Performance of Deep Learning in Searches for New Physics Phenomena in Events with Leptons and Missing Transverse Energy with the ATLAS Detector at the LHC

Mona Anderssen

Thesis submitted for the degree of
Master in Nuclear and particle physics
60 credits

Department of Physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

Performance of Deep Learning in Searches for New Physics Phenomena in Events with Leptons and Missing Transverse Energy with the ATLAS Detector at the LHC

Mona Anderssen

Abstract

In this thesis we have searched for new physics phenomena predicted by Supersymmetry and Dark Matter simplified models. Both traditional cut and count analysis and Machine Learning(ML) based methods, such as Boosted Decision Trees and Neural Networks, were performed. The analysed run-2 13 TeV data, corresponding to an integrated luminosity of 139 fb^{-1}, were collected by the ATLAS experiment at the LHC between 2015 and 2018. The training was performed on different compositions of mass splittings (difference between the new particles involved in each new physics model) and features (low- and high-level kinematic variables). To achieve a good performance, we made use of an advanced computing infrastructure including both CPU's and GPU's. The results obtained have shown a better performance of the ML methods as compared to the more traditional cut and count analysis, especially in the low mass splitting region which so far has been a challenge for the cut and count analysis. Slightly better sensitivities were obtained with BDT but neural networks have so far not yet been fully exploited. Another future challenge.

Acknowledgements

First and foremost, I would like to thank my supervisor Farid Ould-Saada for introducing me to particle physics and helped me find both an exciting and challenging project. My co-supervisor, Eirik Gramstad, have been a great support and help when errors have occurred during the work done for this thesis. Thank you both for never giving up on me and for dragging me over the finish line.

A big thank you goes to the HEP-group which have welcomed me with open arms since the day I started at HEP in 2018. I appreciate the great social environment and that all of you are always ready to answer my questions, both big and small. And of course, thank you for making me a quiz nerd.

Even though the whole HEP-group probably should have been named, I need to give an extra thank you to Eli, Helén and Oda! You have been with me on the highest highs and lowest lows, so thank you for always encouraging, motivating, pushing, and also feeding me.

Of course a huge thank you goes to "my brother from another mother", Federico! You have been a huge support both on and off campus, and I am grateful for sharing this journey with you. Particle physics brought us together and started our friendship, and even though we are now going different ways, I hope it will continue forever. I would also like to thank both you and Jake for a lot of memorable times this last year, and pushing me to be more confident in another language.

Knut Oddvar Høie Vadla, thank you! Without your expertise and patience, I would never have finished this master thesis. You have always taken the time to explain things to me, even when at your new job. To know that you would always be willing to help, have been a relief in challenging situations.

Last but not least, I would like to thank my family, especially my parents and my sister. You have always been nothing but proud of me, and having a sister close by during this degree have been an invaluable support. Thank you very much.

The research presented in this thesis has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

Contents

Introduction

The Standard Model (SM) [1] is a well known theory of particle physics lately further confirmed after the Higgs boson was discovered at CERN in 2012 by the ATLAS [2] and CMS [3] collaborations. The SM can describe all visible matter and contains matter particles, quarks and leptons, and the fundamental forces acting between them. Even though the SM can describe all the matter we can see around us, it can't describe the whole universe. There are of course many theories which attempt to addvers the shortcomings of the SM, but in this thesis we mainly focus on Supersymmetry (SUSY) and non-SUSY Dark Matter (DM) simplified models. SUSY is an extension of the SM which predicts that every particle in the SM has a supersymmetric partner with equal quantum numbers, except for the spin. One of the consequences of discovering SUSY may be that it provides a DM candidate, which we are going to look more into in this thesis.

For many years the standard way to perform an analysis have been to simply apply cuts kinematical variables, which efficiently discriminate new physics signal from SM background. The set of cuts define signal regions, which are optimized to lead to the best sensitivity, i.e. the highest signal over background ratio. However, these methods have limitations when it comes to discovering and using correlations between variables. Cut and count methods usually apply cuts evaluating one variable at the time or use so-called rectangular cuts evaluating the correlation between maximum two variables. With computers and new algorithms constantly being improved and developed, particle physicists have tried to look for other possibilities to perform searches for new physics, namely by using Machine Learning (ML) based methods and algorithms. ML methods, in contrast to the cut and count methods, can investigate multidimensional correlations and place cuts using boundaries in higher dimensions. ML has therefore become very popular and is widely used in many fields of research. In this thesis we explore Boosted Decision Trees (BDTs) and Neural Networks (NNs), and compare the results against each other and with a more traditional cut and count analysis.

In chapter 1 a short introduction to the SM and new physics will be given. Chapter 2 contains a short presentation of the detector and the computing infrastructure to analyze 13 TeV data collected by the ATLAS experiment at the Large Hadron Collider.. We will then move to the search for new physics with the cut and count method in chapter 3. The next chapters (4 and 5) introduce Machine Learning algorithms and apply them to the searches introduced in chapter 3. In the end present our results and compare the traditional and ML-based methods in chapter 6 and conclude in chapter 7.

Chapter 1

The Standard Model and Beyond

Although the Standard Model (SM) of particle physics is one of the greatest triumphs of modern physics, it cannot describe all physical phenomena observed in nature. There are several shortcomings of the this model, which implies that we need an extension of the SM to understand and explain these problems. There are several different suggested theoretical solutions to these, and one of them is called supersymmetry (SUSY). SUSY introduces multiple new particles as an extension to the SM particles, but no SUSY particle has so far been detected. Many supersymmetric theories inlcude a weakly interacting massive particle (WIMPs) and thus a candidate to explain the dark matter (DM) in our Universe, a brief overview of which will be provided in this thesis. Before we discuss SUSY and DM, we begin with a run-through of the SM.

1.1 The Standard Model

The current best description we have of the fundamental constituents of our Universe is the Standard Model. It includes the elementary particles and the forces acting between them and can explain all visible matter we have around us (sans gravity).

The particles in the SM are organized in three generations, as shown in figure 1.1. It consists of two different types of particles, namely fermions and bosons. Fermions are the matter particles, and the bosons carry the forces that act between the fermions and are usually called force particles.

The elementary particles interact via some force if they carry the *charge* corresponding to the force. Only particles that are electrically charged interact via the electromagnetic force, only particles with (weak) isospin interact via the weak force, and only particles with color charge interact via the strong force.

1.1.1 Standard Model symmetries

Formally, the three interactions in nature are related to three different symmetry groups. The collective symmetry group of the SM is often represented as

$$SU(3)_C \otimes SU(2)_L \otimes U(1)_Y, \tag{1.1}$$

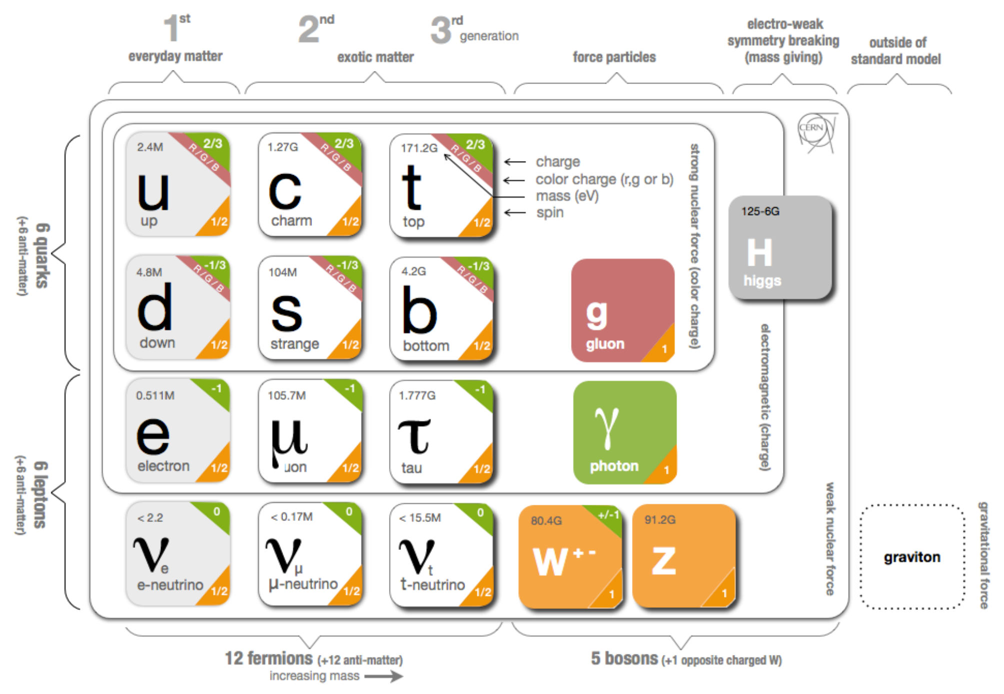
where $SU(3)_C$ is the symmetry group of the strong force[1], $SU(2)_L$ is the symmetry group of the weak force[2], and $U(1)_Y$ is related to the electromagnetic force[3]. The Y represents (weak) hypercharge, which is related to electric charge Q and the third component of the isospin I_3 through $Y = 2(Q - I_3)$.

Since $SU(2)$ symmetry transformations are defined in terms of 2 x 2 matrices, they need a two-dimensional vector to act upon. Analogously, the $SU(3)$ transformations need to act upon a three-dimensional vector. These vectors are referred to as $SU(2)$ *doublets* and $SU(3)$ *triplets.*

[1]More precisely quantum chromodynamics (QCD), which is the quantum field theory of the strong interaction. The C corresponds to color charge being conserved.

[2]With isospin I as the conserved quantity. The subscript L will be explained later.

[3]More precisely quantum electrodynamics (QED), which is the quantum field theory of the electromagnetic interaction.



1st everyday matter
2nd exotic matter
3rd generation
force particles
electro-weak symmetry breaking (mass giving)
outside of standard model
6 quarks (+6 anti-matter)
6 leptons (+6 anti-matter)
2.4M 2/3 R/G/B u up 1/2
1.27G 2/3 R/G/B c charm 1/2
171.2G 2/3 R/G/B t top 1/2
charge
color charge (r,g or b)
mass (eV)
spin
4.8M -1/3 R/G/B d down 1/2
104M -1/3 R/G/B s strange 1/2
4.2G -1/3 R/G/B b bottom 1/2
g gluon 1
125-6G H higgs
0.511M -1 e electron 1/2
105.7M -1 μ μon 1/2
1.777G -1 τ tau 1/2
γ photon 1
< 2.2 0 νe e-neutrino 1/2
< 0.17M 0 νμ μ-neutrino 1/2
< 15.5M 0 ντ τ-neutrino 1/2
80.4G +/-1 W+- 1
91.2G Z 1
graviton
strong nuclear force (color charge)
electromagnetic (charge)
weak nuclear force
gravitational force
12 fermions (+12 anti-matter)
increasing mass
5 bosons (+1 opposite charged W)

Figure 1.1: An overview of the particles in the Standard Model [4].

1.1.2 Fermions

If we look at figure 1.1, we can see that there are 12 different fermions split up in two groups called quarks and leptons, which again are split up into three generations. The lightest quarks and leptons are in the first generation, and the most massive quarks are in the third generation. All of the fermions are $\frac{1}{2}$-spin[4] particles and differ by mass and electric charge. They also obey the Pauli exclusion principle, which means that only one fermion can occupy a given quantum state for any given set of quantum numbers.

Further, it is well known that parity[5] is violated for the weak interaction. This can be explained by introducing *chirality* which is, using projection operators P_L and P_R, defined as

[4]Spin is an intrinsic property of all fundamental particles, which can be seen as some kind of internal angular momentum.

[5]Parity is spatial inversion.

$$P_L = \frac{1}{2}(1-\gamma^5) \text{ and } P_R = \frac{1}{2}(1+\gamma^5). \tag{1.2}$$

From this, fermions can be separated into a left-chiral and right-chiral part, where only the left-chiral part of a fermion is charged under the weak force (corresponding to the L in equation 1.1). It is thus the left-chiral parts of fermions that are put into $SU(2)$ doublets[6].

Quarks

We have six different quarks in the SM, and they differ by mainly mass, but also charge (both electrical and color charge). All up-type quarks, which are the three quarks in the first row in figure 1.1, have electric charge +2/3, and all down-type quarks, which is in the second row in figure 1.1, have electric charge -1/3. Since they carry a non-zero electric charge, they interact via the electromagnetic force.

They can also interact with the weak force, which allows all possible combinations of quarks that differ by one unit of charge in doublets:

$$\begin{pmatrix} u_L \\ d_L \end{pmatrix}, \begin{pmatrix} u_L \\ s_L \end{pmatrix}, \begin{pmatrix} u_L \\ b_L \end{pmatrix}, \begin{pmatrix} c_L \\ d_L \end{pmatrix}, \begin{pmatrix} c_L \\ s_L \end{pmatrix}, \begin{pmatrix} c_L \\ b_L \end{pmatrix}, \begin{pmatrix} t_L \\ d_L \end{pmatrix}, \begin{pmatrix} t_L \\ s_L \end{pmatrix}, \begin{pmatrix} t_L \\ b_L \end{pmatrix}. \tag{1.3}$$

The two quarks in the first generation, namely up (u) and down (d), are the constituents of the protons (uud) and neutrons (ddu) - which then combine in various ways to form atomic nuclei. The second and third generation of quarks (charm, strange, top and bottom) are more massive than the quarks in the first generation. Since the quarks in the second and third generations are more massive, they need more energy to exist and will therefore decay to lighter particles very fast.

Quarks also carry color charge, which allows them to couple to different gluons through the strong interaction in quantum chromodynamics (QCD). Quarks are the only known particles that interact with all the fundamental forces.

[6]The right-chiral parts are said to be $SU(2)$ singlets, meaning that they are not affected bt the weak force.

Further, quarks can mix between generations, as described by the CKM matrix [1].

Leptons

Leptons are the other six fermions of the SM. Like quarks, they come in three generations and can be arranged in doublets:

$$\begin{pmatrix} \nu_{eL} \\ e_L \end{pmatrix}, \begin{pmatrix} \nu_{\mu L} \\ \mu_L \end{pmatrix}, \begin{pmatrix} \nu_{\tau L} \\ \tau_L \end{pmatrix}. \tag{1.4}$$

They also differ by mass and electric charge where the lower component has charge -1, and the upper has no charge. Because of that, neutrinos only interact with weak bosons, while for the electron, muon and, tau, electromagnetic interactions are also possible. In the SM the neutrinos are assumed to be massless, but we know from neutrino oscillation experiments that this is not true. However, exactly how the neutrinos acquire mass and why they are so much lighter than all of the other particles in the SM is not yet understood.

Analogously to quarks, fermion mass and weak eigenstates do not coincide and they are related by the PMNS matrix [1]. This phenomenon provides an explanation of neutrino oscillations, where after some distance from the interaction point we measure a different neutrino flavor.

1.1.3 Bosons

On the right-hand side of figure 1.1, the integer spin particles (spin-0, 1, 2) known as bosons are shown. Bosons follow Bose-Einstein statistics and contain the force carriers for the electromagnetic force, the weak force, and the strong force. If gravity was included in the Standard Model, it would have been through an extra boson, the graviton (G). The graviton is believed to have spin-2, which together with the Higgs boson that has spin-0 is the only two bosons that differs in spin from the force carriers with spin-1. Higgs is also so far the only scalar particle detected.

The strong nuclear force

The strong nuclear force is mediated by the gluon (g) and couples to the three color charges r (red), g (green), b (blue). Moreover, as suggested by the name, it has a very strong coupling to quarks, and it is responsible for the fact that quarks do not appear as free, unbounded particles. The gluon is massless, and unlike the other forces, it can couple with itself, as you can see in the Feynman diagrams[7] in figure1.2.

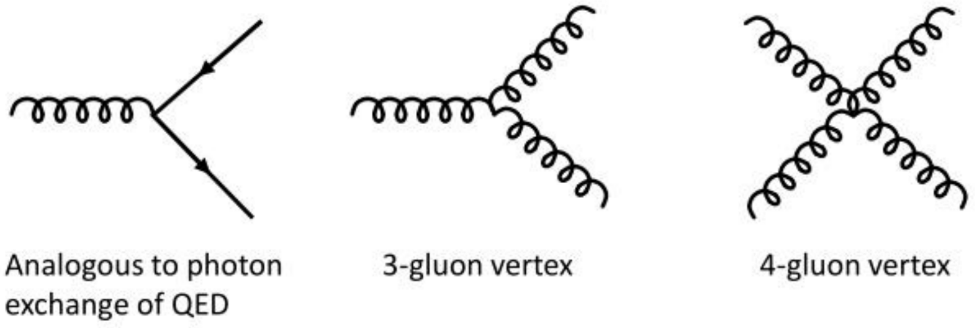


Figure 1.2: Feynman diagrams for the strong force vertices, where the curly lines represent the gluons [5].

The electromagnetic force

The electromagnetic force is mediated by the photon (γ) and couples to all fermions with electric charge, namely all the quarks and leptons except for the neutrinos. As for the gluon, it has no mass, but since it is electrical neutral it cannot couple to itself.

[7]Feynman diagrams are a way to graphically represent interactions between particles.

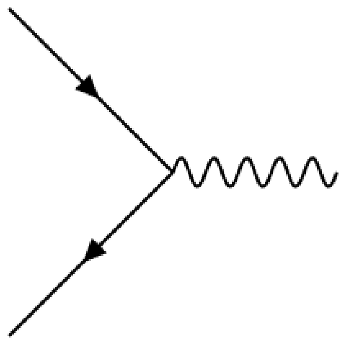


Figure 1.3: A Feynman diagram of the electromagnetic vertex, where the wiggly line represent the photon [6].

The weak nuclear forces

The weak nuclear forces are mediated by the $W^{\pm}$-bosons and the Z^0-boson. They couple to all particles in the SM, and unlike the other force particles, they have mass. This feature is explained in the GWS model [7] through a spontaneous symmetry breaking due to a non-zero vacuum expectation value that adds an extra degree of freedom to otherwise massless force carriers. It is also the only interaction known to allow flavor change, which means that e.g., an up quark can become a down quark or a muon become an electron as shown in figure1.4.

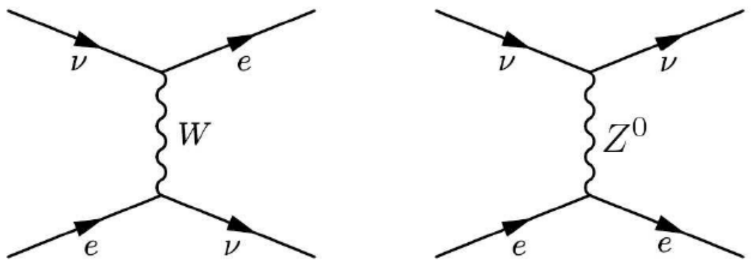


Figure 1.4: Feynman diagrams of the weak force vertices [8].

Note that the physical bosons $\gamma, Z^0, W^{\pm}$ are not the mediators associated to the single symmetry groups, but are instead a linear combination of those states explained by the spontaneous symmetry breaking leading to the electroweak theory.

The Higgs boson

The final piece, the Higgs boson (H), that was missing in the SM was discovered in 2012 at CERN [2, 3]. The discovery gives increased credence to the SM and explains why the weak force carriers, Z^0 and $W^{\pm}$, have mass. The fermion masses can also be explained by couplings to the Higgs boson.

1.2 Beyond the Standard Model

As mentioned earlier, the SM is not enough to explain every phenomena observed in experiments. There are several challenging aspects of the SM; for instance, many of the parameters in the model are made to fit experimental data and do not come from theoretical principles. The SM does not offer a solution to unify the framework with gravity. The large difference between the weak energy scale and the Planck scale, known as the hierarchy problem, is another shortcoming of the SM. Moreover, the SM does not provide any explanation for the dark matter observed in our Universe nor does it explain the small, but non-zero, masses of neutrinos.

Given these shortcomings, various theories have been proposed addressing some of the open questions. Several extensions of the Standard Model exist, and there are pros and cons to all of them. Furthermore, even though the search has been going on for years, no concluding scientific evidence has been found to favor one particular SM extension over another. Below we detail an outline of one possibility: Supersymmetry, which is often denoted SUSY for short.

1.2.1 Supersymmetry

One of the most interesting theories is Supersymmetry (SUSY) [9]. Supersymmetry proposes a symmetry between fermions and bosons. In SUSY, each SM particle has a superpartner "sparticle", which only differs from the particle by half a unit of spin. All other quantum numbers are the same. An illustration of the particles in SM and their superpartner in SUSY can be found in figure 1.5.

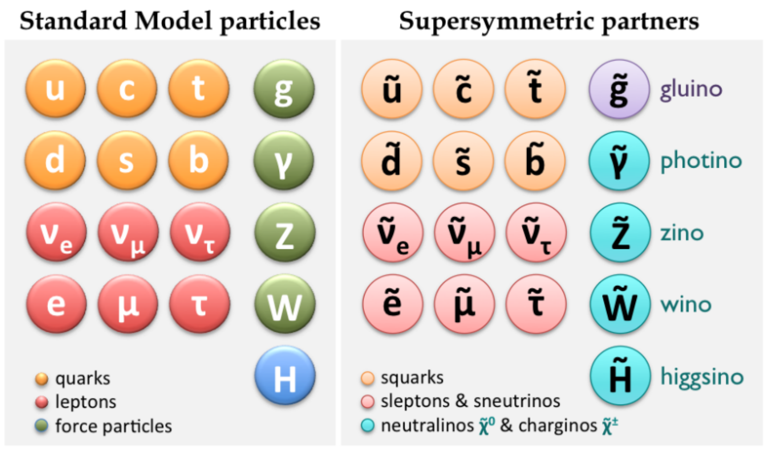


Figure 1.5: An illustration of the content of particles in the SM and sparticles in SUSY [10].

In SUSY, we do not have to introduce any new gauge groups, which means we do not have to handle any new fundamental forces. Because of this we can, in a way, say that we can describe supersymmetry with the help of a supersymmetry operator Q that alters the spin of the SM particles by $1/2$ and commutes with the gauge transformations of the SM:

$$Q\left|\text{fermion}\right\rangle = \left|\text{bosons}\right\rangle, \; Q\left|\text{bosons}\right\rangle = \left|\text{fermions}\right\rangle. \tag{1.5}$$

SUSY possibly provides a solution to the SM's hierarchy problem, which involves the need to reconcile the very different scales of electroweak symmetry breaking and the gravitational Planck scale (M_{Pl}). SUSY allows the unification of the electroweak and strong interactions, proposes dark matter particle candidates, and requires five Higgs bosons (three neutral and two charged ones). One of these proposed DM particles is the lightest supersymmetric particle (LSP) and is assumed to be stable. To make the LSP stable, we have constructed a new quantum number called R-parity such that no SUSY particles can decay into only SM particles. R-parity is defined as

$$P_{\mathrm{R}} = (-1)^{3B+L+2s} \tag{1.6}$$

or

$$P_{\mathrm{R}} = (-1)^{3(B-L)+2s}, \tag{1.7}$$

where B is the baryon number[8], L is the lepton number[9] and s is the spin of the particle.

In addition to the LSP, we are going to look at some different SUSY particles in our processes, namely neutralino and chargino. They are a mixture of the sparticle components photino, zino, and neutral higgsino, and wino and the charged higgsino, respectively.

1.2.2 Dark matter

We have many unanswered mysteries in physics today, but probably the greatest one is the nature of dark matter (DM) [11]. In the previous section's SUSY processes, we have seen that a "consequence" of SUSY may give us some viable DM candidates, namely the LSP neutralino. In this section, we will look at DM particles produced in a more simplified model; that is, a non-supersymmetry model. In this model, we assume that we have a DM mediator in addition to the DM particles in the final state. This mediator can be, among others, a scalar or a vector. This process's signature consists of detecting a well known SM particle recoiling against missing energy-momentum carried away by DM particles.

[8]Baryon number is a conserved quantity in the SM and is defined by $\frac{1}{3}(n_q - n_{\bar{q}})$, where n_q is the number of quarks and $n_{\bar{q}}$.

[9]The lepton number tells us the difference between the number of leptons and anti-leptons. This is also conserved in the SM.

Chapter 2

LHC and ATLAS

The data used in this thesis comes from the ATLAS experiment at the LHC. In this chapter, we briefly introduce the experiment and the accelerator as well as the computing infrastructure used to analyse the data. We will also get a small incite into the organization behind the experiment and the accelerator, namely CERN.

2.1 CERN

The European organisation for nuclear research, CERN, started as a research facility for mainly nuclear physics. It was built on the border between France and Switzerland near Geneva in 1954. CERN has 22 member states, where Norway is one of the founding members, but it welcomes people from all over the world to take part in the different experiments and accelerator developments.

CERN quickly became the biggest and leading research centre in particle physics as well, and the most famous discoveries done at CERN are from high energy particle collisions. Some of the biggest discoveries at CERN are: the weak neutral currents mediated by the hypothetical Z-boson in 1973 [12] and of course the discovery of the actual Z- and the $W^{\pm}$-bosons, mediators of the weak force in 1983/84 [13–15]. The most famous and recent discovery is certainly that of a first scalar boson, the Higgs boson in 2012 by the ATLAS and CMS experiments [2,3]. The Higgs boson was the missing piece to confirm the Standard Model of particle physics.

Throughout the years many different accelerators have been built at CERN. E.g. the Super Proton Synchrotron (SPS), which are still in use, accelerates and collides protons and antiprotons, and enabled the UA1 and UA2 experiments to discover the Z- and W-bosons. The Large electron positron collider (LEP) was built in a 27 km long tunnel about 100 m below ground and was the largest accelerator at the time. LEP allowed important SM precision measurements, in particular confirming the presence of exactly three low mass neutrino flavours in the SM and stringent limits on the top and Higgs masses. The last run at LEP was done in 2000 paving the road for the start of the building of the Large Hadron Collider (LHC). LHC is the accelerator producing the proton-proton collisions being recorded by the ATLAS detector and which have been used in the analysis presented in this thesis.

2.2 The Large Hadron Collider

The Large Hadron Collider (LHC) [16] is a 27km long particle accelerator about 100 meters below ground and is the most powerful of its kind in the world. Since the LHC replaced the other LEP-collider, the tunnel already existed which made the building a bit less comprehensive. The collider consists of superconducting magnets with accelerating structures to boost the proton velocity close to the speed of light, in an environment cooled down to 1.85K (-271.3^{o}C) to ensure superconductivity. The LHC accelerator is shown in figure 2.1 as part of a complex of CERN accelerators.

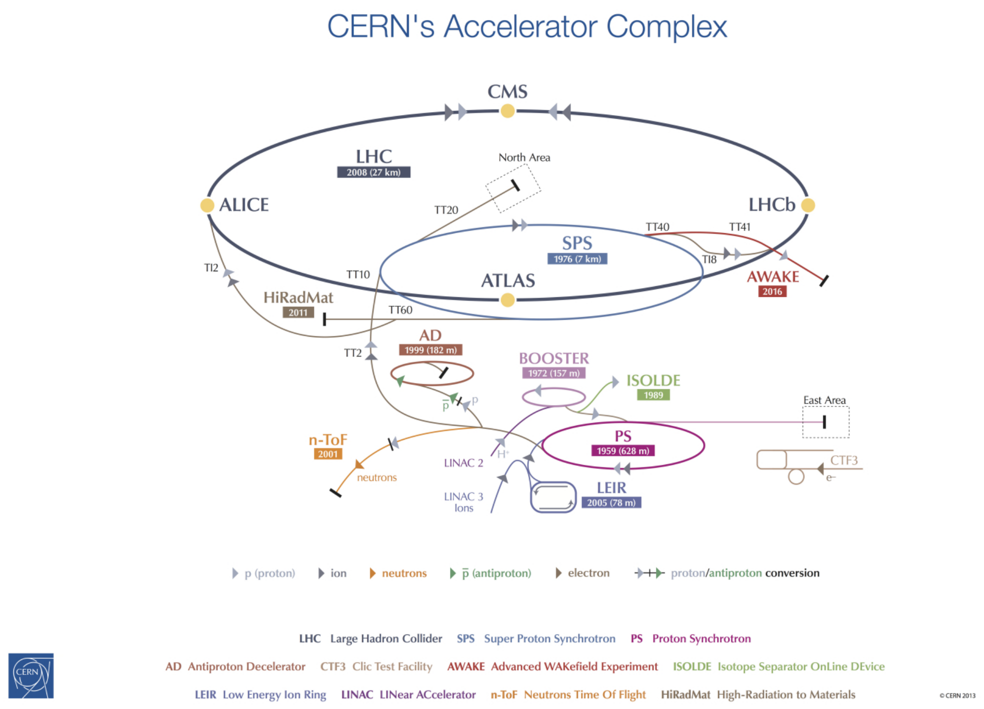


Figure 2.1: An illustration of the accelerator complex at CERN [17]

The protons extracted from a hydrogen bottle go through smaller accelerators to gain more energy before they, in the end, are injected in opposite directions into the LHC, which is the biggest circle in figure 2.1. The particles get accelerated to about 99.99% of the speed of light and a maximum energy of 6.5 TeV. In the end the two beams of protons collide in the centre of four experiments around LHC, which are marked with yellow dots in figure 2.1, namely ATLAS, ALICE, CMS and LHCb.

The different experiments focus on different research goals. ATLAS (A Toridal LHC ApparatuS) and CMS (Compact Muon Solenoid) are multipurpose detectors mainly focusing on SM measurements and searches for new physics, i.e. discovery of new particles and phenomena, e.g. Supersymmetry and Dark Matter. The most famous achievement of ATLAS and CMS is the discovery of the Higgs boson. ALICE (A Large Ion Collider Experiment) focuses on heavy ion collisions with lead-lead and lead-proton to study the quark-gluon plasma. LHCb (Large Hadron Collider beaty) focuses on processes related to b-quarks to precisely measure CP violation and oscillation phenomena.

At the LHC we focus on proton-proton collisions (and heavy ion collisions) to study various high energy final states via electroweak and strong interactions with the hope to discover new phenomena. The internal structure of the proton allows to register a large amount of events in a single collision and thus collect a large enough statistics for the experiments. The number of collisions per area per second is defined through the instantaneous luminosity $\mathscr{L}$, given by

$$\mathscr{L} = f\frac{n_1 n_2}{4\pi\sigma_x\sigma_y}, \tag{2.1}$$

where f is the crossing rate of the proton bunches, n_i is the number of colliding particles in each bunch and $\sigma_{x,y}$ is the spread of the bunch along the x- and y-directions.

Using the integrated luminosity over time we can predict the number of expected events N produced by the LHC. This is given by

$$N = \sigma\int\mathscr{L}(t)dt, \tag{2.2}$$

where σ is the cross section for a certain process.

When the instantaneous luminosity increases we get more collisions happening in the detector. This gives us a lot of interactions at the same time, which can introduce further systematic uncertainties and challenges. This phenomenon is called *pile-up*. We need to consider this to know which particles in the final state comes from which interaction. This is a constantly evolving problem since we are further developing the LHC infrastructures and apparatus towards higher and higher luminosity, e.g. HL-LHC (High Luminosity LHC).

In this thesis we are looking at data collected by the ATLAS detector from 2015-2018 (full run 2) at the LHC from proton-proton collisions at 13 TeV center of mass energy with different pile-up conditions, following the increase of instantaneous luminosity $\mathscr{L}$. The data corresponds to an integrated luminosity of 139 fb^{-1}, where we have 36.2 fb^{-1} for 2015-2016, 44.3 fb^{-1} for 2017 and 58.5 fb^{-1} for 2018.

2.3 The ATLAS detector

The ATLAS [18] detector is a massive 44 m long detector, 25 m in diameter and weighing about the same as the Eiffel tower ($\sim$ 7000 tons). It is designed to handle proton-proton collisions up to 14 TeV with a luminosity of a few times $10^{34}\text{cm}^{-2}\text{s}^{-1}$. We can see an illustration of the whole detector in figure 2.2, where the most important components are marked.

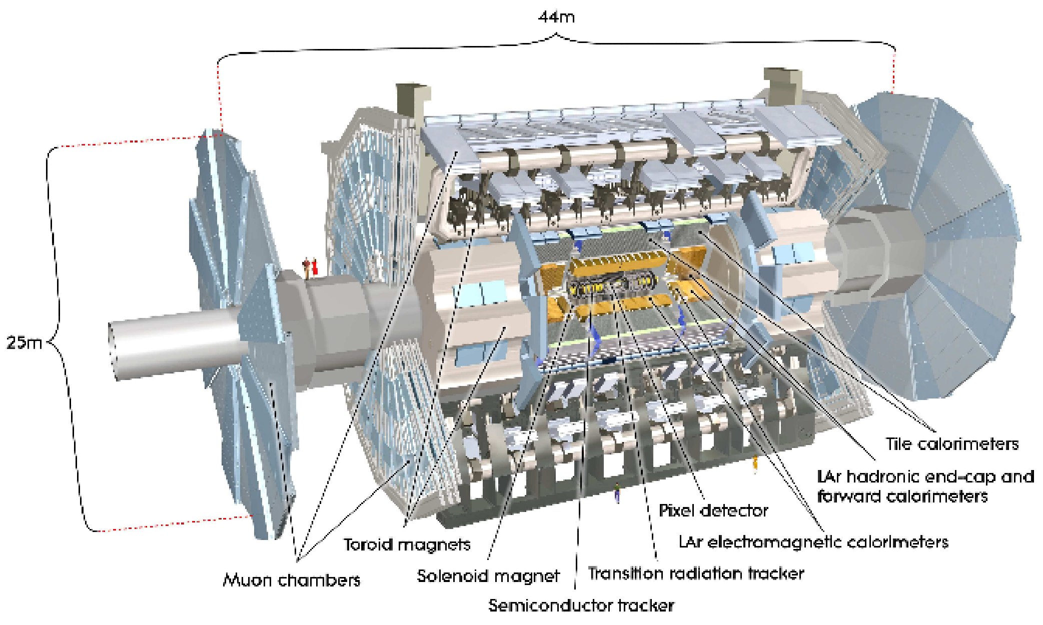

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Transition radiation tracker
Muon chambers
Solenoid magnet
Semiconductor tracker

Figure 2.2: An illustration of the ATLAS detector [18].

The detector is built up of three main layers. Two inner tracking layers, that provide the information about particle trajectories and allow to determine, with good resolution, the interaction point and secondary vertices[1]. The inner detector is composed of a pixel and strip silicon tracker and a transition radiation tracker. A good tracking resolution is in fact needed for particle track momentum determination and primary and secondary vertex measurement purposes: the whole inner detector is inserted in a solenoid magnet that generates a magnetic field along the beam direction. The bending trajectory of charged particles in magnetic field leads to a determination of momentum and electric charge of the

[1]Secondary vertices is the interaction vertices for particles that decay after the collision or collide into decays from other collisions.

particles.

The tracker is followed by two layers of calorimeters. The innermost is the electromagnetic calorimeter and consists of alternating layers of lead and liquid argon. The purpose of the layer is to stop incoming photons, positrons and electrons by inducing electromagnetic showers that allow to measure the energy of these particles. The outer calorimeter is the hadronic calorimeter, composed of three different parts. The hadrons produced in the event interact with the calorimeter material and produce hadronic showers, also called *jets*, which lose their whole energy in the hadronic calorimeter.

The outer layer consists of the muon chambers, which surround the whole detector as a barrel with two end-caps at the edges. Muons are the only detectable particles that are able to travel through all the other layers, only depositing a minimum ionisation energy in the detector material along the trajectory.

All the particles we can not track in the detector layers are referred to as missing energy/-momentum, essentially inferred from energy-momentum conservation and the measurement of all visible particle energy and momenta $E_{miss} = -\sum_i E_i$ and $\vec{p}_{miss} = -\sum_i \vec{p}_i$.

A sketch of the sub-detector layers presented above is shown in figure 2.3.

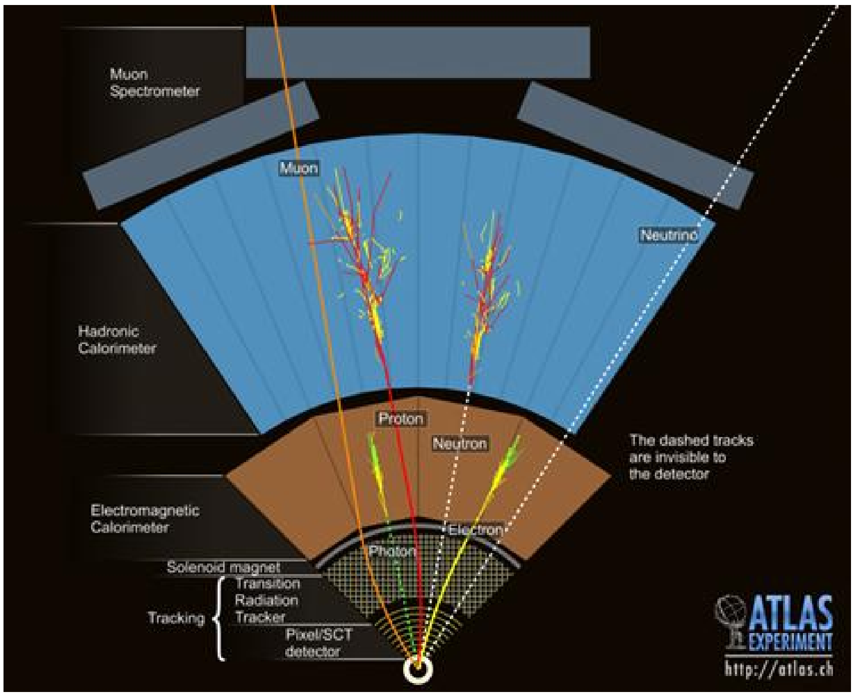


Figure 2.3: An illustration on how we see the tracks of the different particles in the detector [19].

2.3.1 Kinematic variables and the ATLAS coordinate system

By combining information from the different sub-detector layers it is possible to calculate various kinematical variables and identify the different particle species. In the beam direction, corresponding to the z-axis. We have two protons with opposite momenta p in each collision and therefore we are interested mostly in the transverse direction, where energy and momentum is conserved. The energy deposited by the particle is measured by the calorimeter and combined with the tracking information to have the vector quantities of p_T and E_T connected by the invariant mass m of the particle by the relation $p_T^2 = E_T^2 - m^2$. As mentioned before we can measure the difference in the transverse energy before and after the collision, which gives us the *missing transverse energy* (MET/E_T^{miss}).

It is also useful to introduce the detector coordinates used to describe an event. A sketch is shown in figure 2.4. The z-coordinate is defined by the beam direction and the x, y-coordinates define the transverse plane. In addition we have the two angles θ (polar) and

ϕ (azimuthal), being the angle between the particle and the z-axis and the particle and the x-axis, respectively. Note that instead of referring to the coordinate θ it is common to introduce the pseudorapidity η defined as

$$\eta = -\ln\tan\left(\frac{\theta}{2}\right). \tag{2.3}$$

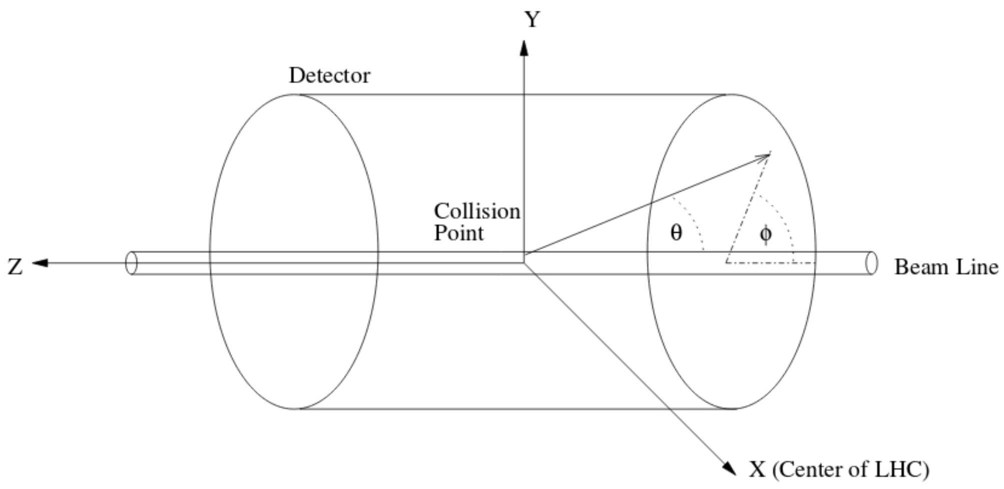


Figure 2.4: An illustration of the coordinate system inside the detector [20].

The other variables we are considering in this thesis, where we are interested in final states with two leptons l^+l^- and missing transverse momentum and energy, are listed below

- $m_{l^+l^-}$ is the invariant mass of the lepton pair in the final state, defined as

$$m_{l^+l^-} = \sqrt{(E_{l^+} + E_{l^-})^2 - (\mathbf{p}_{l^+} + \mathbf{p}_{l^-})^2}. \tag{2.4}$$

- m_{T2}, transverse mass defined as

$$m_T = \sqrt{2|\mathbf{p}_{T,1}| \cdot |\mathbf{p}_{T,2}| \cdot \big(1 - \cos(\Delta\phi)\big)}, \tag{2.5}$$

  where $\mathbf{p}_{T,1}$ and $\mathbf{p}_{T,2}$ are the transverse momentum vectors of the two leptons in the final state and $\Delta\phi$ is defined below.

- m_{T2} is the stransverse mass [21, 22] and is used to describe the masses of a particle pair that is assumed to have decayed to one visible and one invisible particle. It is defined as

$$m_{T2}(\mathbf{p}_{T,1}, \mathbf{p}_{T,2}, \mathbf{p}_T^{miss}) = \min_{q_{T,1}+q_{T,2}=p_T^{miss}} \left\{ \max \left[m_T(\mathbf{p}_{T,1}, \mathbf{q}_{T,1}), m_T(\mathbf{p}_{T,2}, \mathbf{q}_{T,2}) \right] \right\},$$

  where m_T is the transverse mass defined in equation **??** and $\mathbf{q}_{T,1}$ and $\mathbf{q}_{T,2}$ are vectors with $\mathbf{p}_T^{miss} = \mathbf{q}_{T,1} + \mathbf{q}_{T,2}$.

- H_T is the scalar sum of the p_T of the leptons we have selected and of the jets in the event.

- $\Delta\phi(\vec{p}_T^{ll}, E_T^{miss})$ is the difference between the azimuthal angles of the two-lepton system and the missing transverse energy direction.

- $\Delta R_{ll} = \sqrt{(\Delta\phi_{ll})^2 + (\Delta\eta_{ll})^2}$ is the distance between the two leptons in the final (ϕ, η) plane.

2.4 Computing infrastructure

The results obtained in this thesis have been very demanding when it comes to computing power. The reason for this is that the analyses consists of several searches using BDT and deep learning methods that require both training and optimization by using large data sets. BDT and deep learning require a lot of CPU-power and memory because of the size of the data. It has not been possible to run the various codes on a regular computer because of limited number of CPU's and memory. Because of these problems we were granted access to the Experimental Infrastructure for Exploration of Exascale Computing at Simula Research Laboratory: financed by the research council of Norway and made available to researchers. This is a computer with two sockets with 8 cores/CPU's, which again have two threads. This gives us in all 32 virtual CPU's (16 physical) because of hyper-threading in each CPU. It also has 60 GiB[2] memory, which have been crucial to handle the data used in the Machine Learning analysis. With this setup, the import of the data, training and testing have taken

[2]GiB is Gibibyte instead of regular gigabyte and is simply a unit byte for digital information and means 2 to the power of 10 (kiB), 20 (MiB), 30 (GiB), 40 (TiB) and 50 (PiB).

approximately 12-13 days, where 7 of these are just for importing the data. All together we have trained and tested 72 ML models (36 BDTs and 36 NNs) and the data sets we have been working on have been massive (almost 200 GB). Because of the huge amount of data, we need that the computer can handle this while training which is why we need the extra memory.

At the later stage of this thesis we made use of a special server that belongs to the ATLAS High Energy Particle Physics (HEPP) group at UiO. It is a Supermicro Ultra Server with both GPU's and CPU's, but we have only taken advantage of the CPU's in this thesis. This server is a much more powerful computer than the one from Simula. It has two sockets with 128 cores/CPU's, which also have two threads in each CPU. This gives a total of 256 virtual CPU's. It also has 2 TiB memory, which has resulted in that we could import the data in parallel and be done in around 3 days instead of a week. We have been able to train around 18-20 ML models at the same time instead of 1-2 which was the maximum for the Simula server.

In this chapter we have introduced the ATLAS detector and the LHC, which collected 13 TeV data between 2015 and 2018, corresponding to 139 fb^{-1}. The data analysis behind the searches for Supersymmetry and Dark Matter to be presented in this thesis make use of both traditional and ML-based algorithms, and necessitate a special computing infrastructure made of powerful CPU's, and GPU's.

Chapter 3

Search for Supersymmetry and Dark Matter in Events with Dileptons and Missing Energy

This chapter will introduce the search for new physics through various processes we explain, how the signal and background samples are produced, and how we perform a traditional so-called cut and count analysis when searching for new particles or phenomena in LHC data. We will show the results obtained by the ATLAS collaboration and compare them to results we otained following a similar strategy as in the publication.

In this thesis we are interested in processes involving superpartners of leptons, gauge bosons, and the Higgs boson. Besides this, we will look at a dark matter particle candidate, which is predicted by both SUSY, to be the lightest supersymmetric particle (LSP), and by simplified non-SUSY DM models requiring a new mediator V. This thesis looks at data from proton-proton collisions at the LHC in final states of two leptons and missing transverse energy. The SUSY processes we are looking at are direct slepton production, chargino production with slepton/sneutrino-mediated-decays and with W-boson-mediated-decays.

Figure 3.1 shows direct slepton production with the sleptons decaying to a final state with two leptons and missing transverse energy (MET/E_T^{miss} i.e. missing energy in the detector) from the lightest neutralinos ($\tilde{\chi}_1^0$). The neutralino is assumed to be stable and not measured directly by the detector. The energy of the neutralinos is therefore interpreted as MET in this process. The neutralino is a mixture of the sparticle components photino, zino, and neutral higgsino. Since it is believed to be 100% stable it constitutes a perfect dark matter candidate as mentioned above.

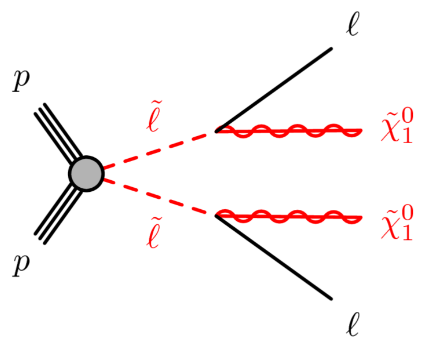


Figure 3.1: Direct slepton production $pp \to \tilde{l}^+\tilde{l}^- \to l^+l^- + \tilde{\chi}_1^0\tilde{\chi}_1^0$.

In Figures 3.2 and 3.3 chargino production with slepton/sneutrino-mediated-decays and W-boson-mediated-decays are shown, respectively. Charginos are a mixture of the sparticle components wino and the charged higgsino. These processes have the same final state as direct slepton production, but here the neutrinos also contribute to the MET, since they connot be observed in the detector.

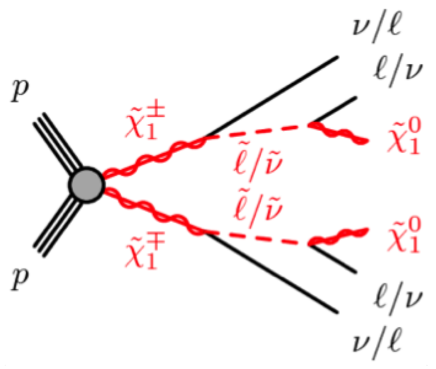


Figure 3.2: Chargino production with slepton/sneutrino-mediated-decays $pp \to \tilde{\chi}_1^+\tilde{\chi}_1^- \to \tilde{l}^+\tilde{l}^-/\tilde{\nu}\tilde{\nu} \to l^+l^- + \nu\bar{\nu} + \tilde{\chi}_1^0\tilde{\chi}_1^0$.

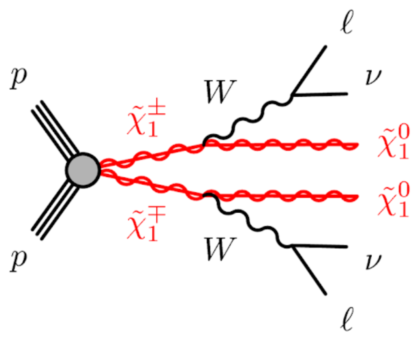


Figure 3.3: Chargino production with W-boson-mediated-decays $pp \to \tilde{\chi}_1^+\tilde{\chi}_1^- \to W^+W^- \to l^+l^- + \nu\bar{\nu} + \tilde{\chi}_1^0\tilde{\chi}_1^0$.

The DM process we are looking at in this thesis is the mono-Z process shown in figure 3.4. Here we have a new mediator V between matter ($q\bar{q}$) and DM (two particles χ). In addition we require a Z-boson, radiated from one of the initial state particles[1], which subsequently decays into two leptons. This gives us the same final state as we had for the SUSY processes above.

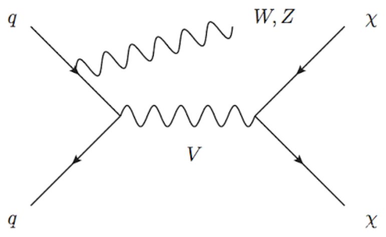


Figure 3.4: Mono-Z process $pp \to Z + MET \to l^+l^- + MET$.

3.1 Monte Carlo simulated events

The data considered is recorded by the ATLAS experiment at the LHC between 2015 and 2018 (Run 2), presented in chapter 2. But, we are also looking at MC simulated SM backgrounds and new physics signals which will be explained in this section, taken from the publications from ATLAS, namely [9] for the SUSY signals and [11] for the mono-Z signal. Tables A.1 - A.12 in section A present an overview of the signal samples that are used.

[1]Initial state radiation means that one of the incoming particles emits a particle before the annihilation, e.g. the Z-boson in our process.

The SUSY signal samples were generated from leading-order (LO) matrix elements with up to two extra partons using MadGraph5_aMC@NLO 2.6.1 [23] interfaced to Pythia 8.186 [24], with the A14 tune [25], for the modelling of the SUSY decay chain, parton showering, hadronisation and the description of the underlying event. Parton luminosities were provided by the NNPDF2.3LO PDF set [26]. Signal cross-sections were calculated to next-to-leading order (NLO) in α_s. The nominal cross-sections and their uncertainties were taken from an envelope of cross-section predictions using different PDF sets and factorisation and renormalisation scales, as described in Ref. [27].

The DM signal is modelled with the leading-order MadGraph5_aMC@NLO matrix element [28] using NNPDF3.0 [29] and showered with Pythia8.186. DM signal events with an axial-vector[2] mediator and fermionic WIMPs (weakly interacting massive particles) are produced for different mediator and DM masses m_V and m_χ, both in a range from 10 to 1000 GeV. As recommended in Ref. [30], the DM events are generated by choosing couplings to quarks $g_q = 0.25$, and to DM $g_\chi = 1$, and a minimal mediator width. The A14 [31] parameter set is used to tune the Pythia8.186 parton-shower for the simulation of the DM signal.

The different SM backgrounds we consider are diboson, triboson, $t\bar{t}$, single top, other top events ($t\bar{t}$ events with a pair of leptons or boson(s)), Higgs, Drell-Yan, Z+jets and W+jets. The MC samples are simulated using different generators that are listed in table 3.1. The goal is to separate these backgrounds from the new physics signal processes discussed earlier in the chapter.

[2]An axial-vector is the cross-product of two vector quantities, which will not change sign under parity transformations because both $\mathbf{v}_1$ and $\mathbf{v}_2$ do. E.g. angular momentum $\mathbf{L} = \mathbf{x} \times \mathbf{p}$, where $\mathbf{x}$ is position and $\mathbf{p}$ is momentum.

Background sample	**Generator**	**Parton shower**	**Normalisation**
Diboson	Sherpa2.2.2 [32–34]	Sherpa2.2.2	NLO [35]
Triboson	Sherpa2.2.2	Sherpa2.2.2	NLO
Z+jets	Sherpa2.2.1 [33, 34, 36]	Sherpa2.2.1	NNLO [37]
W+jets	Powheg-Box v2 [38, 39]	Pythia8.186 [24]	NLO
Drell-Yan	Sherpa2.2.1	Sherpa2.2.1	NNLO
$t\bar{t}$	Powheg-Box v2	Pythia8.186	NNLO
Single top	Powheg-Box v2	Pythia8.186	NLO
topOther	MG5_aMC@NLO [23]	Pythia8.186	NLO
Higgs	Powheg-Box v2	Pyhtia8.186	NLO

Table 3.1: An overview of the different generators used to simulate the MC background samples.

Before we move to the analysis searching for SUSY and DM signals exploiting machine learning techniques we need to make sure that our input (i.e data, SM background and new physics signal MC) looks reasonable. We also need a baseline analysis to check whether or not the ML analysis perform better than the more standard cut and count analysis, which will be outlined in the following sections.

3.2 The standard way: cut and count

The first part of the analysis done in this thesis is a traditional cut and count analysis. *Cut and count* is probably the most known method used in particle physics and has proved to be very useful in the discoveries we have done so far. Since the data become more and more massive and complex, and the processes we are looking at more and more complicated, we also need to develop further and improve the way we perform the analysis. In this thesis, we are mainly going to focus on machine learning algorithms. Therefore we have not tried to improve this standard way to analyze data and have based the cut and count on already published analyses from ATLAS [9, 11].

In cut and count, we select events sensitive to new physics, by reducing as much as possible any SM background processes which could mimic the signal. After applying several cuts, the selection of events we are left with form the so-called *signal region.* We then see whether the expected signal is significantly separated from the expected Standard Model (SM) background in this region. We can calculate an expected significance Z, which will be explained in section 6.1 later in the thesis, to check if we can expect to claim a discovery in this region if a particular signal model turns out to be realized in nature. If we are lucky and have cut away enough background and kept sufficient signal, we can check if the observed events in data are compatible with the signal+background hypothesis or if they match the background-only hypothesis (i.e. no signal) instead. Let us consider the case where the data differ from the background and tends to follow the signal: we know that there is most likely something interesting in this region.

Of course, there are advantages and disadvantages with every method, and this is also the case for cut and count. In cut and count, we need a theory or hypothesis as a reference to know what kind of signals we should look for. We are also only able to do cuts in one or two dimensions and adjust the different variables to our purposes to a certain complexity. It is therefore unfortunately limited by the human understanding of what we are looking at. The lack of human understanding is where Machine Learning (ML) comes to help. The ML methods are expected to help us better separate the signal from the background and can look at the data in several dimensions and with more complexity. This is further explained in the following chapters, where we will look at what the different ML methods do.

3.3 Reproducing the ATLAS publications

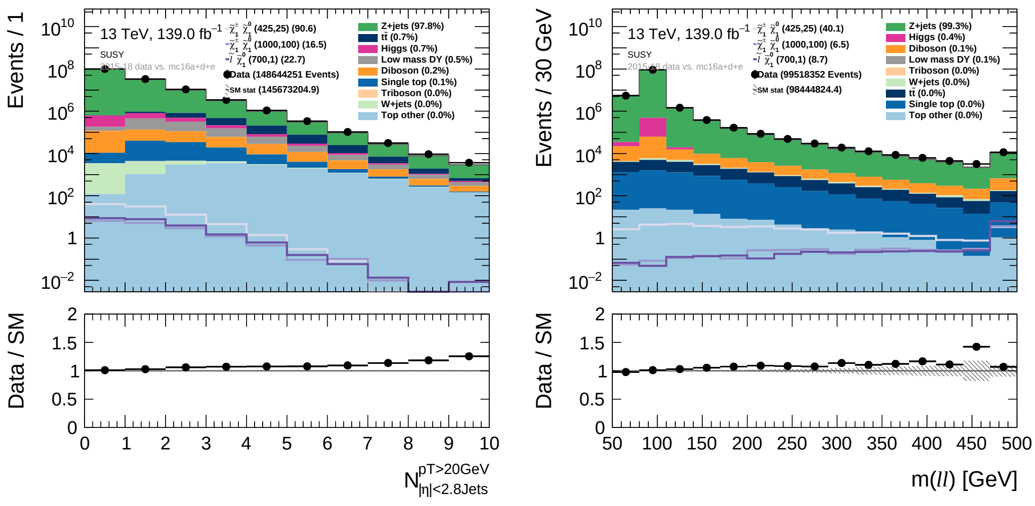
The first part of this analysis was done by cut and count and the goal was to reproduce the results from publications done by ATLAS [9, 11]. Here we will compare our results to the official results from the ATLAS experiment, and hopefully achieve a good agreement between the two.

Since the goal is to reproduce the results, we implement more or less the same analysis procedure as described in the publications. The cuts done for the SUSY processes are listed in table 3.2 and for the mono-Z process in table 3.4, where both tables are taken from the publications [9, 11]. All the kinematic variables we cut on are presented in chapter 2, except the tagging of b-jets which are jets initiated by bottom quarks. All of the results are presented with a systematic uncertainty. The first cut we do for both processes is to demand exactly two leptons with opposite signs in the final state.

Variables	**Cuts**
Two leptons	Same flavor (SF) and opposite sign (OS)
n_{jets}	0
m_{ll} [GeV]	121.2
E_T^{miss} [GeV]	> 110
E_T^{miss} significance	> 10
m_{T2} [GeV]	160

Table 3.2: Cuts added in the cut and count analysis taken from the publication for the SUSY processes [9].

For the SUSY processes we have applied the cuts in table 3.2, where we, in addition to having only two leptons with opposite sign in the final state, demand that they have to have the same flavor as well. Here we get the Z+jets as the dominating background as we can see in both figure 3.5a and table 3.3. Now we want to reduce all of the background, especially Z+jets, and we apply the next cut in table 3.2 which is demanding no jets (both b-tagged and non-b-tagged). As we can see in figure 3.5b, the Z+jets background is still the dominating background, but if we look at table 3.3, we can see that it is reduced a lot. The reason for Z+jets is still dominating is that the jet-cut reduced around the same percentage from all of the different backgrounds.

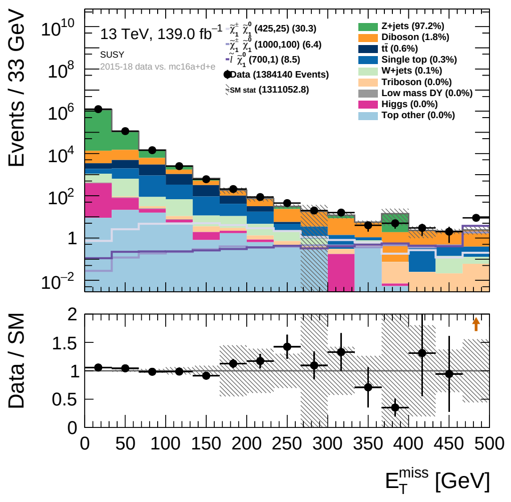


(a) Number of jets with a $p_T > 20$ GeV.

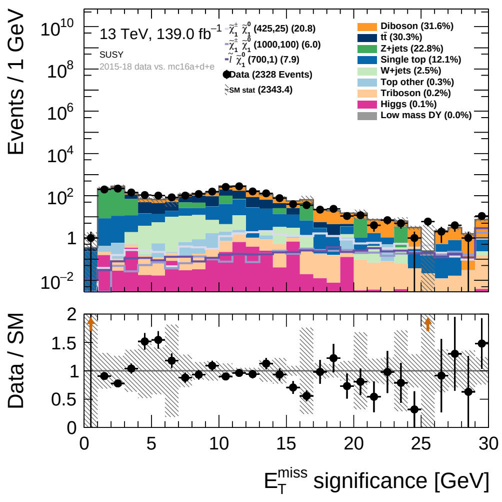
(b) The invariant mass of the two leptons.

Figure 3.5: Plot of different distributions after applying the cuts on 2L, SF, OS (a) and no jets (b).

The next cut we have applied is on the invariant mass of the two leptons in the final state. The results are shown in figure 3.6a and as we can see, most of the background is reduced by a lot. We have also done a cut requiring large missing transverse energy. This is done because it cuts away more background than signal, which entails obtaining a more significant separation between the signal and background. By applying this cut, we can see that the Z+jets are no longer the dominating background and the results are shown in figure 3.6b and table 3.3.



(a) Missing transverse energy.



(b) Missing transverse energy significance.

Figure 3.6: Plot of different distributions after applying the cuts on the invariant mass (a) and MET (b).

The last two cuts applied is a cut on the MET significance and m_{T2}. The results after applying the MET significance cut is shown in figure 3.7 and as we can see, the diboson is still the dominating background. The last cut that are applied for the SUSY processes are on the m_{T2} variable. This is done to get rid of the rest of the $t\bar{t}$ background and leave us more or less with only diboson. This is part of our final result and are shown in figure 3.14 later in this section.

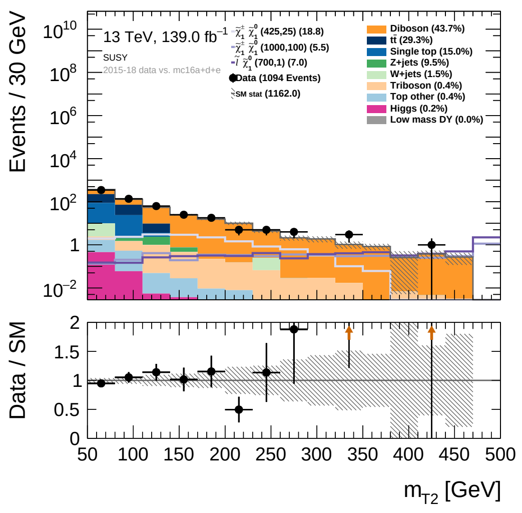


Figure 3.7: Plot of the distribution of m_{T2} after applying the cuts on MET significance.

As we can see in figure 3.5-3.7 and table 3.3, the signal have been reduced, but not as much as all of the background contributions. This implies that we have been able to get rid of the background without affecting the signal too much, which also was our goal by doing this.

Sample	**OS+SF+2L**	**jet-veto**	$\mathbf{m_{ll}}$	**MET**	**MET sign**	$\mathbf{m_{T_2}}$
Drell-Yan	733055.427	64177.536	40.384	0.000	0.000	0.000
Higgs	1055360.610	442535.035	451.966	3.222	2.411	0.005
Single top	96745.087	7383.056	3313.872	283.960	174.644	0.000
$t\bar{t}$	948716.615	16272.119	7648.974	710.246	340.903	0.000
Z+jets	142460122.600	97803688.386	1273912.476	533.182	109.925	-0.004
Top other	14730.742	120.671	56.003	8.079	4.353	0.034
W+jets	8978.117	3252.185	1359.621	58.245	17.834	0.182
Triboson	345.791	65.545	25.860	5.757	4.633	0.653
Diboson	355149.910	107329.899	24243.651	740.741	507.329	42.588
Data	148644251.000	99518352.000	1384140.000	2328.000	1094.000	40.000
$(\tilde{l}, \tilde{\chi}_1^0)(700, 1)$	22.710	8.692	8.519	7.873	6.987	6.062
$(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0)(1000, 100)$	16.475	6.546	6.355	5.997	5.468	4.347
$(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0)(425, 25)$	90.581	40.109	30.305	20.816	18.756	6.692

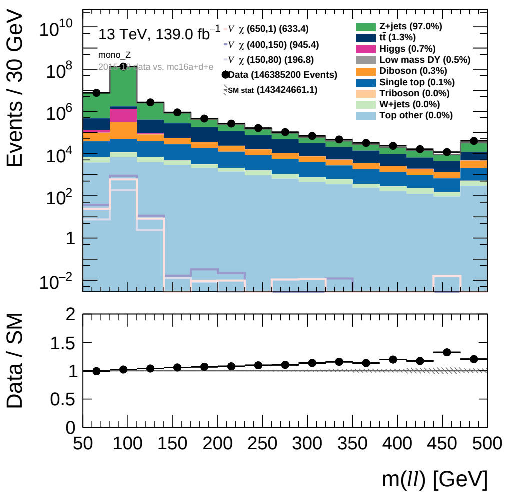
Table 3.3: A cut flow overview after applying the different cuts in table 3.2 with one signal sample from each of the three SUSY processes.

The same procedure was done for the mono-Z process, where we have applied the cuts from table 3.4 and we can see how much each cut affect the different contributions in table 3.5.

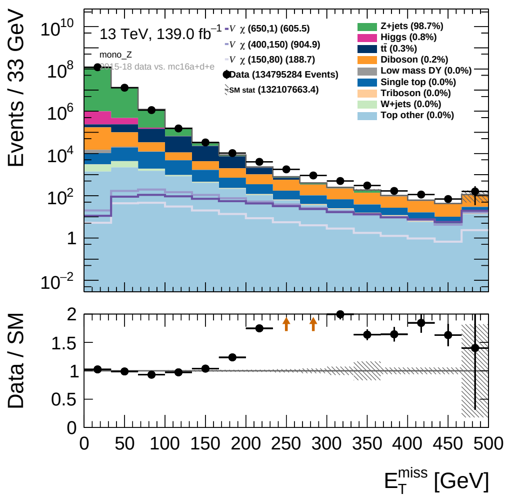
Variables	**Cuts**
Two leptons	OS with leading (subleading) $p_T > 30$ (20) GeV
m_{ll}	$76 < m_{ll} < 106$ GeV
E_T^{miss}	> 90 GeV
E_T^{miss}/H_T	> 0.6
$\Delta\phi(\vec{p}_T^{ll}, E_T^{miss})$	> 2.7 radians
ΔR_{ll}	< 1.8
Fractional p_T difference	$\lvert p_T^{ll} - p_T^{miss,jets}\rvert / p_T^{ll} < 0.2$
b-jets	0

Table 3.4: Cuts added in the cut and count analysis taken from the publication for the mono-Z process [11].

For the DM process, we have applied the cuts in table 3.4, where we, as for the SUSY processes, demand to only have two leptons with opposite sign in the final state together with missing transverse energy. In addition to the cut on number of leptons, we cut on the p_T of the leptons, for both the leading and subleading lepton. The cut on the subleading lepton will not affect anything because it is already done a cut at 25 GeV while handling the data for this thesis. This is shown in figure 3.8a and table 3.5, where we can see that the distribution looks very similar as for the SUSY processes earlier in this chapter. We have demanded to have a Z-boson which we can see the results from in figure 3.8b. For both these cuts, we can see that Z+jets are the dominating background, where all the different backgrounds are reduced.



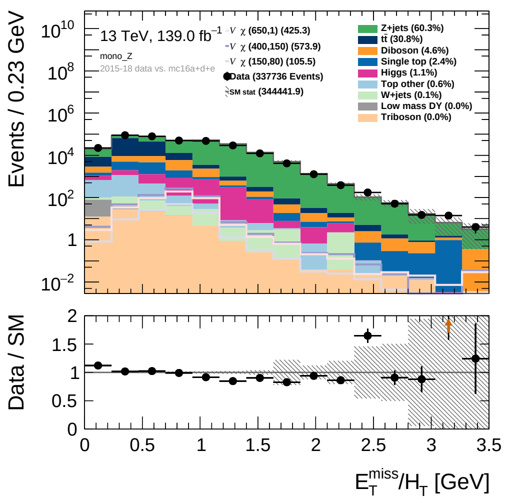
(a) Invariant mass.



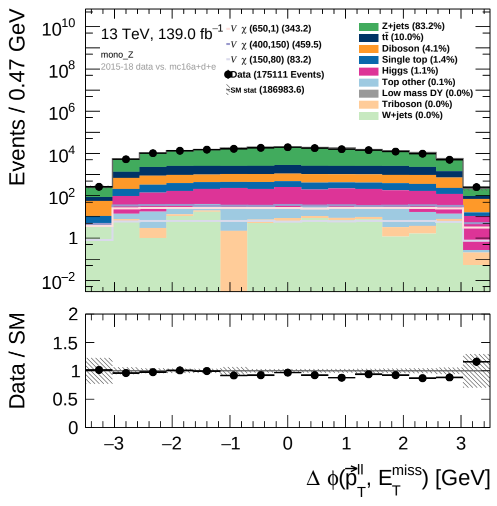
(b) Missing transverse energy.

Figure 3.8: Plot of different distributions after applying the cuts on 2L, OS, p_T of the two leptons (a) and invariant mass (b).

We also do a slightly more gentle cut on the missing transverse energy for this process than the SUSY processes because we have several other MET dependent variables for mono-Z. One of these are E_T^{miss}/H_T. After applying the MET cut, we can see that the Z+jets are less dominating, but since the MET/H_T reduces the $t\bar{t}$ background, the Z+jets becomes more dominating again. The results are shown in figure 3.9.



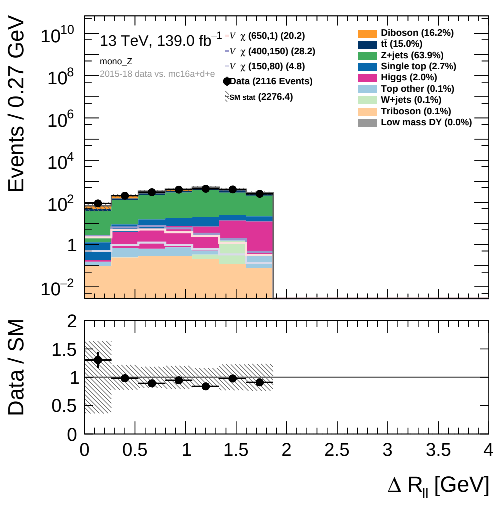
(a) The ratio between E_T^{miss} and H_T.



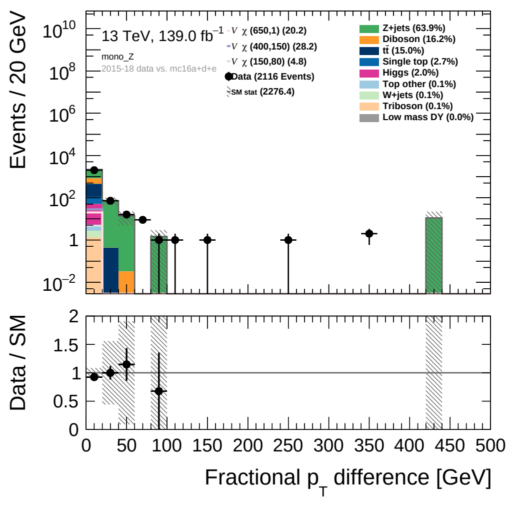
(b) The azimuthal angle difference between the dilepton system and E_T^{miss}.

Figure 3.9: Plot of different distributions after applying the cuts on MET (a) and MET/H_T (b).

The next cut that is applied is $\Delta\phi(\vec{p}_T^{ll}, E_T^{miss})$, where the two leptons also have to be close to each other, which can be demanded by ΔR_{ll}. $\Delta\phi$ is also one of the MET dependent variables we have used for the mono-Z process. The results after applying these cuts are presented in figure 3.10 and as we can see, the Z+jets keeps being the dominating background.



(a) Distance between the two leptons.



(b) The difference between the dilepton p_T, the selected jets p_T and the vector sum of $\vec{E}_T^{miss}$.

Figure 3.10: Plot of different distributions after applying the cuts on $\Delta\phi$ (a) and ΔR_{ll} (b).

The last cuts we apply are on the fractional p_T difference and a b-jet veto. The results from adding the p_T cut are presented in figure 3.11, while the final results including b-jet veto is presented in figure 3.14 later in this section. We can also see for both these cases that the dominating background have become diboson, which is the same as for the SUSY processes.

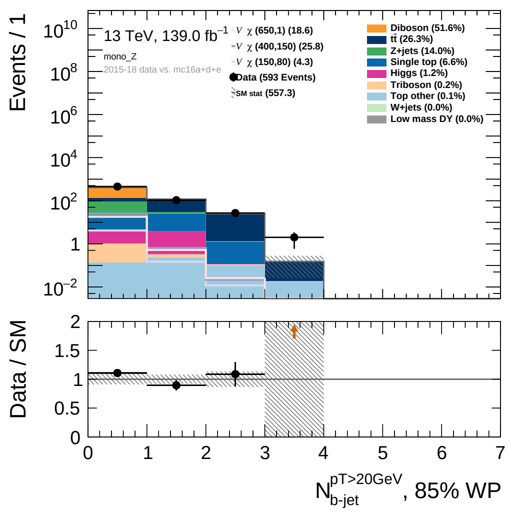


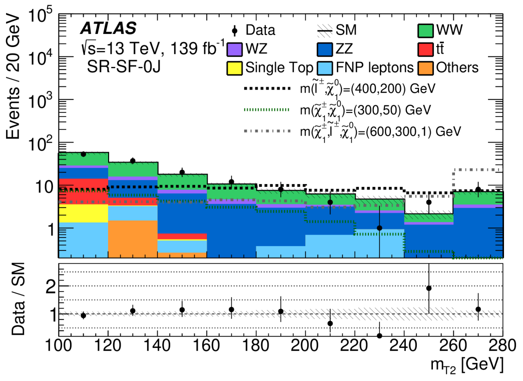
Figure 3.11: Plot of the distribution of number of b-tagged jets after applying the cuts on fractional p_T difference.

For the signals in this case, which is maybe easier to see in table 3.5, we can see that it is reduced a lot. This is of course not what we want to obtain, but it is also expected after doing so many cuts as we have done. In the end of this we are going to calculate the expected significance, where we can see how much sensitivity we actually have to the signal.

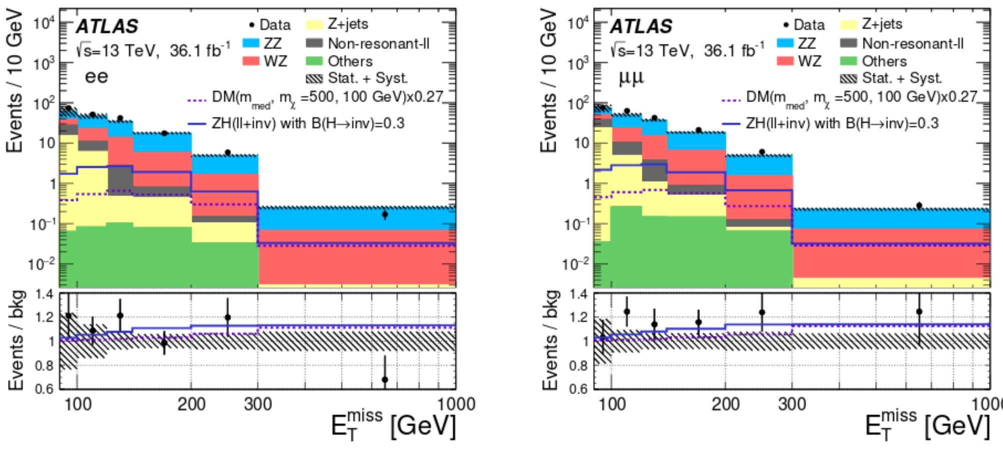
Sample	**2L + OS + $\mathbf{p_T}$**	**$\mathbf{m_{ll}}$**	**MET**	**met/$\mathbf{H_T}$**	**$\boldsymbol{\Delta\phi}$**	**$\mathbf{\Delta R_{ll}}$**	**$\mathbf{p_T}$ diff**	**b-jets**
DY	674647.170	7035.028	79.252	0.448	0.000	0.000	0.000	0.000
Higgs	1052061.840	1014012.858	3834.027	2116.006	73.416	46.198	6.942	3.467
Single top	189826.764	38267.988	8246.807	2620.670	73.820	60.859	36.895	14.845
ttbar	1854383.568	398898.217	106187.071	18667.204	439.750	342.455	146.530	42.570
Zjets	139141315.583	130368076.145	207856.769	155565.299	2613.607	1454.461	77.839	72.669
Top other	26411.801	6865.785	2104.230	211.226	3.570	1.880	0.492	0.136
Wjets	17813.414	4191.918	198.881	72.288	5.306	1.421	0.000	0.000
Triboson	528.623	211.477	81.766	27.851	1.406	1.332	0.999	0.900
Diboson	467672.290	270103.987	15853.093	7702.570	394.866	367.781	287.632	278.123
Data	146385200.000	134795284.000	337736.000	175111.000	3287.000	2116.000	593.000	457.000
$(V,\chi)(150,80)$	196.844	188.682	105.505	83.150	4.866	4.765	4.325	4.166
$(V,\chi)(400,150)$	945.406	904.916	573.904	459.476	28.740	28.194	25.789	25.124
$(V,\chi)(650,1)$	633.417	605.474	425.291	343.235	20.533	20.193	18.566	18.019

Table 3.5: A cut flow overview after applying the different cuts in table 3.4.

All of the cuts are applied to get fewer background events while we, at the same time, do not want to cut away too much of the signal events. The publications have applied more or less the same cuts as listed above in table 3.2 and 3.4, and their results are presented in figure 3.12.



(a) Stransverse mass for direct slepton production, chargino production with $\tilde{l}/\tilde{\nu}$-mediated decays and with W-boson-mediated decays.



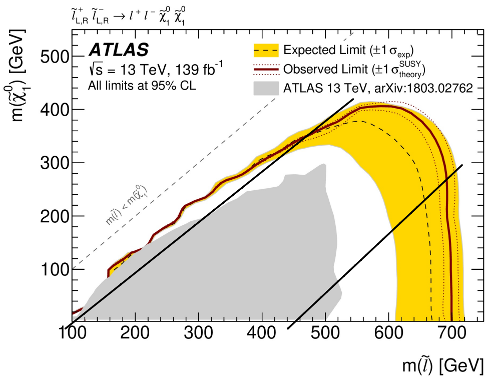
(b) Missing transverse energy for the mono-Z process, where the left plot is the electron channel and right is the muon channel.

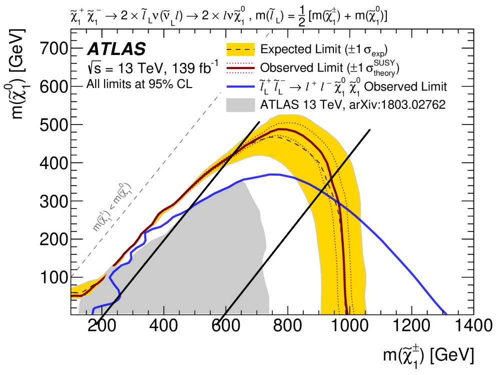
Figure 3.12: Results from the ATLAS publications for the four processes considered in this thesis.

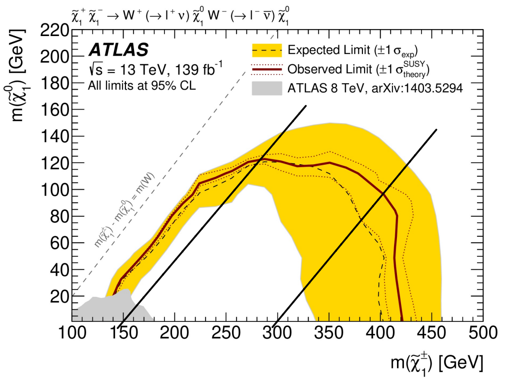
In figure 3.12a, we can see that the signal is separated from the background for both the direct slepton production and the chargino production with W-boson-mediated-decay. However, they have not obtained any significant separation for chargino production with slepton/sneutrino-mediated-decay, which means that we should not expect to claim any discovery for the signal model shown in these plots.

In figure 3.12b, we can see the plots for both electron channel and muon channel for the mono-Z process. It is not as much separation between signal and background as for the direct slepton production and the chargino production with W-boson-mediated-decay, but there is some.

Later in this thesis, we will calculate the expected significance for the different processes we are looking at with both cut and count and ML. This is not done in the publications we are looking at, but they have included an exclusion plot which we can use for comparison later. This is shown in figure 3.13. Here we can see the expected exclusion limit with $\pm 1\sigma$ uncertainty bin together with the observed exclusion limit. The expected exclusion curve in these plots follow the boundary where the significance is 1.36 (i.e. 95% CL exclusion). We are not going to reproduce these results in this thesis. However, we have used it to pick out some benchmark signals around the expected exclusion limit.


(a) Slepton pair production ($\tilde{l}\tilde{l}$) [9].


(b) $\chi_1^+\chi_1^-$ production via $\tilde{l}/\tilde{\nu}$ [9].


(c) $\chi_1^+\chi_1^-$ production via W-boson [9].

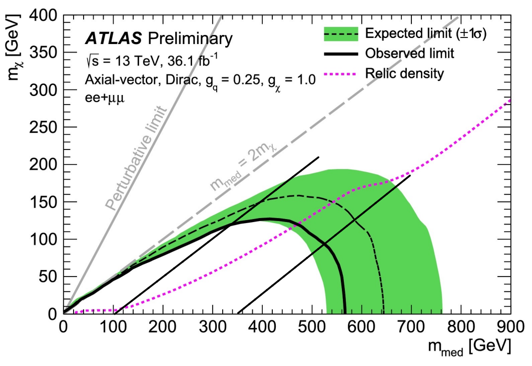

(d) Mono-Z process [11].

Figure 3.13: The observed and expected exclusion limits for both the SUSY simplified models in 3.13a, 3.13b and 3.13c and for the DM model in 3.13d. The lines drawn in the plot is to show the mass splittings for each process.

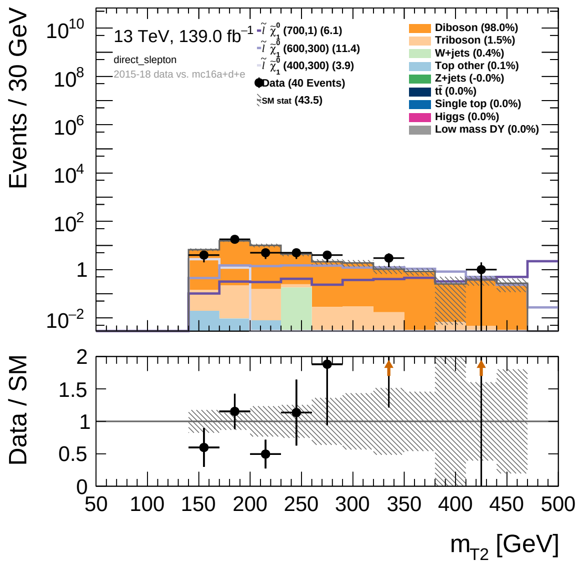
The two black lines in each plot in figure 3.13 shows the division of the signal samples into different mass splittings in the ML analysis later in the thesis. To compare our ML results with the cut and count analysis, we have chosen one representative signal sample from each part of the plots in figure 3.13. This gives us the signal samples listed in table 3.6.

$\mathbf{m}(\tilde{\mathbf{l}}, \tilde{\chi}_1^0)$	$\mathbf{m}(\tilde{\chi}_1^\pm, \tilde{\chi}_1^0)$	$\mathbf{m}(\tilde{\chi}_1^\pm, \tilde{\chi}_1^0)$	$\mathbf{m}(\mathbf{V}, \chi)$
(400, 300)	(300, 200)	(150, 25)	(150, 80)
(600, 300)	(800, 400)	(350, 100)	(400, 150)
(700, 1)	(1000, 100)	(425, 25)	(650, 1)

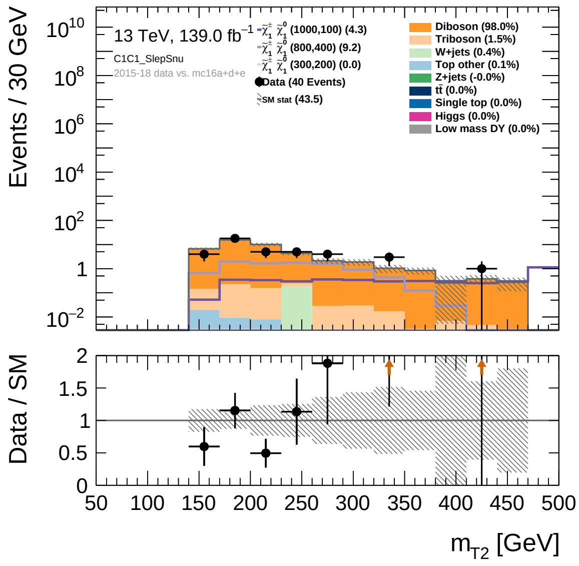
Table 3.6: An overview of the masses of the signals that are used for this thesis.

These signal samples are going to follow us through both analysis (cut and count and ML). They are also shown in figure 3.14 where we have followed the same procedure as the ATLAS publications. The main difference between the publications and our results is that we have chosen some other signal samples than the publications. The reason for this is because we want to have one benchmark signal from each mass splitting made from looking at figure 3.13. The publications have $m(\tilde{l}, \tilde{\chi}_1^0) = (400, 200)$ GeV, $m(\tilde{\chi}_1^\pm, \tilde{\chi}_1^0) = (300, 50)$ GeV, $m(\tilde{\chi}_1^\pm, \tilde{\chi}_1^0) = (600, 300)$ GeV and $m(V, \chi) = (500, 100)$ GeV. Since we have chosen to look at different signal samples than the publications [9, 11], we have to see if the backgrounds for all four processes are compatible.

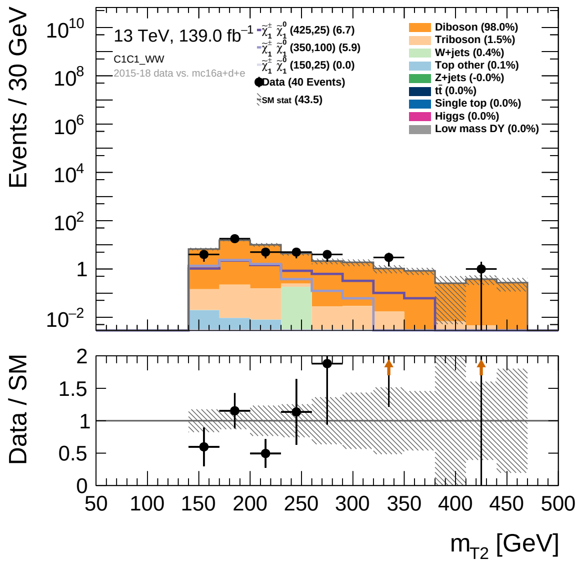
In figure 3.14 we can see our final results for the same variables as in the publications, from the cut and count analysis.



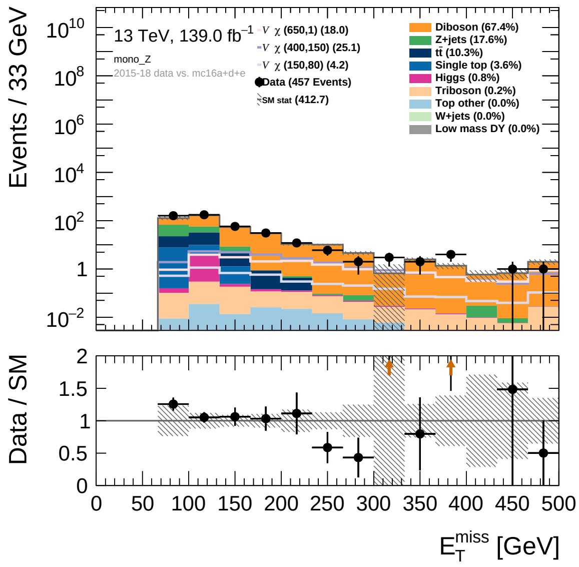
(a) Stransverse mass for direct slepton production.



(b) Stransverse mass for chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Stransverse mass for chargino production via $W^{\pm}$.



(d) Missing transverse energy for mono-Z.

Figure 3.14: Results from the cut and count analysis for all four processes, where we have followed the same procedure as the publications done by ATLAS.

In figure 3.14 we can see that for the direct slepton production and for the chargino production with slepton/sneutrino-mediated-decay we are able to separate some signal samples from the background in the high end of the m_{T_2} variable. In 3.14a both the high and intermediate mass splitting samples are separated from the background and could lead to detection. In 3.14b only the high mass splitting sample is above the background and could be possible to detect. While for the chargino production with W-boson-mediated-decay and mono-Z we have not been able to separate the signal from the background. In comparison to the results from the publications (shown in figure 3.12), we can see that we have roughly the same amount of events in the different bins and can therefore conclude with that they are compatible and we have managed to reproduce the published results.

Now we are going to move on to the ML part, which is the main part of this thesis, and see if we can do the same analysis with some different tools. In the end we will compare the results obtained in this chapter to see if ML can do a better separation than we have managed to do with cut and count. Before presenting the ML results in chapter 5, we first give an introduction to Machine Learning in chapter 4.

Chapter 4

Introducing Machine Learning

In this chapter, we will introduce the different Machine Learning algorithms which are used in the analysis.

The future of scientific experiments faces a problem: on one hand, we collect more data than we have ever done in the history of science. It opens up a whole new range of possibilities and increases the potential for new discoveries. On the other hand, the quantity of data is so vast that no human being, or group of human beings, could hope to sift through and analyze all the data in a single lifetime. However, it's not good enough to have a "dumb" computer algorithm sort through the data, since if the computer cannot adapt and pick out interesting features, a human would need to double check the results anyway. We need a technique that is both efficient and "smart". For this reason, and others, Machine Learning (ML) is fast becoming the standard way of analyzing data in science, and in high energy physics in particular.

Indeed, compared to the cut and count method, ML techniques often provide a more efficient way to perform analyses. It can handle massive datasets and give results in a reasonable time frame. This is needed since the datasets have become increasingly large after each experimental upgrade at for example the LHC. The ML methods may be better at distinguishing between background and signal since they have the potential to pick up characteristics in the data which are difficult for a human to identify (because it might pick out characteristics in the data that a human would never notice), which we are trying to optimize in our analysis. The regular cut and count analysis is implemented by a person, and they analyze the data. For us, however, we will try to separate the signal from the background using ML algorithms. A discussion of these is presented in the following sections.

Broadly speaking, an ML algorithm can be explained as shown in figure 4.1. We have an input variable X that we send into an ML model of our choice, and we get the actual output Y (the ML output) and the predicted output Y' (the output we expect). Let us say that we give the ML model a variable with label 1; then the predicted value is 1 and the actual output will (hopefully) be either 1 or close to 1.

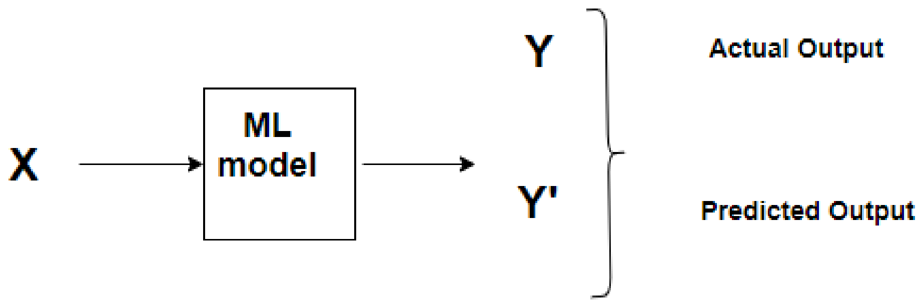


Figure 4.1: A very simple illustration of an ML algorithm [40].

4.1 Machine Learning basics

In this thesis, we are focusing on using ML algorithms to separate the signal from the background. This is done by the ML algorithms called Boosted Decision Trees (BDT) and Neural Networks (NN). Before we go into the specifics, we will give a short introduction to some ML basics.

4.1.1 Training and testing

To obtain our BDT and NN results, we have to split our datasets into training and test sets. We split the dataset into 2/3 for training and 1/3 for testing. It is common practice to divide the train and test set like this, but it is also possible to choose a smaller training set and a larger test set. The reason for doing this is that we want to check if our model is learning and is able to classify the background as background and the signal as signal - before we, in the end, give it actual data.

Validation test

In addition to a standard test set, we need a validation set to avoid overtraining (this is not used to test whether the model works, but rather to prevent overfitting - see below). A validation set is a set which hasn't been used to train the algorithm. We use the validation set to see when the model cannot do better with calculating a validation loss, which is the loss for the validation set - and will be explained later in this chapter - for each epoch or estimator the network or BDT goes through. An epoch is the number of cycles the NN does the training and estimator is the number of the times we want to boost the tree. If the validation loss does not become better after e.g., ten steps, another function breaks the training before it has finished all of the epochs or estimators. This is done by a function called *early stopping*, which stops the training. This is very useful for the NN and BDT since there is one less parameter we have to worry about, namely the epochs or estimators. We only have to set a value high enough for the model to get far enough through the training.

4.1.2 Overfitting and underfitting

In ML we encounter a very common problem while training our model, namely overfitting and underfitting. The goal when training a model is to get it to fit the data as well as possible; but this training can't be too precise or the model will be useless if your data doesn't look exactly the same the next time you are using it. That would lead to loss of generality and our model would become dataset specific. This is overfitting. Underfitting is of course the opposite problem: the model is too simple to adapt to slightly more complex datasets, and the algorithm will simply use a simple function to fit data. These ideas are very well illustrated in figure 4.2.

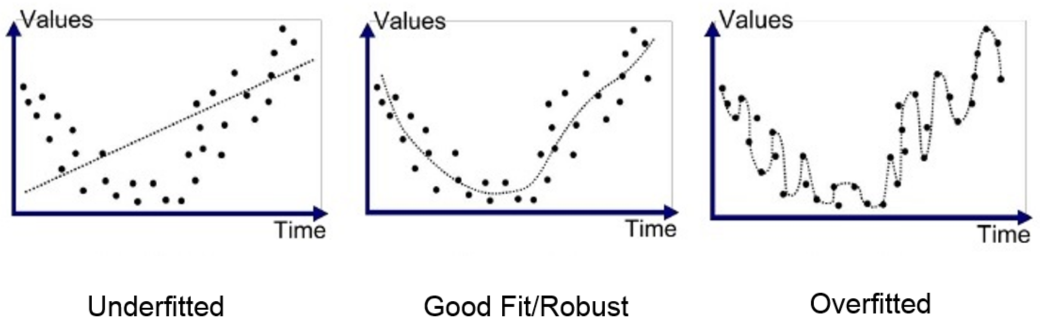


Figure 4.2: An illustration of underfitting and overfitting [41].

The goal in our case is to optimize the ML model so that it efficiently can separate signal from background. If we overfit, it means that it can only find the signal for which we have trained on, and if it turns out that the new physics we are looking for is slightly different, our network will not be able to distinguish it from background. This is particularly important in our case since we train the network on simplified SUSY/DM models which we know are not the true realization of new physics in nature, but we believe that they have similar phenomenology. However, if we overtrain, our ML model will not be able to find any possible new physics in the data which looks slightly different from what we have trained on.

4.1.3 Evaluating metrics

Accuracy

In ML we have many ways to measure how well or badly our algorithm is performing; one of them is the *accuracy* [42]. The accuracy expresses in percentage how accurate your algorithm is performing and can be calculated as

$$accuracy = \frac{correct}{correct + incorrect}, \tag{4.1}$$

where *correct* is all data points that are classified as true positives (TP) and *incorrect* as false positives (FP)[1]. This could be the only metric needed to evaluate a model; it gives a very basic measure on how, well, accurate - or reliable - the ML model is.

Loss and cost function

Earlier we mentioned the validation loss, which is calculated for the whole training set during training. The loss is calculated for each estimator (BDT) and epoch (NN). The loss can be calculated in many different ways and we should choose the one best suited for the problem. In our case we want to classify the background and signal as two variables, namely 0 and 1. This means that we are looking at a binary classification problem and we can use a binary classification loss function [43], namely binary *cross-entropy*. Mathematically it can be derived as follows. We have a true probability p_i and a distribution of the predicted values q_i. The probability to get the outcome $y = 1$ is given by

$$q_{y=1} = \hat{y} = g(\mathbf{w} \cdot \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}, \tag{4.2}$$

where $\mathbf{x}$ is a vector of input features, $\mathbf{w}$ is a vector of optimized weights and $g(\mathbf{w} \cdot \mathbf{x})$ is a logistic function.

To find the probability of $y = 0$ we can write

$$q_{y=0} = 1 - \hat{y}. \tag{4.3}$$

This gives us the notation setup, $p \in \{y, 1 - y\}$ and $q \in \{\hat{y}, 1 - \hat{y}\}$ and we can get the difference between p and q by using the cross-entropy $H(p, q)$, which leads to

$$H(p, q) = -\sum_i p_i \log q_i = -ylog\hat{y} - (1 - y)log(1 - \hat{y}) \tag{4.4}$$

[1] *True positives* are all the datapoints that are correctly classified as positives, while *false positives* are the datapoints which are classified as positive when they actually are negative. The same labeling holds for the negative classifications (TN/FN).

While we are training, logistic regression optimizes the log loss J[2] which means that we optimize the cross-entropy in the sample. This is given by

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n log\hat{y}_n + (1-y_n)log(1-\hat{y}_n)\right], \tag{4.5}$$

where $\hat{y}_n$ is defined by equation 4.2.

Cross-entropy is the best loss function to use in our case and is the preferred function to use in similar cases. Ideally, the cross-entropy loss should be 0 during training - this corresponds to the minimum possible loss. In practice, this does not happen; in fact, a validation loss of 0 normally indicates a different problem - likely overfitting.

ROC-curve and AUC-score

To analyze our results from both ML algorithms, we are going to use the ROC-curve and the AUC-score, which is short for Receiver Operating Characteristic and Area Under Curve. AUC refers to the area under a ROC curve, and this curve is a plot of the True Positive Rate (TPR) against the False Positive Rate(FPR). TPR is the actual positives that are identified correctly, called *signal efficiency*, and FPR is the rate between the negative events categorized as positive and the negative events. In figure 4.3, we can see how this looks.

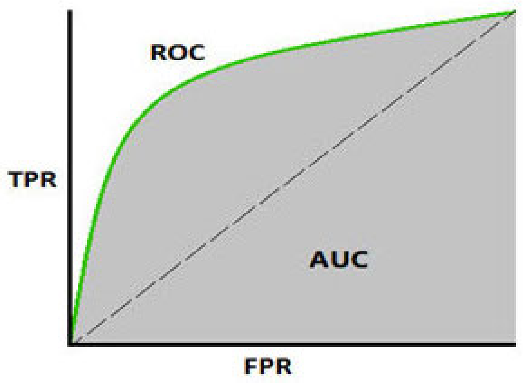


Figure 4.3: An example of a ROC curve [44].

[2]Here the loss function is called the cost function J. This might be a bit confusing because the cost function is defined as the loss function for all training sets, but in our case the loss and cost are the same.

If we look at the green line in figure 4.3, which is changing from model to model, we can see it has a smooth curve. If the green line creates a right angle in the top left corner, we will get an AUC score of 1, meaning the ML algorithm classifies correctly every signal and background event. In other words, the algorithm picks correctly every time. If the green line follows the diagonal dashed line, the algorithm is correct approximately 50 % of the time, which is statistically the same as having the model randomly classify objects. If the curve goes below the dashed line, the model classifies the background events as signal and the signal events as background, precisely the opposite of what we are trying to do.

4.2 Boosted Decision Trees

Boosted decision trees (BDT's) are one of the easiest ways to implement Machine Learning. It's typically not very sensitive to its hyperparameters, and is relatively easy to set up. In this thesis we have used XGBoost [45], which is one specific way to boost a decision tree. Before we explain boosted DTs however, we have to understand a regular DT.

4.2.1 Decision Tree

A decision tree is something we often do when thinking, even if we don't refer to the process as such. When we make a choice, we use a decision tree. Let us say that we have a picture of a person and we want to figure out if it is a boy or a girl. To make this conclusion we identify different features like whether the person has long hair, a beard, makeup, and so on. Then we set up a decision tree that will split our input data, which in this case is our picture, and make decisions based on the features you give it. Of course, we are not going to look on gender recognition in this thesis, but in principle decision trees can classify many things. This was just a small illustration of the simplicity and broad usage of BDT's.

Our goal is to use these principles to separate the signal from the background events. We give our decision tree some input data [46], which consists of Monte Carlo (MC) simulated background and different signal samples (which are also obtained by MC simulations). Our decision tree will then split our data based on features we provide, e.g. invariant mass and missing transverse energy.

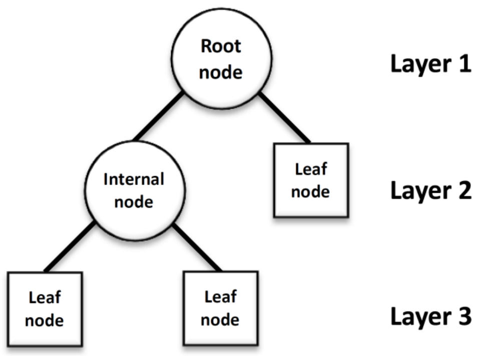


Figure 4.4: A sketch of how a decision tree is set up. [47]

If we look at figure 4.4, we can see that we have three layers and three different types of nodes. The *root node* is our input node, i.e. our input data, which for the first round is signal and background samples. Then the decision tree splits up our input data depending on the features we give it. This part will happen continuously through *internal nodes* until we hit the depth of the tree, i.e. the last layer, or only have *leaf nodes* left. As mentioned in section 4.1.1, we split our input data into train and test parts where we save some data to test how well our model is trained. If we are happy with our results, we can test our pretrained model on real data and potentially extract some signal.

4.2.2 Boosting

Decision trees are weak learners (meaning that they typically perform badly), but if we ensemble several weak learners, we can get a strong learner. This process gives what it is called a boosted decision trees. Boosting creates a collection of predictors, which means that we fit consecutive trees, and at every step, we try to solve for the net error from the prior tree. If a hypothesis misclassifies the input, its weight will increase, so it would more likely classify it correctly next time. In figure 4.5, we can see a sketched diagram of a BDT.

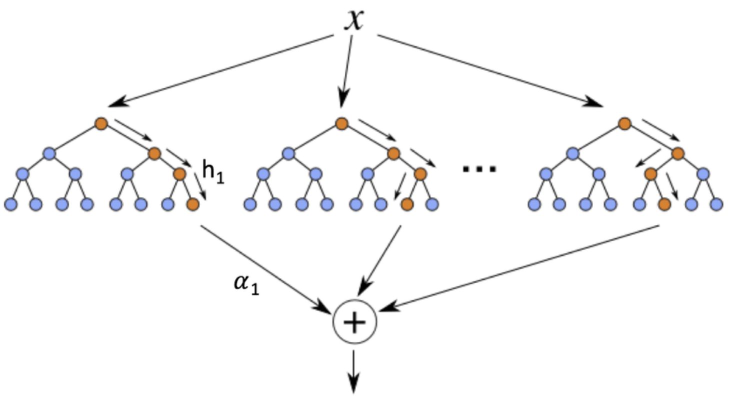


Figure 4.5: Illustration of a BDT [48].

There are many ways to boost a decision tree, but in this thesis we have used a Python software library called XGBoost [45], which is short for eXtreme Gradient Boosting. This library uses a gradient descent while boosting the decision trees. The gradient descent is an algorithm that can optimize the differential loss function, i.e., the next tree will try to recover the loss from the previous tree. Note that the loss is defined as the difference between actual and predicted values [49].

XGBoost is optimized to be highly efficient and flexible [45]. It is written in C++, but offers a user interface in python and other languages. In this thesis we are using a class in XGBoost called `XGBClassifier` which, as implied by the name, classifies the data. We give all our input data a label 0 or 1 for background and signal respectively and train our model, with the different features, on what should have label 0 or 1 in the end. This is what we call *supervised learning* because we help our model by providing a label in the input. The `XGBClassifier` takes a lot of arguments, which can be found in the Scikit-Learn API reference guide [50], but in this thesis we use just the ones that are necessary for our purpose. The arguments that we have used are mentioned in the analysis chapter.

4.2.3 Feature importance

We are also going to look at the feature importance in the analysis. This is a way to see which features were important while constructing our BDT. This is an interesting aspect of the analysis because this will change from process to process and we will also be able to see if it matters if we train on different signal samples or not.

4.3 Neural Networks

Neural Networks (NN) are probably the most popular machine learning algorithms that we use today. This is also what is often used for face and speech recognition by companies like Google and Apple [51]. NN does basically the same as a BDT, but instead of splitting the input data into background and signal at each node, each node is trained on different features.

NN are a collection of nodes, like the ones we have in BDT. 4.2. Instead of many trees with nodes, we create a network with these nodes as we can see in figure 4.6.

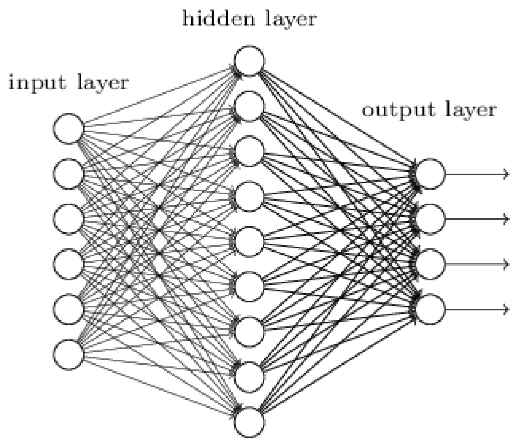


Figure 4.6: Illustration of a neural network [52].

The network consists of one input layer, one hidden layer and one output layer. This is a very simple NN and is often called a *shallow neural network* (SNN) since it has only one hidden layer. If we increase the number of hidden layers to two or more we have a deep neural network (DNN) as we can see in figure 4.7

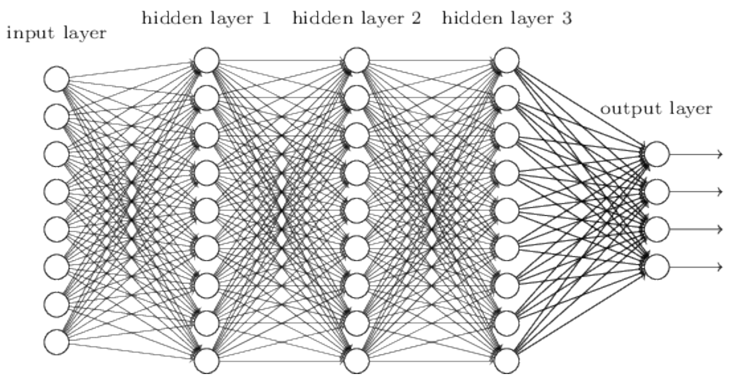


Figure 4.7: Illustration of a deep neural network [52].

In this thesis we have built a NN with a tool called `KerasClassifier`[3] which, as for the BDT, classifies the data. For our network to be able classify, we have to send in some pre-labeled data to train it. This is done by simply giving the background label 0 and signal label 1. Additionally, we have to understand what happens in the NN to know what parameters we should adjust in order to train and analyse the data.

The NN has many more dependencies than the BDT. It is therefore unfortunately more sensitive to possible under- and overtraining, but for correct composition of parameters, it will typically be better trained than a BDT. To achieve the optimal results, we need a lot of computing power and time to optimize.

There are many advantages and disadvantages with both SNN and DNN but it is in principle possible to achieve the same results with both. In a SNN you can increase the number of nodes and get the same results as for a DNN with several layers and fewer nodes in each layer. The problem with doing it that way is that we get more free parameters by increasing the number of nodes, which again increases the possibility that our model gets biased and hence over or underfits.

This is unfortunately not as trivial as for a BDT, but maybe easier to adapt to different purposes. In our case we are going to use it to distinguish signal from background.

With the basics in place, we can now describe NNs in more detail. Next we provide an overview of the activation function, how the network evolves, and how it gets better with

[3]Keras [53] is a NN library for Python.

training.

4.3.1 Activation functions

To get the output for the different nodes in a layer, we need an activation function. This output is again used as input for the next layer until you reach the last layer, which then will give us our final output/results.

Since Machine Learning doesn't come from a theoretical prediction which is implemented numerically, the choice of activation function for different problems is found by trial and error. The activation function will provide a number between 0 and 1 [54], which is sent into the next nodes in the next layer. Some activation functions provide a number between -1 and 1 instead, but this depends on how the activation function we choose behaves, as we can see in figure 4.8. In the next sections we are going to look more closely at the different activation functions we have used in this thesis. We are also going to consider some other activation functions that are not used in this thesis but still of very common use in ML.

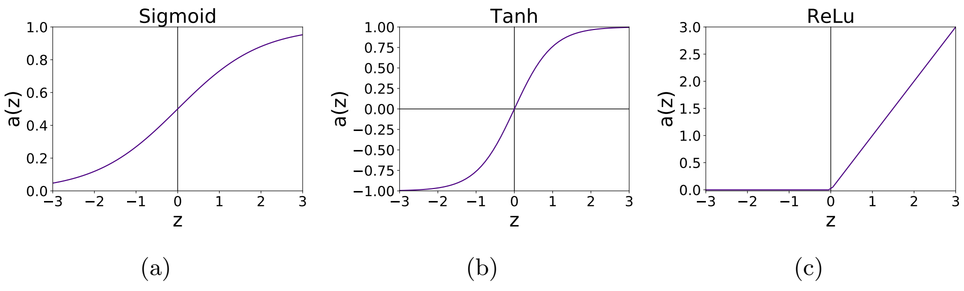


Figure 4.8: This is an illustration of the behaviour of three different activation functions. In (a) we can see the Sigmoid function, in (b) the Tanh function, and in (c) the ReLU function.

Sigmoid

The Sigmoid function, figure 4.8a, is a logistic activation function and is given by equation 4.6 below.

$$a_i = \frac{1}{1 + e^{(-z_i)}}, \tag{4.6}$$

where z_i is the output value from the i-th node in the output layer.

As we can see in figure 4.8a, the Sigmoid function gives us a value between 0 and 1. This makes this activation function a good choice if we have to predict a probability for the output. A drawback, and the reason we have to use other activation functions sometimes, is that the Sigmoid can get the NN stuck on its training time because of the calculation time of the exponential. This is not an optimal activation function to use inside the network, but in the output layer it is the preferred one. We want our network to classify our output as signal or background which we have labeled as 1 and 0.

Tanh (Hyperbolic tangent)

In addition to the Sigmoid function we have the tanh function, figure 4.8b, which is shaped the same way as the Sigmoid function, but it gives values between -1 and 1 instead of between 0 and 1. This we can see in figure 4.8b and it is given by equation 4.7 below.

$$a_i = \tanh(z_i) \tag{4.7}$$

Since we also can get negative numbers from this activation function, it is preferred instead of the Sigmoid if we have negative numbers in the input. This function returns negative values as negative unlike e.g. Sigmoid which only returns 0 or 1. Since we don't care for negative values in our case, this activation function is not used in this thesis.

ReLU (Rectified Linear Unit)

One of the most popular activation functions is the ReLU function. This function outputs 0 for any negative value, and if the value is positive, the function returns the value as it is. This is shown in figure 4.8c. Since all data that is below zero becomes zero, it can be hard do get a good read out of the data because the data may sometimes assume negative values. The ReLU function is given by equation 4.8.

$$a_i = max(0, z_i). \tag{4.8}$$

Since we're not interested in negative values we are using this activation function in our hidden layers. The Sigmoid function would also be a good choice when we are thinking about our wanted form of the output. But, as mentioned earlier, it can get stuck in its training time.

Other activation functions

The three functions mentioned above are not the only activation functions one can use. Since every problem has different approaches, we have to adjust these functions for our needs. There are, for example, many different adjustments done with the ReLU function to get a better activation output in problems that need adjustment to perform optimally. One of the newest activation functions is called Swish [55], which is simply a modification of the Sigmoid function given by equation 4.9.

$$a(x) = x \cdot Sigmoid(x) \tag{4.9}$$

The developers of the Swish function state that it does overall a better job than the widely used ReLU function, but until this time there is no evidence in literature supporting this claim. As such, we will stick with ReLU in this thesis.

4.3.2 Feed forward and back propagation

Every node in the neural network has an activation function connected to itself and if the node is connected to another node it also has a weight for each node it is connected to. To create the new node we simply multiply the nodes activation with the weight and sum them together like so

$$a_1 \cdot w_1 + .. + a_n \cdot w_n = \text{new node} \tag{4.10}$$

This operation is done for every node in every layer, and fed to the next layer until we hit the the output layer. This is what we call *feed forward* because we always send the obtained information forward from input to output layer.

We also have a method to go back in the network to optimize it, namely *back propagation*. What we do is go back to the previous layers to optimize the weights. This is not what we call an optimizer and should not be confused with that; it's merely the process of going backwards. Back propagation is probably the most important step in a NN, because this optimizes our model and makes sure that our training goes well.

4.3.3 Optimizers

When training a NN, we need to know how well it performs during and after the training. For this we use a loss function L. There are many different types of functions that are used, depending on what we want to do with the network.

We want to minimize the loss function to make the prediction error as small as possible, i.e. optimize the network. And to do this, we use an optimizer.

Stochastic Gradient Descent [56]

The gradient descent method uses the fact that a function $F(x)$, where $x = (x_1, x_2, ..., x_n)$, will increase fastest in the direction of the gradient of the function, ∇F. We want to move in the opposite direction of the gradient to make sure that we decrease the loss function.

We then multiply this gradient with a number η called "the learning rate", and subtract this from the current weights w_j. In the case of neural networks, $F(x)$ is the loss function L. This leads to equation 4.11

$$w_j = w_j - \eta \frac{\partial L}{\partial w_j} \tag{4.11}$$

To speed up the training we can also add a stochastic part to the method, which means that we randomly choose a batch size of the data that we use to approximate the derivative, and use that to update the weights. This is also called Mini Batch Gradient Descent.

The data that we want to use might be very noisy, so we can use a technique to de-noise the data [57]. This is done by adding a momentum, which is a moving average of our gradients.

We now subtract this from the weights. The equations now become

$$V_i = \beta V_i + (1 - \beta)\nabla_w L w_i = w_i - \eta V_i \tag{4.12}$$

One way to look at this last addition is viewing the movement in parameter space as an object that moves one step at a time downhill. When we add momentum, it is like giving this object some physical momentum. Now when the object finds the minimum, it will continue a little bit back and forth to make sure it has found the minimum. A good reason to add this is if there are many saddle-points where the optimizer might stop. With momentum, it will move past, and then continue the descent on the other side without stopping.

Adam [58]

The learning rate is one of the most difficult parameters to optimize, because any change to its value may have a huge impact on a method's performance. To circumvent this, we can use a method that has an adaptive learning rate.

Adam is currently the most popular optimization algorithm today, and the name is derived from Adaptive Moments. It is an update to another optimization algorithm called RMSProp, which stands for Root Mean Square Propagation. RMSProp uses the running average of recent gradient magnitudes, and divides the learning rate by this average. This is done for every weight in the NN.

Adam takes this a step further. It uses an estimation of the first-order moments of the gradient as momentum, and also adds a bias correction to both the momentum and the second-order moments.

Chapter 5

Machine learning analysis implementation

In this chapter we are going to look at how the ML algorithms are implemented, trained and how they perform before we in the end are going to test it on real data.

5.1 Preparations and expectations

For both machine learning algorithms, BDT and NN, we have done a few preselection cuts on the data before we feed it into our ML classifiers. It is necessary to make the datasets smaller in order to get a more reasonable training time and to avoid consuming too much memory when converting the input to dataframes[1]. The cuts we have applied on the data are given in table 5.1 below.

[1]Pandas dataframes [59] is a two-dimensional frame of your data which also includes the corresponding labels.

Precuts
Number of leptons = 2
$E_T^{miss} > 40$ GeV

Table 5.1: A table of the precuts we have done before sending the data into the BDT and NN.

The features included, that is the kinematical variables, in the training of the ML algorithms have been selected using the different variables available in the MC nTuples[2] and are listed in table 5.2.

Low level features	**High level features**
lep p_{T_1}	mll
lep p_{T_2}	mt2
lep η_1	H_T
lep η_2	E_T^{miss}/H_T
lep ϕ_1	$\Delta\phi(\vec{p}_T^{ll}, E_T^{\text{miss}})$
lep ϕ_2	ΔR_{ll}
nJet20	Fractional p_T difference
nJet30	
$\text{n}_{\text{b-tagged jets}}$	
E_T^{miss}	
E_T^{miss} significance	
Electric charge	
Flavor	

Table 5.2: List of features chosen for the training of the ML model, where high level features are more complicated kinematical variables than low.

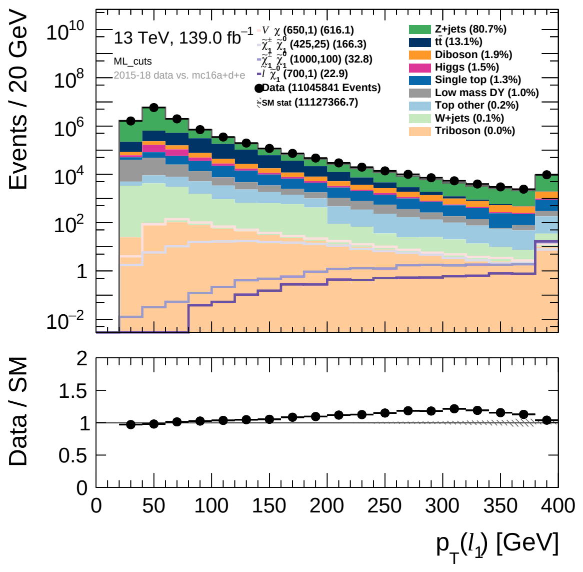
We have also included different variables to weight each event according to its cross-section, average number of collision per bunch crossing (pile–up) and other characteristic of the event. The weights are listed in table 5.3.

[2]nTuple is the data format used in the high energy physics community to store large amounts of data for analysis within the framework known as ROOT [60].

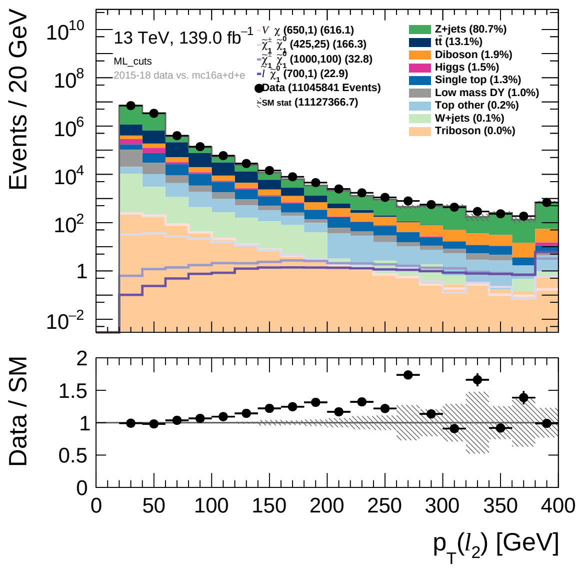
Weights
event weight
pileup weight
b-tag weight
generator weight
jvt weights (jet vertex tagger)
global dilepton trigger SF (same flavor)
Luminosity for MC 2015-2016 = 36.2 fb^{-1}
Luminosity for MC 2017 = 44.3 fb^{-1}
Luminosity for MC 2018 = 58.5 fb^{-1}

Table 5.3: List of weights to weight each event in the dataframes and the luminosity for the data taken in 2015-2018.

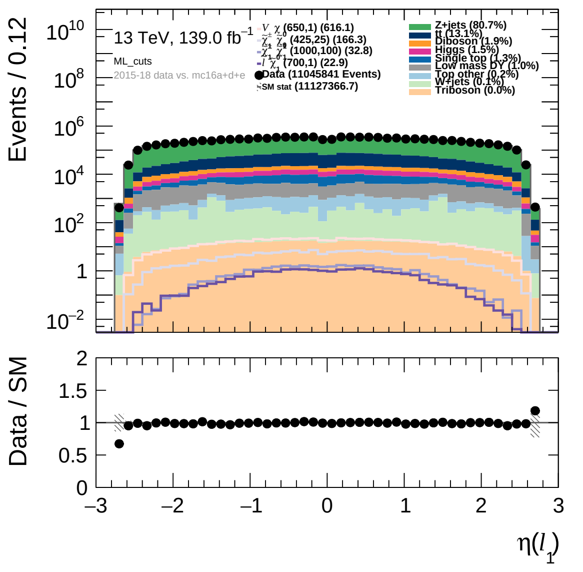
Figure 5.1 shows the distribution of the variables/features for data, the simulated SM background and some selected new physics signal models. Overall, it seems like there is good agreement between data and MC, except for some missing MC in the high MET tail (figure 5.1o and 5.1p) and high b-jet multiplicity (figure 5.1l). Since it is overall good agreement we can trust the data we are putting into the ML.



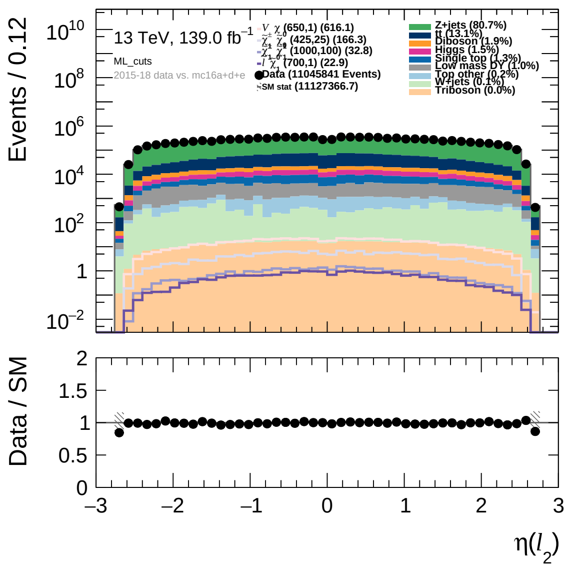
(a) The transverse momentum for lepton 1.



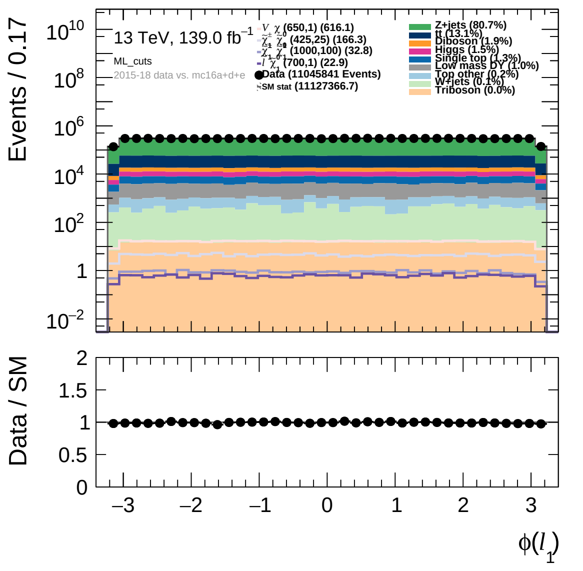
(b) Missing transverse energy.



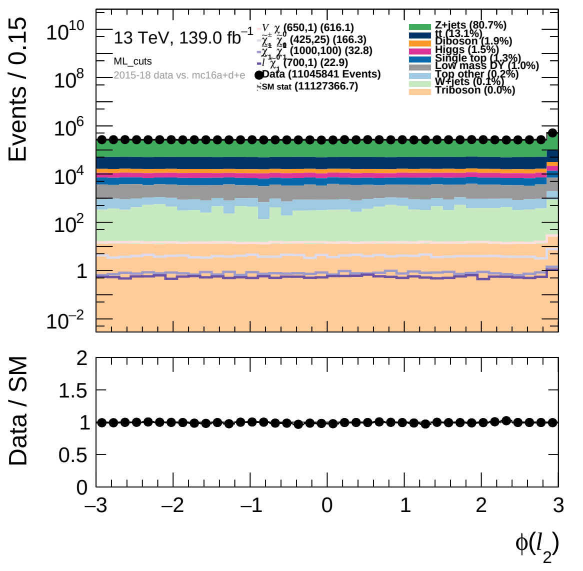
(c) The pseudorapidity for lepton 1.



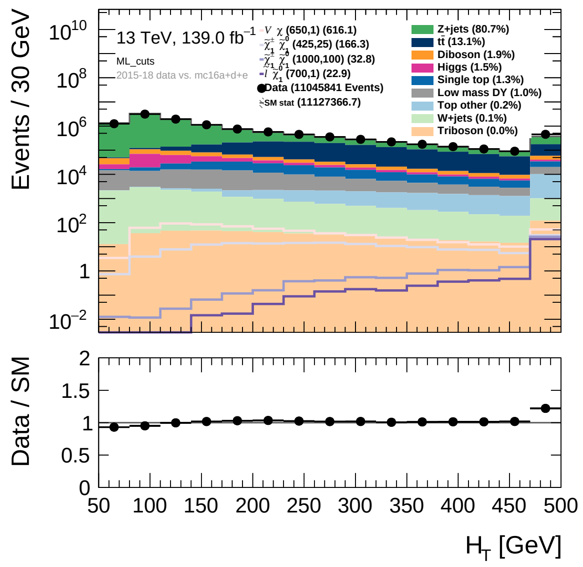
(d) The pseudorapidity for lepton 2.

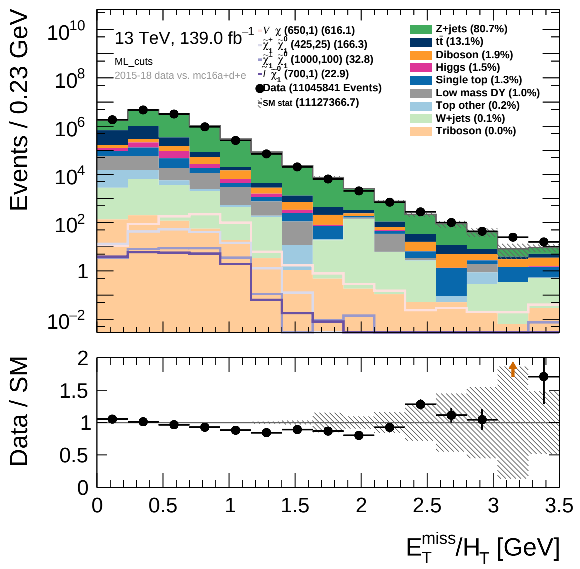


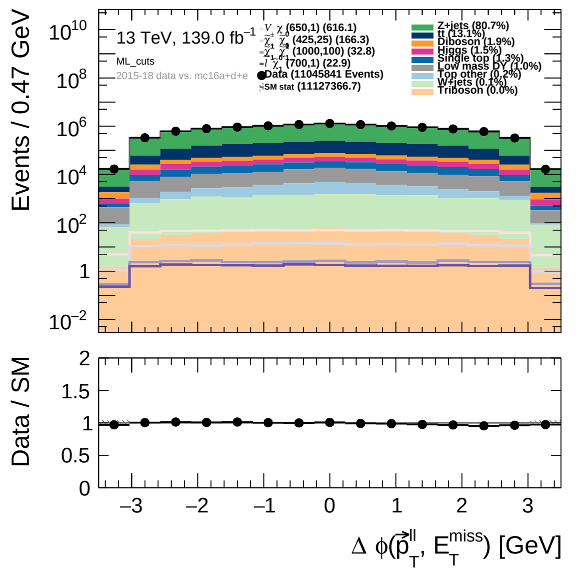
(e) The azimuthal angle for lepton 1.

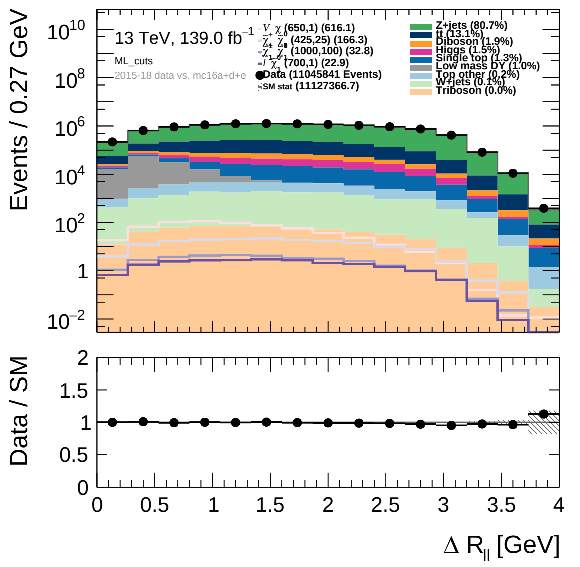


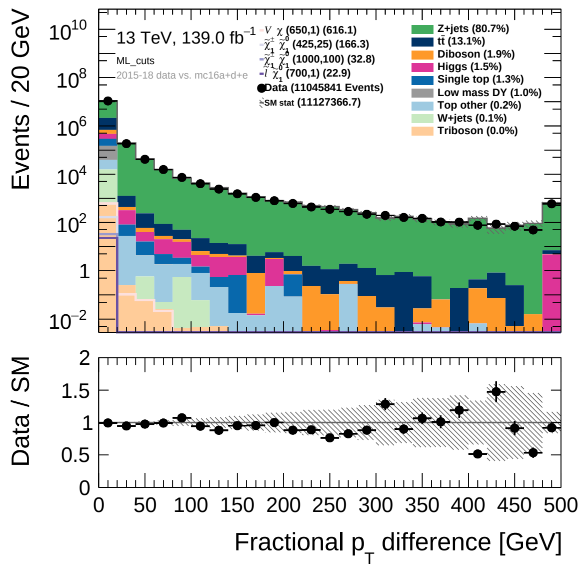
(f) The azimuthal angle for lepton 2.

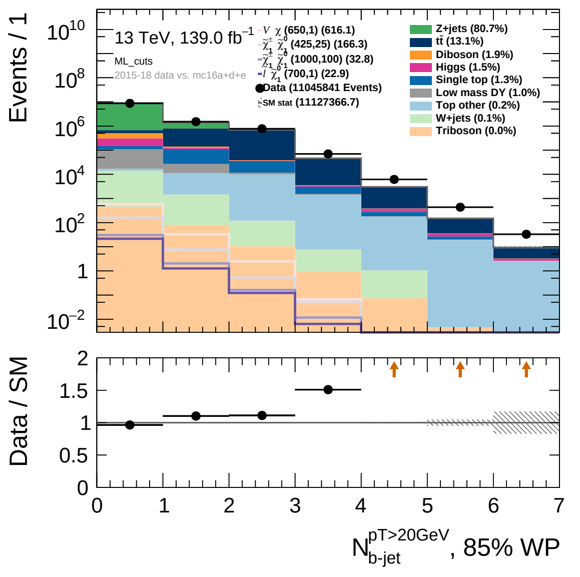

(g) The scalar sum of the p_T of the selected jets and leptons.

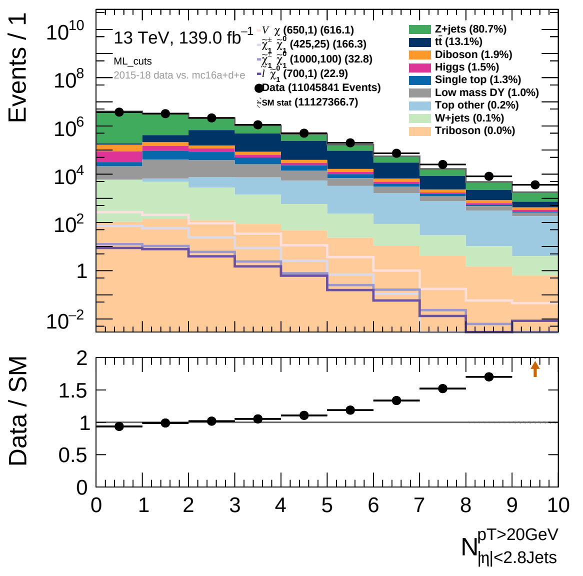

(h) The ratio between E_T^{miss} and H_T.


(i) The azimuthal angle difference between the dilepton system and E_T^{miss}.

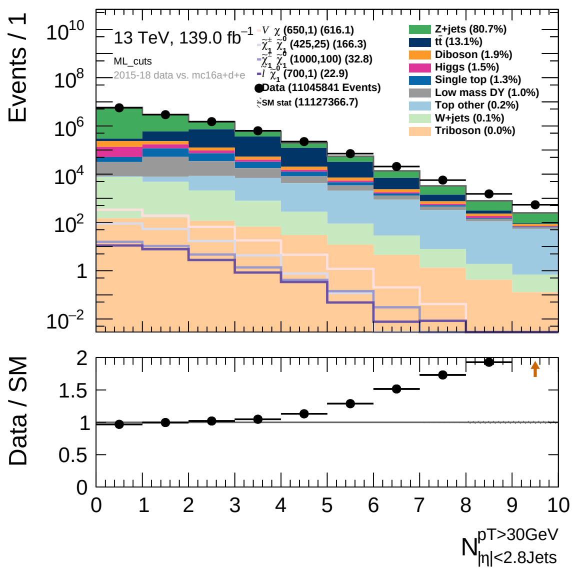

(j) Distance between the two leptons.


(k) The difference between the dilepton p_T, the selected jets p_T and the vector sum of $\vec{E}_T^{miss}$.

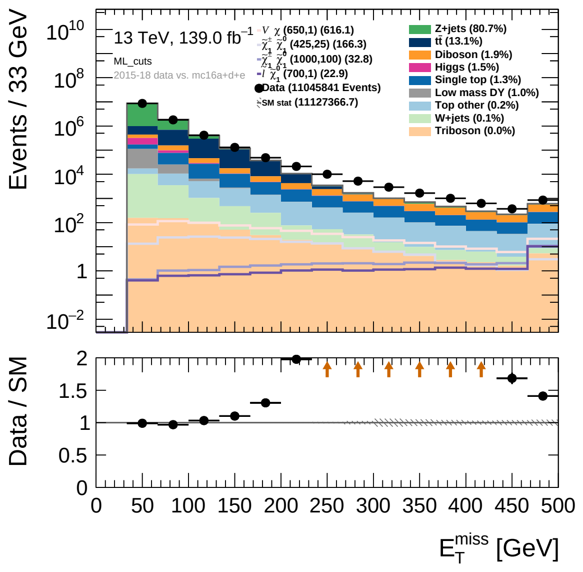

(l) number of b-tagged jets.



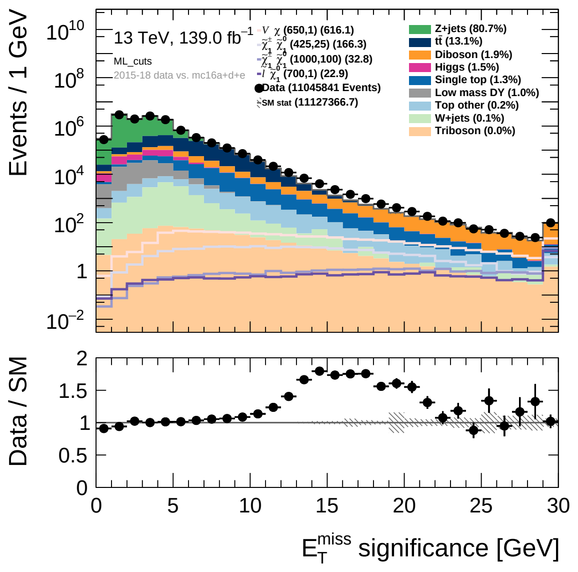
(m) Jets with $p_T > 20$ GeV.



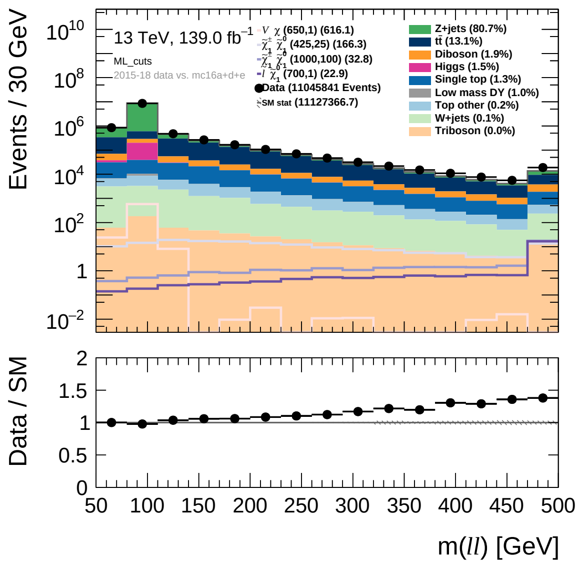
(n) Jets with $p_T > 30$ GeV.



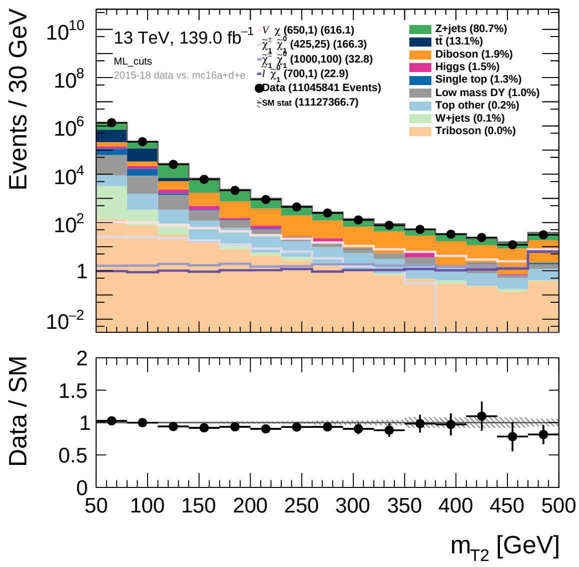
(o) Missing transverse energy.



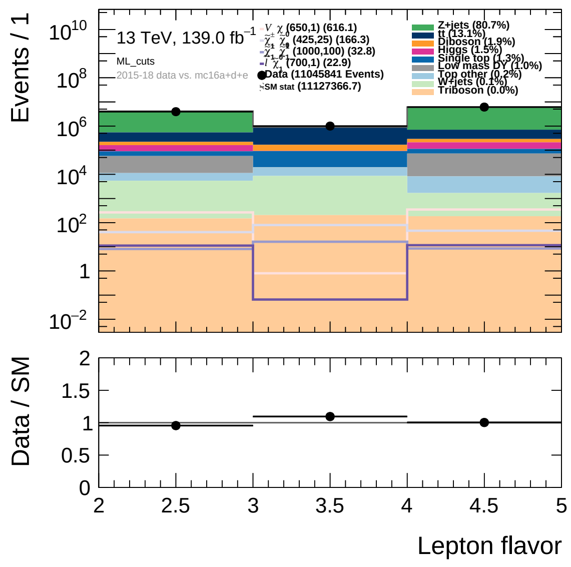
(p) Missing transverse energy significance.



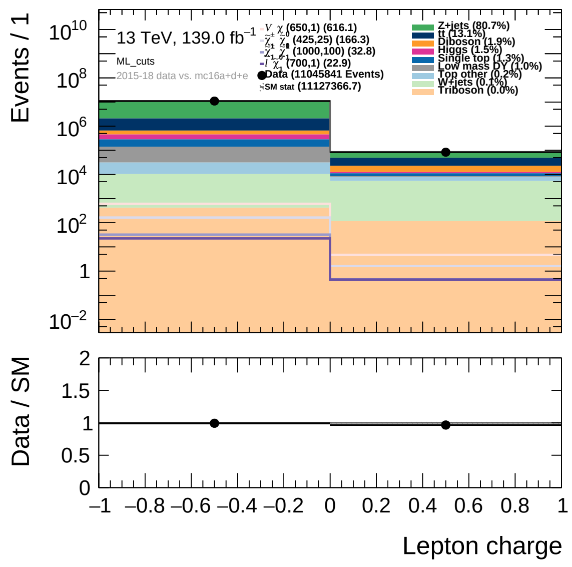
(q) Invariant mass.



(r) Stransverse mass.



(s) Flavor combinations of the two leptons, with the first, second and third bin corresponding to ee, $e\mu$ and $\mu\mu$ events, respectively.



(t) Events with two opposite sign (OS, first bin) and same sign (SS, second bin) leptons.

Figure 5.1: Plots of the different variables used as features in the ML algorithms after the preselection cuts in Table 5.1 have been applied.

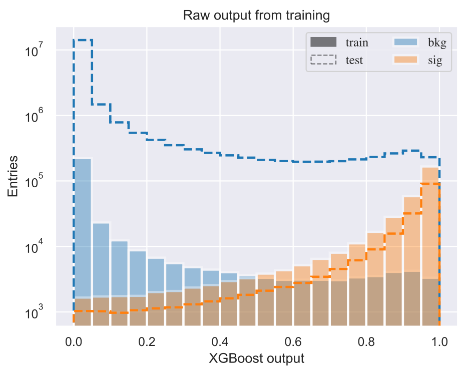
To make our ML analysis code more user friendly and unbiased (give the ML the opportunity to figure it out), we have used the same precuts for all four processes.

The goal with the ML analysis is to teach the computer to separate the signal from the background and then calculate the expected significance and compare it with the cut and count results. Since we are handling a huge amount of data and each step takes some time, we have separated some of the most time consuming steps to save some time to optimize the ML part. The first thing we did was to import all the nTuples and put them into pandas dataframes. Then we did the precuts in table 5.1, giving the dataframes the features in table 5.2, weighting and scaling up each event to correct luminosity from table 5.3 before we in the end stored all of the processed dataframes temporarily into HDF5 files [3]. To avoid huge memory issues in the computer, this is done chunk by chunk since importing the whole dataset in one go is too much for our computers to handle.

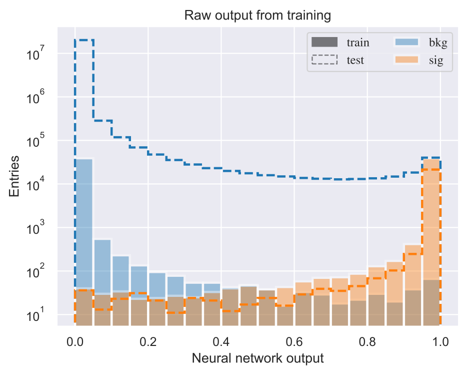
Importing the data is the most time consuming step, but luckily we don't have to do this every time we do some adjustments which is mainly done in the next step, the ML part. In the ML part we are training two different algorithms, namely a BDT and a NN.

As mentioned in chapter 4 we have to divide our input data into train and test sets, which are done by simply taking 1/3 of the input (randomly drawn) and call it the test set. This data is not going to be touched until we are finished with the training of our model except for 1/10 of the test set which is used as a validation set. To both save some time in the training and to give the BDT and NN the opportunity to train more on actual signal samples, we have chosen to train on the same amount of background events as we have for signal. This gives us the results shown in figure 5.2, which is the raw output from the trained model. The data sent into the ML are scaled between 0 and 1, where the MC background have gotten a label 0 and the signal a label 1. This gives us also an output between 0 and 1 which is the x-axis in 5.2. Looking at the raw output gives us a good indication on how little MC background we need during training to obtain the results we have in this thesis and have made the training a lot less time consuming. The reduction of the background does not have much impact on the results for the BDT except for some reduced training time, but for the NN the results improves a lot by doing this.

[3]HDF5 [61] is a file format to store large, complex and heterogeneous data



(a) A plot of the raw output from the BDT.



(b) A plot of the raw output from the NN.

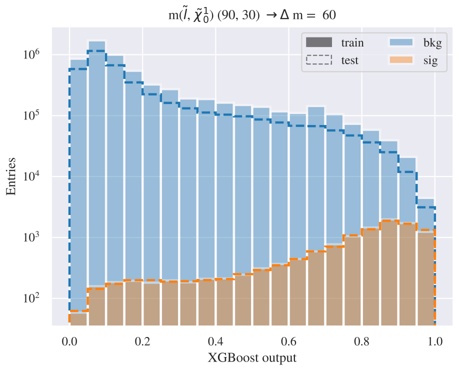
Figure 5.2: Examples on how the raw output from a BDT (left) and NN (right) look after training and testing. This is obtained by using the signal samples from direct slepton production with low mass splitting and all level variables. These plots look very similar for all of the different signal processes and trained model and are therefore only these are shown here.

Since one signal sample alone is pretty small, we have merged the signal samples into different mass splittings for each process. The reason for this is that with several samples together we get a lot more data to train our BDT and NN on and in this way our model will also handle other datasets much better. The mass splittings are defined by the lines drawn in figure 3.13 in chapter 3. This gives us the mass splittings as shown in table 5.4 for each process.

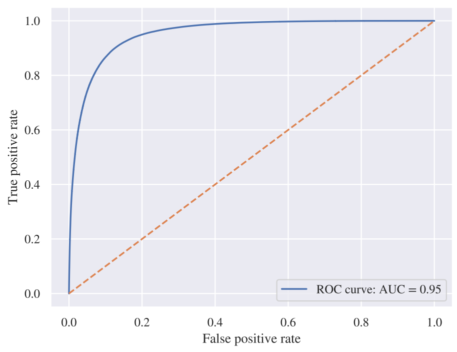
Process	**Low**	**Intermediate**	**High**
Direct slepton production	Δm $\leq$ 100 GeV	100 $<$ Δm $<$450 GeV	Δm $\geq$ 450 GeV
Chargino production via $\tilde{l}/\tilde{\nu}$	Δm $<$ 200 GeV	200 $\leq$ Δm $<$600 GeV	Δm $\geq$ 600 GeV
Chargino production via $W^{\pm}$	Δm $<$ 150 GeV	300 $\leq$ Δm $<$300 GeV	Δm $\geq$ 300 GeV
Mono-Z	Δm $<$ 100 GeV	100 $\leq$ Δm $<$350 GeV	Δm $\geq$ 350 GeV

Table 5.4: Table defines the different mass splittings based on figure 3.13 for the different processes.

When training the ML models we have different means for evaluating how good or bad the models are doing. We can for example scale the test set up to be as big as the training set to see how good the fit between the up-scaled training and testing sample is, shown in figure 5.3a. We can also look at the ROC curve and AUC score, figure 5.3b, to see how well the model can classify background as background and signal as signal. The goal in figure 5.3a is that the test set (dotted line) is matching the training set (filled bins) and in figure 5.3b we want an AUC-score close to 1.



(a) A plot of the training and test set, where the test set is scaled up to match the training set.

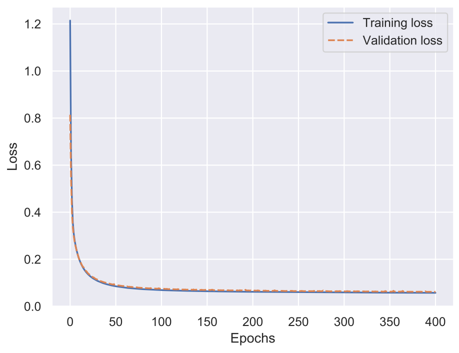


(b) A plot of the ROC curve together with the AUC score.

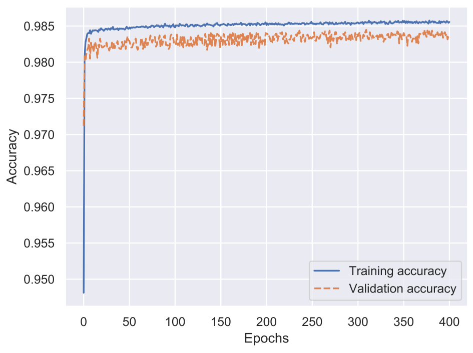
Figure 5.3: In this figure you can see results from a BDT trained on low mass splittings and with low level features for the direct slepton production.

Both of the plots in figure 5.3 are merely shown to illustrate what we are judging the training on. The ROC-curve will not be shown for every trained model, but the AUC-score will be presented in a table for every trained model. For reference later in the results, we say that an AUC-score $\gtrsim 0.85$ is a good score.

In addition to the ROC-curve and AUC-score, we have plotted the training loss vs the validation loss and the training accuracy vs the validation accuracy for the NN as shown in figure 5.4. The loss should go towards zero and the accuracy towards one.



(a) A plot of the training loss vs the validation loss.



(b) A plot of the training accuracy vs validation accuracy.

Figure 5.4: In this figure we can see results from a NN trained on high mass splittings and with high level variables for the direct slepton production.

As we can see in figure 5.4a, both the training and validation losses goes quickly towards zero after the NN starts training and does not improve much after $\sim$ 100 epochs. This means that we probably could have stopped the training after only 100 epochs and would still have gotten a fairly good result, but we want the best result as possible and therefore we let it minimize the loss further. We can also see that the validation loss is completely flat in the end and would not improve further (continue towards zero), while the training loss continues down towards zero (this is not that easy to see because of the scale on the y-axis, but nice to follow during training where you can see the loss for each epoch). Let us say we have given the NN a patience on 50 epochs instead of 10, we would most likely see that the validation loss would tend to move away from zero. This is not that easy to see in this plot, but same point is in the accuracy plot in figure 5.4b, where we can see this more clearly. It is a bit hard to determine if the validation accuracy tends to go down again because of the fluctuations[4], but it is fairly easy to see that the training accuracy continue towards one. That the training loss and accuracy continues to improve is often due to overtraining/overfitting which we want to avoid.

The plots in figure 5.4 for every process will not be presented in this thesis since they are going to look very similar for every process and ensemble of training compositions. All of the results that are not included in this thesis or the appendix can be found in the GitHub

[4]The fluctuations can be avoided by having a bigger validation set, but then you will also have a smaller test set to test our ML model on.

repository `https://github.com/monaanderssen/MasterML` together with the ML code.

5.2 Building, training and testing the BDT

Now we know how a BDT and NN works and we are aware of the results to expect. In this section we will get an understanding on how the BDT is built up, how we train it and, in the end, testing how well the training went.

5.2.1 Building

The BDT, as mentioned in chapter 4.2, is built up using an existing python library called XGBoost. The hyperparameters of the method are listed in table 5.5 below.

Parameter	**Value**
Maximum depth of tree	3
Learning rate	0.1
Number of estimators	10000000
Verbosity	3
Objective	binary:logistic
Scale position weight	1

Table 5.5: An overview of the different parameters used during training to obtain the results for the BDT.

Most of these parameters are self explanatory, but not all of them so we will take a brief explanation before we move on. The first parameter that is not so intuitive is the verbosity level, which we set to let the BDT know how much information we want from the tree during training. This can be set to 0, 1, 2 or 3, where 0 is silent, 3 is debug and 1 and 2 will give us parts of the information. It is nice to see what is going on during the training, so we have chosen to set it to 3. The next parameter is the objective, which specifies the learning task we are doing and the learning objective. As we can see in table 5.5, we have chosen to use something called `binary:logistic` which simply is a logistic regression for a binary classification. And last, we have the scale position weight which controls the balance between negative and positive weights in the tree. These parameters are chosen by simply trial and error (especially depth of tree and learning rate), and they are optimized for the direct slepton production process.

In addition to the parameters that we have to give to the BDT, we have different compositions of models we are training. As mentioned earlier in this chapter we are training on three different mass splittings (low, intermediate and high), but we are also training on high level, low level or all the features we listed up in table 5.2. This is done to see how important the different features are during the next step, namely the training.

5.2.2 Training

While we train our BDT, we have to be aware of overtraining/overfitting. This can easily be done by putting in a demand on the loss and stop the training early. This is simply done by calculating the loss from the validation set and give the BDT a patience parameter of 20 steps which means that it will stop if the loss has not improved in 20 steps, and save the model made at the best iteration. This is also the reason for the high number of estimators in table 5.5.

It is not much more we can do while the BDT is training, other than wait to see how the testing looks. If we are not happy with how the model is trained, we have to go back to the building of the tree and adjust the parameters in table 5.5 and do the training again until we are happy with our results.

5.2.3 Testing

The last step of the BDT is to test it on the test set. We have a lot of results from the BDT to look at since we have trained the model with low, intermediate and high mass splittings and with high level, low level and all features for four different processes. We are going to look at all four processes at the same time and see how the results evolves when changing the features and mass splittings. All the results which are not shown in this chapter can be found in the appendix B.

AUC score

The main selection of results is done based on the AUC-score. If the AUC-score is the same for high level, low level and all features, we have made the selection on looking at which features that have been important for the trained model. The AUC-scores are listed in table 5.6.

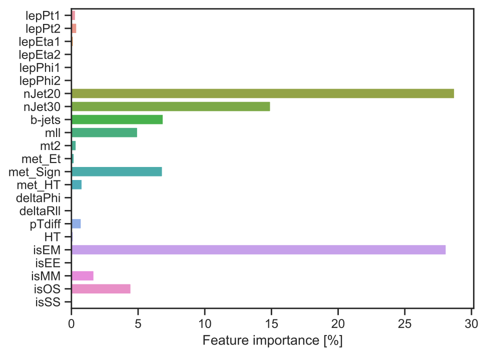
Level	$\mathbf{\Delta m}$	$\tilde{\mathbf{l}}\tilde{\mathbf{l}}$	$\tilde{\chi}_1^\pm \to \tilde{\mathbf{l}}/\tilde{\nu}$	$\tilde{\chi}_1^\pm \to \mathbf{W}^\pm$	**Mono-Z**
	Low	0.91	0.91	0.91	0.95
High	Intermediate	0.99	0.98	0.94	0.96
	High	1.00	1.00	0.96	0.97
	Low	0.95	0.93	0.93	0.95
Low	Intermediate	0.99	0.98	0.95	0.97
	High	1.00	1.00	0.97	0.97
	Low	0.97	0.95	0.94	0.96
All	Intermediate	1.00	0.99	0.96	0.97
	High	1.00	1.00	0.98	0.98

Table 5.6: The AUC score for all four processes with different compositions of features and mass splittings for the BDT.

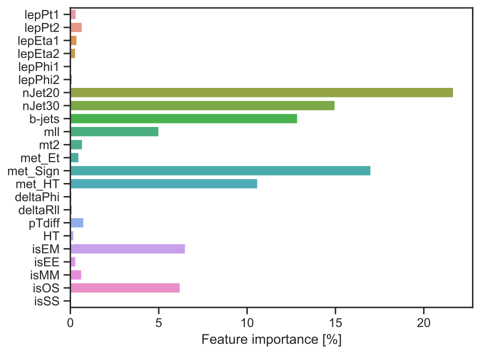
As we can see in table 5.6, the preferred composition of features is in most cases to use all the features, except for some few exceptions where all three options have the same AUC-score. As mentioned above, this is then determined by looking at the feature importance plot that also are presented in the following sections. The results are presented for each mass splitting with all the processes together for easier comparison. The signal samples shown here are the same as we looked at in the cut and count analysis presented in chapter 3.

Low mass splittings

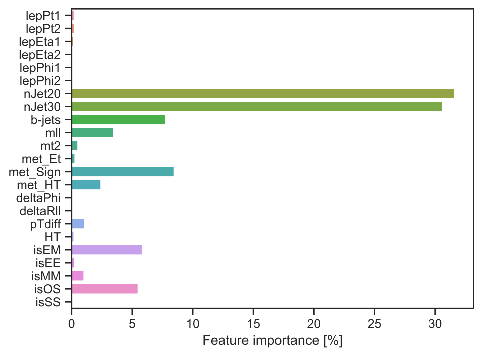
The first results presented is for the BDTs trained on low mass splittings. The preferred composition of features is all features and in figure 5.5 we can see how these different features contributes during training.



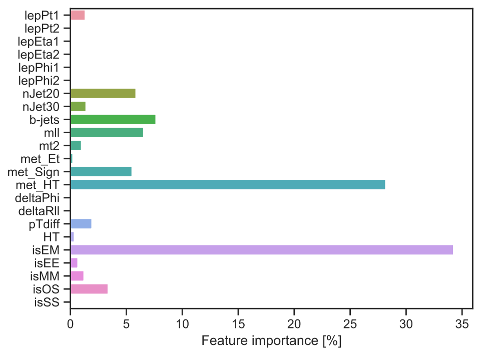
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.

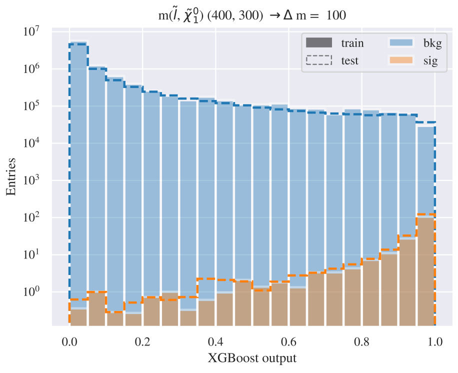


(d) Mono-Z.

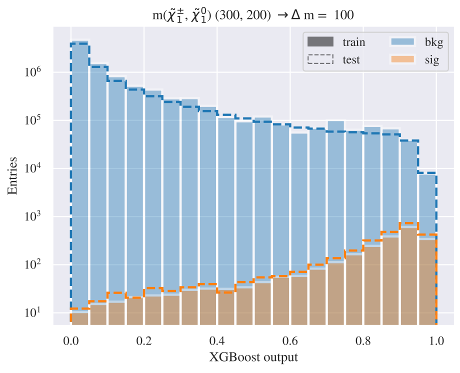
Figure 5.5: Feature importance for low mass splittings for all four processes using all features during training.

If we now look at the different feature contributions in figure 5.5, we can see that jets with a $p_T > 20$ GeV is an important feature when the BDT is trained on SUSY processes. This means that the BDT evaluate this as a powerful variable to distinguish signal from background. This is in accordance with the cut and count analysis where a cut on jets is applied. In addition we can see that the jets with a $p_T > 30$ GeV, b-jets, m_{ll} and MET significance have been used a lot. These are the variables we used in the cut and count analysis, but they do not contribute equally much for every SUSY process. For the mono-Z process the same features as for the SUSY processes are used, however, MET$/H_T$ is regarded as the second most important one. This again is reflected in the cut and count analysis where a cut on this variable is applied. For all of the four processes the tree finds the different flavor and opposite sign variable powerful when trying to separate the signal from background. To conclude the feature importance for low mass splittings we can say that there are no big surprises when comparing with the cuts applied in cut and count analysis.

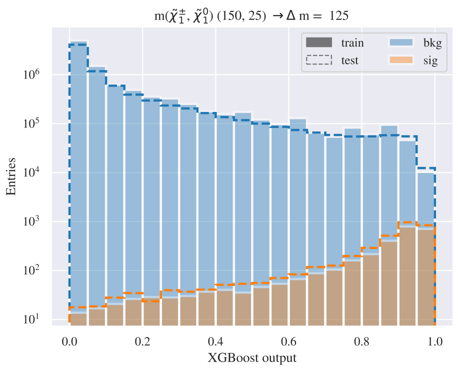
The next step is to see how well the testing of the model went. This can be seen in figure 5.6 below.



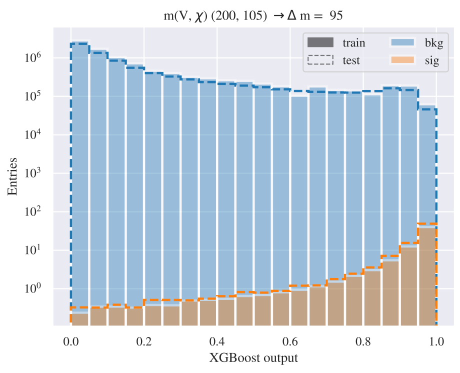
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



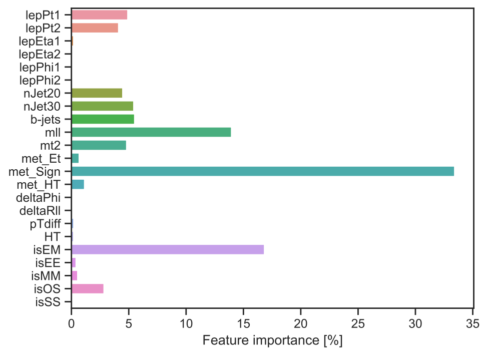
(d) Mono-Z.

Figure 5.6: Test vs train for low mass splittings with all features done with the BDT. Here the test set is scaled up to match the number of training events.

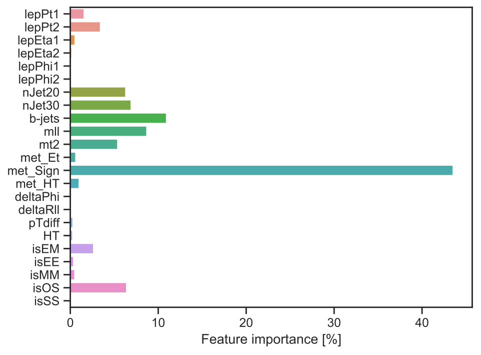
As we can see in figure 5.6, the test set (dashed line) match the training set pretty well for all processes. It differs a bit in some places, but overall it is a good fit. We can also see that the distinction between signal and background is not that good, but it is at least most signal close to one, which is the value we have labeled the signal with. The relatively poor separation of signal and background is due to the closer of the masses of the final states particles from our signal to the masses of the particles in the SM, which makes them harder to separate. This is a major challenge in searches for new physics in high energy particle physics and is something we hope to improve in the future exploiting more advanced techniques.

Intermediate mass splittings

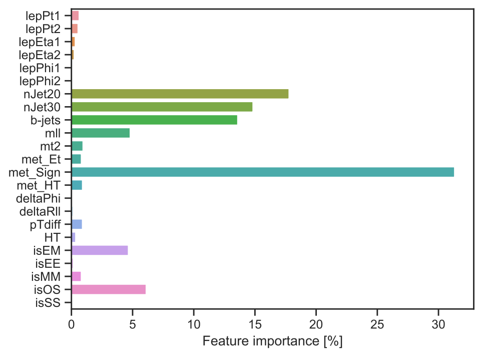
The next mass splitting we are looking at is the intermediate mass splittings. In figure 5.7 we can see the importance of the different features where all features are used for training.



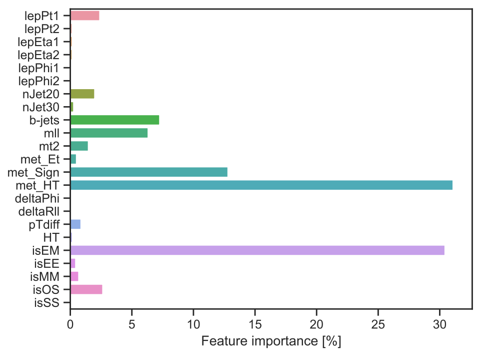
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



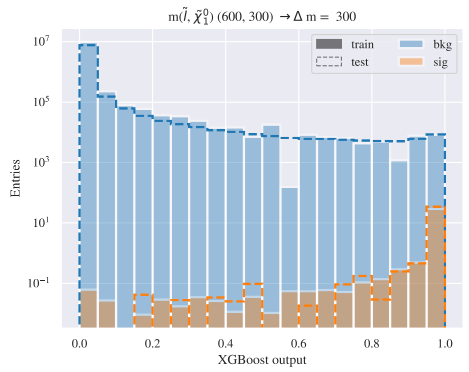
(c) Chargino production via $W^{\pm}$.



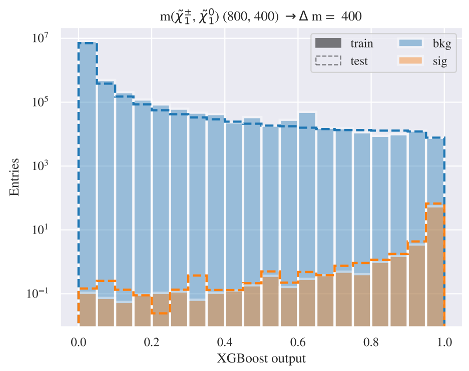
(d) Mono-Z.

Figure 5.7: Feature importance for intermediate mass splittings for all four processes using all features during training.

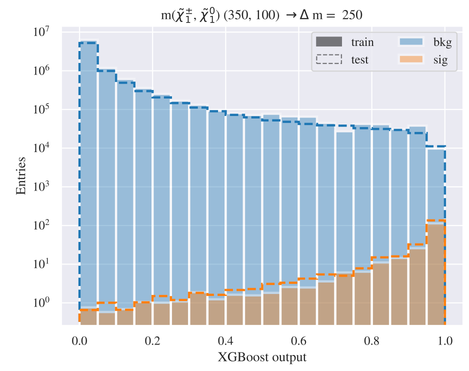
In figure 5.7 we can see that MET significance is the most important feature for the SUSY processes and the third most important feature for mono-Z. This might be a bit more interesting result than for the low mass splittings, since we, in cut and count, do a very gentle cut on this variable while the BDT finds it very interesting in the training. For later work on a similar analysis, this could be interesting to test in the cut and count analysis as well. The other features the BDT finds interesting, for all four processes, are more or less the same as for low mass splittings.



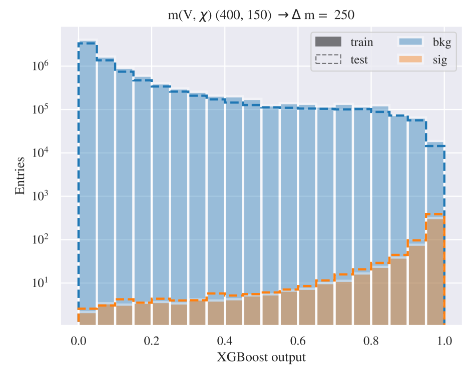
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



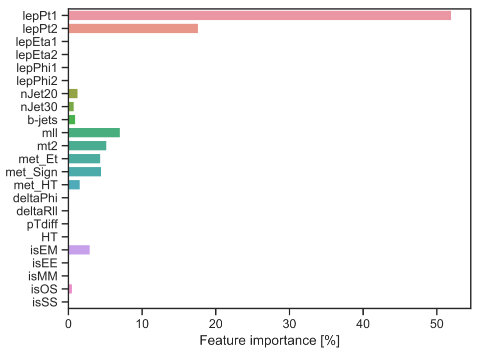
(d) Mono-Z.

Figure 5.8: Test vs train for intermediate mass splittings with all features done with the BDT. Here the test set is scaled up to match the number of training events.

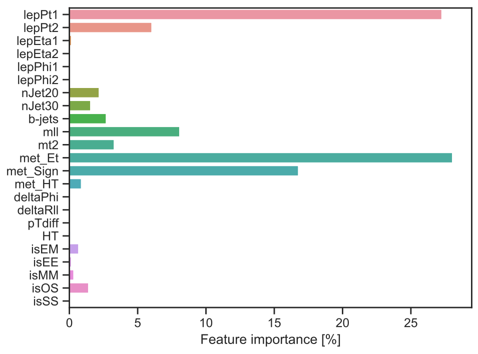
The results from testing the trained model is shown in figure 5.8. As we could see for the low mass splittings, it is overall a pretty good match except for a couple of bins in figure 5.8a. Here we can see that some of the bins are lacking some background, but the test has managed to do a pretty good fit anyway. This is a result of limited statistics in the training set since it should match the amount of signal you have available. Therefore the training set is subject to statistical fluctuations. In the signal it is a bit more fluctuations and train and test does not have that good compliance. We can also see that we have a bit more signal close to one and a bit less background in this region. This is not that surprising considering that we are in a region where our final state particles have typical masses larger than the SM particles.

High mass splittings

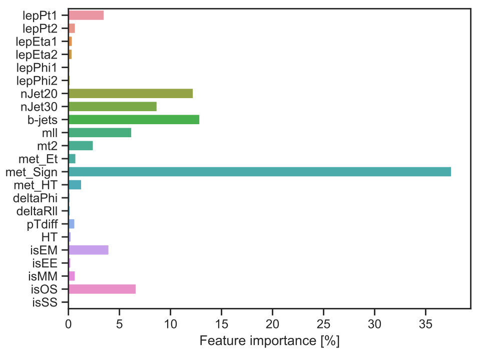
The last BDT we have trained is the one on high mass splittings. Figure 5.9 shows some interesting differences in the importance of the various features compared to the observations for the other mass splittings.



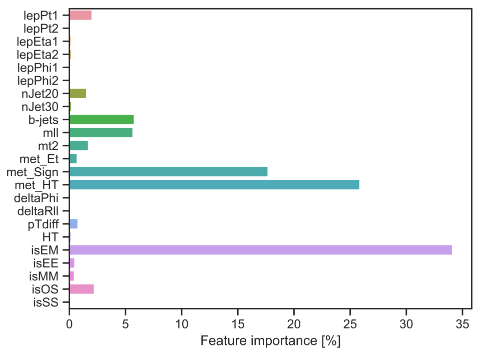
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



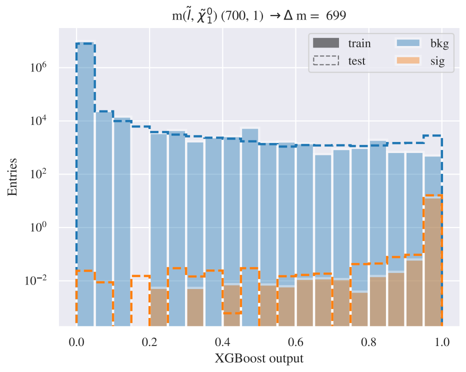
(c) Chargino production via $W^{\pm}$.



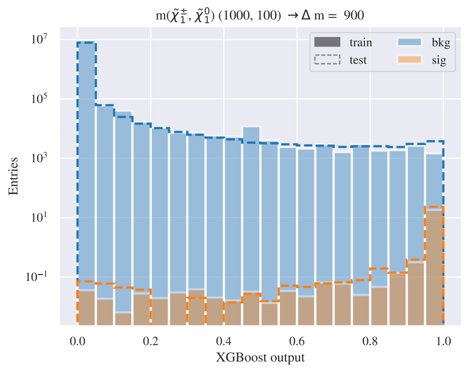
(d) Mono-Z.

Figure 5.9: Feature importance for high mass splittings for all four processes using all features during training.

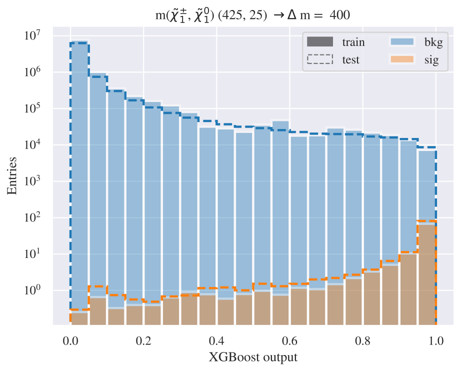
As we can see in figure 5.9, the momentum for the leading lepton have become a lot more interesting for the BDT especially for the direct slepton production and chargino production with slepton/sneutrino-mediated-decay. In addition the momentum for the subleading lepton have also gotten interesting for the two processes including sleptons. The rest of the features seems to be recurring for the processes no matter what mass splitting we look at.



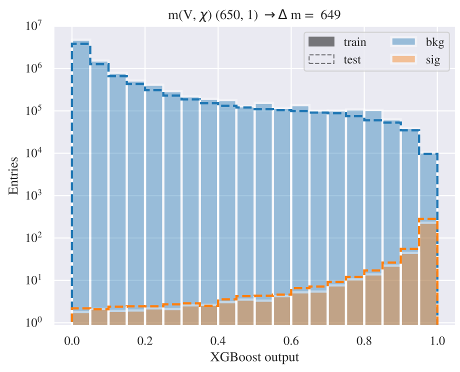
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



(d) Mono-Z.

Figure 5.10: Test vs train for high mass splittings with all features done with the BDT. Here the test set is scaled up to match the number of training events.

In figure 5.10 we can see the results from testing our BDT. In figure 5.10c and 5.10d the match is overall pretty good between test and training, while for the processes in figure 5.10a and 5.10b this does not match that well. It can be many reasons for this e.g. having too little input data to perform a proper training, the features that are chosen might be less efficient for the process with a given mass splitting or tendency to over/underfit. Nevertheless, it has succeeded in correctly classifying the signal and background to some extent. We can see for all four processes that we have more signal in the last bin than in any of the other models shown earlier in this chapter. This is of course not that surprising since these masses are outside of the SM masses we already know very well.

5.3 Building, training and testing the Neural Network

In the following sections we are going to look at how the Neural Network (NN) is built up, trained and tested before we in the end go to the last part of the ML where we are going to test both trained algorithms on real data.

5.3.1 Building

As for the BDT, the NN uses some already existing libraries. The main parts of the building is covered in chapter 4.3 and the parameters we us to optimize the NN is presented in table 5.7.

Parameter	**Value**
Number of nodes	300
Number of hidden layers	5
Dropout rate	0
Batch size	32
Epochs	100000
Learning rate	10^{-5}
L1	0
L2	10^{-3}

Table 5.7: An overview of the different parameters used during training to obtain the results for the NN.

Most of the parameters are self explanatory, but we are going to go through them step by step so we are sure that we have understood every one of them. The first one is the number of nodes in each layer (and not the total number in the network). Then we have the number of hidden layers which is the number of layers we want inside the network in addition to the input and output layer. Dropout rate is a number between 0 and 1 and is taken into consideration if you want to drop a percentage of your input when you start training. As we can see, we have not used the dropout rate other than telling our NN that it should not drop any of the input events, so we are not discussing the advantages or disadvantages in using this here. The next parameter is the batch size which is the number of iterations we go through the data we give the NN. This can be any value above 1 but it is common to set it to a number of power of 2. The epochs and learning rate should be clear from chapter 4.3, where the epochs is not a very interesting parameter due to the early stopping and the learning rate gets optimized by Adam. The last two parameters we have to take into consideration are L1 and L2, which is what we call *layer weight regularizes.* These are applied to add a penalty on the layers kernel and are simply added as a term in the cost/loss function in equation 4.5.

In addition to the parameters, we have to give our network an activation function for each layer in the network. In chapter 4 we looked at different activation functions, where we have used two of them in our network, namely ReLU and Sigmoid. We have given the hidden layers ReLU as activation function and Sigmoid is only used for the output layer.

As for the BDT, we have different compositions of mass splittings and features in the different NNs as well.

5.3.2 Training

The training of the NN is the most time consuming step of all of the ML done in this thesis. It uses more time on solving the same problem as the BDT, which is one of the disadvantages of the NN compared to the BDT. The advantage is that the NN can do a better job than the BDT if we are able to choose the right parameters for training. Since the training of the NN takes some time, we have chosen to give it a patience of only 10 steps, instead of 20. After some trial and error during the work with this thesis we have concluded that when the network has not improved in 10 steps, it will not improve much more after 20 either. So after 10 steps, the early-stopping will come in and stop the training and save the model at the best epoch.

5.3.3 Testing

The next step is to test the NN on our test set. This is again done on the same signal samples we used in the cut and count analysis. The results are presented with all four processes together with low, intermediate and high mass splittings and which set of features (low, high or all) performing best are evaluated by looking at the AUC-scores in table 5.8.

AUC-score

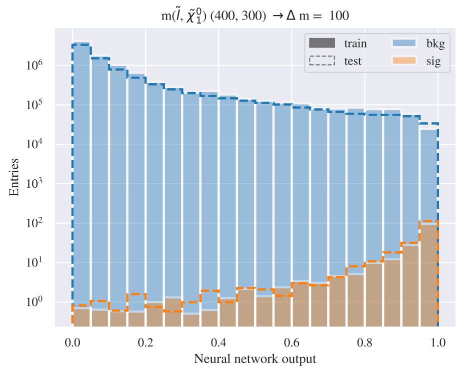
In table 5.8, we can see the AUC-score for each process and all combinations of mass splittings and features. There are several scores that are the same for the different features, but here we don't have the opportunity to look at the feature importance (because this is not relevant for NN) to make the decision. Instead we have chosen to show the same compositions as for the BDTs because the ones with the best score in NN are the same as with BDTs.

Level	**Δm**	$\tilde{\mathbf{l}}\tilde{\mathbf{l}}$	$\tilde{\chi}_1^{\pm} \rightarrow \tilde{\mathbf{l}}/\tilde{\nu}$	$\tilde{\chi}_1^{\pm} \rightarrow \mathbf{W}^{\pm}$	**Mono-Z**
	Low	0.91	0.90	0.90	0.94
High	Intermediate	0.99	0.97	0.93	0.96
	High	1.00	1.00	0.96	0.96
	Low	0.96	0.93	0.93	0.96
Low	Intermediate	0.99	0.98	0.95	0.97
	High	1.00	1.00	0.97	0.97
	Low	0.97	0.94	0.94	0.96
All	Intermediate	1.00	0.98	0.96	0.97
	High	1.00	1.00	0.97	0.98

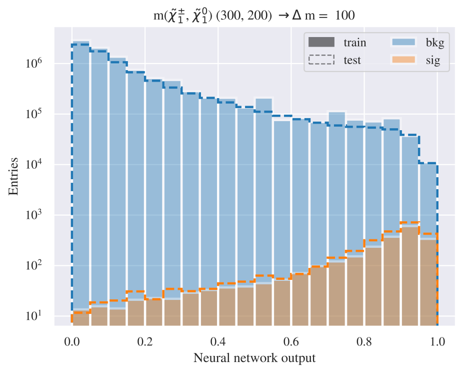
Table 5.8: The AUC score for the different processes trained on different compositions of features and mass splittings for the NN.

Low mass splittings

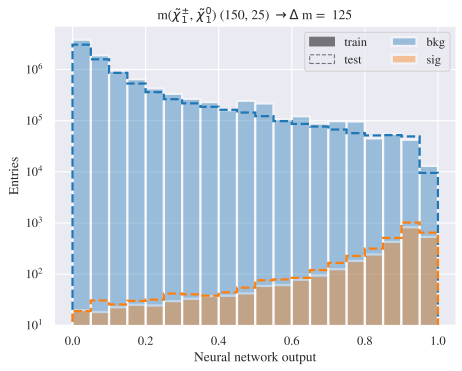
The first results from the NN we are going to look at are the ones with low mass splittings and trained on all features.



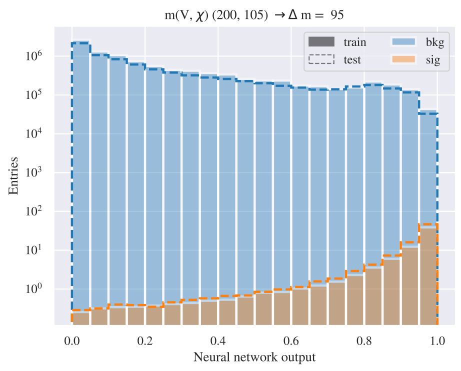
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^\pm$.



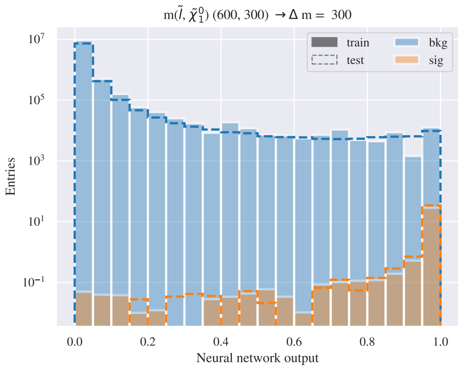
(d) Mono-Z.

Figure 5.11: Test vs train for low mass splittings with all features done with the NN. Here the test set is scaled up to match the number of training events.

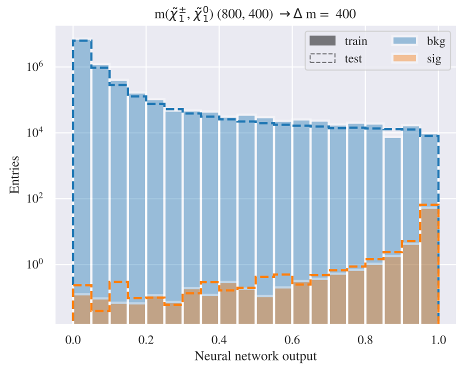
Figure 5.11 shows a pretty good agreement between the test and training, but we don't have much signal for either of the four processes except for the chargino production processes, where we have been able to get more signal and less background towards 1.

Intermediate mass splittings

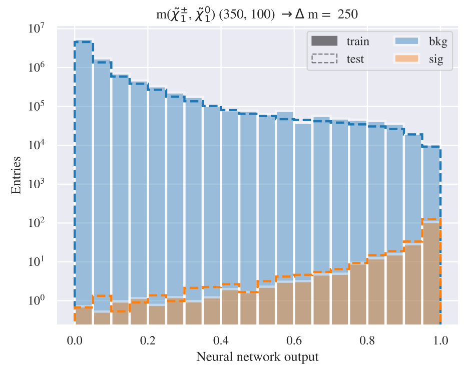
The next results we are going to look at are from the model trained on intermediate mass splittings and all features, which are shown in figure 5.12.



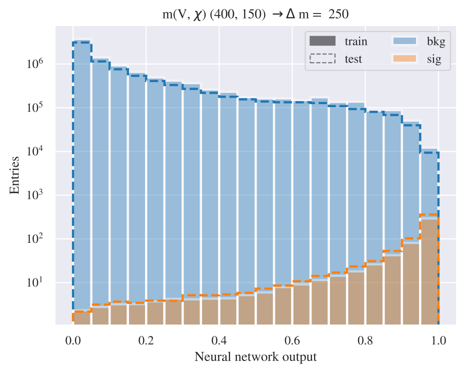
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



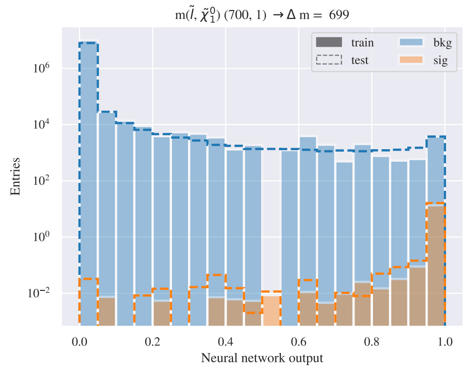
(d) Mono-Z.

Figure 5.12: Test vs train for low mass splittings with all features done with the NN. Here the test set is scaled up to match the number of training events.

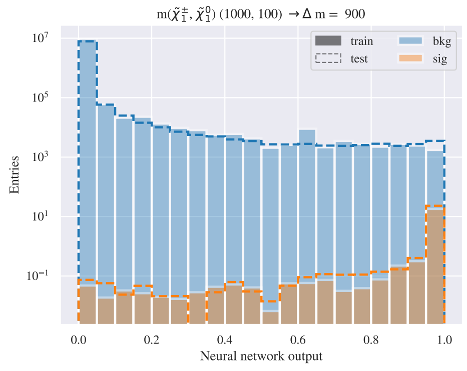
As for the BDT there is a good compliance between train and test with some fluctuations. What might be interesting in these results compared to the results in figure 5.11 for low mass splittings, is that we actually got less signal towards one for intermediate mass splittings than we had for low. We would expect that this would be the other way around because of the bigger mass splittings between these particles.

High mass splittings

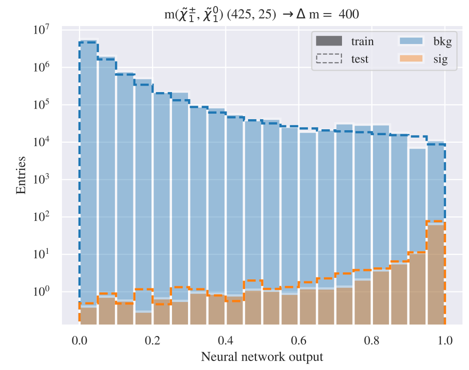
The last trained NN we are looking at is the one trained on high mass splittings and all features. The results are presented in figure 5.13.



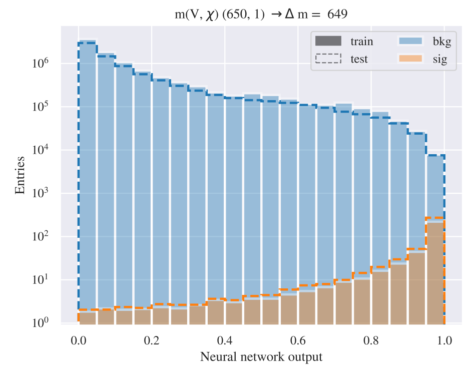
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



(d) Mono-Z.

Figure 5.13: Test vs train for low mass splittings with all features done with the NN. Here the test set is scaled up to match the number of training events.

For high mass splittings we can see that we have less background in the last bin for all four processes, which makes sense since it should be easier to separate the signal from the background at these masses. We still have some fluctuations, especially in the signal test set, but overall it seems that the test sample is not too much affected of this.

5.3.4 Summarizing the ML performance

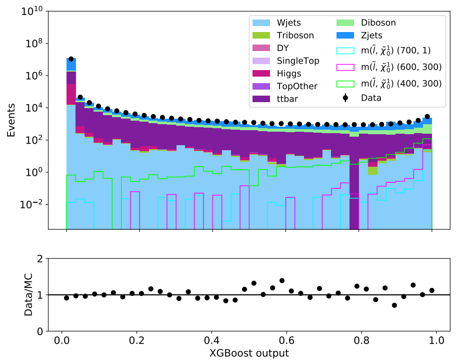
Before we move on to testing the trained ML algorithms on actual collected data from ATLAS, we are going to summarize the BDT and NN a bit. The BDT and NN in general perform well even though the separation of background and signal could have been improved. To discuss some of the problems that have occurred, we can see that some bins are partly or completely empty for the background training set. We suspect that this can come from the reduction of the MC set to match the number of signal events, where some of the background that probably should have been in these bins are taken out before training. The reduction of the MC set was necessary to train the ML methods to perform well. But, since the test set overall seems to be unaffected by this, we have chosen to leave it like this.

For the the different mass splittings, we have also experienced that the ML have found it, in some cases, hard to classify the signal as signal. It is not just the masses of the particles themselves. As we have mentioned earlier in this chapter, it might have something to do with the fact that the masses are close to or similar to the SM particle masses. If the mass differences in the decay are large, more phase space is available in the decay, which typically give larger momentum to the decay products - in case of SUSY and DM this means more MET. When the mass splitting approaches the W-mass, it is almost impossible to distinguish the direct slepton decay and a SM WW event (where each W decays to lepton+neutrino). That is why mass splittings around and below 100 GeV are extremely difficult since one has to fight with an almost irreducible background from WW.

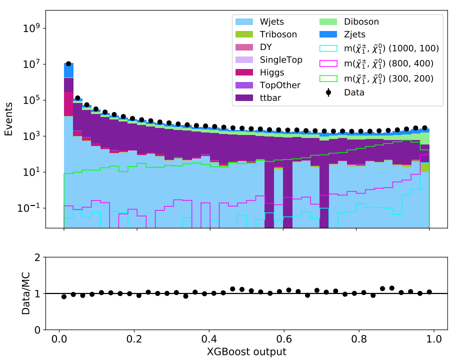
5.4 Testing the BDT and NN on real data

In this section we are going to see how the BDT and the NN perform on non-simulated data, which is an extra test to see how well they are trained. We will look at all three signal samples for each process together, where the background and data come from the ML model trained on high mass splittings. This is, of course, tested on the ML models trained on low and intermediate mass splittings as well. However, exactly which trained model we use for the comparison of background and data did not have a big impact and therefore we have chosen to rather show all signals together for a better comparison.

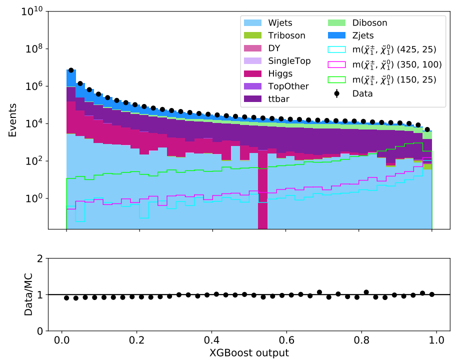
First we are going to look at the results from the BDT which are presented in figure 5.14, where we have stacked the different backgrounds to see how much each contribute in different areas.



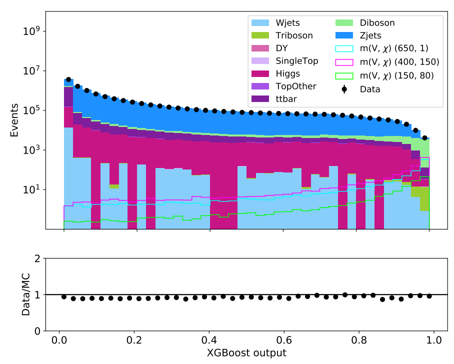
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



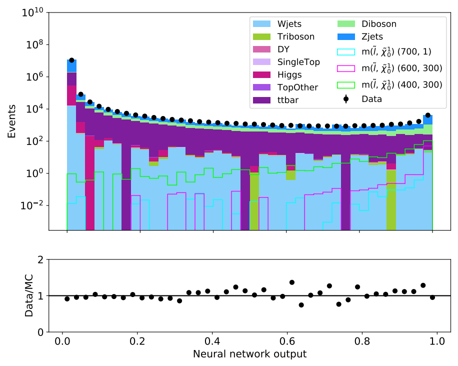
(d) Mono-Z.

Figure 5.14: The output from the test set together with real data using BDTs trained on all features. Here the different backgrounds are stacked to see how much they contribute together with the three benchmark signals for each process.

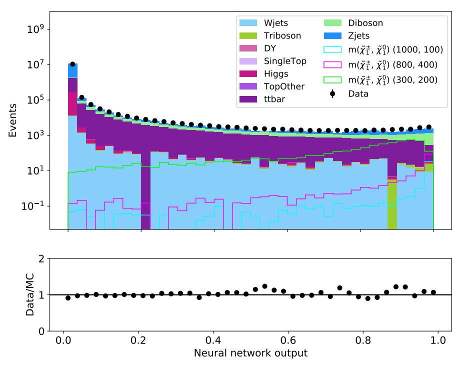
In figure 5.14, we can see that the data and background have a pretty good compliance with some minor fluctuations, which might be caused by the weights for each event. Some of the events have gotten negative weights when we have multiplied them together, but since the differences are well within $\pm 20\%$, we have not considered this as a problem. We can also see that there are some fluctuations in the different background contributions in the bins, but since we are looking at around 30% of the MC background we have to expect some statistical fluctuations. This seems not to affect overall the performance, however, which means that we can trust the results we have obtained.

It might be a bit difficult to see because of the logarithmic scale, but in the last bin we have most contribution from dibosons. This is a very good sign because this background is the one that looks most like the signature in question and suggests that the BDT handles the different background contributions well. This can also be drawn back to the cut and count analysis, where the diboson background definitely is the dominating background after applying all cuts.

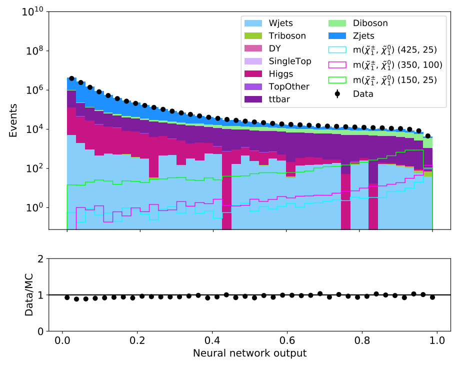
We have also tested the NN on data which are presented in figure 5.15.



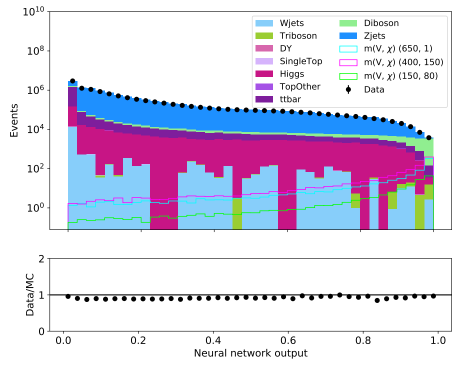
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



(d) Mono-Z.

Figure 5.15: The output from the test set together with real data from the NNs trained on all features. Here the different backgrounds are stacked to see how much they contribute together with the three benchmark signals for each process.

When comparing the results for the BDT and the results for the NN in figure 5.14 and 5.15, we cannot see much differences except for a couple of fluctuations in the different background contributions. The data and the MC seem to have a good compliance and diboson is the dominating background where we expect to have most of our signal.

It looks like our BDT and NN do a good job in separating the signal and background, and it handled the real data very well as well. The next and last step in this thesis is to check if our models are sensitive to actually discover new physics and compare the results from using ML with the results from the cut and count analysis.

Chapter 6

Machine Learning or Cut and Count?

Until this point we have looked at how well the different analyses perform on the same dataset and we are now going to see how sensitive they are to the signal samples we have chosen. To be able to do this we have to know how we can quantify the sensitivity before we move on to compare ML to cut and count analysis in searching for Supersymmetry and Dark Matter.

6.1 Quantifying the sensitivity

The overall goal of the analysis carried out in this thesis is to optimize the sensitivity to discover new physics and possibly in the end to claim discovery of new particles. To check if we have succeeded in making our model sensitive to new physics we can check the expected p-values and significances for the signal regions considered.

Each event in the data will either pass or fail the cuts defining the signal region used in the analysis. The number of events therefore follows a binomial distribution, and the probability of finding n events in the signal region is given by equation 6.1.

$$P(n|N,p) = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n}, \tag{6.1}$$

where N is the total number of events and p the probability for the event to pass the cuts. If the number of total events gets very big and the probability p gets very small, we can approximate equation 6.1 as a Poisson distribution, which is given by

$$P(n|\nu) = \frac{\nu^n}{n!} e^{-\nu}, \tag{6.2}$$

where ν is given by a hypothesis. This equation (eq. 6.2) gives us a probability of n observed events when the expectation is to observe ν events.

When we talk about hypotheses in this thesis, we mean *background-only* (b-only) and *signal + background* (s+b) hypotheses. The b-only hypothesis is the SM, which is a well known theory. We are going to check this against the s+b hypothesis which in our case will be one of the three SUSY processes or the mono-Z process. This gives us the opportunity to introduce the p-values. A p-value is a measure to judge the deviation of the observed number of events from the b-only hypothesis expectations. If we assume that only the SM processes contribute we can use the p-value to get the probability of observing as many, or more, events as we can find in the data. With a perfectly known background the p-value can be found by

$$p = \sum_{n=n_{obs}}^{\infty} f(n;b), \tag{6.3}$$

where $f(n;b)$ is the Poisson distribution given in equation 6.2. It is common to convert the p-value into an observed significance z with a unit Gaussian Φ. This is given by

$$z = \Phi^{-1}(1-p) \tag{6.4}$$

and the unit is expressed with a σ as in number of standard deviations from the center of the Gaussian distribution. To claim discovery of a new particle we need an observed significance above 5 σ. We are not going to check the observed significance in this thesis, but calculate the expected significance to see if there is any hope to discover new physics for the models we are looking at. The expected significance is the significance associated with the median number of expected events under the s+b hypothesis. It quantifies how well the two hypotheses we are looking at are separated. This is given by equation 6.4, where the p-value now is expressed as a *confidence level* (CL) given as

$$CL_{s+b} = \sum_{n=0}^{q_{obs}} \frac{(s+b)^n}{n!} e^{-(s+b)}, \tag{6.5}$$

where q_{obs} is the number of observed events. If the CL_{s+b} is below 5% we exclude that we can find any new physics with the selection criteria we have applied. Putting equation 6.4 and 6.5 together we get the expression

$$Z_N = \Phi^{-1}(1 - CL_{s+b}), \tag{6.6}$$

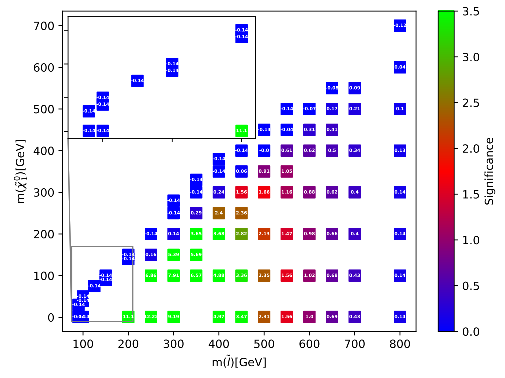
where Z_N is the expected significance. To claim exclusion in the signal region we need a $CL_{s+b} \leq 5\%$ which corresponds to $Z_N \leq 1.64\sigma$. In this thesis the Z_N is calculated by an already existing tool in ROOT called `RooStats.NumberCountingUtils.BinomialExpZ` which simply takes the number of events for both background and signal as input together with a systematic background uncertainty of e.g. 20%.

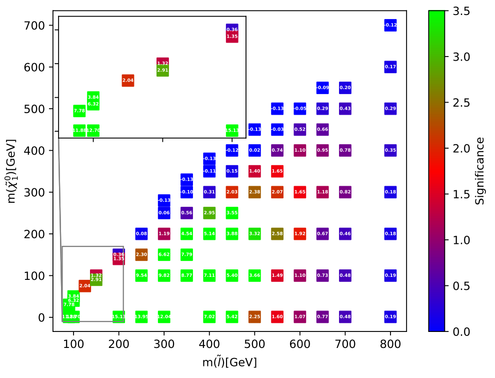
6.2 Results

In this section the expected significance for all four new physics processes and the three analysis methods are presented. The significance is calculated as described in the previous section. For the ML methods we have studied the three benchmark signals (from high, intermediate and low mass splittings) for each process and optimized the sensitivity by applying a cut at each bin in figure 5.14 and 5.15. The optimization procedure starts by calculating the significance when integrating over all the bins (i.e complete distribution of ML scores from 0 to 1). Then we move to the next bin, integrate up to 1 and compute the corresponding significance. This procedure is repeated until only the last bin remains in the significance calculation. At the end we see which cut in the ML score gave the best significance and use this to calculate the significance for all signals for the respective mass splitting. The bin which is used to place the cut for each optimization varies a bit, but is usually within the last 10% of the bins in figure 5.14 and 5.15. This is expected since we see that we have a lot more signal in this region of the output than in the first bins.

6.2.1 Direct slepton production

The first results we are going to look at is for the direct slepton production. The results are presented in figure 6.1, where we have the mass of the slepton on the x-axis and the mass of the neutralino on the y-axis.


(a) Cut and count.


(b) Boosted Decision Tree.

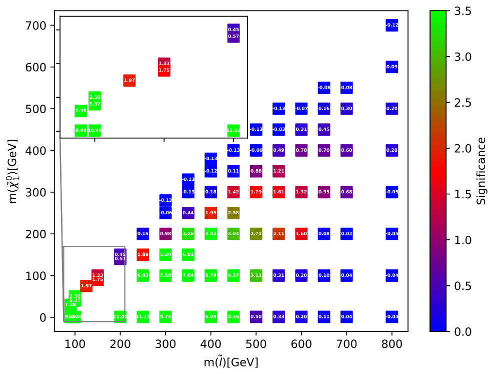

(c) Neural Network.

Figure 6.1: Significance plots for direct slepton production where all features are used during training the ML models. Since we have an overlap on some of the samples in the lower left corner, we have zoomed in at this area and added it in a empty area in the upper left corner.

This figure shows the expected significance for cut and count (6.1a), BDT (6.1b) and NN (6.1c). The color bar is fixed at 3.5 in all plots to make it easier to compare the different results. Exactly how much greater than 3.5 the significance is is not so interesting since for these points we already have very good sensitivity (the 95% CL exclusion limit, in figure 3.13, corresponds to a significance of 1.63). Nevertheless, the exact significance value for each point is shown in the plot for each point. As we can see, the expected significance is greater for several signal samples for both the BDT and NN compared with the cut and count method. In particular, signal samples with low mass splittings ($\Delta m \leq 100$ GeV), found along the diagonal in figure 6.1, show big improvements for both ML methods compared to cut and count. As stated in section 5.3.4 signals with low mass splittings are particularly hard to distinguish from SM background, making these results very interesting. To see how much better both ML algorithms are performing, we have presented some signal samples from the lower left corner in figure 6.1 in table 6.1.

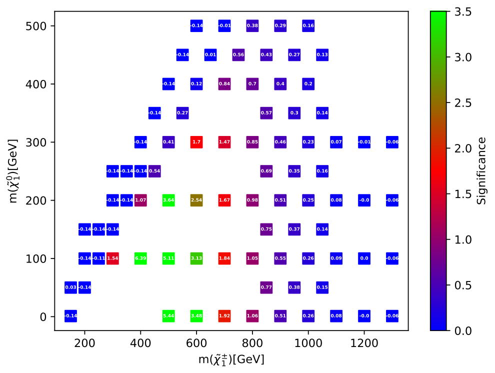
$\mathbf{m(\tilde{l}, \tilde{\chi}_1^0)}$ **[GeV]**	**|BDT/C&C|**	**|NN/C&C|**	**|BDT/NN|**
(90, 1)	84.29	67.79	1.25
(90, 30)	55.57	52.71	1.05
(100, 1)	90.71	74.71	1.21
(100, 40)	45.57	38.36	1.19
(100, 50)	27.43	25.00	1.10

Table 6.1: Differences in expected significance for a selection of signals in lower left corner in figure 6.1. This is done to compare how much better the ML performed for low mass splittings than the cut and count.

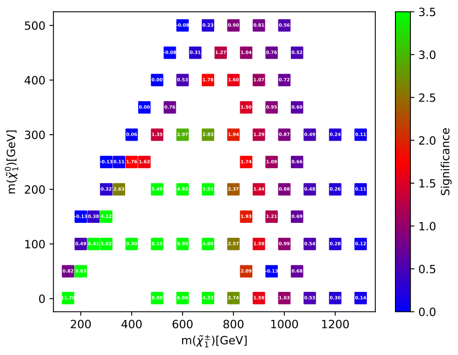
From the table we can see that the sensitivity for both ML methods are much better than for the cut and count. We can also see that the BDT overall have some better sensitivity than the NN, but there is no doubt that they are both preferable analysis methods in this area of mass splittings.

6.2.2 Chargino pair with slepton/sneutrino-mediated-decay

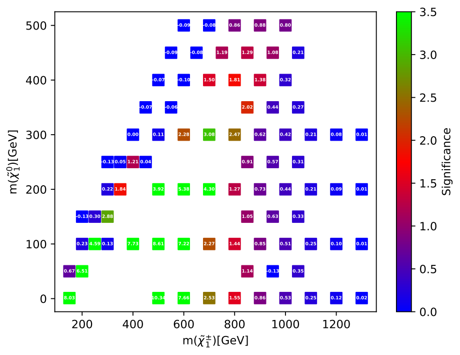
The results for the chargino production with slepton/sneutrino-mediated-decays are presented in figure 6.2. Here are the x-axis the mass of the chargino, while the y-axis remains the mass of the neutralino.



(a) Cut and count.



(b) Boosted Decision Tree.



(c) Neural Network.

Figure 6.2: Significance plots for chargino production with slepton/sneutrino-mediated-decay where all features are used during training of the ML models.

Figure 6.2 shows the expected significance for all three analysis methods used in this thesis. As for the direct slepton production, we can see that the ML have a greater sensitivity than the cut and count. The ML methods also have a greater sensitivity for low mass splittings ($\Delta m < 200$ GeV) compared to cut and count, which we can easily see in table 6.2.

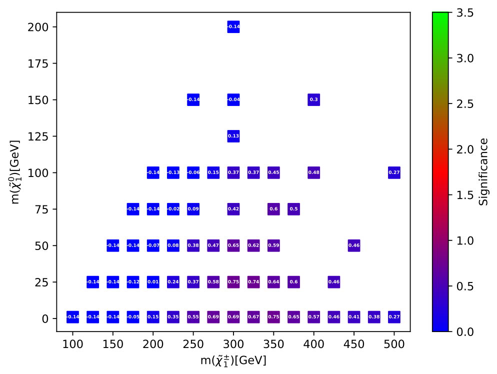
$\mathbf{m(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0)}$ **[GeV]**	**|BDT/C&C|**	**|NN/C&C|**	**|BDT/NN|**
(150, 1)	83.57	57.36	1.46
(150, 50)	27.33	22.33	1.22
(200, 50)	68.79	46.50	1.48
(200, 100)	3.50	1.64	2.13
(250, 100)	58.27	41.73	1.40
(300, 100)	3.26	0.08	38.62

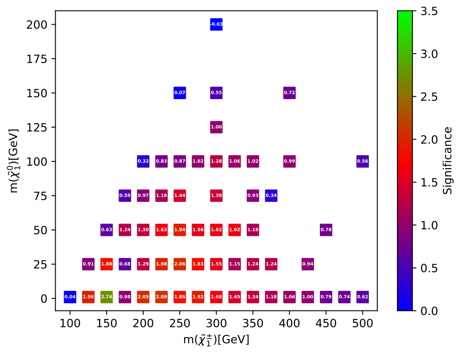
Table 6.2: Differences in expected significance for a selection of signals in lower left corner in figure 6.2. This is done to compare how much better the ML performed for low mass splittings than the cut and count.

In table 6.2, we have chosen signal samples in the same area of the plot as for the direct slepton production. As we can both see and expect from figure 6.2, the sensitivity is not that good for all of the signals presented in 6.2. However, the ML methods, and maybe especially the BDT, is more prefereable for these mass splittings in the process we are looking at.

6.2.3 Chargino pair with W-boson-mediated-decay

The next results we are going to look at are for the chargino production with W-boson-mediated-decay, which are presented in figure 6.3. The x- and y-axis are the same as for the chargino production with slepton/sneutrino-mediated-decay.


(a) Cut and count.


(b) Boosted Decision Tree.

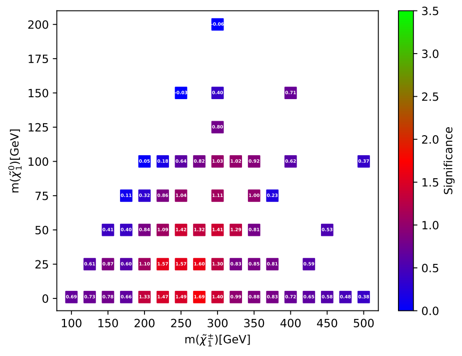

(c) Neural Network.

Figure 6.3: Significance plots for chargino production with W-boson-mediated-decay where all features are used during training the ML models.

From figure 6.3, we can easily see that the sensitivity are better for most signal samples for the ML methods compared with the cut and count method. However, the sensitivity over all is not very good, and we should not expect to make any discoveries at any mass splitting. This is somewhat expected from the fact that the cross-section for W-mediated decay is much smaller since we loose quite a lot by ignoring the quark decays of the W (only 11% of the BR is to leptons). In addition the W-mediated decay is identical to SM WW production which is a challenging background. More optimization would be needed to increase the sensitivity in this channel, which also are discussed in the publication [9]. Even though the sensitivity is not that good for this process, it still interesting to see how much better the sensitivity is for the ML methods than for the cut and count. Some selections are presented in table 6.3.

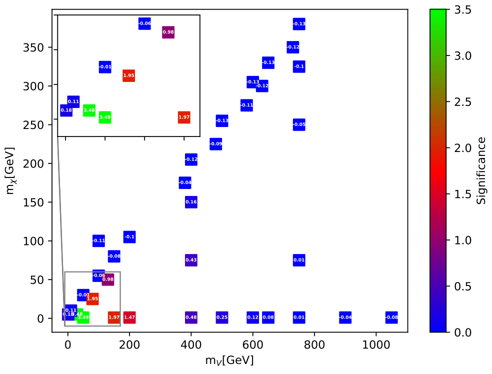
$\mathbf{m}(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0)$ **[GeV]**	**|BDT/C&C|**	**|NN/C&C|**	**|BDT/NN|**
(100, 1)	0.29	4.93	0.06
(125, 1)	14.00	5.21	2.69
(125, 25)	6.50	4.36	1.49
(150, 1)	19.57	5.57	3.51
(150, 25)	13.43	6.21	2.16

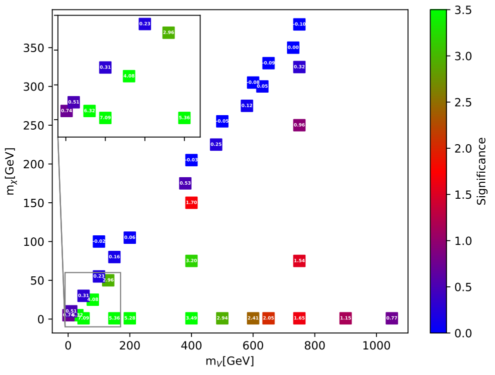
Table 6.3: Differences in expected significance for a selection of signals in lower left corner in figure 6.3. This is done to compare how much better the ML performed for low mass splittings than the cut and count.

As we can see in table 6.3, we can see that both ML methods have a better sensitivity than the cut and count. This is also the case for the BDT at $m(\tilde{\chi}_1^{\pm}, \tilde{\chi}_1^0)$ = (100, 1) GeV, even though it looks like the cut and count have a better sensitivity for this signal. The reason for this is that the significance for cut and count is negative for this signal and we have calculated the absolute value of the ratio to avoid negative ratios. However, the NN seem to have a better sensitivity for this signal, so a combination of BDT and NN would maybe be beneficial in this process.

6.2.4 Mono-Z

The last results we are going to look at is for the mono-Z process. The results are presented in figure 6.4, where we have the mass of the mediator on the x-axis and the mass of the DM particle on the y-axis.


(a) Cut and count.


(b) Boosted Decision Tree.

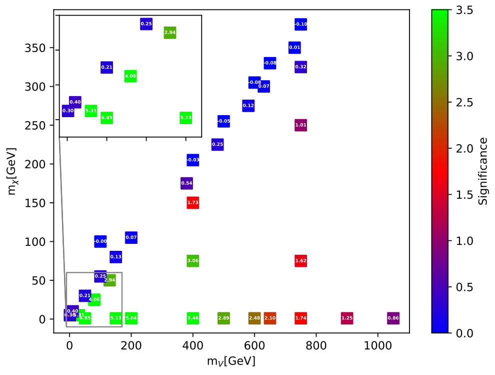

(c) Neural Network.

Figure 6.4: Significance plots for the mono-Z process where all features are used during training the ML models.

The figure shows that the cut and count method is somewhat sensitive to signals with low mass splittings ($\Delta m < 100$ GeV). However, the ML methods are still more sensitive to a wider range of signals with low mass splittings, which we can have a closer look at in table 6.4.

$\mathbf{m(V,\chi)}$ **[GeV]**	**|BDT/C&C|**	**|NN/C&C|**	**|BDT/NN|**
(1, 5)	4.11	1.67	2.47
(10, 10)	4.64	3.64	1.28
(30, 5)	1.82	1.53	1.19
(50, 1)	2.03	1.96	1.04
(50,30)	31	21	1.48
(80,25)	2.09	2.08	1.01

Table 6.4: Differences in expected significance for a selection of signals in lower left corner in figure 6.4. This is done to compare how much better the ML performed for low mass splittings than the cut and count.

The table shows that the BDT is constantly a bit more sensitive than the NN, which have been the case for almost all signals we have had a closer look at. Both ML methods are more sensitive than the cut and count in this range of masses, which also is the case for all four processes.

6.2.5 Summarizing the results

From our studies it seems that overall the ML methods show a better sensitivity compared to the cut and count method, especially for low mass splittings. Low mass splittings is what we are most interested in improving with ML techniques since the cut and count analyses are less sensitive here. This is also the case for the ML methods trained on low level and high level features, but they are not as good as the ones trained on all features. These results can be found in appendix D.

The difference between the performance of the BDT and NN are not very noticeable, but it seems like the BDT is slightly more sensitive overall. But, as mentioned earlier in this thesis, the NN are much more sensitive to the parameters we give it than the BDT. This means that with some more time for optimization of the hyperparameters, we would probably be able to get just as good results as for the BDT or even better.

In this thesis we have not done a full analysis of the cut and count method as is presented in chapter 3. We have only used one signal region (for SUSY $m_{T2} > 160$ GeV) while in the publication they do a multi-bin likelihood fit including the control and signal regions. This is why the exclusion curves from the publications are better than what we have obtained for the cut and count. Nevertheless, the fact that the ML methods show a better performance towards low mass splitting is very promising and with more optimization the sensitivity could be further increased. For direct slepton, especially, it seems like we are able to get a sensitivity which is better than what was achieved in the publication in the lower left part of the parameter space.

Chapter 7

Conclusions and outlook

In this thesis we have been searching for Supersymmetry (SUSY) and Dark Matter (DM) using two Machine Learning (ML) based algorithms, namely Boosted Decision Trees (BDT) and Neural Networks (NN). We have also reproduced the results from some published work by the ATLAS collaboration [9, 11], which have searched for the same signal processes using a simple cut and count method. The results from the cut and count method are used to evaluate how well the ML methods performed by comparing the expected sensitivity. The BDT and NN were trained on different compositions of mass splittings (high, intermediate and low) and features/kinematical variables, and for four different signal processes (three SUSY and one DM). Both ML methods have overall performed very well and reached AUC-scores above 0.90 for every trained model. An AUC-score of 0.90 means that the ML methods are able to classify the signal as signal and background as background 90% of the time.

Looking at the achieved sensitivity of the three analysis methods and comparing them to each other it is clear that the ML methods overall have more sensitivity for the signals than the cut and count method,i n particular for low mass splittings, which is the experimentally challenging because it requires working with low p_T leptons and lower E_T^{miss}. To achieve a high sensitivity for low mass splittings is difficult in the cut and count method and it is therefore very satisfying that the ML perform rather well for these signals. We can, with these results, conclude that ML may indeed be an efficient and rewarding technique when performing searches for new physics phenomena.

For future research using these or similar ML methods, it would be interesting to see what we could have done to make the performance better. In this thesis we have used a couple of precuts, which made the ML methods somewhat biased and thus not able to learn everything from a less selective data input. It would be interesting to see how it would perform without these precuts and with more features included. However, this would imply more powerful computing infrastructure and more time to perform our analysis because of the massive datasets.

It would also be interesting to do a hyper parameter scan to see which parameters are the best to use for our BDT and NN. However, although this way to perform a ML analysis is definitely very useful, the most preferable way would have been to make the ML method independent of a hypothesis. This can be done through a NN where we do a so-called *anomaly detection*. Instead of training on MC-samples for both signal and background, we train only on background before we test with data. The network would then tell us if there is something interesting to see in the data or not. However, this introduces many new challenges. One of them is to know what we have actually found because there is no hypothesis to confirm. Unfortunately, today's ML algorithms are not complex enough to handle this problem, but this is going to be a very interesting research to follow in the future.

Appendix A

Signal sample tables

A.1 Direct slepton production

$(\mathbf{m}(\tilde{l}),\ \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**	$(\mathbf{m}(\tilde{l}),\ \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**
(90, 30)	60	(350, 330)	20
(90, 1)	89	(350, 300)	50
(100, 50)	50	(350, 250)	100
(100, 40)	60	(400, 380)	20
(100, 1)	99	(400, 350)	50
(125, 75)	50	(400, 300)	100
(150, 100)	50	(450, 400)	50
(150, 90)	60	(450, 350)	100
(200, 150)	50	(500, 450)	50
(200, 140)	60	(500, 400)	100
(250, 240)	10	(550, 500)	50
(250, 200)	50	(550, 450)	100
(250, 150)	100	(600, 500)	100
(300, 280)	20	(650, 550)	100
(300, 250)	50	(800, 700)	100
(300, 200)	100		

Table A.1: Table of signal samples with low mass splitting ($\Delta m \leq 100$ GeV) for the direct slepton production.

$(\mathbf{m}(\tilde{l}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**	$(\mathbf{m}(\tilde{l}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**
(200, 1)	199	(500, 100)	400
(250, 100)	150	(550, 400)	150
(250, 1)	249	(550, 350)	200
(300, 150)	150	(550, 300)	250
(300, 100)	200	(550, 200)	350
(300, 1)	299	(600, 450)	150
(350, 200)	150	(600, 400)	200
(350, 150)	200	(600, 300)	300
(350, 100)	250	(600, 200)	400
(400, 250)	150	(650, 500)	150
(400, 200)	200	(650, 450)	200
(400, 100)	300	(650, 400)	250
(400, 1)	399	(650, 300)	350
(450, 300)	150	(700, 550)	150
(450, 250)	200	(700, 500)	200
(450, 200)	250	(700, 400)	300
(450, 100)	350	(700, 300)	400
(450, 1)	449	(800, 600)	200
(500, 350)	150	(800, 500)	300
(500, 300)	200	(800, 400)	400
(500, 200)	300		

Table A.2: Table of signal samples with intermediate mass splitting (100 GeV $< \Delta m <$ 450 GeV) for the direct slepton production.

$(\mathbf{m}(\tilde{l}),\, \mathbf{m}(\tilde{\chi}_1^0))$ [GeV]	Δm [GeV]	$(\mathbf{m}(\tilde{l}),\, \mathbf{m}(\tilde{\chi}_1^0))$ [GeV]	Δm [GeV]
(500, 1)	499	(700, 200)	500
(550, 100)	450	(700, 100)	600
(550, 1)	549	(700, 1)	699
(600, 100)	500	(800, 300)	500
(600, 1)	599	(800, 200)	600
(650, 200)	450	(800, 100)	700
(650, 100)	550	(800, 1)	799
(650, 1)	649		

Table A.3: Table of signal samples with high mass splitting ($\Delta m \geq 450$ GeV) for the direct slepton production.

A.2 Chargino pair with slepton/sneutrino-mediated-decay

$(\mathbf{m}(\tilde{\chi}_1^\pm),\, \mathbf{m}(\tilde{\chi}_1^0))$ [GeV]	Δm [GeV]	$(\mathbf{m}(\tilde{\chi}_1^\pm),\, \mathbf{m}(\tilde{\chi}_1^0))$ [GeV]	Δm [GeV]
(150, 1)	149	(300, 250)	50
(150, 50)	50	(350, 250)	100
(200, 50)	150	(400, 250)	150
(200, 100)	100	(400, 300)	100
(200, 150)	50	(450, 350)	100
(250, 100)	150	(500, 400)	100
(250, 150)	100	(550, 450)	100
(300, 150)	150	(600, 500)	100
(300, 200)	100		

Table A.4: Table of signal samples with low mass splitting ($\Delta m < 200$ GeV) for chargino production with slepton/sneutrino-mediated-decay.

$(\mathbf{m}(\tilde{\chi}_1^{\pm}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**	$(\mathbf{m}(\tilde{\chi}_1^{\pm}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**
(300, 100)	200	(700, 200)	500
(400, 100)	300	(700, 300)	400
(450, 250)	200	(700, 400)	300
(500, 1)	499	(700, 500)	200
(500, 100)	400	(750, 450)	300
(500, 200)	300	(800, 300)	500
(500, 300)	200	(800, 400)	400
(550, 350)	200	(800, 500)	300
(600, 1)	599	(850, 350)	500
(600, 100)	500	(850, 450)	400
(600, 200)	400	(900, 400)	500
(600, 300)	300	(900, 500)	400
(600, 400)	200	(950, 450)	500
(650, 450)	200	(1000, 500)	500

Table A.5: Table of signal samples with intermediate mass splitting (200 GeV $\leq \Delta m < 600$ GeV) for chargino production with slepton/sneutrino-mediated-decay.

$(\mathbf{m}(\tilde{\chi}_1^{\pm}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**	$(\mathbf{m}(\tilde{\chi}_1^{\pm}), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	Δ**m [GeV]**
(700, 1)	699	(1000, 300)	700
(700, 100)	600	(1000, 400)	600
(800, 1)	799	(1050, 50)	1000
(800, 100)	700	(1050, 150)	900
(800, 200)	600	(1050, 250)	800
(850, 50)	800	(1050, 350)	750
(850, 150)	700	(1050, 450)	600
(850, 250)	600	(1100, 1)	1099
(900, 1)	899	(1100, 100)	1000
(900, 100)	800	(1100, 200)	900
(900, 200)	700	(1100, 300)	800
(900, 300)	600	(1200, 1)	1199
(950, 50)	900	(1200, 100)	1100
(950, 150)	800	(1200, 200)	1000
(950, 250)	700	(1200, 300)	900
(950, 350)	600	(1300, 1)	1299
(1000, 1)	999	(1300, 100)	1200
(1000, 100)	900	(1300, 200)	1100
(1000, 200)	800	(1300, 300)	1000

Table A.6: Table of signal samples with high mass splitting ($\Delta m \geq 600$ GeV) for chargino production with slepton/sneutrino-mediated-decay.

A.3 Chargino via W-bosons

$(\mathbf{m}(\tilde{\chi}_1^\pm), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**	$(\mathbf{m}(\tilde{\chi}_1^\pm), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**
(100, 1)	99	(175, 75)	100
(125, 1)	124	(200, 75)	125
(125, 25)	100	(200, 100)	100
(150, 1)	149	(225, 100)	125
(150, 25)	125	(250, 150)	100
(150, 50)	100	(300, 200)	100
(175, 50)	125		

Table A.7: Table of signal samples with low mass splitting ($\Delta m < 150$ GeV) for chargino production with W-mediated-decay.

$(\mathbf{m}(\tilde{\chi}_1^\pm), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**	$(\mathbf{m}(\tilde{\chi}_1^\pm), \mathbf{m}(\tilde{\chi}_1^0))$ **[GeV]**	**Δm [GeV]**
(175, 1)	174	(275, 25)	250
(175, 25)	150	(275, 50)	225
(200, 1)	199	(275, 100)	175
(200, 25)	175	(300, 1)	299
(200, 50)	150	(300, 25)	275
(225, 1)	224	(300, 50)	250
(225, 25)	200	(300, 75)	225
(225, 50)	175	(300, 100)	200
(225, 75)	150	(300, 125)	175
(250, 1)	249	(300, 150)	150
(250, 25)	225	(325, 50)	275
(250, 50)	200	(325, 100)	225
(250, 75)	175	(350, 75)	275
(250, 100)	150	(350, 100)	250
(275, 1)	274	(400, 150)	250

Table A.8: Table of signal samples with intermediate mass splitting (150 GeV$\leq \Delta m < 300$ GeV) for chargino production with W-mediated-decay.

$\left(\mathbf{m}(\tilde{\chi}_1^{\pm}),\, \mathbf{m}(\tilde{\chi}_1^0)\right)$ **[GeV]**	Δ**m [GeV]**	$\left(\mathbf{m}(\tilde{\chi}_1^{\pm}),\, \mathbf{m}(\tilde{\chi}_1^0)\right)$ **[GeV]**	Δ**m [GeV]**
(325, 1)	324	(400, 100)	300
(325, 25)	300	(425, 1)	424
(350, 1)	349	(425, 25)	400
(350, 25)	325	(450, 1)	449
(350, 50)	300	(450, 50)	400
(375, 1)	374	(475, 1)	474
(375, 25)	350	(500, 1)	499
(375, 75)	300	(500, 100)	400
(400, 1)	399		

Table A.9: Table of signal samples with high mass splitting ($\Delta m \geq 300$ GeV) for chargino production with W-mediated-decay.

A.4 Mono-Z

$\left(\mathbf{m}(V),\, \mathbf{m}(\chi)\right)$ **[GeV]**	Δ**m [GeV]**	$\left(\mathbf{m}(V),\, \mathbf{m}(\chi)\right)$ **[GeV]**	Δ**m [GeV]**
(50, 1)	49	(50, 30)	20
(30, 5)	25	(100, 55)	45
(80, 25)	55	(150, 80)	70
(130, 50)	80	(200, 105)	95
(1, 5)	4	(100, 100)	0
(10, 10)	0		

Table A.10: Table of signal samples with low mass splitting ($\Delta m < 100$ GeV) for mono-Z process.

$\big(\mathbf{m}(V),\, \mathbf{m}(\chi)\big)$ **[GeV]**	**Δm [GeV]**	$\big(\mathbf{m}(V),\, \mathbf{m}(\chi)\big)$ **[GeV]**	**Δm [GeV]**
(150, 1)	149	(600, 305)	295
(200, 1)	199	(400, 250)	150
(380, 175)	205	(400, 75)	325
(480, 225)	255	(400, 150)	250
(580, 275)	305	(630, 300)	330
(400, 205)	195	(650, 330)	320
(500, 255)	245		

Table A.11: Table of signal samples with intermediate mass splitting (100 GeV$\leq \Delta m < 350$ GeV) for mono-Z process.

$\big(\mathbf{m}(V),\, \mathbf{m}(\chi)\big)$ **[GeV]**	**Δm [GeV]**	$\big(\mathbf{m}(V),\, \mathbf{m}(\chi)\big)$ **[GeV]**	**Δm [GeV]**
(400, 1)	399	(730, 350)	380
(500, 1)	499	(750, 380)	370
(600, 1)	599	(750, 75)	675
(750, 1)	749	(750, 250)	500
(900, 1)	899	(750, 325)	425
(1050, 1)	1049	(650, 1)	649

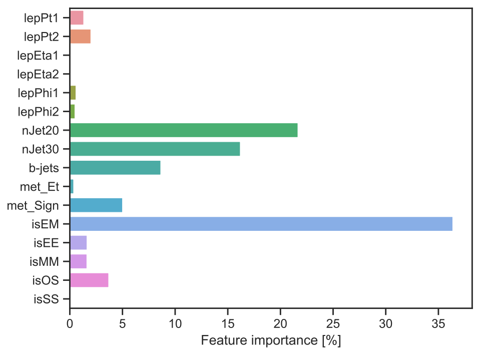
Table A.12: Table of signal samples with intermediate mass splitting ($\Delta m \geq 350$ GeV) for mono-Z process.

Appendix B

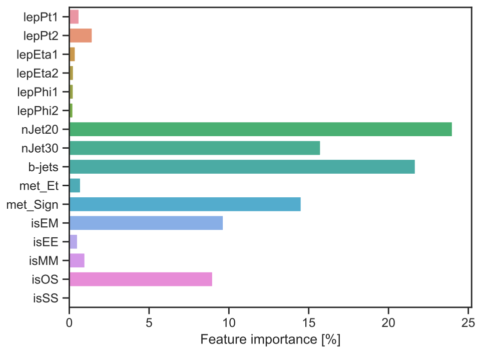
BDT plots

B.1 Low mass splittings

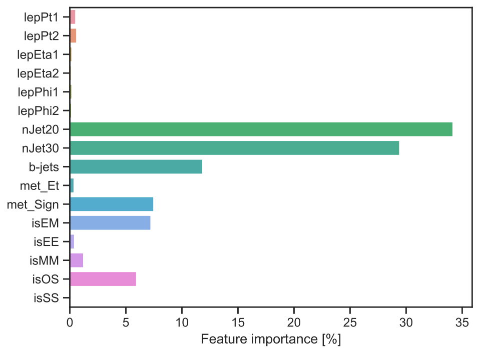
B.1.1 Low level features



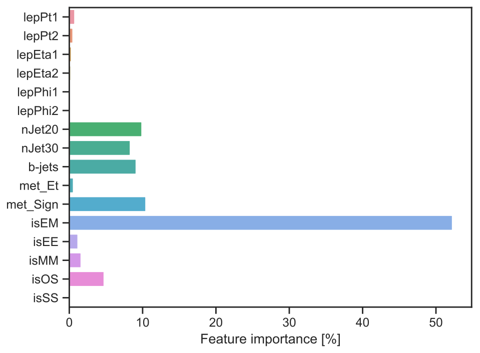
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

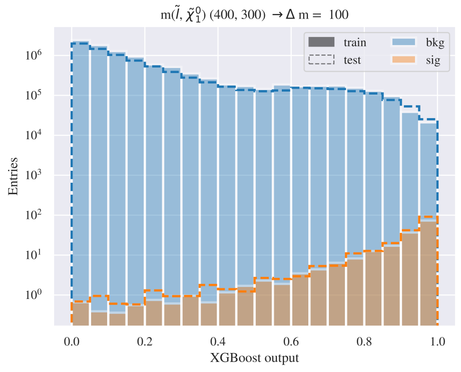


(c) Chargino production via $W^{\pm}$.

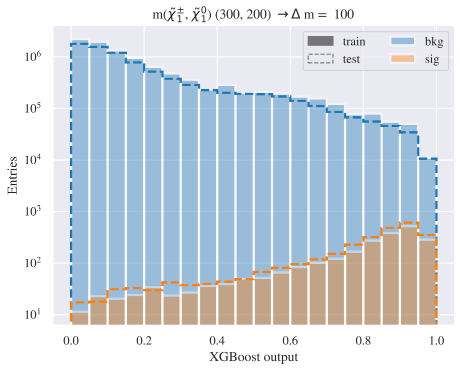


(d) Mono-Z.

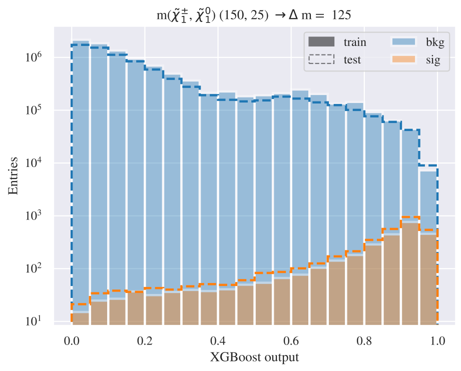
Figure B.1: Feature importance for low mass splittings for all four processes using low level features during training.



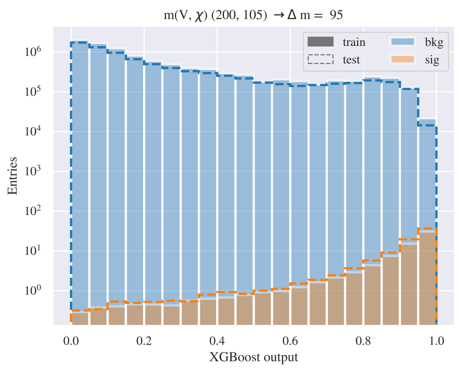
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



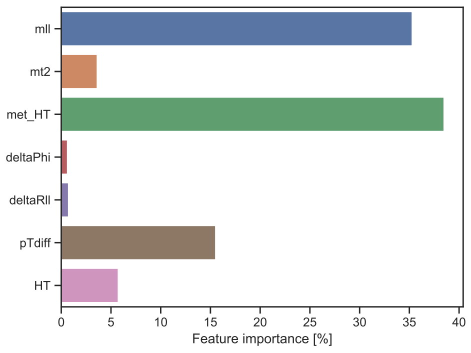
(c) Chargino production via $W^{\pm}$.



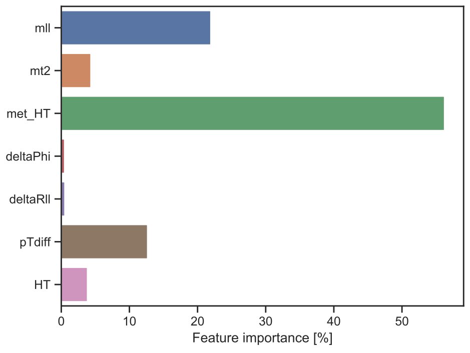
(d) Mono-Z.

Figure B.2: Test vs train for low mass splittings done with the BDT using low level features during training. Here the test set is scaled up to match the number of training events.

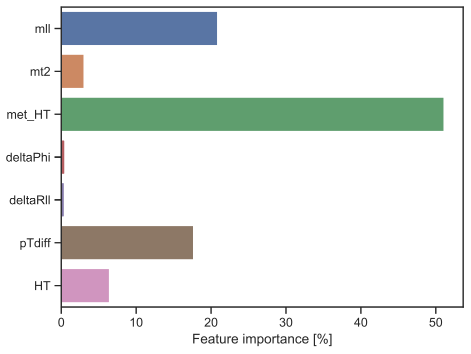
B.1.2 High level features



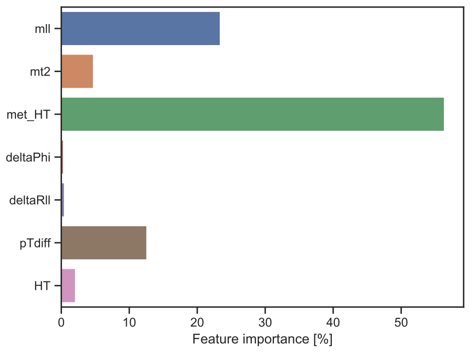
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

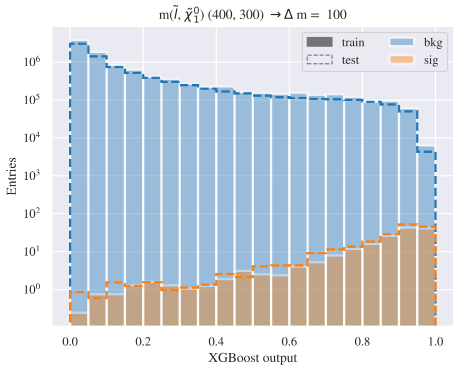


(c) Chargino production via $W^{\pm}$.

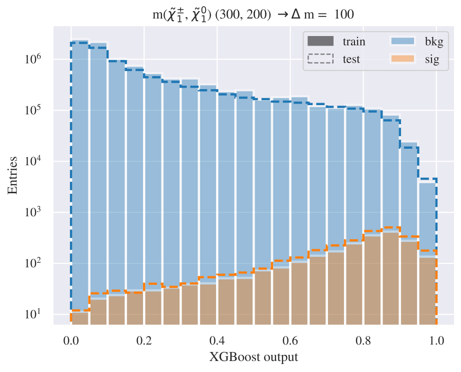


(d) Mono-Z.

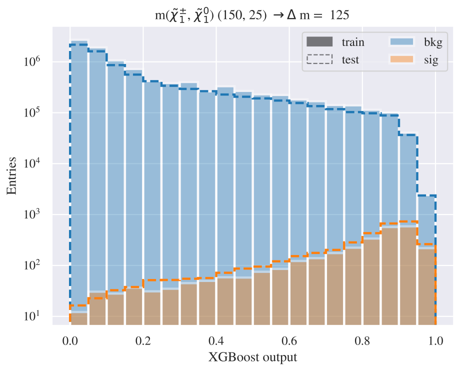
Figure B.3: Feature importance for low mass splittings for all four processes using high level features during training.



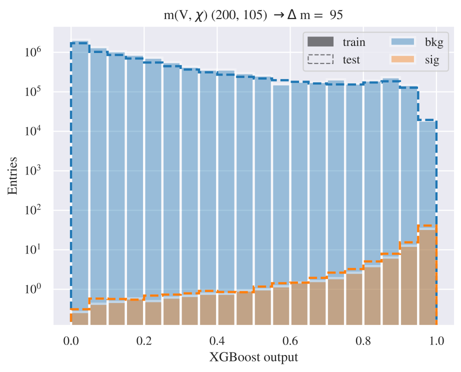
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.

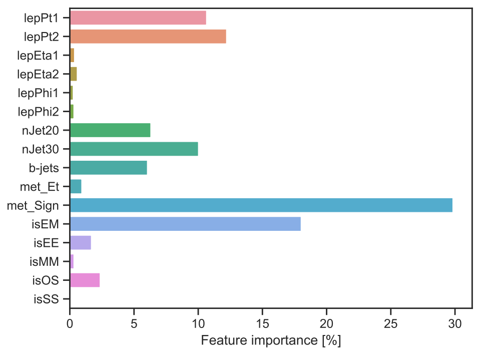


(d) Mono-Z.

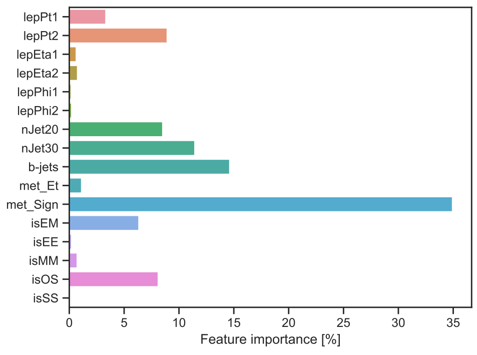
Figure B.4: Test vs train for low mass splittings done with the BDT using high level features during training. Here the test set is scaled up to match the number of training events.

B.2 Intermediate mass splittings

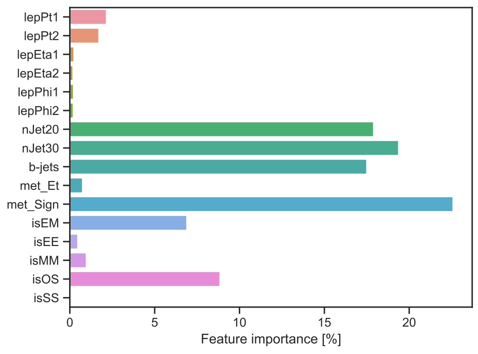
B.2.1 Low level features



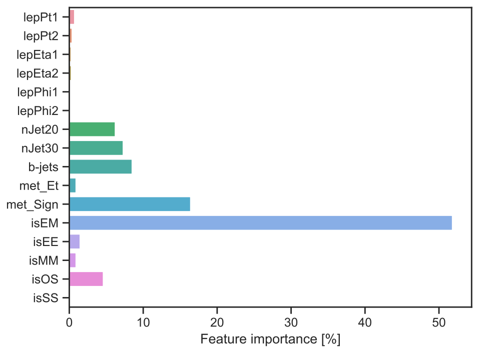
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

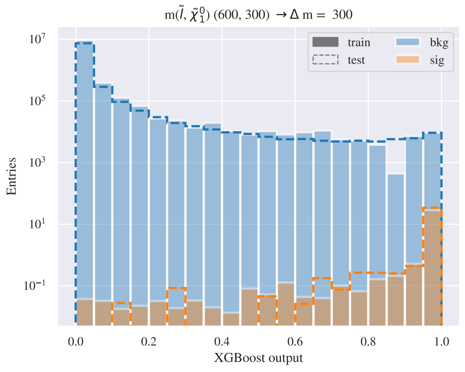


(c) Chargino production via $W^{\pm}$.

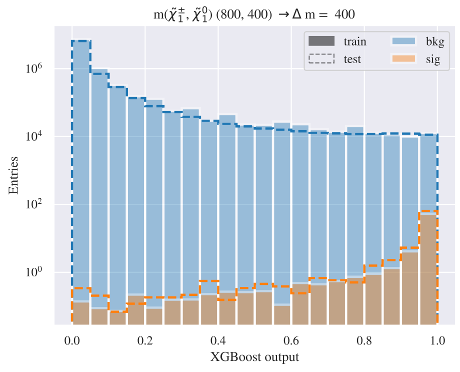


(d) Mono-Z.

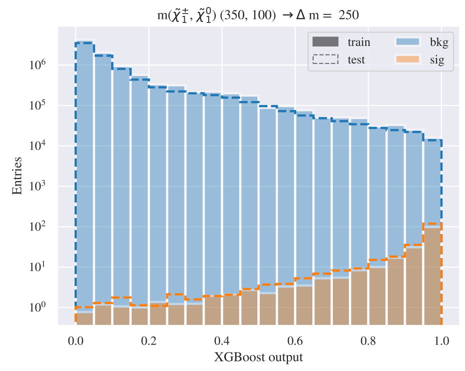
Figure B.5: Feature importance for low mass splittings for all four processes using low level features during training.



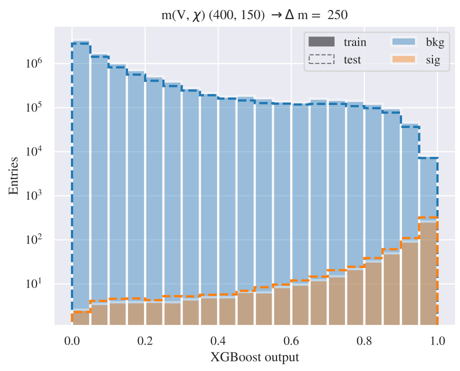
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



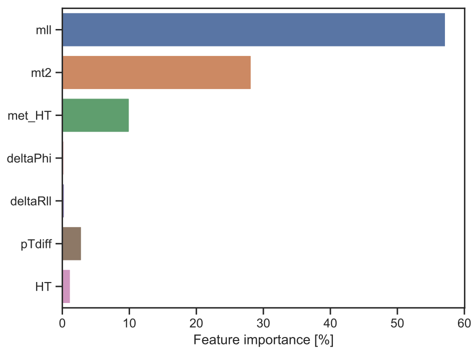
(c) Chargino production via $W^\pm$.



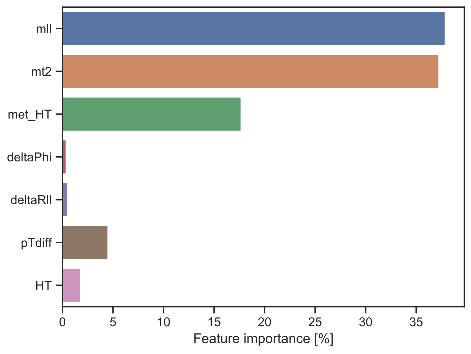
(d) Mono-Z.

Figure B.6: Test vs train for low mass splittings done with the BDT using low level features during training. Here the test set is scaled up to match the number of training events.

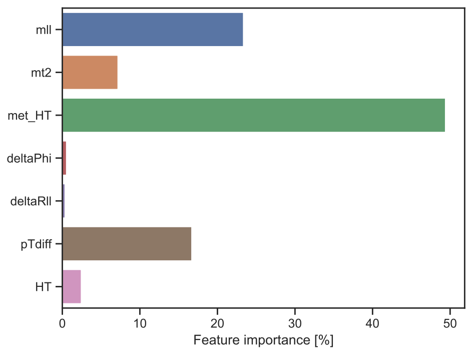
B.2.2 High level features



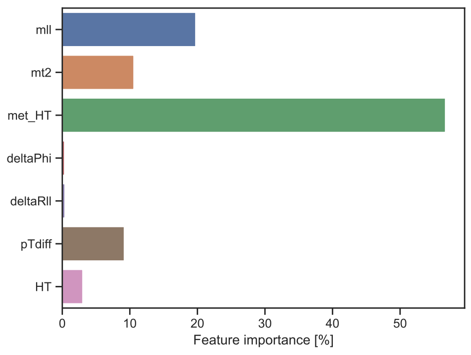
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

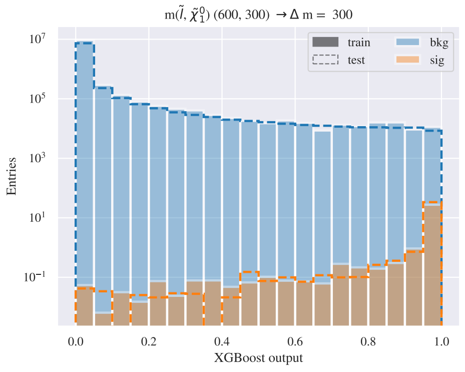


(c) Chargino production via $W^{\pm}$.

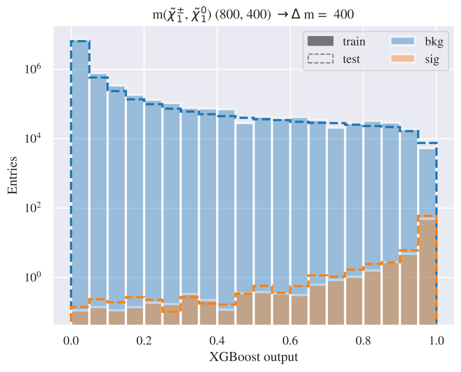


(d) Mono-Z.

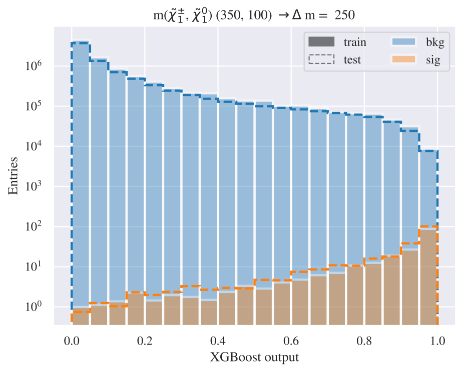
Figure B.7: Feature importance for low mass splittings for all four processes using high level features during training.



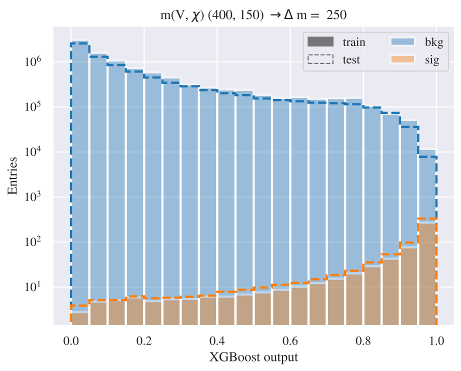
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.

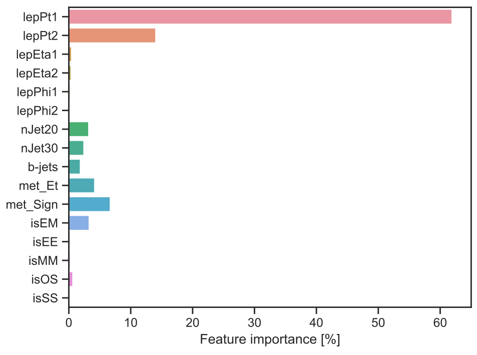


(d) Mono-Z.

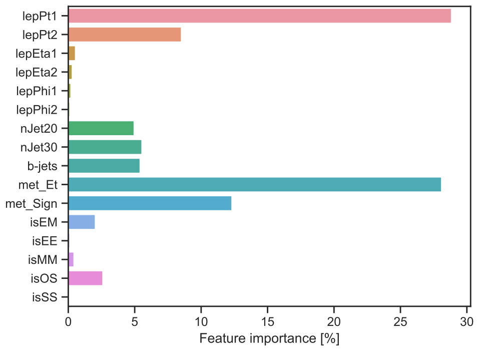
Figure B.8: Test vs train for low mass splittings done with the BDT using high level features during training. Here the test set is scaled up to match the number of training events.

B.3 High mass splittings

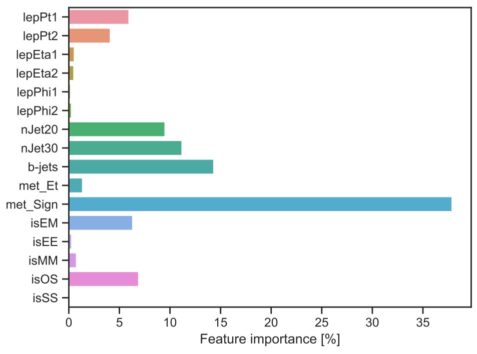
B.3.1 Low level features



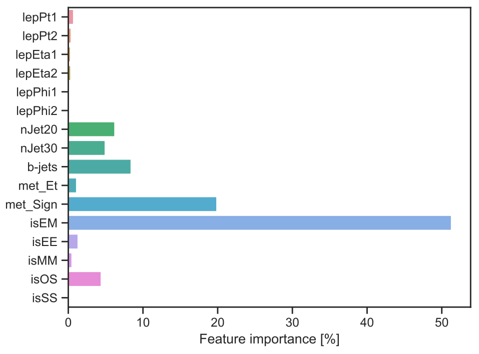
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

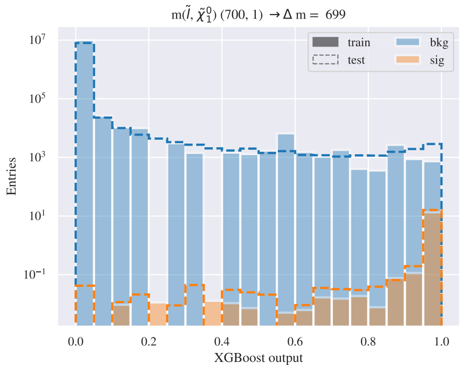


(c) Chargino production via $W^{\pm}$.

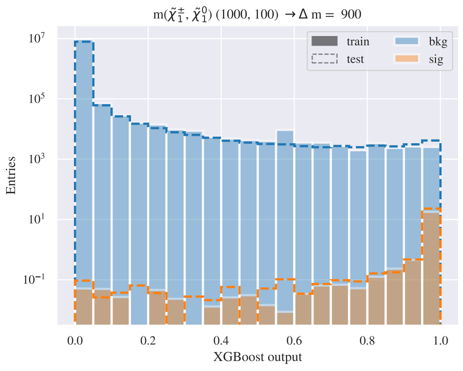


(d) Mono-Z.

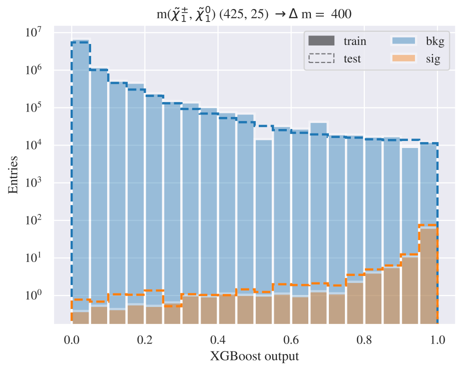
Figure B.9: Feature importance for low mass splittings for all four processes using low level features during training.



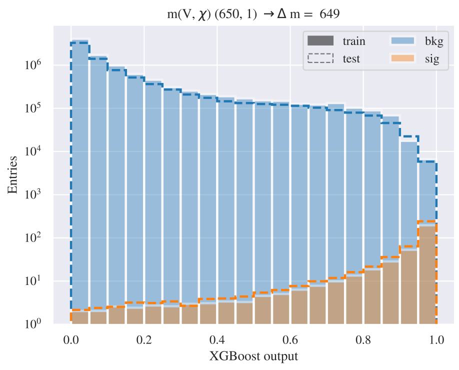
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



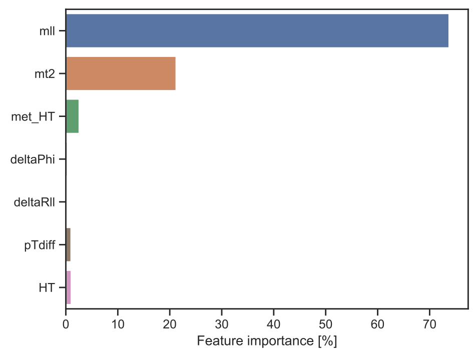
(c) Chargino production via $W^{\pm}$.



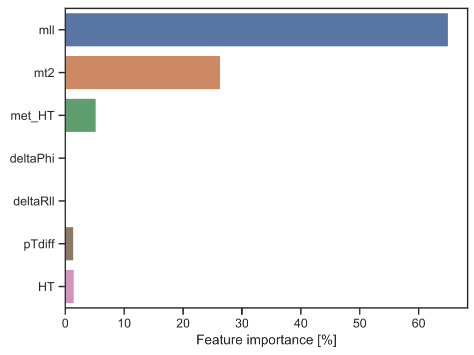
(d) Mono-Z.

Figure B.10: Test vs train for low mass splittings done with the BDT using low level features during training. Here the test set is scaled up to match the number of training events.

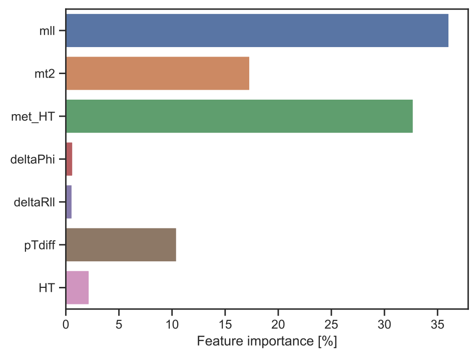
B.3.2 High level features



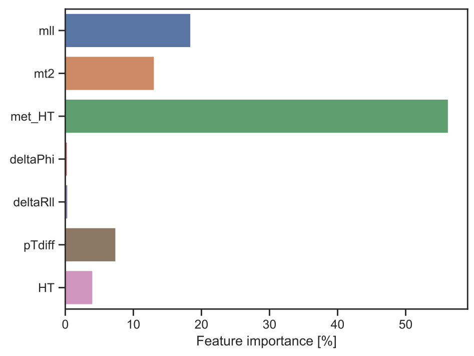
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.

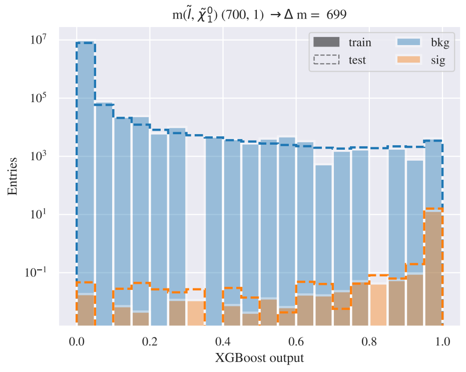


(c) Chargino production via $W^{\pm}$.

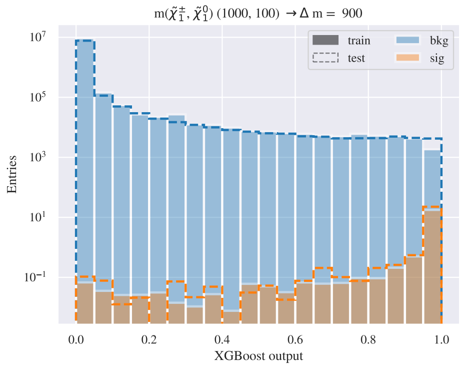


(d) Mono-Z.

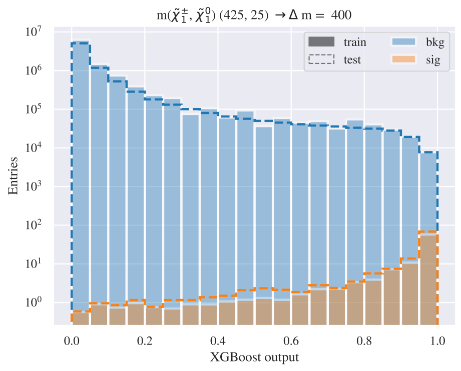
Figure B.11: Feature importance for low mass splittings for all four processes using high level features during training.



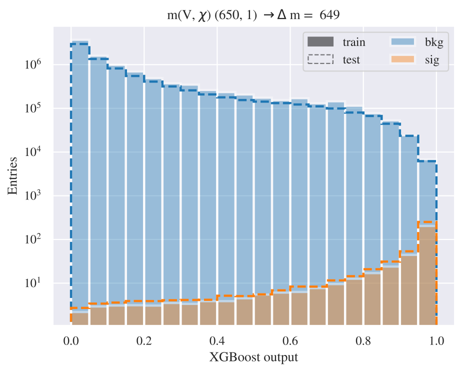
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^\pm$.

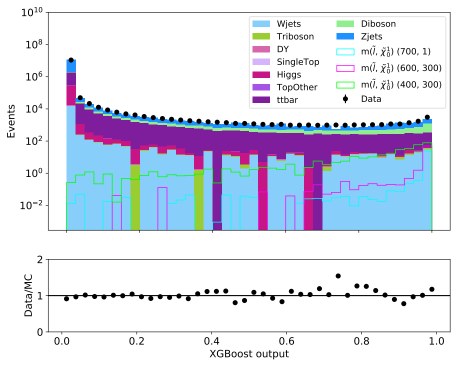


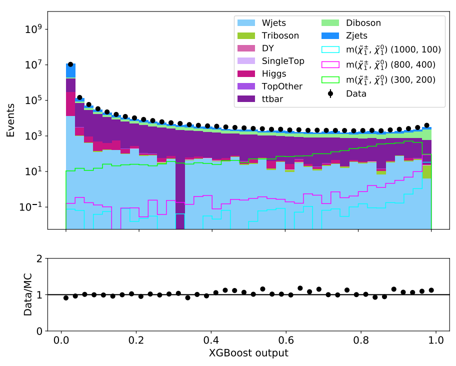
(d) Mono-Z.

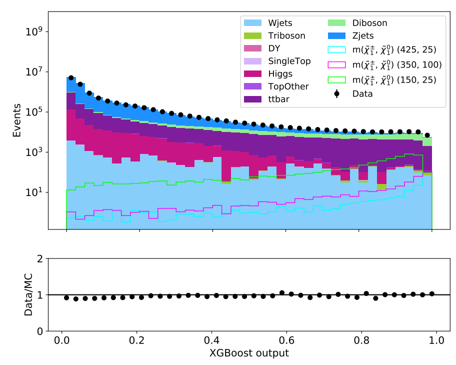
Figure B.12: Test vs train for low mass splittings done with the BDT using high level features during training. Here the test set is scaled up to match the number of training events.

B.4 Stacked background with data

B.4.1 Low level features


(a) Direct slepton production.


(b) Chargino production via $\tilde{l}/\tilde{\nu}$.


(c) Chargino production via $W^{\pm}$.

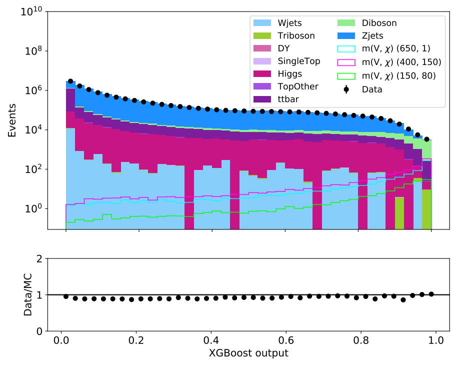
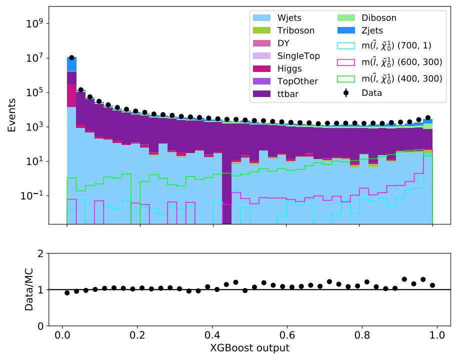
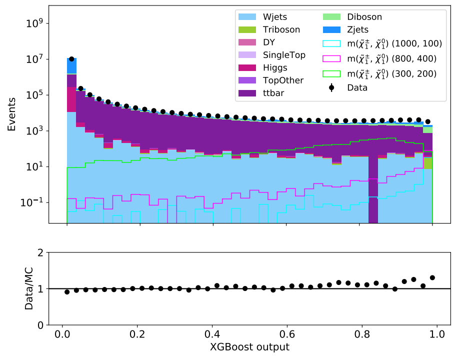

(d) Mono-Z.

Figure B.13: Test vs train for low mass splittings done with the BDT using low level features during training. Here the test set is scaled up to match the number of training events.

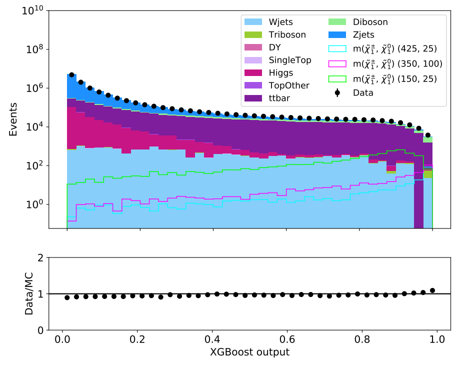
B.4.2 High level features



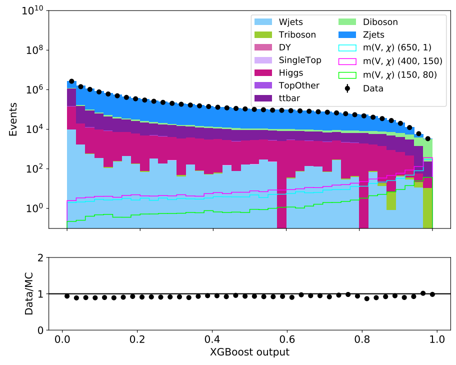
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



(d) Mono-Z.

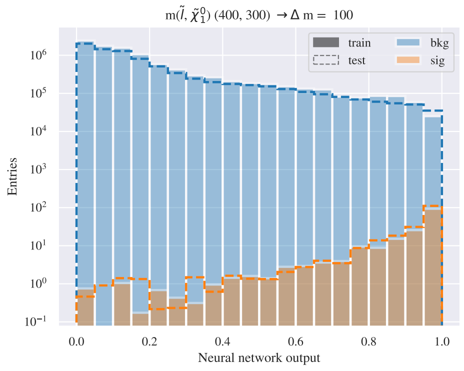
Figure B.14: Test vs train for low mass splittings done with the BDT using high level features during training. Here the test set is scaled up to match the number of training events.

Appendix C

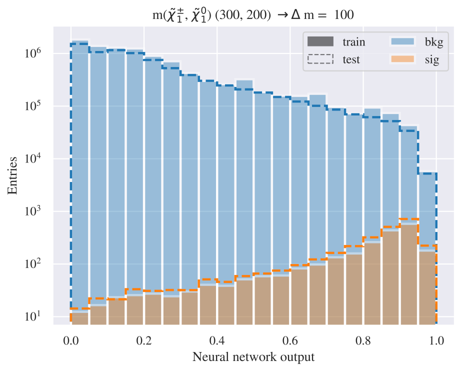
NN plots

C.1 Low mass splittings

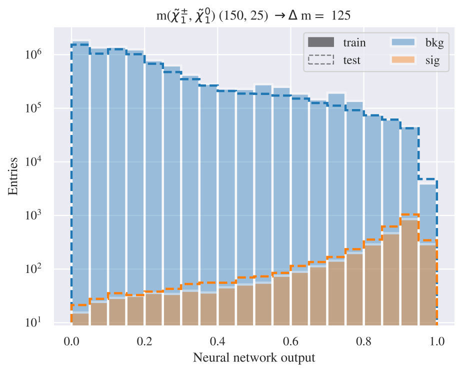
C.1.1 Low level features



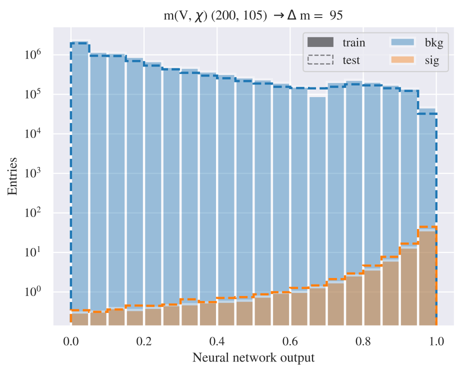
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



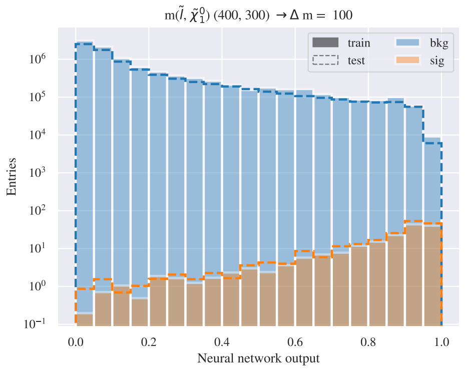
(c) Chargino production via $W^\pm$.



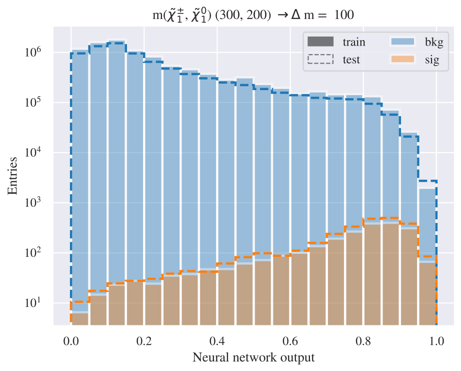
(d) Mono-Z.

Figure C.1: Test vs train for low mass splittings done with the NN using low level features during training. Here the test set is scaled up to match the number of training events.

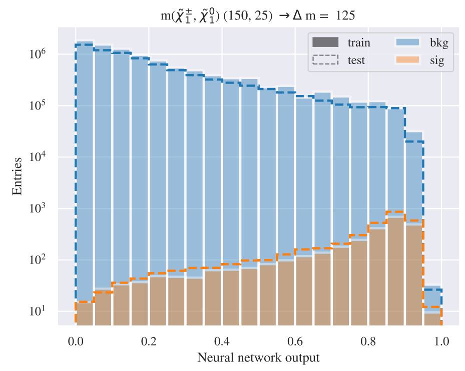
C.1.2 High level features



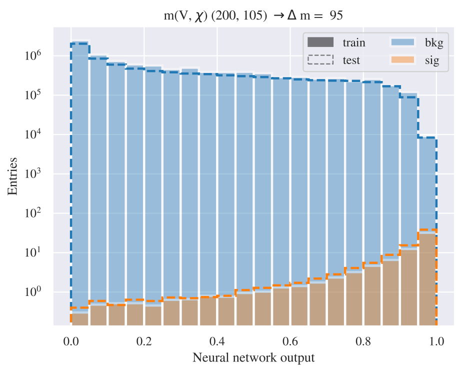
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^\pm$.

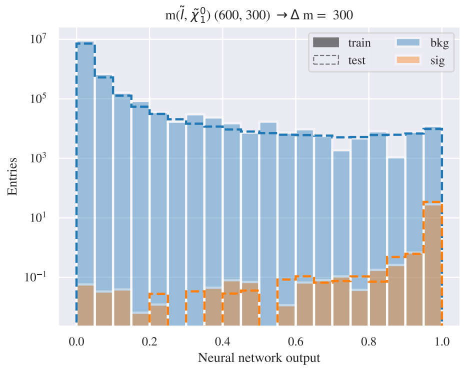


(d) Mono-Z.

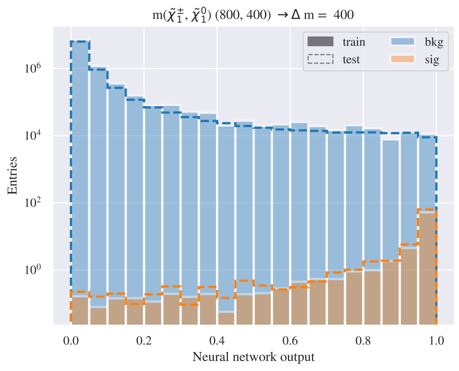
Figure C.2: Test vs train for low mass splittings done with the NN using high level features during training. Here the test set is scaled up to match the number of training events.

C.2 Intermediate mass splittings

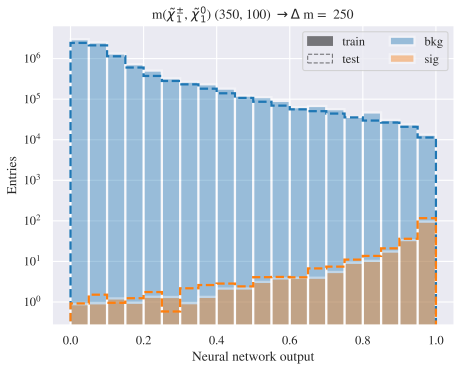
C.2.1 Low level features



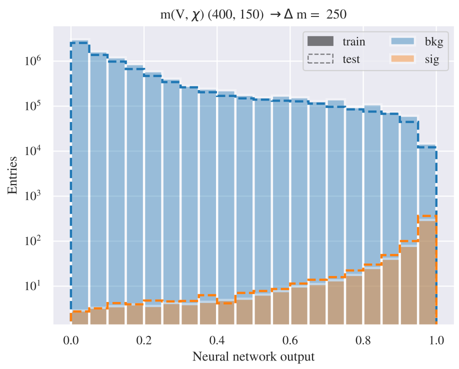
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



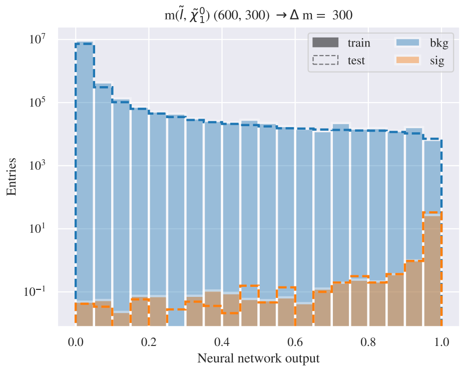
(c) Chargino production via $W^{\pm}$.



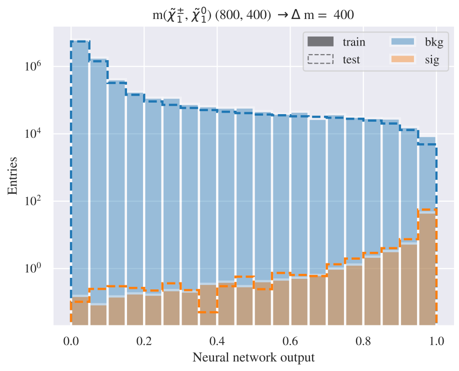
(d) Mono-Z.

Figure C.3: Test vs train for intermediate mass splittings done with the NN using low level features during training. Here the test set is scaled up to match the number of training events.

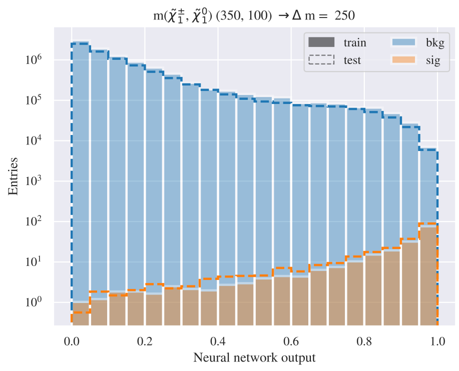
C.2.2 High level features



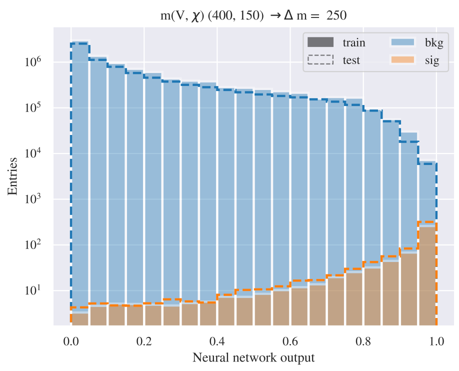
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.

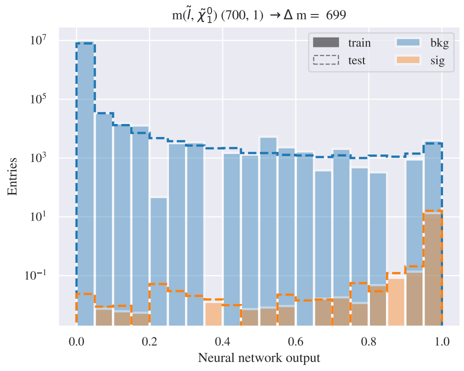


(d) Mono-Z.

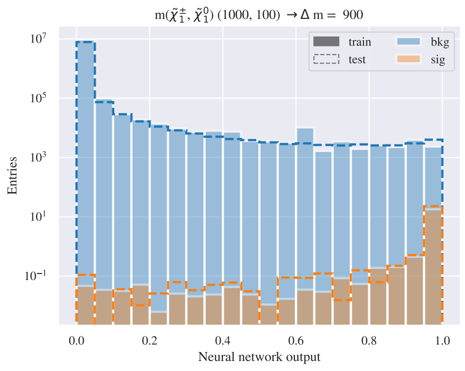
Figure C.4: Test vs train for intermediate mass splittings done with the NN using high level features during training. Here the test set is scaled up to match the number of training events.

C.3 High mass splittings

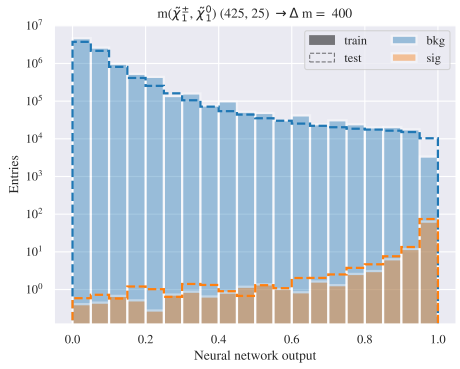
C.3.1 Low level features



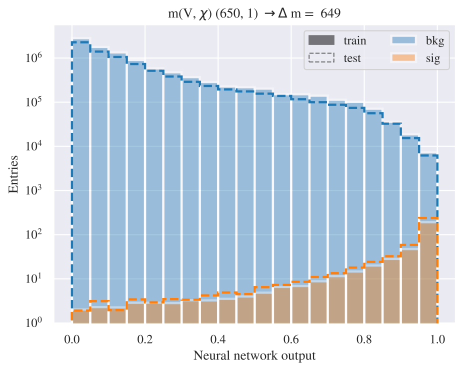
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



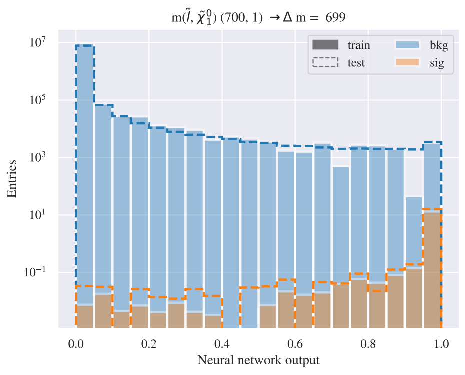
(c) Chargino production via $W^{\pm}$.



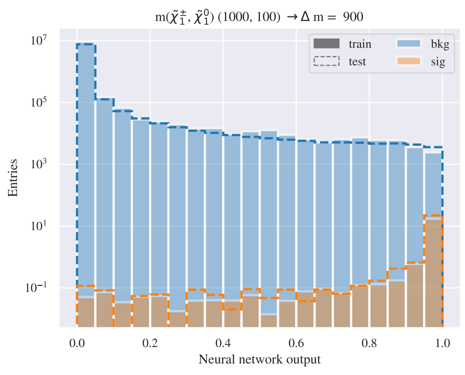
(d) Mono-Z.

Figure C.5: Test vs train for high mass splittings done with the NN using low level features during training. Here the test set is scaled up to match the number of training events.

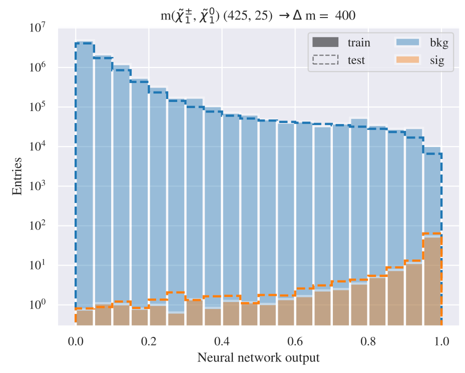
C.3.2 High level features



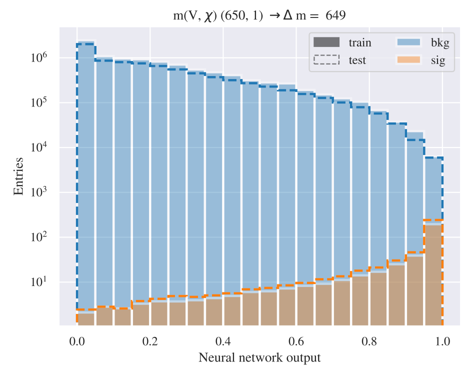
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.

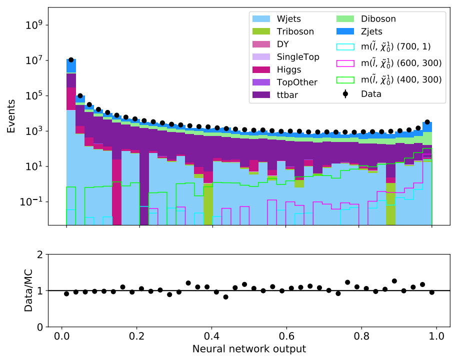


(d) Mono-Z.

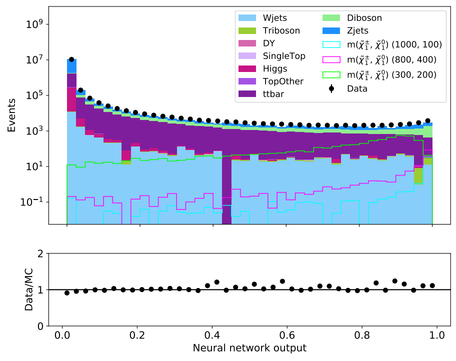
Figure C.6: Test vs train for high mass splittings done with the NN using high level features during training. Here the test set is scaled up to match the number of training events.

C.4 Stacked background with data

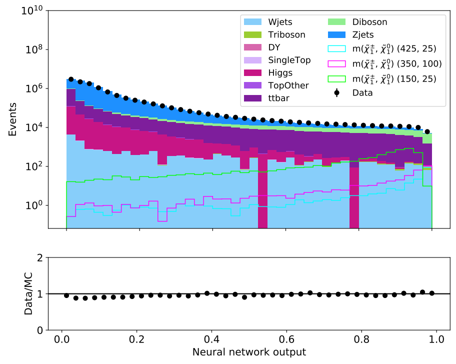
C.4.1 Low level features



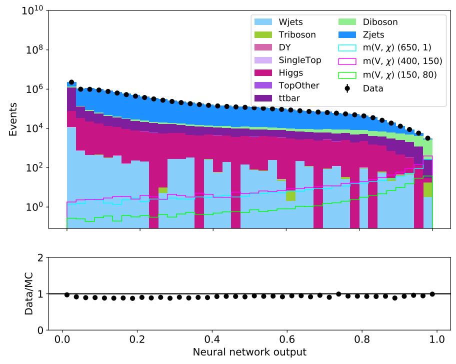
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



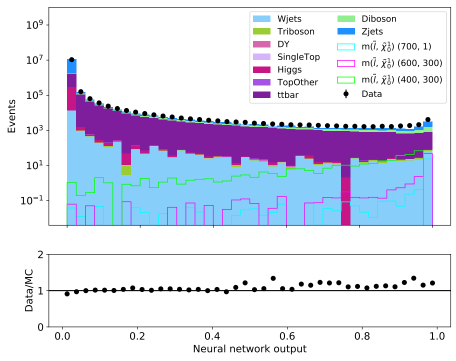
(c) Chargino production via $W^{\pm}$.



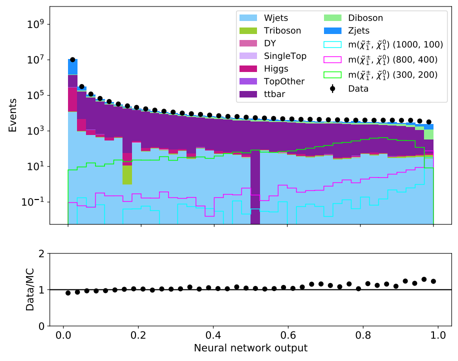
(d) Mono-Z.

Figure C.7: Test vs train for low mass splittings done with the NN using low level features during training. Here the test set is scaled up to match the number of training events.

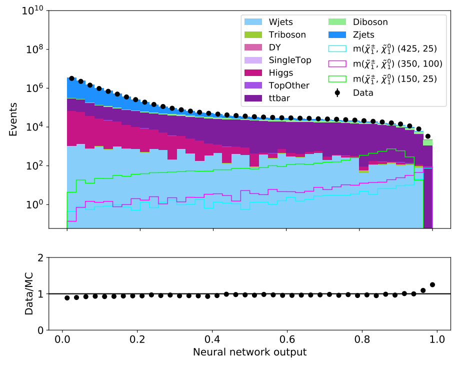
C.4.2 High level features



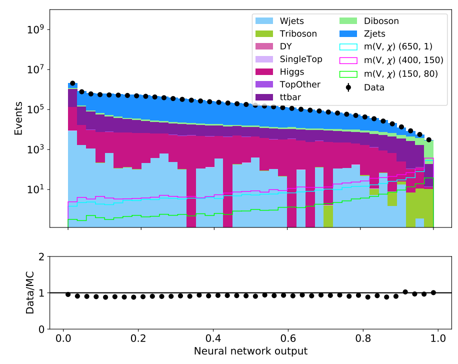
(a) Direct slepton production.



(b) Chargino production via $\tilde{l}/\tilde{\nu}$.



(c) Chargino production via $W^{\pm}$.



(d) Mono-Z.

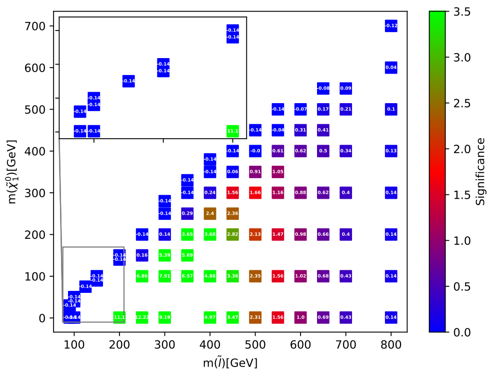
Figure C.8: Test vs train for low mass splittings done with the NN using high level features during training. Here the test set is scaled up to match the number of training events.

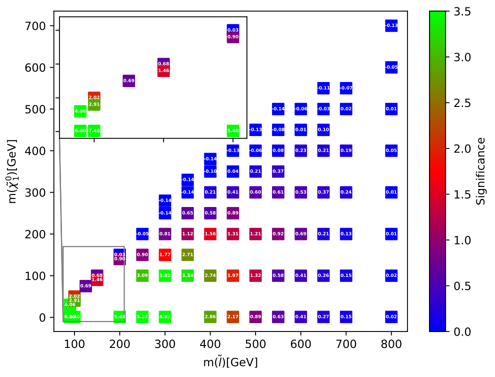
Appendix D

Significance plots

D.1 Direct slepton production

D.1.1 Low level features


(a) Cut and count.


(b) Boosted Decision Tree.

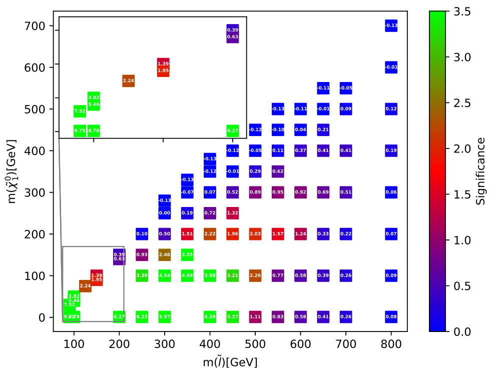
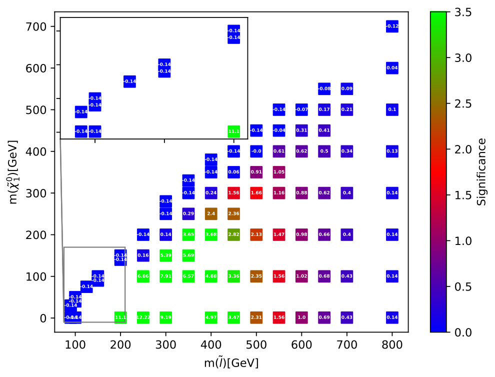
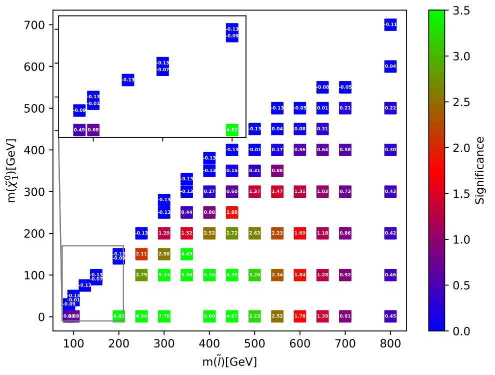

(c) Neural Network.

Figure D.1: Significance plots for direct slepton production where low level features are used during training the ML models.

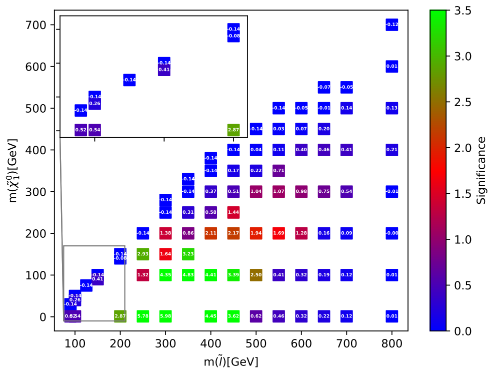
D.1.2 High level features



(a) Cut and count.



(b) Boosted Decision Tree.

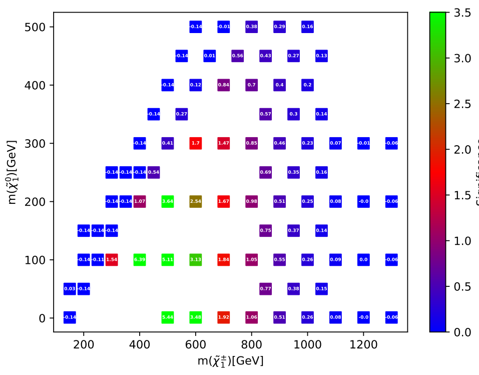


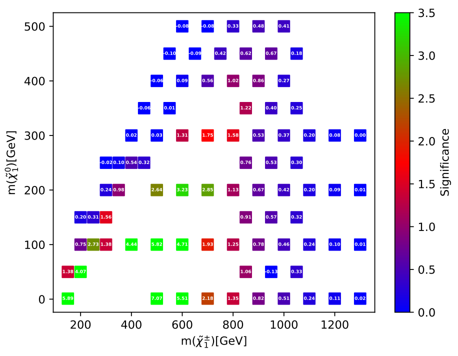
(c) Neural Network.

Figure D.2: Significance plots for direct slepton production where high level features are used during training the ML models.

D.2 Chargino pair via slepton or sneutrino

D.2.1 Low level features


(a) Cut and count.


(b) Boosted Decision Tree.

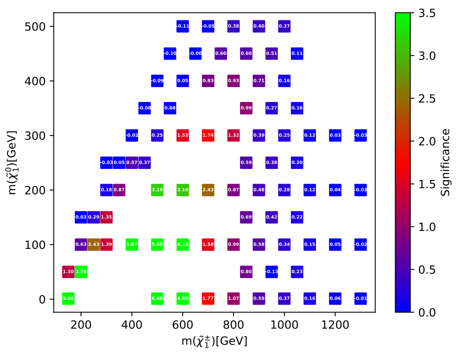
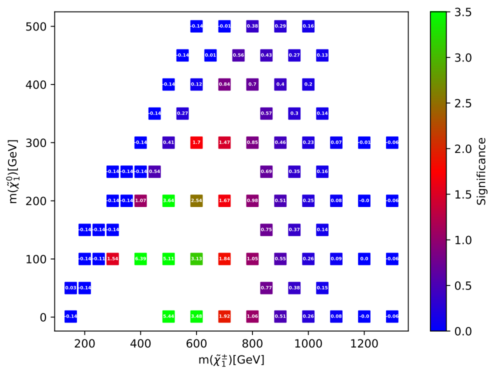
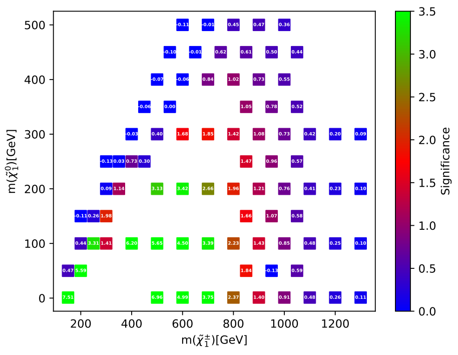

(c) Neural Network.

Figure D.3: Significance plots for chargino production with slepton/sneutrino-mediated-decay where low level features are used during training the ML models.

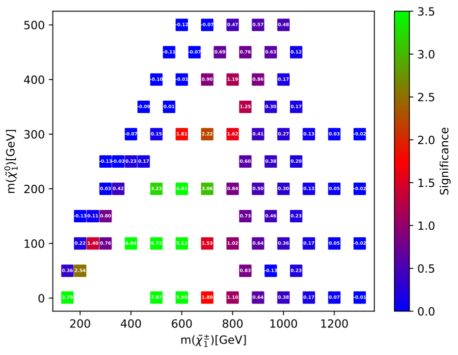
D.2.2 High level features



(a) Cut and count.



(b) Boosted Decision Tree.

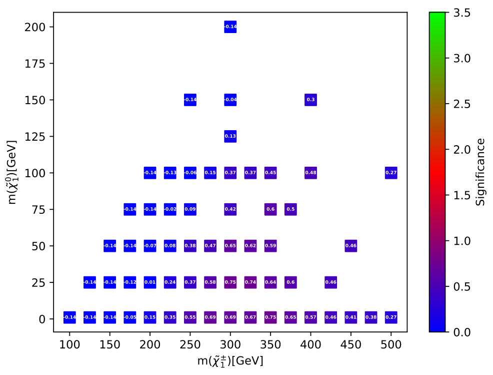


(c) Neural Network.

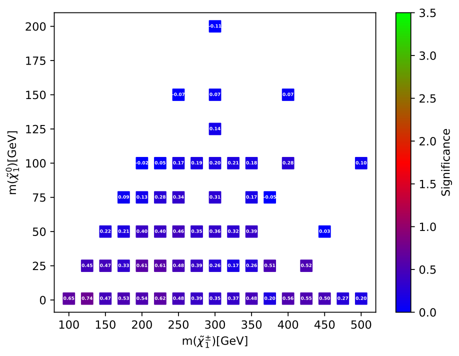
Figure D.4: Significance plots for chargino production with slepton/sneutrino-mediated-decay where high level features are used during training the ML models.

D.3 Chargino pair via W-bosons

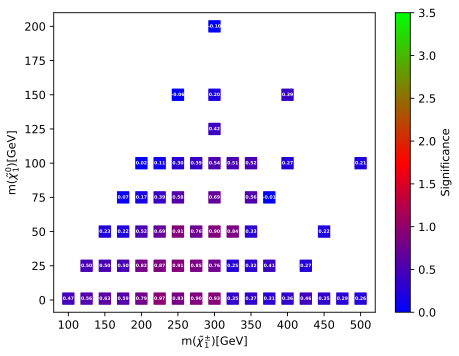
D.3.1 Low level features



(a) Cut and count.



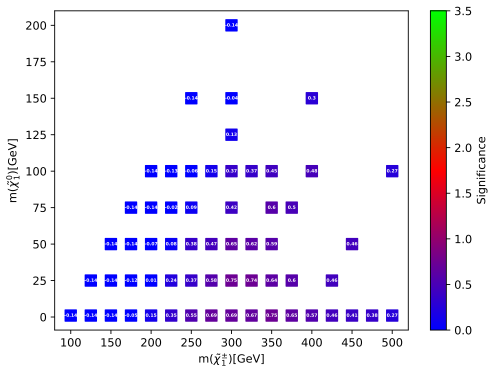
(b) Boosted Decision Tree.



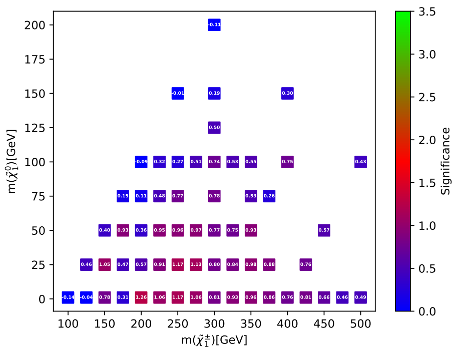
(c) Neural Network.

Figure D.5: Significance plots for chargino production with W-boson-mediated-decay where low level features are used during training the ML models.

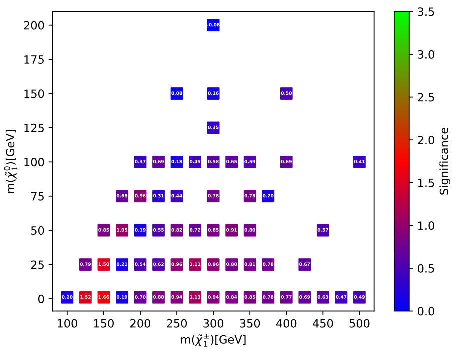
D.3.2 High level features



(a) Cut and count.



(b) Boosted Decision Tree.

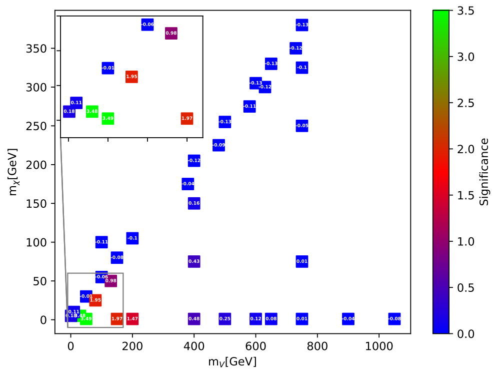


(c) Neural Network.

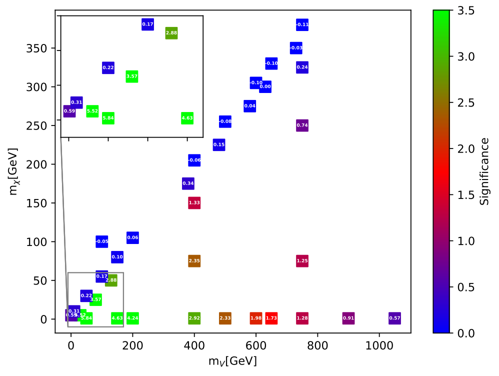
Figure D.6: Significance plots for chargino production with W-boson-mediated-decay where high level features are used during training the ML models.

D.4 Mono-Z

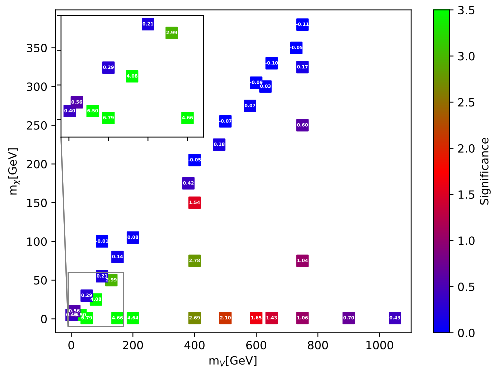
D.4.1 Low level features



(a) Cut and count.



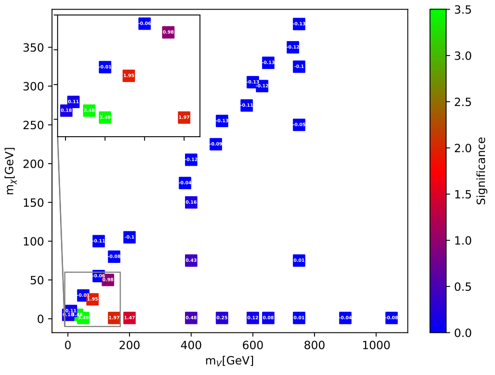
(b) Boosted Decision Tree.

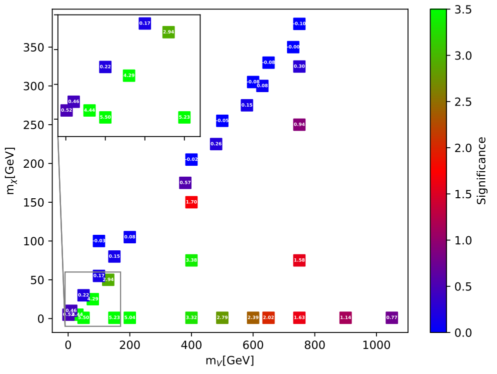


(c) Neural Network.

Figure D.7: Significance plots for the mono-Z process where low level features are used during training the ML models.

D.4.2 High level features


(a) Cut and count.


(b) Boosted Decision Tree.

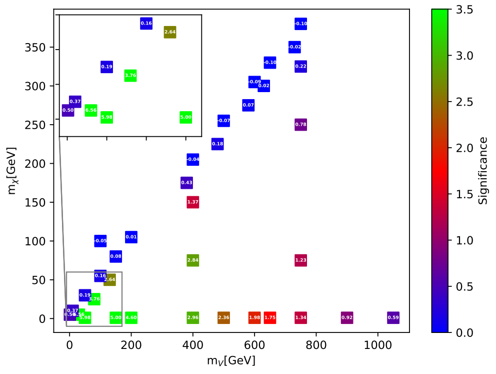

(c) Neural Network.

Figure D.8: Significance plots for the mono-Z process where high level features are used during training the ML models.

Bibliography

[1] Mark Thomson. *Modern particle physics.* Cambridge University Press, New York, 2013.

[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. *Phys. Lett.*, B716:1–29, 2012.

[3] Serguei Chatrchyan et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys. Lett.*, B716:30–61, 2012.

[4] CERN. The Standard Model and beyond. `http://united-states.cern/physics/standard-model-and-beyond`.

[5] Michael Mattern. The Strong Interaction. `https://www.slideserve.com/wells/the-strong-interaction`, October, 2014.

[6] The Avocado. A Primer on Particle Physics, Part 3: Feynman Diagrams. `https://the-avocado.org/2020/07/21/a-primer-on-particle-physics-part-3-feynman-diagrams/`, June, 2020.

[7] Xianhao Xin. Glashow-Weinberg-Salam Model: An Example of Electroweak Symmetry Breaking. 2007.

[8] Konstantin Kouzakov and Alexander Studenikin. Theory of Neutrino-Atom Collisions: The History, Present Status, and BSM Physics. *Advances in High Energy Physics*, 2014, 06 2014.

[9] The ATLAS collaboration. Search for electroweak production of charginos and sleptons decaying into final states with two leptons and missing transverse momentum in $\sqrt{s} = 13$

TeV pp collisions using the ATLAS detector. `https://arxiv.org/pdf/1908.08215.pdf`, February, 2020.

[10] ATLAS Silicon IFIC. SUSY Physics. `https://ific.uv.es/sct/physics_susy`.

[11] The ATLAS collaboration. Search for an invisibly decaying Higgs boson or dark matter candidates produced in association with a Z-boson in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. `https://arxiv.org/pdf/1708.09624.pdf`, December, 2017.

[12] F.J. Hasert, S. Kabe, W. Krenz, J. Von Krogh, D. Lanske, J. Morfin, K. Schultze, H. Weerts, G.H. Bertrand-Coremans, J. Sacton, W. Van Doninck, P. Vilain, U. Camerini, D.C. Cundy, R. Baldi, I. Danilchenko, W.F. Fry, D. Haidt, S. Natali, P. Musset, B. Osculati, R. Palmer, J.B.M. Pattison, D.H. Perkins, A. Pullia, A. Rousset, W. Venus, H. Wachsmuth, V. Brisson, B. Degrange, M. Haguenauer, L. Kluberg, U. Nguyen-Khac, P. Petiau, E. Belotti, S. Bonetti, D. Cavalli, C. Conta, E. Fiorini, M. Rollier, B. Aubert, D. Blum, L.M. Chounet, P. Heusse, A. Lagarrigue, A.M. Lutz, A. Orkin-Lecourtois, J.P. Vialle, F.W. Bullock, M.J. Esten, T.W. Jones, J. McKenzie, A.G. Michette, G. Myatt, and W.G. Scott. Observation of neutrino-like interactions without muon or electron in the gargamelle neutrino experiment. *Physics Letters B*, 46(1):138 – 140, 1973.

[13] G. Arnison, A. Astbury, B. Aubert, C. Bacci, G. Bauer, A. Bézaguet, R. Böck, T.J.V. Bowcock, M. Calvetti, T. Carroll, P. Catz, P. Cennini, S. Centro, F. Ceradini, S. Cittolin, D. Cline, C. Cochet, J. Colas, M. Corden, D. Dallman, M. DeBeer, M. Della Negra, M. Demoulin, D. Denegri, A. Di Ciaccio, D. DiBitonto, L. Dobrzynski, J.D. Dowell, M. Edwards, K. Eggert, E. Eisenhandler, N. Ellis, P. Erhard, H. Faissner, G. Fontaine, R. Frey, R. Frühwirth, J. Garvey, S. Geer, C. Ghesquière, P. Ghez, K.L. Giboni, W.R. Gibson, Y. Giraud-Héraud, A. Givernaud, A. Gonidec, G. Grayer, P. Gutierrez, T. Hansl-Kozanecka, W.J. Haynes, L.O. Hertzberger, C. Hodges, D. Hoffmann, H. Hoffmann, D.J. Holthuizen, R.J. Homer, A. Honma, W. Jank, G. Jorat, P.I.P. Kalmus, V. Karimäki, R. Keeler, I. Kenyon, A. Kernan, R. Kinnunen, H. Kowalski, W. Kozanecki, D. Kryn, F. Lacava, J.-P. Laugier, J.-P. Lees, H. Lehmann, K. Leuchs, A. Lévêque, E. Linglin, E. Locci, M. Loret, J.-J. Malosse, T. Markiewicz, G. Maurin, T. McMahon, J.-P. Mendiburu, M.-N. Minard, M. Moricca, H. Muirhead, F. Muller, A.K. Nandi, L. Naumann, A. Norton, A. Orkin-Lecourtois, L. Paoluzi, G. Petrucci,

G.Piano Mortari, M. Pimiä, A. Placci, E. Radermacher, J. Ransdell, H. Reithler, J.-P. Revol, J. Rich, M. Rijssenbeek, C. Roberts, J. Rohlf, P. Rossi, C. Rubbia, B. Sadoulet, G. Sajot, G. Salvi, J. Salvini, J. Sass, A. Saudraix, A. Savoy-Navarro, D. Schinzel, W. Scott, T.P. Shah, M. Spiro, J. Strauss, K. Sumorok, F. Szoncso, D. Smith, C. Tao, G. Thompson, J. Timmer, E. Tscheslog, J. Tuominiemi, S. Van der Meer, J.-P. Vialle, J. Vrana, V. Vuillemin, H.D. Wahl, P. Watkins, J. Wilson, Y.G. Xie, M. Yvert, and E. Zurfluh. Experimental observation of isolated large transverse energy electrons with associated missing energy at $\sqrt{s} = 540$ GeV. *Physics Letters B*, 122(1):103 – 116, 1983.

[14] G. Arnison, A. Astbury, B. Aubert, C. Bacci, G. Bauer, A. Bézaguet, R. Böck, T.J.V. Bowcock, M. Calvetti, P. Catz, P. Cennini, S. Centro, F. Ceradini, S. Cittolin, D. Cline, C. Cochet, J. Colas, M. Corden, D. Dallman, D. Dau, M. DeBeer, M.Della Negra, M. Demoulin, D. Denegri, A. Di Ciaccio, D. Dibitonto, L. Dobrzynski, J.D. Dowell, K. Eggert, E. Eisenhandler, N. Ellis, P. Erhard, H. Faissner, M. Fincke, G. Fontaine, R. Frey, R. Frühwirth, J. Garvey, S. Geer, C. Ghesquière, P. Ghez, K. Giboni, W.R. Gibson, Y. Giraud-Héraud, A. Givernaud, A. Gonidec, G. Grayer, T. Hansl-Kozaecka, W.J. Haynes, L.O. Hertzberger, C. Hodges, D. Hoffmann, H. Hoffmann, D.J. Holthuizen, R.J. Homer, A. Honma, W. Jank, G. Jorat, P.I.P. Kalmus, V. Karimäki, R. Keeler, I. Kenyon, A. Kernan, R. Kinnunen, W. Kozanecki, D. Kryn, F. Lacava, J.-P. Laugier, J.-P. Lees, H. Lehmann, R. Leuchs, A. Lévêque, D. Linglin, E. Locci, J.-J. Malosse, T. Markiewicz, G. Maurin, T. McMahon, J.-P. Mendiburu, M.-N. Minard, M. Mohammadi, M. Moricca, K. Morgan, H. Muirhead, F. Muller, A.K. Nandi, L. Naumann, A. Norton, A. Orkin-Lecourtois, L. Paoluzi, F. Pauss, G.Piano Mortari, E. Pietarinen, M. Pimiä, A. Placci, J.P. Porte, E. Radermacher, J. Ransdell, H. Reithler, J.-P. Revol, J. Rich, M. Rijssenbeek, C. Roberts, J. Rohlf, P. Rossi, C. Rubbia, B. Sadoulet, G. Sajot, G. Salvi, G. Salvini, J. Sass, J. Saudraix, A. Savoy-Navarro, D. Schinzel, W. Scott, T.P. Shah, M. Spiro, J. Strauss, J. Streets, K. Sumorok, F. Szoncso, D. Smith, C. Tao, G. Thompson, J. Timmer, E. Tscheslog, J. Touminiemi, B. Van Eijk, J.-P. Vialle, J. Vrana, V. Vuillemin, H.D. Wahl, P. Watkins, J. Wilson, C. Wulz, G.Y. Xie, M. Yvert, and E. Zurfluh. Experimental observation of lepton pairs of invariant mass around 95 GeV/c2 at the CERN SPS collider. *Physics Letters B*, 126(5):398 – 410, 1983.

[15] M. Banner, R. Battiston, Ph. Bloch, F. Bonaudi, K. Borer, M. Borghini, J.-C. Chol-

let, A.G. Clark, C. Conta, P. Darriulat, L. Di Lella, J. Dines-Hansen, P.-A. Dorsaz, L. Fayard, M. Fraternali, D. Froidevaux, J.-M. Gaillard, O. Gildemeister, V.G. Goggi, H. Grote, B. Hahn, H. Hänni, J.R. Hansen, P. Hansen, T. Himel, V. Hungerbühler, P. Jenni, O. Kofoed-Hansen, E. Lançon, M. Livan, S. Loucatos, B. Madsen, P. Mani, B. Mansoulie, G.C. Mantovani, L. Mapelli, B. Merkel, M. Mermikides, R. Møllerud, B. Nilsson, C. Onions, G. Parrour, F. Pastore, H. Plothow-Besch, M. Polverel, J.-P. Repellin, A. Rothenberg, A. Roussarie, G. Sauvage, J. Schacher, J.L. Siegrist, H.M. Steiner, G. Stimpfl, F. Stocker, J. Teiger, V. Vercesi, A. Weidberg, H. Zaccone, and W. Zeller. Observation of single isolated electrons of high transverse momentum in events with missing transverse energy at the CERN pp collider. *Physics Letters B*, 122(5):476 – 485, 1983.

[16] CERN. The Large Hadron Collider. `https://home.cern/science/accelerators/large-hadron-collider`, 2020.

[17] Julie Haffner. The CERN accelerator complex. `https://cds.cern.ch/images/OPEN-PHO-ACCEL-2013-056-1`, October, 2013.

[18] The ATLAS Collaboration et al. The ATLAS Experiment at the CERN Large Hadron Collider. `https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003/pdf`, 2008.

[19] Joao Pequenao. Event Cross Section in a computer generated image of the ATLAS detector. Mar 2008.

[20] Matthias Schott and Monica Dunford. Review of single vector boson production in pp collisions at $\sqrt{s} = 7$ TeV. *The European Physical Journal C*, 74(7), Jul 2014.

[21] C.G. Lester and D.J. Summers. Measuring masses of semiinvisibly decaying particles pair produced at hadron colliders. *Phys. Lett. B*, 463:99–103, 1999.

[22] Alan Barr, Christopher Lester, and P. Stephens. m(T2): The Truth behind the glamour. *J. Phys. G*, 29:2343–2363, 2003.

[23] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014.

[24] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A brief introduction to PYTHIA 8.1. *Computer Physics Communications*, 178(11):852–867, Jun 2008.

[25] ATLAS Pythia 8 tunes to 7 TeV datas. Technical Report ATL-PHYS-PUB-2014-021, CERN, Geneva, Nov 2014.

[26] Richard D. Ball, Valerio Bertone, Stefano Carrazza, Christopher S. Deans, Luigi Del Debbio, Stefano Forte, Alberto Guffanti, Nathan P. Hartland, José I. Latorre, Juan Rojo, and et al. Parton distributions with LHC data. *Nuclear Physics B*, 867(2):244–289, Feb 2013.

[27] Christoph Borschensky, Michael Krämer, Anna Kulesza, Michelangelo Mangano, Sanjay Padhi, Tilman Plehn, and Xavier Portell. Squark and gluino production cross sections in pp collisions at $\sqrt{s} = 13$, 14 and 100 TeV. *The European Physical Journal C*, 74(12), Dec 2014.

[28] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), Jul 2014.

[29] Richard D. Ball, Valerio Bertone, Stefano Carrazza, Christopher S. Deans, Luigi Del Debbio, Stefano Forte, Alberto Guffanti, Nathan P. Hartland, José I. Latorre, and et al. Parton distributions for the LHC run II. *Journal of High Energy Physics*, 2015(4), Apr 2015.

[30] Daniel Abercrombie, Nural Akchurin, Ece Akilli, Juan Alcaraz Maestre, Brandon Allen, Barbara Alvarez Gonzalez, Jeremy Andrea, Alexandre Arbey, Georges Azuelos, Patrizia Azzi, and et al. Dark Matter benchmark models for early LHC Run-2 Searches: Report of the ATLAS/CMS Dark Matter Forum. *Physics of the Dark Universe*, 27:100371, Jan 2020.

[31] ATLAS Pythia 8 tunes to 7 TeV datas. Technical Report ATL-PHYS-PUB-2014-021, CERN, Geneva, Nov 2014.

[32] Multi-Boson Simulation for 13 TeV ATLAS Analyses. Technical Report ATL-PHYS-PUB-2017-005, CERN, Geneva, May 2017.

[33] Enrico Bothmann, Gurpreet Singh Chahal, Stefan Höche, Johannes Krause, Frank Krauss, Silvan Kuttimalai, Sebastian Liebschner, Davide Napoletano, Marek Schönherr, Holger Schulz, Steffen Schumann, and Frank Siegert. Event Generation with Sherpa 2.2. *SciPost Phys.*, 7:34, 2019.

[34] Monte Carlo Generators for the Production of a W or Z/γ^* Boson in Association with Jets at ATLAS in Run 2. Technical Report ATL-PHYS-PUB-2016-003, CERN, Geneva, Jan 2016.

[35] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), Jul 2014.

[36] T Gleisberg, S Höche, F Krauss, M Schönherr, S Schumann, F Siegert, and J Winter. Event generation with SHERPA 1.1. *Journal of High Energy Physics*, 2009(02):007–007, Feb 2009.

[37] Ryan Gavin, Ye Li, Frank Petriello, and Seth Quackenbush. FEWZ 2.0: A code for hadronic Z production at next-to-next-to-leading order. *Computer Physics Communications*, 182(11):2388–2403, Nov 2011.

[38] Simone Alioli, Paolo Nason, Carlo Oleari, and Emanuele Re. NLO Higgs boson production via gluon fusion matched with shower in POWHEG. *Journal of High Energy Physics*, 2009(04):002–002, Apr 2009.

[39] Paolo Nason and Carlo Oleari. NLO Higgs boson production via vector-boson fusion matched with shower in POWHEG. *Journal of High Energy Physics*, 2010(2), Feb 2010.

[40] Parul Pandey. Understanding the Mathematics behind Gradient Descent. `https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e`, March, 2019.

[41] Anup Bhande. What is underfitting and overfitting in machine learning and how to deal with it. `https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76`, March, 2018.

[42] Rishi Sidhu. Understanding ML Evaluation Metrics — Precision Recall. https://medium.com/x8-the-ai-community/understanding-ml-evaluation-metrics-precision-recall-2b3fb915b666, June, 2019.

[43] Jason Browniee. How to Choose Loss Functions When Training Deep Learning Neural Networks. https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/, January, 2019.

[44] Ajay Kumar, Rama Sushil, and Arvind Kumar Tiwari. Significance of Accuracy Levels in Cancer Prediction using Machine Learning Techniques. http://bbrc.in/bbrc/significance-of-accuracy-levels-in-cancer-prediction-using-machine-learning-techniques/, August, 2019.

[45] dmlc XGBoost. About XGBoost. https://xgboost.ai/.

[46] Katherine Woodruff. Introduction to boosted decision trees. https://indico.fnal.gov/event/15356/contributions/31377/attachments/19671/24560/DecisionTrees.pdf, September, 2017.

[47] Mei-Hung Chiu, Yuh-Ru Yu, Hongming Liaw, and Lin Hao. THE USE OF FACIAL MICRO-EXPRESSION STATE AND TREE-FOREST MODEL FOR PREDICTING CONCEPTUAL-CONFLICT BASED CONCEPTUAL CHANGE. 01 2016.

[48] Mahsa Shoaran, Benyamin Allahgholizadeh Haghi, Milad Taghavi, Masoud Farivar, and Azita Emami. Energy-Efficient Classification for Resource-Constrained Biomedical Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, PP:1–1, 06 2018.

[49] Anuja Nagpal. Decision Tree Ensembles - Bagging and Boosting. https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9, October, 2017.

[50] XGBoost developers. Python API Reference. https://xgboost.readthedocs.io/en/latest/python/python_api.html, 2020.

[51] Larry Hardesty. Explained: Neural networks. April, 2017.

[52] Michael A. Nielsen. Neural Networks and Deep Learning. http://neuralnetworksanddeeplearning.com/chap5.html, 2015.

[53] Keras API reference. https://keras.io/api/.

[54] Sagar Sharma. Activation Functions: Neural Networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[55] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for Activation Functions. *CoRR*, abs/1710.05941, 2017.

[56] Vitaly Bushaev. How do we 'train' neural networks. https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73.

[57] Vitaly Bushaev. Stochastic Gradient Descent with momentum. https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[59] Mirko Stojiljković. The Pandas DataFrame: Make Working With Data Delightful. https://realpython.com/pandas-dataframe/, April, 2020.

[60] ROOT project. https://github.com/root-project/root/#cite.

[61] Leah A. Wasser. Hierarchical Data Formats - What is HDF5? https://www.neonscience.org/about-hdf5.