNormalization	N/A	N/A
Dense	Units: 1	Linear

Table 11: Architecture of the regressor CNN model for the mono-Z data, and the hyperparameters of each layer.

Compilation and training	
Optimizer	adam
Learning rate	0.00007
Loss function	Mean squared error
Metric	MAE, MAPE, RMSE
Training size	1916
Validation size	479
Testing size	600
Batch Size	32

Table 12: Hyperparameters of the compilation and training of the regressor CNN for the mono-Z data.

Total parameters of architecture	
Mono-Z regressor	2405

Table 13: Total parameters for the mono-Z regressor.

As for the monojet data, the model were trained on a maximum of 2000 epochs, with patience 80.

3 Results and Discussion

In this section, the results of the project are presented. This includes parton level event generation results from **MadGraph** and **MadAnalysis**, as well as the performance of the classifier and regression models.

The scope of this project is limited to exclusively analyze parton levels, using a simplified MSSM, and use of a small-scale dataset. On the part of physics, this makes the results slightly different from the realistic experiments produced at the LHC. Even if the model is simplified, it represents a real model with results corresponding to real-time events. Because of the time limitation of this project, the simplification had to be done. The simulation aids of choice provide a plethora of parameters to adjust or introduce to experiment with new conditions and new physics. Of the many available, the mass of the DM candidate was chosen as it was believed to be the most interesting and have the most effect on the kinematic distributions. Other parameters, such as coupling constants, spin and mass of mediator were consciously ignored to stay within the time frame of this project.

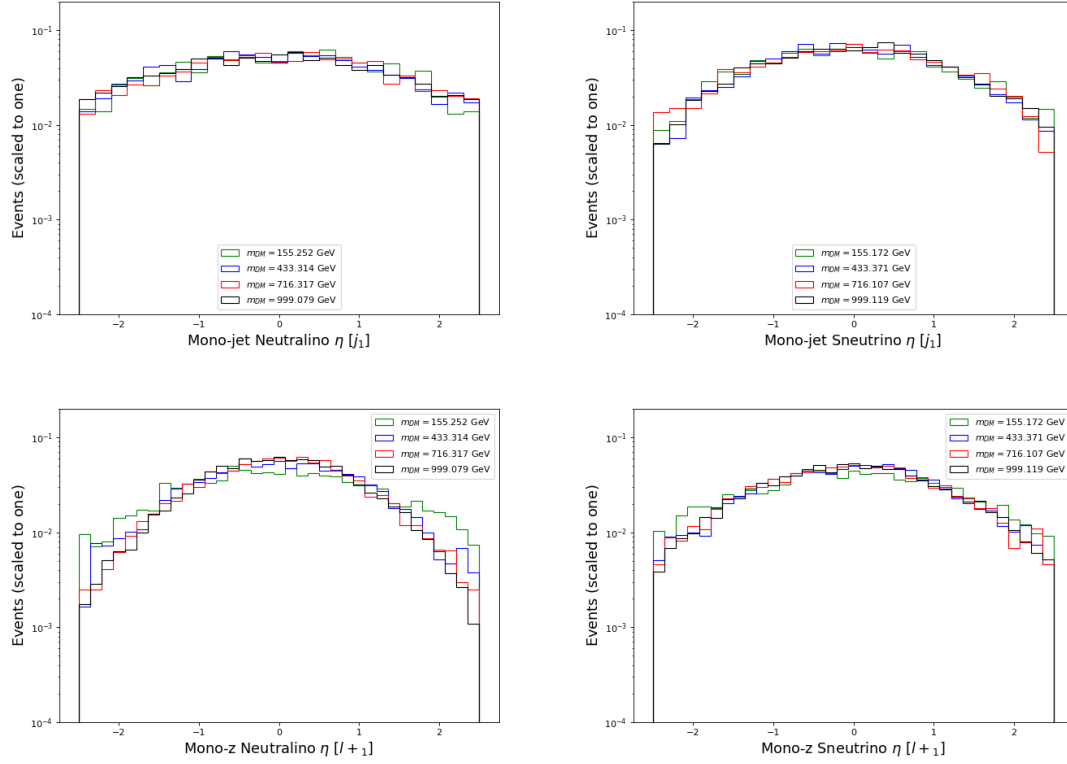
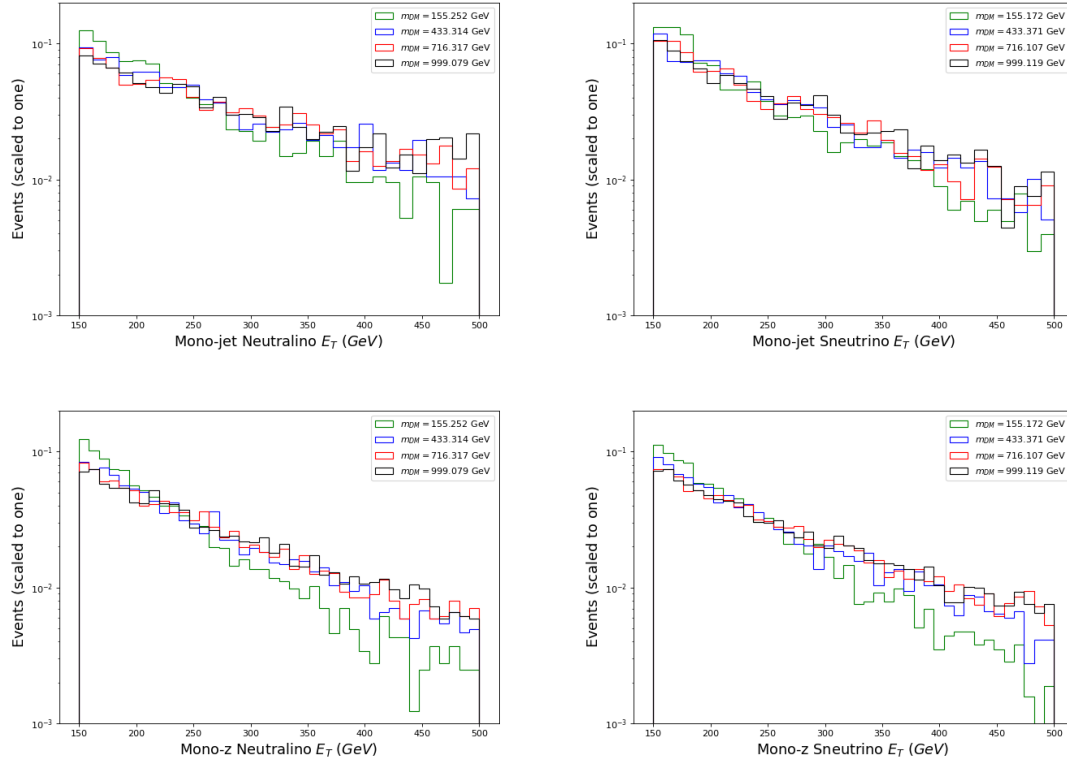
3.1 Kinematic Distributions

MadAnalysis enables the conversion of **.lhe** files to presentable and comprehensive normalized arrays of the kinematic distributions. Presented in figure 9 are the η distributions for both signatures and both DM candidates for varying masses of the candidate. Similarly, figure 10 presents the E_T distribution for all categories with a varying mass of the DM candidate.

Each histogram presented in the figures below represents different dark matter candidates with distinct signatures, which naturally leads to variations in the observed outcomes. Sneutrinos and neutralinos possess different spin properties. As mentioned in Section 2, sneutrinos and anti-sneutrino pairs have spin-0, while neutralino pairs have spin-1/2.

In the context of the monojet events, the high-energy jet originates from the initial state protons, and the overall event characteristics are determined by the combination of the jet and the energy of the transverse momentum. In Figure 10, the distribution of the total transverse momentum is depicted. It can be observed that there is not a significant difference in the distribution for different masses. This can be attributed to the fact that the jet originates from the initial state, leading to similar characteristics across different mass scenarios.

In contrast, in the case of the mono-Z signature, where the Z boson decays after the collision, distinguishable shape differences can be observed for masses up to a few hundred GeV. This is due to the varying decay processes and kinematics involved in the mono-Z signature, compared to the monojet case.

Figure 9: η distribution for all categoriesFigure 10: E_T distribution for all categories

3.2 CNN classifier

The results of the CNN classifiers are summarized in table 14.

Signature	Accuracy	AUC	Recall
Mono-jet	0.995	0.999	0.995
Mono-z	0.978	0.998	0.978

Table 14: Performance of particle classifiers

Based on the findings in Table 14, we can conclude that the trained 1D CNN effectively classifies the various models described in Section 2. The analysis of the kinematic distribution shown in Figure 10, 19, 20 reveals that there are minimal differences between the models. We therefore believe that the CNN assigns greater importance, more weights, to the ETA distribution depicted in Figure 9, where discernible distinctions between the models can be observed. Specifically, the monojet neutralino distribution appears flatter for all masses, while the monojet sneutrino distribution exhibits a dome-like shape and contains fewer normalized events at the endpoints and a higher peak in the center compared to the monojet neutralino ETA distribution. The same is noticeable for the mono-Z case where mono-Z neutralino is more dome-shaped and contains a larger peak than mono-Z sneutrino.

Another interesting observation pertains to the disparity in the number of events at 150 GeV in the E_T distribution of the monojet models. In the case of monojet neutralino, nearly all of the plotted distributions at 150 GeV fall below 10^{-1} events. Conversely, in monojet sneutrino, the amount of normalized event at 150 GeV consistently reaches or exceeds 10^{-1} events. Although the overall shapes may appear similar, we consider that the model primarily assigns weights to the parameters that highlight the smallest noticeable differences for each model, enabling it to make predictions based on those distinctions.

The outstanding performance of the classifier at the parton level suggests that deep learning is a valuable tool for distinguishing between different theoretical models, including dark matter candidates, in proton-proton collisions. However, to gain a deeper understanding of the applicability of deep learning to real events at the LHC, further research must be conducted at the hadronization level. The kinematic distributions obtained at hadronization level would differ significantly from those at the parton level, as the hadronization process involves the rearrangement of partons into color-neutral hadrons due to the strong force, leading to variations in their energies. Hence, it is essential to evaluate the models when applied to parton level data before considering their applicability to hadronization level. If deep learning proves ineffective at parton level, its utility for hadronization level would be questionable.

Many different architectures for the CNN classifiers were tested and compared to each other in terms of performance. Ultimately a model with two `Conv1D` and `MaxPooling1D` consecutive layers was chosen, followed by three `Dense` layers. The classifiers were not as sensitive to the learning rate compared to the regression models, see section 3.3, but the learning rate had to be relatively low in order to

achieve a smooth convergence to a global minimum. This gives multiple insights into the CNN classifiers as well as the data used for training. Since complex and noisy data can introduce fluctuations in gradients it can become more difficult for the model to find a stable solution. Complex data distributions and relationships may also lead to the model exploring more complicated optimization landscapes which demands a lower learning rate in order to achieve a smooth convergence.

Early stopping was used as a regularization technique with a maximum number of epochs set to 1000 and using a patience of 2 in order to prevent overfitting, thus improving the model's ability to generalize to unseen data. In the classifier case the patience was set to be relatively low, with a value of 2 since the validation loss converged smoothly without discrepancy as in the regression case.

3.2.1 Monojet

The ROC curve and the convergence of the training and validation losses, for both monojet in Figure 11 and 13 and mono-Z in Figure 14 and 16, provide additional evidence of the outstanding performance of the respectively trained CNN models. The ROC, with an AUC of approximately 1 in both models, signifies that the models achieve an excellent balance between true positive rate and false positive rate, indicating accurate classification.

The convergence of the training and validation losses toward zero reinforces the robustness and accuracy of the trained models. The decreasing values during each epoch of both training and validation losses indicate that the models are consistently improving their performance and minimizing their errors, as validated by the results in Table 14. Notably, the rate of loss reduction differs between the models of monojet and mono-Z. In the initial epochs, the monojet classifier improves at approximately twice the rate compared to the mono-Z classifier. This can be attributed to the complexity of the mono-Z classifier model and its increased number of input channels relative to the monojet classifier. Overall, both models converge within a similar amount of epochs, despite the implementation of an adaptive early stopping during the training process. The monojet classifier exhibits a more gradual convergence, while the mono-Z classifier has a greater capability for rapid improvement towards the later epochs as it contains more parameters. One factor that may explain some of the discrepancies in the figures could be the relatively small size of the training dataset. This could lead to the model not generalizing or learning the underlying patterns. A lower learning rate can help prevent overfitting the small dataset, and thus improve the model's ability to converge smoothly.

The combination of a nearly perfect ROC curve and the convergence of training and validation losses toward zero confirms the two model's high precision and reliability. These results provide strong evidence that the trained deep learning model, CNN, can be used to distinguish theoretical models at parton level including dark matter candidates in proton-proton collisions.

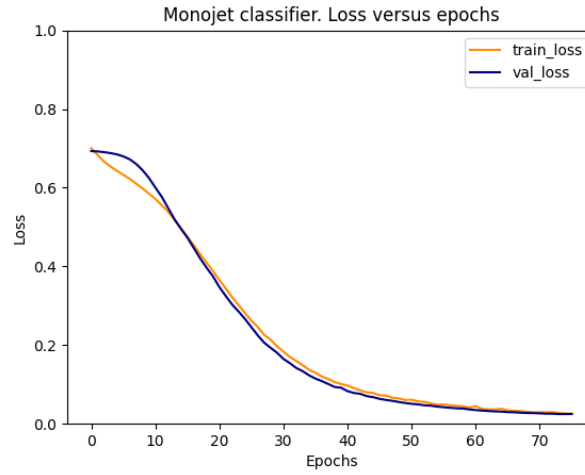


Figure 11: Loss function (categorical crossentropy) of the model 5 training on the monojet data, versus number of epochs.

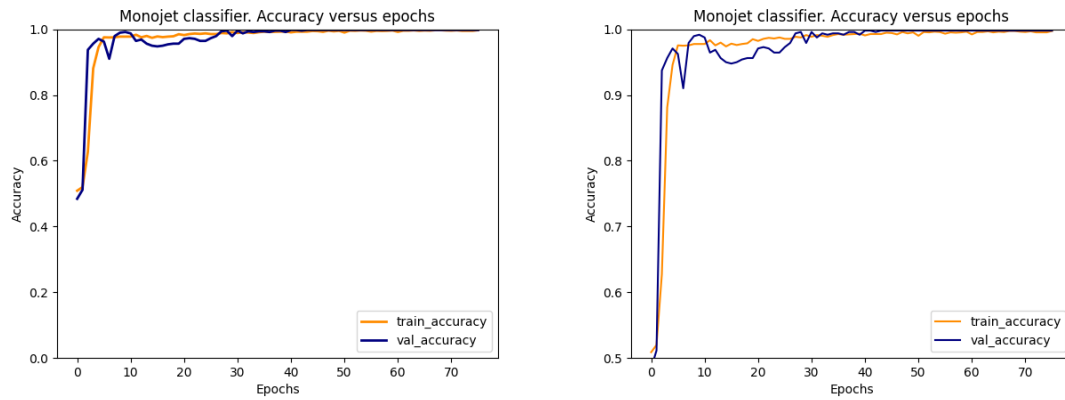


Figure 12: Accuracy of the model 5 training on the monojet data, versus number of epochs.

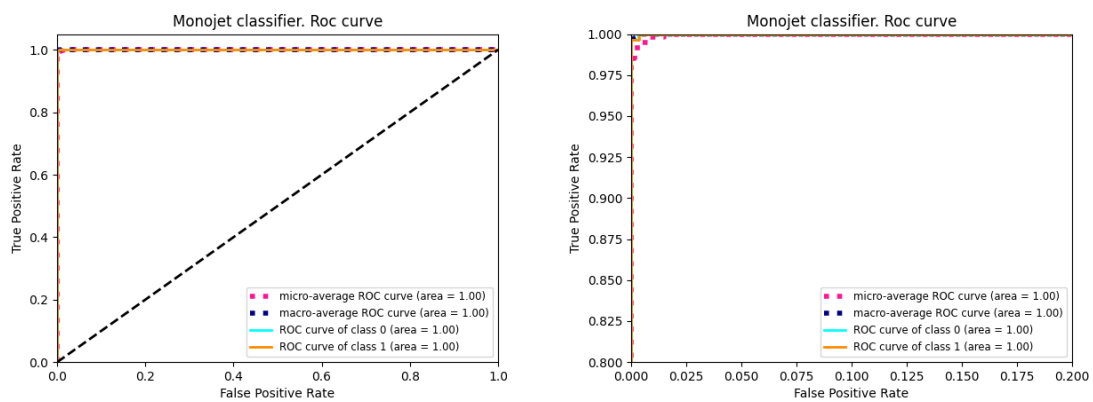


Figure 13: ROC curve of the model 5 training on the monojet data.

3.2.2 Mono-Z

The figures in this section clearly demonstrate that the model achieves convergence to a global minimum. Analyzing the shape of the trained accuracy and validated accuracy throughout the training process for each epoch provides valuable insights. Figure 15 reveals that the shape of the validation accuracy curve is distinct from that of the monojet case shown in Figure 12. Drawing definite conclusions based solely on the small discrepancy in the number of parameters between the two models, 12514 for mono-Z and 9914 for monojet, would be challenging. Deep learning models typically contain a larger number of parameters, which means that a marginal difference may not be a significant factor in why the shape differs from one another. Therefore, it is reasonable to infer that the higher complexity of the mono-Z dataset, containing four input channels, compared to the monojet dataset, containing two input channels, is the primary reason behind the difference in the shape of the validation accuracy curves. One other aspect that could help explain the discrepancy in the accuracy curve of the mono-Z model is the limitation of training data. Since the mono-Z data contains higher complexity and more dimensions compared to the monojet data the model might need more training examples to learn the underlying complex relationship, while the monojet data is enough.

Consequently, training a model with fewer parameters and smaller channels is comparatively easier. The uneven shape of the validation accuracy curve arises from the model's training on the training data and subsequent testing on the validation data. Due to the differences in data points between the two sets, variations in the validation accuracy occur during the learning process. However, an analysis of the overall shape in Figure 15 reveals that the validation accuracy follows the training accuracy, indicating that the model learns with each epoch. Additionally, towards the end of the epochs, the validation accuracy continues to increase at a similar rate as the training accuracy, which means that overfit does not occur.

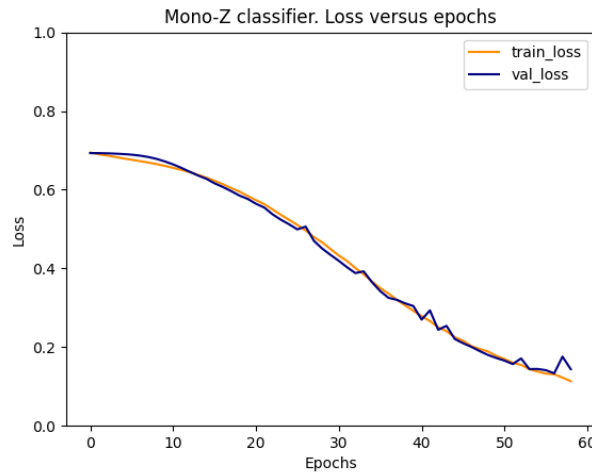


Figure 14: Loss function (categorical crossentropy) of the model 5 training on the mono-Z data, versus number of epochs.

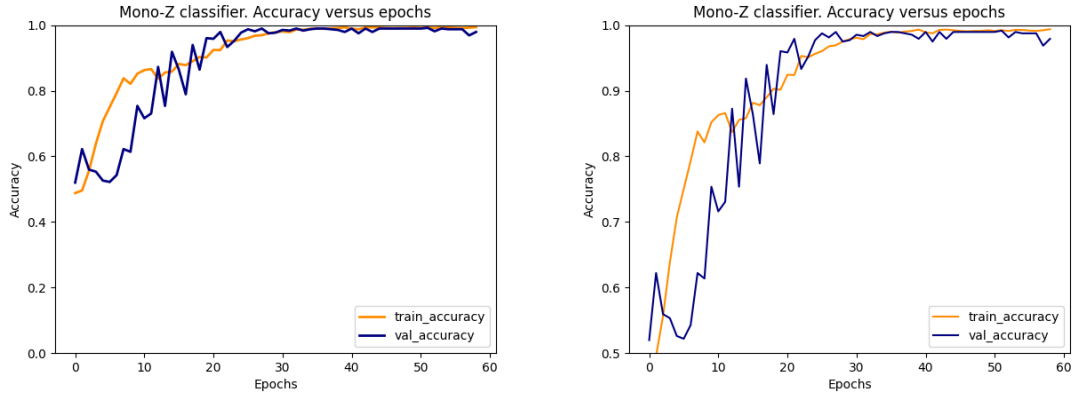


Figure 15: Accuracy of the model 5 training on the mono-Z data, versus number of epochs.

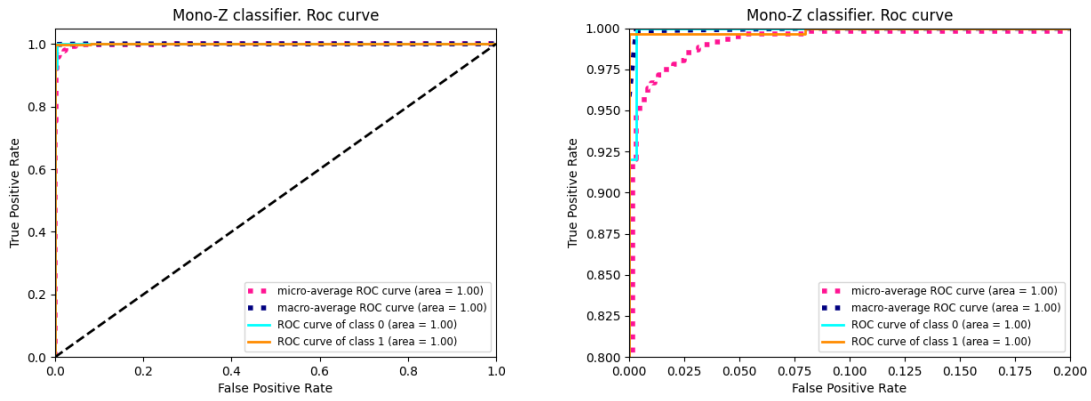


Figure 16: ROC curve of the model 5 training on the mono-Z data.

3.3 CNN regressor

The results of the CNN regressor models are summarized in table 15.

Signature	MAPE	MAE	RMSE
Monojet	17.9	101.4	126.6
Mono-Z	8.5	51.8	69.0

Table 15: Performance of mass regressors

As we can see from table 15 both the regressor models can efficiently predict the mass of the DM candidates. However, the regressor model for the mono-Z data achieves a better score across all metrics compared to the monojet regressor. This could be due to the difference in the input arrays to the models, where the mono-Z

data contains more information that could be of predictive nature regarding the mass of the DM candidates.

Another plausible explanation for the performance of the mono-Z regression model compared to the monojet regression model is because of their distinctive kinematic distribution behaviors. As previously discussed in Section 3.1, the kinematic distribution associated with the monojet signature exhibited minimal variations for different masses. The kinematic distribution of the mono-Z signature displayed more distinction between various masses. Based on these observations, it is reasonable to assume that the mono-Z regression model surpasses the performance of the monojet regression model due to its increased capability to discern and exploit the patterns presented in the kinematic distributions and due to more information from the larger input tensors.

Early stopping, similarly to the classifier models, was used as a regularization technique. To combat the higher complexity of the regressors the maximum number of epochs was set to 2000. Due to the emerging discrepancy in the loss functions for the regressor models the patience of the early stopping was set to 80.

Another observation is that the regressor models required significantly more epochs in order to converge compared to the classifier models. However, this is expected given the inherent higher complexity of the regression task contra the classification task.

For both the regressor models the learning rate had to be sufficiently low in order to get a relatively clear convergence. The high discrepancy in the models could be reduced by lowering the learning rate even more sacrificing computational costs where the number of epochs required to achieve convergence reached over 5000.

3.3.1 Monojet

For the monojet regressor the architecture that was chosen includes one `Conv1D` layer together with one `MaxPooling1D` layer followed by two `Dense` layers. The regression model was unstable when the number of parameters was relatively large, which is why the number of filters for the convolutional layer was set to 10. By reducing the feature space the model became more robust to noise leading to a stable model.

From figure 17 we can see that the regressor model for monojet converges after around 1300 epochs. The discrepancy is noticeable, but it is clear that the validation loss decreases, though fluctuating, simultaneously with the training loss which indicates convergence.

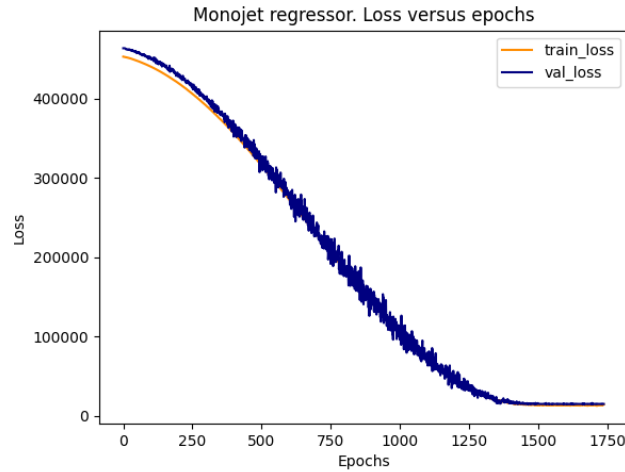


Figure 17: Loss function (MSE) of the model 8 training on the mono-jet data, versus number of epochs.

3.3.2 Mono-Z

For the mono-Z regressor the architecture has been modified in order to deal with higher complexity data. The model includes a total of three **Conv1D** layers with decreasing filter size together with three **MaxPooling1D** layers and finally two **Dense** layers. Stacking three convolutional layers leads to so called hierarchical feature extraction where each layer captures different levels of abstraction. The first layer with 32 filters allows the network to capture a wide range of low-level features, whereas the subsequent layers with fewer filters, 12 and 8, focus on learning higher-level features. This architecture ultimately leads to the network being able to find patterns and underlying relationships in the complex input data.

In figure 18 we can see that the loss function over the number of epochs clearly converges, however, just as for the classification case there is more discrepancy compared to the monojet equivalent.

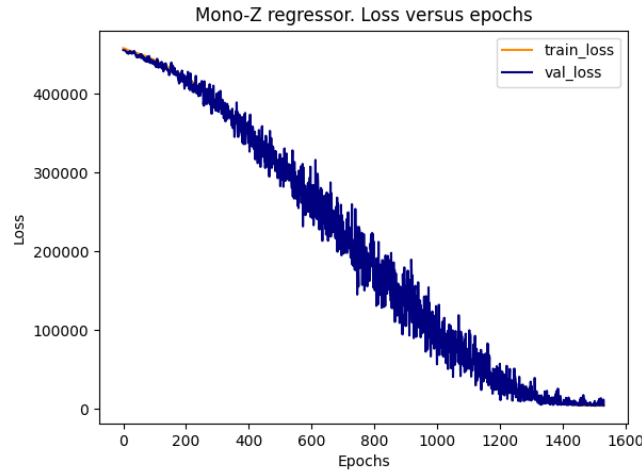


Figure 18: Loss function (MSE) of the model training on the mono-jet data, versus number of epochs.

3.4 Sustainability aspect

This project spanned across three months and was the culmination of a Bachelor's degree in Engineering Physics. Though its effect on global environmental, social and economic sustainability is negligible, the principles applied in this project are performed on large scale all across the globe daily and is therefore worth discussing. The main focus of this project and therefore the main topic of discussion in this section is the use of Artificial Intelligence.

The Artificial Intelligence Index Report is an initiative of Stanford Institute for Human-Centered Artificial Intelligence and tracks and visualizes data related to AI. AIAAIC, a database which tracks incidents of AI misuse has seen a rapid increase in reported incidents. As use of AI goes up, so does misuse. The acceleration is in large due to the shift from an academia dominated field to an industry one, partly due to the huge amounts of data gathered by the largest tech companies today. The race to build the most sophisticated AI has also taken a toll on the environment. The training of BLOOM, a multilingual language model, emitted 25 times more carbon dioxide than a single air traveller on a one way trip from New York to San Francisco [10].

Sustainability in academia and research is no straightforward topic. Resources are poured into research in various areas and fields. CERN alone, being the worlds largest science experiment, consumes enough power throughout a year to power 300,000 UK homes. It usually operates at around 200 MW, which is about a third of the power consumption of nearby city Geneva. [27]. The aim of research is to prospect for future returns, whether financial gain or of non-tangible nature such as the human strive for knowledge. One does therefore not know beforehand whether the expenses implied by a project will diminish in comparison to its result, or if the project will fail. This, of course, is the fundamental complication of research funding.

4 Conclusions

We have proven that a DL classifier can be trained to distinguish DM candidates from monojet and mono-Z signatures on a parton level, and to do so with near perfect accuracy. The results thus enforces the fact that DL can be a great asset in classifying data produced by physics experiments, if trained on sufficient and *iid* data. On the other hand, interpreting the result of the mass regressors is not as binary. The monojet and mono-z models obtained a MAPE of of 17.9 % and 8.5 % respectively. In general, regressors require more data compared to a classifier in order to produce results of the same magnitude of use. Obtaining more data is often difficult, time consuming or outright impossible. It can therefore be concluded that DL can in fact be a useful tool in classifying data from HEP simulation aids.

This proves that highly accurate regressors require large amounts of training data in comparison to an accurate classifier.

Obtaining more data is often difficult, time consuming or outright impossible. It can therefore be concluded that

This report enforces the fact that DL can and should in fact be utilized when analyzing data produced by experiments

Judging from the performance of both the classifier and regression models there is great potential in serializing these models, which wouldn't require more training of the models, but would require some programming bal bla bla

ML kan användas i upptäckandet av ny fysik, Classification iaf

5 Acknowledgements

The computations at UPPMAX were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Uppsala University partially funded by the Swedish Research Council through grant agreement no. 2022-06725. We would also like to thank Harri Waltari, Prashant Singh, Stefano Moretti, Rikard Enberg, Yong Sheng Koay and Luka Panizzi for the help along the way and making this project possible.

6 Populärvetenskaplig sammanfattning

Trots att det idag håller en etablerad plats i populärkultur och *science fiction*, så är mörk materia ett mycket relevant forskningsämne i framkanten av modern partikelfysik. Dess upptäckt krediteras ofta till den schweiziske vetenskapsmannen Fritz Zwicky när han under tidigt 1930-tal observerade ovanligt höga rotationshastigheter av himlkroppar kring galaxkluster. Zwicky konstaterade då förekomsten av osynlig, eller mörk, materia. Sedan dess har ett stort antal experiment genomförts med målet att direkt detektera mörk materia, främst med av CERN, men utan resultat. Forskare tror att mörk materia partikeln antingen kan gömma sig i form av partiklar vi redan vet interagerar väldigt lite med vanligt materia, eller så kallade supersymmetriska partiklar som vi ännu inte observerat. Dessa partiklar kallas därför för kandidater till mörk materia.

Utvecklingen av sofistikerade verktyg som kan simulera partikelacceleratorer, specifikt förhållandena som råder inuti CERN's flaggskepp Large Hadron Collider, har öppnat nya möjligheter inom partikelforskning. LHC är en 27 km lång ring av avancerad teknologi, placerad 100 meter under gränsen mellan Frankrike och Schweiz. Inuti LHC accelereras protoner till 99.9999991% av ljusets hastighet, för att sedan kollideras till dess beståndsdelar vilket kan fördjupa vår förståelse om universum. I denna rapport undersöker vi hur AI kan utnyttjas på data från dessa simuleringsverktyg för att förutspå egenskaper hos mörk materia. Eftersom den mörka materian i sig inte interagerar med vanligt materia, så påstår forskare att dess närvaro i efter-spelet av en proton-proton krock uppkommer i form av *saknad energi*. Målet med denna rapport är att från bland annat denna saknade energi kunna förutspå vilken sorts mörk materia kandidat som producerats, samt vilken massa den har.

References

- [1] *A framework for fast simulation of a generic collider experiment*. 2021. URL: <https://cp3.irmp.ucl.ac.be/projects/delphes>.
- [2] Georges Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett. B* 716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].
- [3] Georges Aad et al. "Performance of the missing transverse momentum triggers for the ATLAS detector during Run-2 data taking". In: *Journal of High Energy Physics* 2020.8 (Aug. 2020). DOI: 10.1007/jhep08(2020)080. URL: <https://doi.org/10.1007%2Fjhep08%282020%29080>.
- [4] Aad, Georges, and et. al. "Search for new phenomena in events with an energetic jet and missing transverse momentum in pp collisions at $\sqrt{s}=13$ TeV with the ATLAS detector". In: *Phys. Rev. D* 103.11 (2021). DOI: 10.1103/PhysRevD.103.112006.
- [5] L. Alzubaidi, J. Zhang, and A.J. et al. Humaidi. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8.53 (Mar. 2021), pp. 2–9. DOI: <https://doi.org/10.1186/s40537-021-00444-8>.

- [6] Richard D. Ball et al. “Evidence for intrinsic charm quarks in the proton”. In: *Nature* 608.7923 (2022), pp. 483–487. DOI: 10.1038/s41586-022-04998-2. arXiv: 2208.08372 [hep-ph].
- [7] Gianfranco Bertone and Dan Hooper. “A History of Dark Matter”. In: *Physics Reports* 90.4 (2018), pp. 13–14. ISSN: 0370-1573. DOI: <https://doi.org/10.1103/RevModPhys.90.045002>.
- [8] Serguei Chatrchyan et al. “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].
- [9] L Chuan. *Construction of the deep neural network (DNN) model*. 2019. URL: https://www.researchgate.net/figure/Construction-of-the-deep-neural-network-DNN-model_fig4_333567419.
- [10] Jack Clark and Ray Perrault. *Artificial Intelligence Index Report 2023*. Stanford University Human-Centered Artificial Intelligence, 2023.
- [11] W. N. Cottingham and D. A. Greenwood. *An Introduction to the Standard Model of Particle Physics*. 2nd ed. Cambridge University Press, 2007. DOI: 10.1017/CB09780511791406.
- [12] *Dark matter*. URL: <https://home.cern/science/physics/dark-matter>.
- [13] B A Dobrescu. “Beyond the Standard Model”. In: (2014). DOI: 10.5170/CERN-2014-003.119. URL: <https://cds.cern.ch/record/2114791>.
- [14] A. Hidaka and T. Kurita. *Consecutive Dimensionality Reduction by Canonical Correlation Analysis for Visualization of Convolutional Neural Networks*. 2017. URL: https://www.researchgate.net/publication/320748406_Consecutive_Dimensionality_Reduction_by_Canonical_Correlation_Analysis_for_Visualization_of_Convolutional_Neural_Networks.
- [15] *How a detector works*. URL: <https://home.cern/science/experiments/how-detector-works> (visited on 05/22/2023).
- [16] *How an accelerator works*. URL: <https://home.cern/science/accelerators/how-accelerator-works>.
- [17] Andrew Zimmerman Jones. *Physics: Fermion Definition*. URL: thoughtco.com/fermion-definition-in-physics-2699188.
- [18] Gerard Jungman, Marc Kamionkowski, and Kim Griest. “Supersymmetric dark matter”. In: *Physics Reports* 267.5 (1996), pp. 195–373. ISSN: 0370-1573. DOI: [https://doi.org/10.1016/0370-1573\(95\)00058-5](https://doi.org/10.1016/0370-1573(95)00058-5). URL: <https://www.sciencedirect.com/science/article/pii/0370157395000585>.
- [19] C. K. Khosa, V. Sanz, and Soughton M. “Using Machine Learning to disentangle LHC signatures of Dark Matter candidates”. In: *Journal of Big Data* 10.151 (Apr. 2021), pp. 2–15. DOI: <https://doi.org/10.48550/arXiv.1910.06058>.
- [20] Dimitrios Sidiropoulos – Kontos. “Monte Carlo production of proton-proton collision events using the ATLAS@Home framework”. Presented 26 Jun 2020. 2018. URL: <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8932453&fileId=8932470>.
- [21] Robert Lea. *What are bosons?* URL: <https://www.space.com/what-are-bosons>.
- [22] *Lecture 3: LHC physics with MadGraph*. Mar. 2021. URL: https://www.fis.cinvestav.mx/~gae/Documents/lecture3_Vicente.pdf.

- [23] *LHC the guide*. URL: <https://press.cern/resources/brochure/knowledge-sharing/lhc-facts-and-figures>.
- [24] *MadAnalysis 5*. 2023. URL: <https://launchpad.net/madanalysis5>.
- [25] MissMJ and Cush. *File:Standard Model of Elementary Particles.svg*. 2019. URL: https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg.
- [26] Jose Miguel No. “Looking through the pseudoscalar portal into dark matter: Novel mono-Higgs and mono-Z signatures at the LHC”. In: *Phys. Rev. D* 93.3 (2016). DOI: 10.1103/PhysRevD.93.031701. arXiv: 1509.01110 [hep-ph].
- [27] *Powering CERN*. URL: <https://home.cern/science/engineering/powering-cern> (visited on 05/23/2023).
- [28] *Pseudo-rapidity*. URL: <https://atlas.cern/glossary/pseudo-rapidity> (visited on 05/22/2023).
- [29] *ROOT: analyzing petabytes of data, scientifically*. 2023. URL: <https://root.cern/>.
- [30] J Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (Oct. 2014), pp. 4–6. DOI: <https://doi.org/10.48550/arXiv.1404.7828>. URL: <https://arxiv.org/abs/1404.7828>.
- [31] *Searching for Dark Matter with the ATLAS detector*. URL: <https://atlas.cern/updates/feature/dark-matter>.
- [32] Mantilla Suarez and Cristina Ana. “Probing new physics using initial state radiation jets at the Large Hadron Collider”. Presented 26 Jun 2020. 2020. URL: <https://cds.cern.ch/record/2745725>.
- [33] *The Standard Model*. URL: <https://home.cern/science/physics/standard-model>.
- [34] P. P Urone and R. Hinrichs. *Physics*. OpenStax, 2020. URL: <https://openstax.org/books/physics/pages/1-introduction>.
- [35] *Welcome to PYTHIA*. 2023. URL: <https://www.pythia.org/>.
- [36] WikiDocs. *Part F. Convolutional Neural Networks*. 2023. URL: <https://wikidocs.net/164365>.
- [37] Julia Woithe, Gerd J Wiener, and Frederik F Van der Veken. “Let’s have a coffee with the Standard Model of particle physics!” In: *Physics Education* 52.3 (Mar. 2017), p. 034001. DOI: 10.1088/1361-6552/aa5b25. URL: <https://dx.doi.org/10.1088/1361-6552/aa5b25>.
- [38] R. Yamashita et al. “Convolutional neural networks: an overview and application in radiology”. In: *Insights into Imaging volume 9* (June 2018), pp. 611–629. DOI: <https://doi.org/10.1007/s13244-018-0639-9>.
- [39] M. et al Yani. *Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry’s Nail*. 2019. URL: https://www.researchgate.net/publication/333593451_Application_of_Transfer_Learning_Using_Convolutional_Neural_Network_Method_for_Early_Detection_of_Terry’s_Nail.

A Appendix

A.1 Machine learning models

In this section more information about the architecture of the models are presented. This include the input and output shape and all of the layers that has been used to build the models.

A.1.1 Classification CNN

Table 16 gives information about shape and layers of the CNN classifier of both models, monojet and mono-Z. The main differences between the models are the amount of bins and channels in the input. Monojet have two input channels with 30 bins, datapoints, for each channel. Mono-Z have four input channels with 40 bins for each channel.

Layer	Monojet		Mono-Z	
	Input Shape	Output shape	Input shape	Output shape
Conv1D	None, 30, 2	None, 28, 32	None, 40, 4	None, 38, 32
MaxPooling1D	None, 28, 32	None, 14, 32	None, 38, 32	None, 19, 32
Conv1D	None, 14, 32	None, 12, 32	None, 19, 32	None, 17, 32
BatchNormalization	None, 12, 32	None, 12, 32	None, 17, 32	None, 17, 32
MaxPooling1D	None, 12, 32	None, 6, 32	None, 17, 32	None, 8, 32
Flatten	None, 6, 32	None, 192	None, 8, 32	None, 256
Dense	None, 192	None, 32	None, 256	None, 32
Dense	None, 32	None, 8	None, 32	None, 8
Dense	None, 8	None, 2	None, 8	None, 2

Table 16: Architecture and shapes of the classification CNN models.

A.1.2 Regressor CNN

In the following sections, the architecture of the regression models, monojet and mono-Z, are presented.

A.1.2.1 Monojet regressor

In Table 17, the architecture of the monojet regression model is presented with the same input as the classification model.

Layer	Input Shape	Output shape
Conv1D	None, 30, 2	None, 28, 10
BatchNormalization	None, 28, 10	None, 28, 10
MaxPooling1D	None, 28, 10	None, 9, 10
Flatten	None, 9, 10	None, 90
Concatenated	None, 90	None, 91
Dense	None, 91	None, 16
BatchNormalization	None, 16	None, 16
Dense	None, 16	None, 1

Table 17: Architecture and shapes of the regressor CNN models for monojet data.

A.1.2.2 Mono-Z regressor

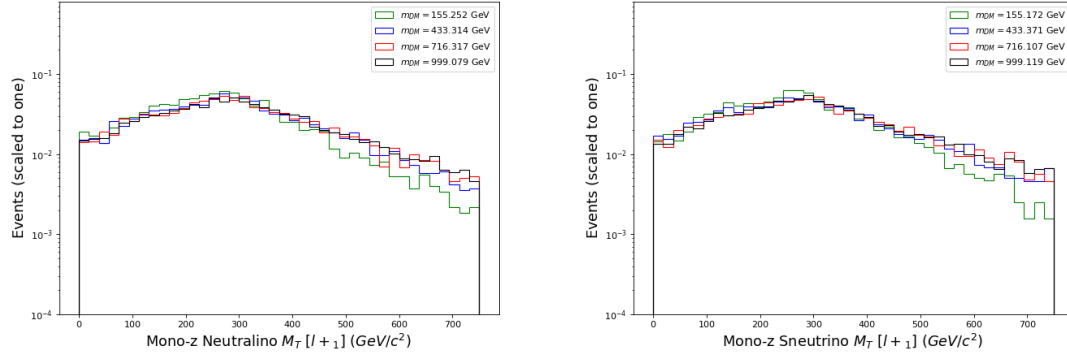
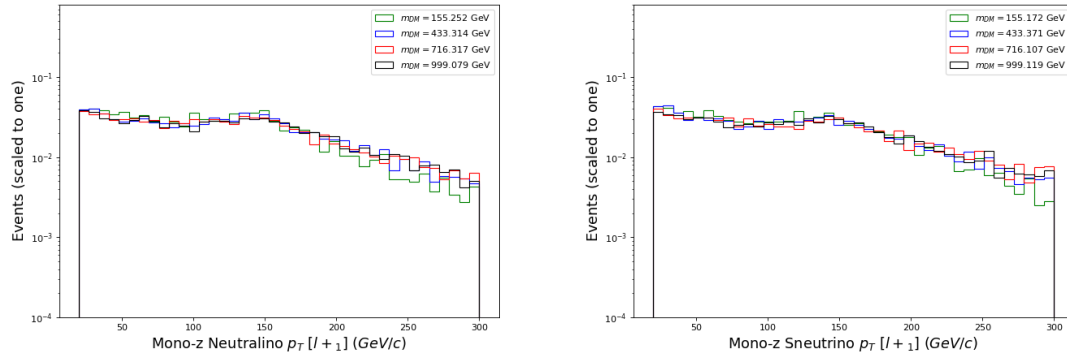
In Table 18, the architecture of the mono-Z regression model is presented with the same input as the classification model.

Layer	Input Shape	Output shape
Conv1D	None, 40, 4	None, 38, 32
MaxPooling1D	None, 38, 32	None, 19, 32
Conv1D	None, 19, 32	None, 17, 12
MaxPooling1D	None, 17, 12	None, 8, 12
Conv1D	None, 8, 12	None, 6, 8
BatchNormalization	None, 6, 8	None, 6, 8
MaxPooling1D	None, 6, 8	None, 3, 8
Flatten	None, 3, 8	None, 24
Concatenated	None, 24	None, 25
Dense	None, 25	None, 16
BatchNormalization	None, 16	None, 16
Dense	None, 16	None, 1

Table 18: Architecture and shapes of the regressor CNN models for mono-Z data.

A.2 Kinematic distributions

The following figures represent the kinematic distributions, p_T and M_T , for different masses for the mono-Z signature with the two different dark matter candidates, sneutrino and neutralino.

Figure 19: M_T distribution of the positive lepton, mono-Z signatureFigure 20: p_T distribution of the positive lepton, mono-Z signature