Modern Artificial Intelligence Tools Applied to searches for Supersymmetry at the ATLAS experiment

Samuel Valentine
201508651

Supervisors:
Monica D'Onofrio and Carl Gwiliiam

April 18, 2024

Contents

1 Abstract

2 Introduction

Evidence for dark matter is littered all across the universe. One of the main indications of its existence is that when observing galaxies, the mass of normal matter is not nearly large enough to hold the galaxies together. There must be an invisible 'dark matter' that compensates for the lack of mass in order to have enough gravity to hold the galaxies together [1]. Evidence suggests dark matter makes up 27% of the universe. It does not interact with the electromagnetic force, nor does it absorb, reflect or emit light, making it incredibly difficult to detect [2].

It is widely accepted that dark matter (DM) exists, yet it is far more unknown about what it is. One prediction is that it is a new particle with some non-gravitational coupling. Supersymmetry (SUSY) is a theory that involves a partner particle for each standard model (SM) particle [3]. The lightest of these SUSY particles would be stable and hence serve as a good candidate for a DM particle. If this is the case, it may be produced and studied at the LHC, however it is yet to be detected. There are ongoing analyses of the data collected by the ATLAS experiment at the LHC to try to change this. In this specific study we consider heavy SUSY particles decaying into DM candidates in association with a W boson and a Higgs boson (H).

The difficulty with this ongoing search is that the events for this process are relatively rare in comparison to the amount of events for known SM processes that serve as background. Not only this, but the events are also similar to the known SM processes, making this event even harder to distinguish. Artificial Intelligence (AI) in the form of machine learning (ML) can learn to recognise the difference between background and signal, helping to optimise analysis of the signal without the need for direct instruction.

Machine Learning has different branches such as branch decision trees and artificial neural networks. Multivariate classifiers are a form of machine learning that are useful for identifying small signals in large data sets containing mostly background. Graph neural networks (GNNs) can analyse the relationships (edges) between different objects (nodes) leading to predictions be able to be made on individual nodes or edges [4].

2.1 Physical Motivation

The standard model encapsulates our current best understanding of how fundamental particles and three of the four fundamental forces are related to each other [5]. The standard model has limitations to the explanations of some physical phenomena observed in reality. The last fundamental force, gravity, is completely unexplained by the SM. Gravity is extremely weak in the subatomic world, however in phenomena such as black holes or the first moments of the big bang, gravity becomes extremely important [6]. On top of this, the SM predicts a massless neutrino. Neutrino oscillation experiments have shown that neutrinos have a small mass and behave like other particles. Rectifying this by adding mass to neutrinos in the SM leads to new theoretical problems, and is therefore unable to incorporate a non-massless neutrino [7]. The hierarchy problem is unexplained by the SM. The Higgs field couples proportionally to mass, in theory massive particles should give a quantum 'loop' that drives the mass up to the Planck scale. In reality however, the Higgs is ten quadrillion times smaller than the Planck scale [8]. This discrepancy between them cannot be explained by the SM as the SM offers no mechanism to negate this. The SM also doesn't explain the imbalance between matter and anti-matter abundance in the universe[7]. If the universe did in fact originate in a big bang, the number of matter and anti-matter should be equal, thus there must be a mechanism unexplained by the SM preventing this from being the case. Perhaps most obviously, the SM only accounts for the 5% of the universe that is made up by visible matter, unable to offer any particles that are good candidates for DM. These are just some of the reasons for the needs of solutions beyond the SM.

Supersymmetry is an extension of the SM that predicts a superpartner particle for each known particle in the SM. Each partner particle would differ by half a unit of spin to that of its SM counterpart, i.e. for each fermion there is a corresponding boson and for every boson there is a corresponding fermion [3]. The additional fermionic and bosonic partners postulated in supersymmetry cancel the contributions to the Higgs mass made by SM particles, causing the Higgs mass to remain low, solving the hierarchy problem [9]. SUSY offers the possibility of 'loop induced' neutrino masses through a theory called the 'soft see-saw mechanism' [10], aswell as having some theories that contain a fermion superpartner to the hypothetical gravitational force carrying particle, the graviton, called the gravitino [11]. Much like the existence of anti-matter solved the question of how the electron could be such a small size, SUSY allows the SM to describe physics down to the Planck length, and makes the unification of the strong, weak and electromagnetic forces possible [12]. Supersymmetry however, also poses new problems, there is currently no explanation for why superpartners are heavier than their ordinary counterparts, why they are so well hidden in rare phenomena and how they can be discovered experimentally. This last problem is currently trying to be solved by multiple ongoing experiments that are attempting to discover DM.

The superpartners of the SM Higgs, Z and W bosons are called higgsinos, binos and winos respectively, collectively they are referred to as electroweakinos [13]. The mass eigenstates of electroweakinos come in two forms: linear combinations of higgsino and wino fields, called charginos, $\tilde{\chi}_i^{\pm}(i = 1, 2)$ and linear combinations of higgsino, wino and bino fields, called neutralinos, $\tilde{\chi}_j^0(j = 1, 2, 3, 4)$. Both of these are ordered in value of increasing mass. SUSY particles have an R-parity (also known as 'matter-parity') with a value of -1. In contrast to this SM particles have an R-parity of $+1$, leading to the theoretical existence of a lightest SUSY particle (LSP) that would be unable to decay into SUSY particles due to R-parity needing to be conserved. This stable LSP therefore serves as a good DM particle.

Detection of dark matter particles at CERN involves colliding beams of normal matter particles, causing dark matter matter particles to be produced [14]. The ATLAS experiment at

CERN (Figure 1) is concerned with analysing the products of high energy particle collisions. At the center of ATLAS, particles collide at a rate of a billion collisions per second [15].

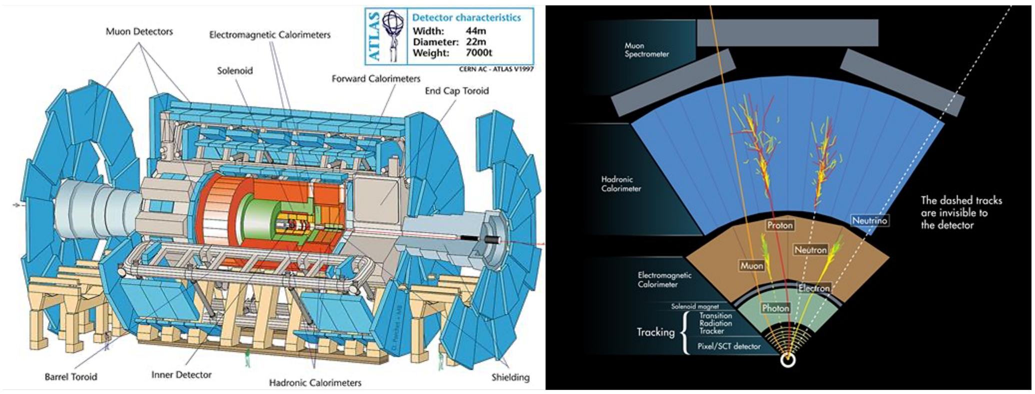

Figure 1: The layout of the ATLAS detector (left) [16] and the paths of each type of particle inside the detector (right) [17].

Transverse momentum, p^T is one of the features measured for events measured by ATLAS. The transverse momentum corresponds to the momentum in the transverse plane of the ATLAS detector, i.e. the directions that the two colliding particles travel. This is a useful measurement as due to the nature of the head on collision of the two protons, the total transverse momentum of all the objects produced should be 0. If this isn't the case, there is some missing transverse momentum that hasn't been detected [13]. The azimuthal angle feature, ϕ corresponds to the position in the ATLAS detector on the z axis that the energy corresponding to an object was detected. The pseudorapidity, η^T is another geometric feature measured by the ATLAS detector. Rapidity is a property used to measure angles at highly relativistic speeds. When dealing with objects moving close to the speed of light, angles tend to grow and shrink, rendering normal angle measurements less useful. Rapidity accounts for these relativistic effects. It has a value of 0 for trajectories perpendicular to the beam and positive and negative values [**?**, PSEUDORAPIDITY]

The incomprehensible number of collisions creates a lot of products, some of which could be SUSY particles through processes such as the focus of this project, shown in Figure 2 . The experimental signature is characterised by energies relating to the presence of two *b*-quarks (produced from the decay of the Higgs), a lepton (produced in the decay of the W boson), along with missing transverse energy (MET). This MET accounts for that taken by the dark matter candidate (pair of neutralinos) and the neutrino (the other decay product of the W boson). This was searched for in run 2 of the LHC using a ML approach [13].

While this creates potential SUSY events, it also produces a high number of background processes, some of which produce signals that are incredibly similar to that of the SUSY process

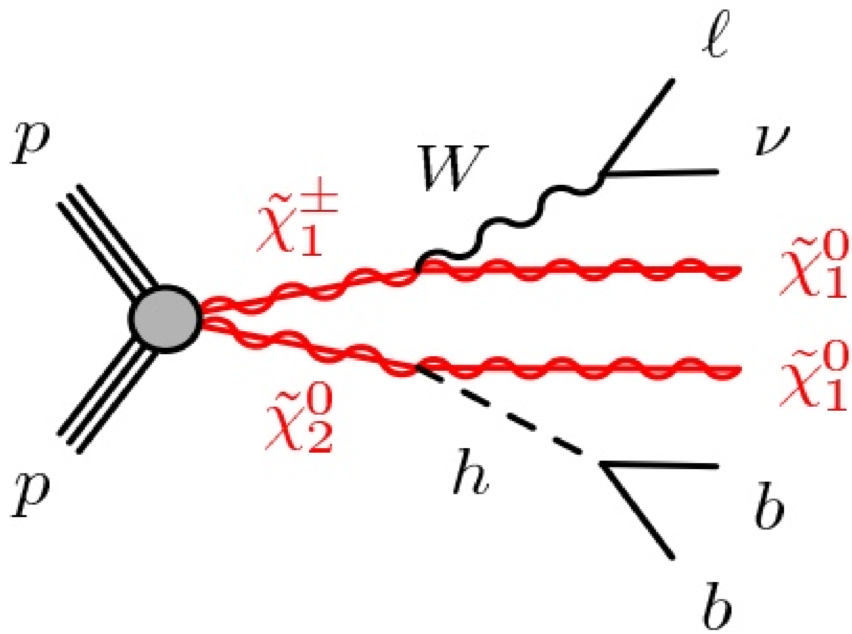

Figure 2: SUSY decay process

being searched for. Two of the main background processes are top, anti-top pair production or single-top production, leading to similar decay products to that of the signal process, shown in Figure 3. The background processes significantly overwhelm the signal, making selections on individual variables to try to separate the background from signal sub-optimal. The analysis of run-2 data previously considered multiple variables together using a multivariate approach. .The Boosted Decision Tree (BDT) output of the background and signal processes from the previous run-2 data analysis [13] are shown in figure 4. From the figure high BDT scores are shown to have picked up the signal, however there is still a lot of background. There is still a need for improvement using more advanced machine learning in order to separate the signal from the background further, this is studied in this project.

2.2 Machine Learning

Deep learning is a machine learning technique that learns from examples. It teaches a computer to filter information through layers in order to predict and classify further information. Neural networks can be classified into two camps. Feedforward networks only filter inputs one way, putting the inputs through a given number of hidden layers until you get an output. Feedback networks contain feedback paths, allowing information to travel in all possible directions and allowing all possible connections. Feedback networks are often utilised for tasks in which opti-

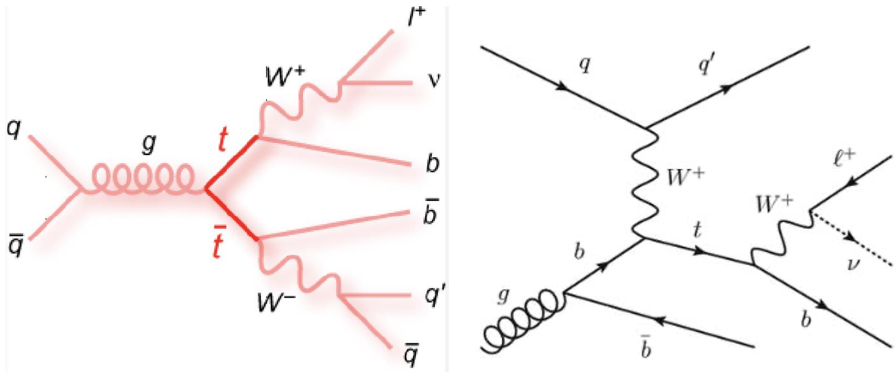

Figure 3: The the top, anti-top decay process (left) and a single-top decay (excluding a light quark from radiation) (right).

misation comes from the best arrangement of interconnected factors. [18].

Generally, feedback networks operate in the following sequence: Neural networks are first trained on known data with expected outputs to learn from. For a classification task like separating signal from background, the neural network would be trained on known signals and backgrounds to learn how to distinguish between the two. Information is given to an input node in the form of an activation value. The activation value is then passed to the next node based on the connection strengths between the nodes. This node now calculates the sum of the weighted inputs (linear function)[19] from the previous layer and applies a transfer function. The node then applies an activation function. This activation function represents the likelihood that the node will pass information. There are different activation functions used depending on the nature of the data being processed. This is repeated until the activation value reaches an output node. The output is compared with the expected output and the difference between them (the cost function) is calculated. The neural network then tweaks the weights between each node with the goal of minimising the cost function (backpropagation). Once the weights have been optimised the neural network moves to the testing phase, in which it makes predictions without expected values using the optimised weights from the training phase.

The overall performance of a neural network can be measured using ROC curves. ROC curves are commonly used to show the performance of a classification model at all classification thresholds [20]. It plots the true positive rate against the false positive rate at varying threshold levels for the probability that classifies a result as positive. The true positive rate, TPR is calculated using:

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

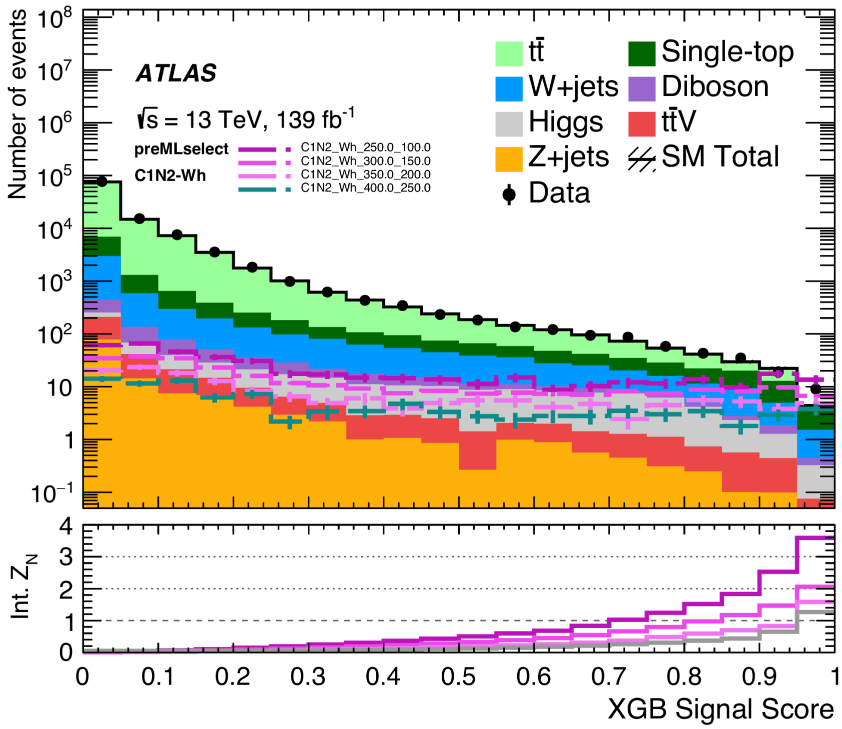

Figure 4: Comparison of SUSY signal and background processes in terms of number of events, signal tends to be closer to an XGB Signal score of 1 while background tends to be closer to a signal score of 0 [13].

where TP is the true positive, i.e. the number of times a classification model correctly classifies a positive result as a positive result. FN is false negative, i.e the number of times a classification model incorrectly classifies a positive result as a negative result. The false positive rate, FPR is calculated using:

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

where FP is the false positive, i.e. the number of times a classification model incorrectly classifies a negative result as a positive one. TN is true negative, i.e the number of times a classification model correctly classifies a negative result. The classification threshold dictates what minimum probability the classification model has to give a result for it to be considered positive. This means that the lower the threshold, the greater the number of false positives and true positives there will be. The area under the curve (AUC) is a measure of the probability that a classification model's prediction is correct. It is calculated by taking the area under the ROC curve plotted by varying the classification threshold across all values from 0 to 1, with a threshold of 0

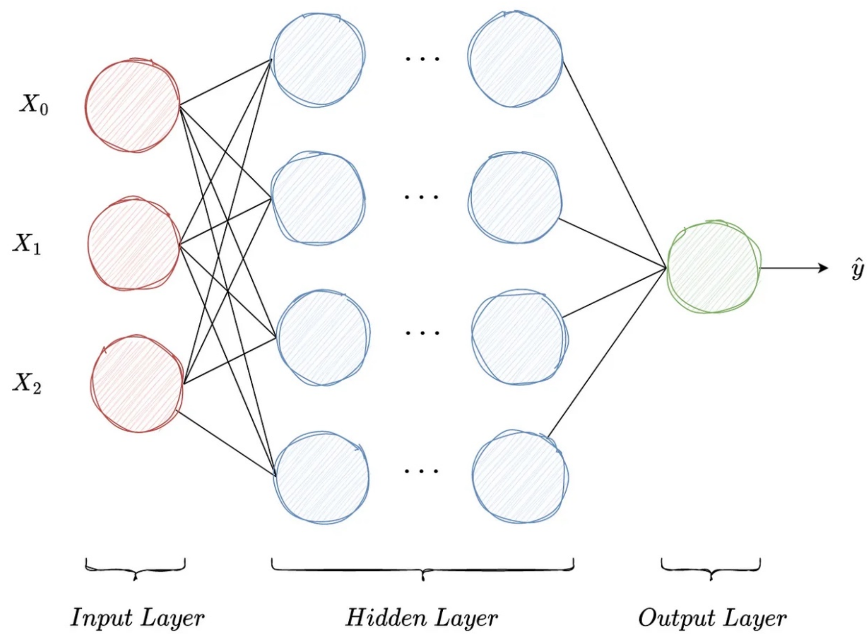

Figure 5: Basic layout of neural networks, many inputs are taken in and the network outputs an optimal value based on the correlations between inputs. [19]

meaning everything will be classified as a positive result and 1 meaning nothing will be classified as a positive result.

A graph can be used to show the relationships (edges) between a collection of objects (nodes). Graphs can be used to represent many different things. The range of types of information that can be represented as a graph is far greater than you might initially think. Graph Neural Networks are a machine learning tool that formats information as graphs in order to make their predictions. GNNs can make predictions about some property of a node (node-level), the property or presence of edges in a graph (edge-level) and also a single property of the whole graph (graph-level). It works by acquiring embeddings from each feature of the graph. These embeddings are pooled together, allowing the nodes to apply a linear function, resulting in an updated embedding for that node. Neighbouring nodes and edges can exchange information to influence each other's updated embeddings (message passing). Message passing GNN layers can be stacked together, potentially resulting in each node prediction eventually being influenced by information from across the entire graph. The GNN's complexity allows it to be a vastly customizable model. The number of GNN layers (depth), the dimensionality of each characteristic, the function used for pooling, and which features of the graph that get updated can all be changed in order to optimise the GNN based on the type and amount of information being processed. Not only this, but it is possible to understand how a GNN is learning by analysing the penultimate layer activations [21].

3 Experimental Method

3.1 Individual features

The measurements taken at ATLAS can be used to provide further features that can indicate more about individual events, these features are the following. The jet quantile is the likelihood that a jet measured at ATLAS corresponds to a b-quark. The two b-jets each have a mass calculated using the relativistic momentum of the jet. It is found by rearranging the relativistic energy-momentum relation to get:

$$m_b = \sqrt{E^2 - p_T^2} \tag{3}$$

where E is the accepted value for the rest mass energy of the b-quark. The missing energy has the feature of significance attached to it, $E^T_{miss}Sig.$. This is a feature that indicates how likely a result of MET is to be attributed to a weakly interacting particle (neutrinos or something more exotic) rather than a sum of resolution smearing [22]. The measurements p^T, η and ϕ outlined in secton 2.1, were combined with these further variables to form all the features that were used as information for each object in each graph.

Graphs sometimes contained 6 objects and sometimes contained 7 depending on whether or not a non b-jet was present in that event. This made object identification for analysis slightly more complicated, as after the jets, the other objects would have different order depending on how many jets an event contained. On top of this, the jets in each graph had an object order number based on the transverse momentum of the jets. The jet with the highest transverse momentum always had the lowest object number, however the jets with the highest two momenta weren't always the two b-jets, so the non b-jets object number could change from graph to graph. Luckily the next two objects were the two b-jets, which had the same properties as the jets representing the two b-quarks. This varying object order can be shown by the layout of the graphs in table 1 which only contains 6 objects and table 2 which contains 7 objects due to the presence of the non b-jet. Therefore equation 4 was used to calculate which object was the non b-jet and allow plots to be created on the properties of it:

$$\Delta\eta = |\eta_i - \eta_j| \tag{4}$$

where η_1 was the pseudorapidity of a given jet and η_2 was the pseudorapidity of a given b-quark object. If $\Delta\eta \approx 0$, then this would indicate that the jet in question was that of a b-quark. From identifying with which jets this happened for for η_{b1} and η_{b2}, the object number of the non b-jet could be identified.

The individual features of each object in each event were investigated to learn which of these were better indicators of the differences between a dark matter process signature and a background signature. The individual features that proved to have the largest difference in signature between signal and background was η of the non b-jet, as shown in figure 6. The feature that

	$p_T[GeV]$	η	$\phi[rads]$	$quantile$	$mass[GeV/c^2]$	$Significance$
jet1	146.949	-2.155	-1.183	5.0	nan	nan
jet2	53.620	-1.249	-0.136	5.0	nan	nan
b1	146.949	-2.155	-1.183	5.0	13.676	nan
b2	53.620	-1.249	-0.136	5.0	7.334	nan
lepton	41.106	-1.907	2.303	nan	nan	nan
energy	189.534	nan	1.794	nan	nan	8.663

Table 1: Objects and features of the 1st graph

	$p_T[GeV]$	η	$\phi[rads]$	$quantile$	$mass[GeV/c^2]$	$Significance$
jet1	112.468	1.507	1.188	5.0	nan	nan
jet2	45.432	-2.212	-0.255	1.0	nan	nan
jet3	43.096	2.157	2.260	5.0	nan	nan
b1	112.468	1.507	1.188	5.0	10.933	nan
b2	43.096	2.157	2.260	5.0	5.007	nan
lepton	42.533	0.218	-0.853	nan	nan	nan
energy	132.924	nan	-2.321	nan	nan	8.686

Table 2: Objects and features of the 486th graph, here the non b-jet was identified as jet 2.

had the least importance for differentiation was ϕ, which was indistinguishable between signal and background for all applicable objects, as shown in figure 7.

By learning this it allowed the knowledge of what the GNN should be learning itself. It also allowed the planning of measuring the change in the GNNs performance when more important or less important features are witheld from the inputs for the GNN. This was measured by plotting histograms of the values for signal and the two main background processes and observing the differences by eye.

3.2 Relational features

Variables between the objects in the events were not initially given to the GNN, but could serve as good indicators of the differences between signal and background processes. This had already been shown by the SHAP (Shapely Additive Values) plot from the BDT analysis carried out in [13] shown in figure 8. The figure shows that among the most important features for signal and background discrimination were the distance between the two b-jets, ΔR_{bb} and most notably, the invariant mass of the two b-jets.
ΔR_{bb} and M_{bb} were calculated using:

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{5}$$

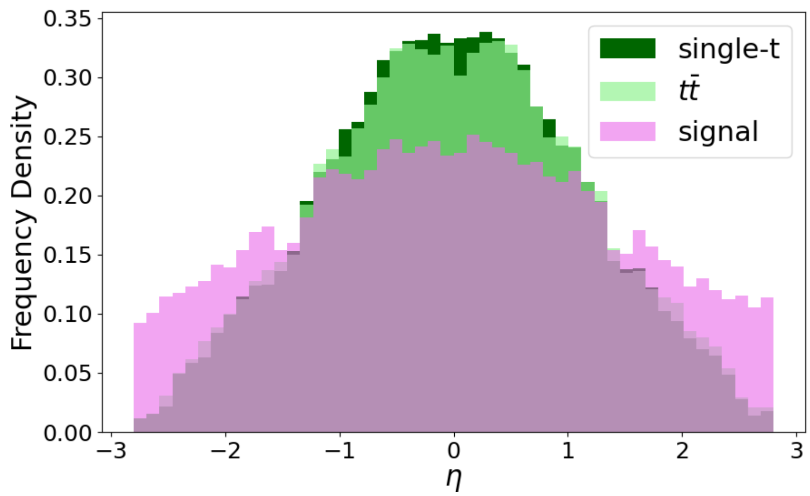

Figure 6: Plot comparing values of eta of the non b-jet for signal and the two main background processes.

and:

$$M = \sqrt{2p_{T_1}p_{T_2}(\cosh(\Delta\eta) - \cos(\Delta\phi))} \tag{6}$$

respectively, where p^T_{b1} and p^T_{b2} are the transverse momenta of each b-jet, η_{b1} and η_{b2} were used in equation 4 to get $\Delta\eta$ and $\Delta\phi$ was calculated using:

$$\Delta\phi = |\phi_{b1} - \phi_{b2}| \tag{7}$$

Finding these types of inter-object features could then be used to measure how well the GNN was learning these relationships on its own, or whether it needed to be given these to begin with. These relational features were analysed in the same way that the individual features of each object were. As a result of this, the plots shown in figures 9 and 10 were produced.

These would be useful once the GNN had made its predictions on the events. If the GNN had learned these properties that it had not been given as inputs, it would reproduce similar looking plots.

3.3 The GNN

In the search for SUSY processes, the decay products' masses, energies and directions of the previously discussed processes could all be applied as the nodes and edges of a graph. This allowed individual variables to all be considered at once meaning the artificial intelligence could better understand the relationships between each variable and polarise the separation of the

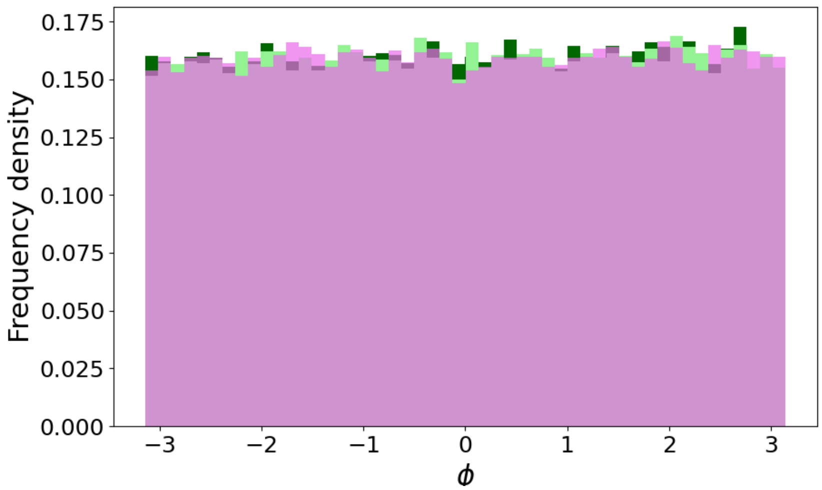

Figure 7: Plot comparing values of phi of the first b-jet for signal and the two main background processes.

signal and background to a more optimal level. The information of simulated data was created so that it was as close to the data collected by ATLAS for this particular experiment as possible. In order to get the best comparison with the analysis previously done using BDT's, as outlined in [13], the simulated data with features across the same feature ranges were used, as outlined in table 3

Graphs for each event were created, with each node representing an object: the 2-3 jets, 2 b-quarks, a lepton and the missing energy all with the intrinsic properties: transverse momentum (pT), pseudorapidity (η), the angle (ϕ), the mass of the b-quarks (for b-jets only) calculated

Variable	Signal Region
E_T^{miss}	$> 50GeV$
N_{lepton}, $p_T > 27GeV$	1
N_{jets}	2-3
N_{bjets}, $p_T > 30GeV$	2
m_{bb}	$\in [95, 140]$
$E_T^{miss}\ Sig.$	> 8

Table 3: Signal region that data was simulated within, taken from the previous study on analysis using a BDT [13]

.

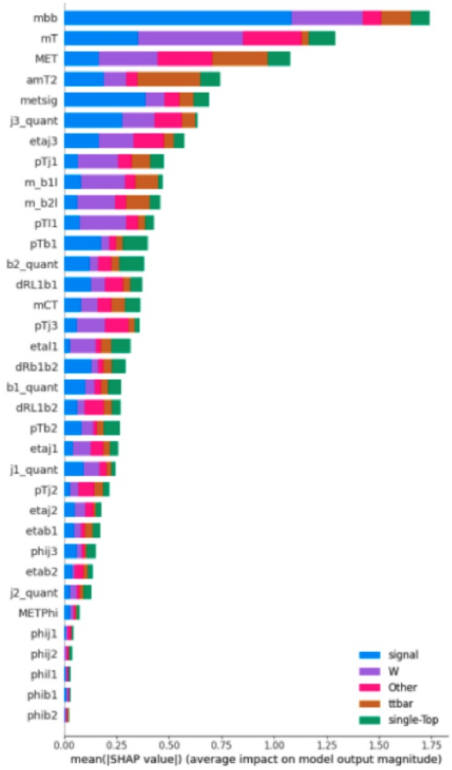

Figure 8: Plot comparing importance of features of objects in terms of how useful they are for indicating the difference between a signal event or a background event from previous analysis.

using relativistic momentum (m), the quantile of each jet and the significance (for missing energy node only). Meanwhile, relationships between these features (e.g. distance apart and invariant masses) were not added to the GNN edges at first. The layout of the graphs inputted in to the GNN is shown in figure 11. The graphs were created so that they were fully connected, meaning all nodes had edges between them. They were also undirected, this meant that information could be passed between all nodes of the graph in any direction. In order to negate the 'not a number' ('nan') values that can be seen in table 1 and table 2 the graphs had to be extended from having a dimension of 6 corresponding to the 6 features, to 12. This was so that each feature value could have a flag associated with it. Flags are used to signal to the computer program that there is a 'nan' value there for the program to deal with [23].

First the data was split into a testing and training part, with 80% of the dataset used for training and 20% of the data used for testing. The uneven split was due to the fact that the testing phase does not improve the GNN at all, so there was no need to use any more of the data than this as it would just be a waste. It was split such that each of the three types of events,

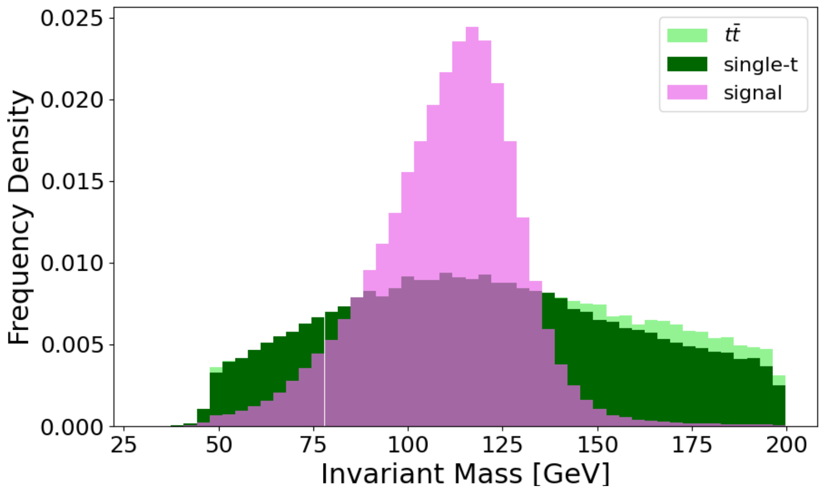

Figure 9: Plot comparing invariant mass of the two b-jets for signal and the two main background processes.

signal, $t\bar{t}$ and t decay had the same 80:20 split between training and testing for their number of events.

Data loaders were used for the GNN. Data loaders are useful when working with large datasets of graphs. They put the graphs into 'batches' to increase the speed at which the data can be processed. In the GNN used for this research they did this by stacking the graphs representing single events in a diagonal fashion. This creates one giant graph holding multiple individual graphs. This is effective because it causes the adjacency matrices to be saved in such a way that only non-zero values are held (those representing the edges), allowing the batches to save space [24]. To start with, a GNN with batch number of 96 was used, i.e. there were 96 graphs per batch, meaning for a train dataset of 240,000 graphs, there were 2500 batches.

Data was inputted into a GNN with 12 input channels, 36 hidden layers and 1 output. The 12 input channels were due to the 12 features of each graph (6 features, 6 flags). The 36 hidden layers were so that the GNN was fully connected, so that there were edges connected to all of the objects. The single output was so that there was a single prediction made by the GNN as to the probability of a graph representing a signal event. The GNN was created in a type of GNN that supports message passing from neighbouring nodes, message passing with one-dimensional edge weight information and supports message passing in static graphs. This meant that the GNN could pass information from node to node (as long as the two nodes were connected which they all were) and could pass information from weight on the edges (useful for when relational

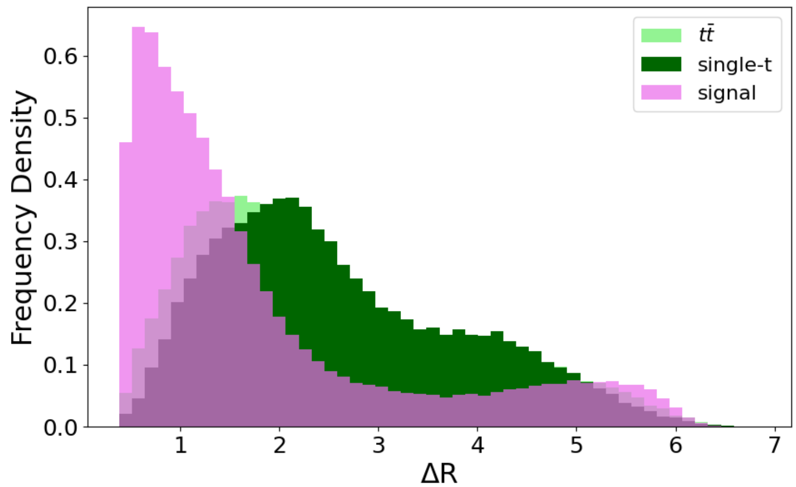

Figure 10: Plot comparing the distance between the two b-jets for signal and the two main background processes.

features were introduced.

Models can be trained and tested on an epoch by epoch basis. A training epoch is when all the information availabe to be inputted into the GNN has been i.e. all the graphs in the training dataset had been passed through the GNN to have an output delivered once. A validation epoch was then carried out where the testing data was ran through the GNN that had now updated it's weightings. The next training epoch was then ran, this time not based on random weightings but the previously learned embeddings, and the cycle was repeated. The loss and accuracy of the graphs as this cycle was carried out was plotted, in order to know that the GNN was getting more accurate.

The GNN output is in terms of 'logits' for each event. 'Logits' are a prediction made by the GNN that is expressed across the whole number line, rather than just between 0 and 1 [25]. Due to this, they needed to be converted to a prediction between 0 and 1. This was done using one of the activation functions. In this case, it was done using a sigmoid function. After all this the GNN had outputted a prediction for a probability of how likely all the graphs were to be signal.

The GNN was trained by giving it a certain amount of events for signal and a certain amount of events for the two main background events. The number of events chosen for each of these

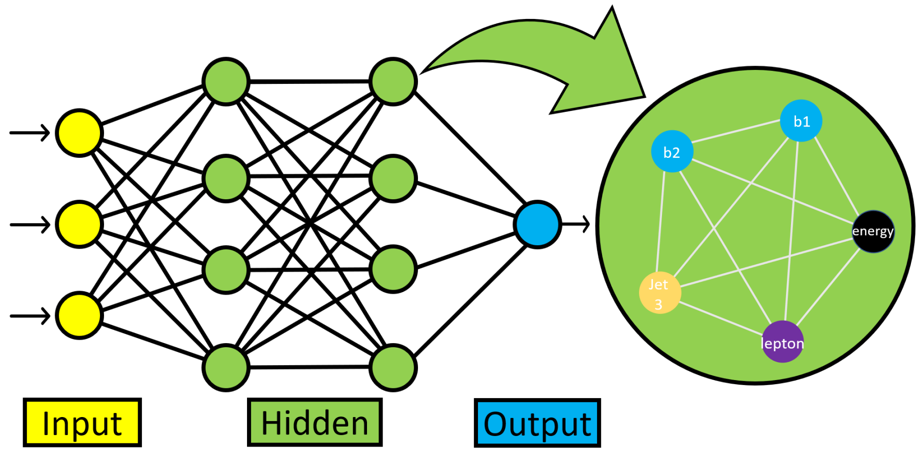

Figure 11: Diagram showing how information was inputted into the GNN as graphs.

was based on a happy-medium between overall performance of the GNN and the time taken to run the GNN using that number of events. This lead to 150,000 signal events, and 75000 events each for the two background events given to the GNN. In reality the number of signal events is orders of magnitude less than that of the two background processes (as can be seen in figure 4) however in order to train the GNN effectively the signal events supplied were comparative in number to that of the background processes.

The performance of the GNN was tested using a Reciever Operating Characteristic curve. In the case of the GNN, the TP was the number of times the GNN correctly classified an event as a signal, the FN was the number of times the GNN classified an event as background when the event was signal, the FP was the number of times the GNN classified an event as a signal when the event was background and the TN was the number of times the GNN correctly classified an event as background. The AUC was used to measure how the performance of the GNN varied when different settings, inputs and dataset sizes were changed.

A histogram was plotted with the predictions for each graph made by the GNN in terms of probability of being signal. Further histograms were made based on specific features, with the feature value being plotted against frequency density. Three different sets of data were initially plotted onto the same histograms: the data that had been signal, the data that had been background that had been assigned a GNN score of less than 0.6, and background events that had a GNN score of above 0.6. After this, similar histograms were plotted with signal data also split by the 0.6 threshold. This was done for the features that had previously been found to be the most important based on previous analysis of the individual features in the method outlined in sections 3.1 and 3.2 as well as the previous research from [13]. It was also done for ΔR and M to get an idea of how well the GNN was recognising the importance of these relationships on its own. If it was, this would be indicated by the histograms of GNN score above the threshold to be clearly different from that of the histogram of the low GNN threshold. It would also follow a similar trend to that of the histogram of signal data for these complex features plotted previously.

4 Results and Discussion

- Talk about AUC in general.
- Comparison of AUC of GNN with the AUC of the BDT
- Talk about the output distribution in comparison to that of the BDT
- Talk about how well GNN has learned about individual features
- Talk about how well the GNN has learned complex features.

4.1 GNN performance

The AUC of the ROC curve from modifying GNN settings only without modifying the input information maximised at a value of 0.83. There was little to zero change in the overall performance by changing the batch number. The only thing the batch number changed was the proportion of data used before the train performance plateaued, where larger batch numbers needed a higher proportion of data to optimise GNN performance. The GNN training performance always plateaued before being trained on at least 50% of the training data. This in theory meant it would be possible to reduce the proportion of data used to train the GNN, however this would have no effect on the GNN's performance. The amount of data used was decided on due to the fact that that amount of data would take around 350s to run, which met the needs for the amount of data with the need to be efficient with time management.

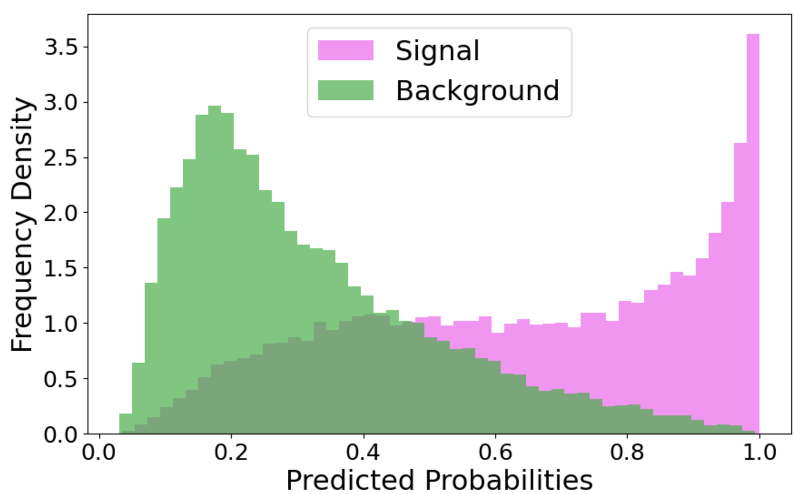

Figure 12: Plot of the model output distribution for the GNN for signal and background processes.

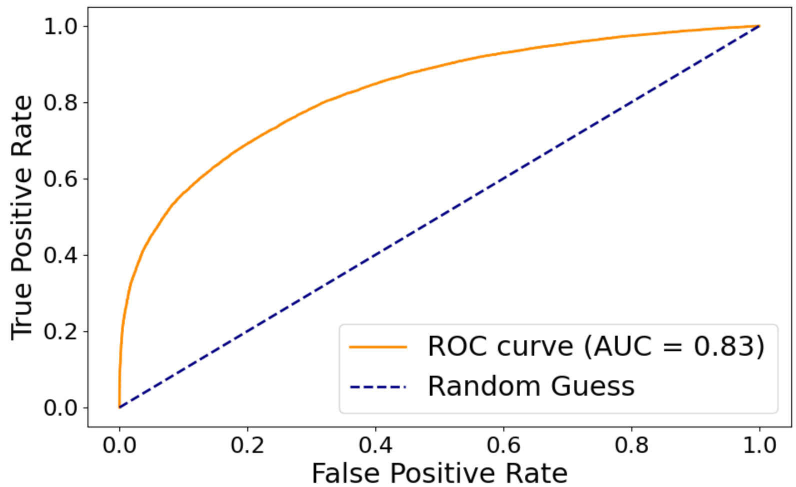

Figure 13: Plot of ROC curve of simple GNN without giving any information of complex variables.

4.2 GNN learning

5 Conclusion

- How effective research was
- What could be done further.

References

[1] W. Clavin. What Is Dark Matter, and Where Is it Hiding? [Online]. Available: https://magazine.caltech.edu/post/where-is-dark-matter-hiding

[2] CERN. (2023) Dark Matter. Accessed: 2023-11-30. [Online]. Available: https://home.cern/science/physics/dark-matter

[3] CERN. (2023) Supersymmetry. Accessed: 2023-10-27. [Online]. Available: https://home.cern/science/physics/supersymmetry

[4] Distill. (2021) A Gentle Introduction to Graph Neural Networks. Accessed: 2023-10-25. [Online]. Available: https://distill.pub/2021/gnn-intro/

[5] CERN. (2024) The standard model. Accessed: 2024-04-16. [Online]. Available: https://home.cern/science/physics/standard-model

[6] Institute of Physics. (2024) The standard model. Accessed: 2024-04-16. [Online]. Available: https://www.iop.org/explore-physics/big-ideas-physics/standard-model

[7] E. Paudel, "Problems of standard model, review," *BMC Journal of Scientific Research*, vol. 4, 03 2022.

[8] M. Strassler. The hierarchy problem. [Online]. Available: https://profmattstrassler.com/articles-and-posts/particle-physics-basics/the-hierarchy-problem/

[9] E. Smith. (2019) The hierarchy problem. Accessed: 2024-04-17. [Online]. Available: https://homes.psd.uchicago.edu/ sethi/Teaching/P445-S2019/Emily$_{S}mith_{Q}FT_{I}II_{F}inal_{P}aper.pdf$

[10] L. Megrelidze and Z. Tavartkiladze, "Soft see-saw: Radiative origin of neutrino masses in susy theories," *Nuclear Physics B*, vol. 914, pp. 553–576, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0550321316303807

[11] H. Pacey. (2019) Supersymmetry. Accessed: 2024-04-17. [Online]. Available: https://www.hep.phy.cam.ac.uk/ hp341/JohnRose2017$_{S}upersymmetry.pdf$

[12] U. of California Berkeley. Introduction to Supersymmetry. Accessed: 2023-12-04. [Online]. Available: http://hitoshi.berkeley.edu/public_html/susy/susy.html

[13] A. Collaboration, "Search for direct production of electroweakinos in final states with one lepton, jets and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ tev with the atlas detector," 2023.

[14] N. Dinmore. Celebrate Dark Matter Day 2022 with CERN,. Accessed: 2023-11-30. [Online]. Available: https://home.cern/news/announcement/cern/celebrate-dark-matter-day-2022-cern

[15] A. E. at CERN. (2015) Detector & Technology. Accessed: 2023-10-27. [Online]. Available: https://home.cern/science/physics/supersymmetry

[16] C. AC, "Layout of ATLAS. Dessin representant le detecteur ATLAS," 1998, Accessed: 2023-12-07. [Online]. Available: https://cds.cern.ch/record/39038

[17] J. Pequenao and P. Schaffner, "How ATLAS detects particles: diagram of particle paths in the detector," 2013, Accessed: 2023-12-07. [Online]. Available: https://cds.cern.ch/record/1505342

[18] A. Bonner, "The Complete Beginner's Guide to Deep Learning: Artificial Neural Networks," Jun 2019. [Online]. Available: https://towardsdatascience.com/simply-deep-learning-an-effortless-introduction-45591a1c4abb

[19] Salar, "A beginner guide to neural networks," Mar 2023. [Online]. Available: https://ai.plainenglish.io/a-beginner-intro-to-neural-networks-52911064fa18

[20] Google, “Classification: ROC Curve and AUC — Machine Learning Crash Course,” 2019, Accessed: 2024-04-15. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[21] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, “A gentle introduction to graph neural networks,” Sep 2021. [Online]. Available: https://distill.pub/2021/gnn-intro/

[22] “Object-based missing transverse momentum significance in the ATLAS detector,” CERN, Geneva, Tech. Rep., 2018, all figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2018-038. [Online]. Available: https://cds.cern.ch/record/2630948

[23] Flag concept. [Online]. Available: https://press.rebus.community/programmingfundamentals/chapter/flag-concept/

[24] pytorch$_{g}eometricdocumentation.Advancedmini - batching.Accessed : 2024 - 04 - 17.[Online].Available : https : //pytorch - geometric.readthedocs.io/en/latest/advanced/batching.html$

[25] Logit. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/logit

6 Appendix

Extra graphs, uncertainty calculations and the like.