

Thesis Presentation: Title of thesis.

Outline: Outline of presentation.

Standard model: The Standard Model of particle physics has been used to explain many observed phenomena in the nature with great precision. The fundamental particles that have been discovered are seen in the Figure with their respective charges, spins and masses. 17 particles divided into fermions and bosons. But the SM does not explain everything, like the theorized graviton particle for gravitation. This figure also shows that the neutrino masses are zero. This does not fit with observations of the neutrinos, which have been discovered to have non-zero masses from neutrino oscillations from experiments involving the Sun.

To explain this non-zero mass of the neutrinos a mechanism with heavy neutrino masses and right-handed neutrinos are introduced. The other fundamental particles have both left-handed and right-handed particles. Left-handed means that the direction of spin and motion of the particle are opposite. This mechanism is called the Inverse Seesaw mechanism. This mechanism can lead to heavy pseudo-Dirac neutrinos with triplepton final states as seen in this Figure: **Next page**

Triplepton final state: Here we see a diagram of a proton-proton collision making a W-boson that decays into a lepton and a heavy neutrino. The heavy neutrino then decays into another lepton with opposite charge with compared to the first lepton and a new W-boson. This W-boson decays into another lepton and a neutrino. Such p-p collisions are produced at the Large Hadron Collider (LHC) at CERN. The produced leptons are detected by detectors like ATLAS, while the neutrinos are not directly detected since they do not interact with matter and will only be detected as missing transverse energy since the total energy of the p-p collision system should be conserved. This decay process is called the charged current Drell-Yan process and is the same neutrino model we will look at as an article by Pascoli et al. The ISS mechanism and the CCDY process give an almost conserved lepton number, and we only consider electrons and muons in this thesis. This means that the amount of opposite-sign and same-sign events for vertex 1 and 2 may differ from the normal seesaw model. This allows for lepton flavor violation between vertex 1 and 2, with an electron and a muon. In the thesis we study two types of simulated neutrino signals where the mass of the heavy neutrino from vertex 1 are 150 GeV and 450 GeV. We expect different LFV for different neutrino mass models. The models can then be used in the future to identify which neutrino mass model is applicable to a case.

It is not always trivial to identify the detected leptons in p-p collisions. Many events of p-p collisions can be simulated computationally and various final state particle properties can be measured, like their momentum, trans-

verse momentum and coordinate angles. These are used to produced new properties connecting the particles, like the angle between two final state particles, the angular distance between them and the invariant mass of pairs of particles.**Next page**

Proton-Proton Collision Data: Data in this thesis consists of Monte Carlo simulated backgrounds and simulated neutrino signals as well as data from p-p collisions at  $\sqrt{s}=13$  TeV produced from the LHC from 2018. The produced data (left side of the figure) are not interesting to us in this thesis, maybe in the future. Only the MC and signals (right side) are used with ML. The signal simulations used to train ML have only gone through the Generation step with all the truth properties of the particles available to us. Here, among others the interactions of quarks and gluons and subsequent decays are simulated for each event.

The following steps are for simulating particle interactions with detectors, converting the simulations into similar to real raw data as histograms and reducing the amount of data. The MC backgrounds are made to best represent all possible production-mechanisms that may give a three lepton final state with a given transverse momentum plus MET. The (reconstructed) simulated MC and signals to be classified with the fully trained ML model(s) are from after the Analysis step with the desired particle physics model.

Machine learning process: Machine learning algorithms can be trained to identify the particles vertices by using pattern recognition in these particle properties. The data to be trained on are truth simulated data where we know the origins and what particles have been produced. The outcome is to train various machine learning algorithms and use the best performing model to predict similar data of simulated backgrounds and signals that have gone through the full data flow (Analysis). Want to identify lepton vertex permutations in the simulated backgrounds and signals. Since we have three vertices with particles in each event we have six different lepton vertex permutations we could get. We have a multiclass classification case with six classes. Multiclass classification have not been studied much previously in particle physics.

How: Use supervised learning and multiclass classification. Train and tune ML algorithms. Evaluate which model performs best, predicts the vertices the best. Predict lepton vertices for simulated backgrounds and signals.

Need good and fast algorithms for classification since we have a lot of data to work with. Need preprocessing of data before classification. Train models on training set. Many hyperparameters to optimize on validation set. Use several evaluation metrics for performance evaluation for both validation and test set. Export best model to predict vertices for backgrounds and signals, skips the need for training the models each time.

End goal: Apply cuts to study some chosen particle features in signals regions to look for LFV between lepton 1 and 2. Compare with a more standard analysis by Pascoli et al.

Preprocessing of Data: Before we start the training of the models, we have to do some preprocessing of the data. First we take a look at the correlations between the features. We do not want any strong correlations/relations between our features, corresponding to a linearly dependence between the features, since they will both have the same effect on the classes. If two features have a strong correlation, dropping one of them should, theoretically, give better results. With the information gain we want higher values between the features and the classes, meaning the features have more information regarding to the classes.

The data sets we have used are imbalanced, meaning we have more events for some of the classes than others. This may lead to some classes being biased, and bad predictions of the minority classes. That is why we use two types of resampling methods to balance out the data sets in number of events for each class.

Then we split the data sets into training, validation and test sets with different purposes. The training set is the training data the models will train on, the validation set is used to test the trained models and tune the hyperparameters of the models. After the models are properly tuned, they are used on the test set that have not been used until now.

Lastly we scale the features, and not the classes, to avoid any weighted favoring of some features. The scaling technique used is standardization with a 0 mean and a standard deviation of 1.

The tuning of the trained models are done with a randomized search combined with cross-validation to find the combinations of the chosen hyperparameters to each model that produce the highest accuracy.

Bias-Variance Tradeoff: One of the problems with supervised learning is the balance between variance and bias= the bias-variance tradeoff. We want the best compromise between variance and bias that gives the best model when the model complexity/number of data points increases. Often we get the case where high bias with low variance (underfitting), or low bias and high variance (overfitting). Overfitting with models are more the case today with improved machinery and algorithms=more complex models. The optimal case of model complexity, as seen in the figure, is at the minimum of the test sample where the predicted error difference between the samples are sufficiently small. This measures the quality of a model by how well it does on data not seen during training.

Page 8 - Classification Algorithms: Test different types of classification models for multiclass classification. The first for types are binary classifiers

that can be expanded to multiclass classification.

Logistic regression as a more simple classification case with linear regression and a logistic function to predict classes.

MLP is a neural network model with an input, several hidden and an output layer. The inputs are transformed in the hidden layers with some weights, biases and a non-linear activation function. Lastly transformed to outputs. Normally have a lot of hyperparameters with regularization terms to control overfitting.

Trees: A simpler single tree model constructing tree-like models with the features with simpler decisions. Use some criterion to value splits for the features leading to the outcome classes (leaves). Control with hyperparameters to avoid overfitting. A tree ensemble with many trees are also used to improve accuracy and overfitting by decreasing variance.

Booster: Booster methods uses weights from each iteration and builds base estimators sequentially and are combined into a better estimator model. The simplest we use is the AdaBoost with adaptive boosting of a weaker classifier. For each iteration, the AdaBoost changes the weights of the classifier to adjust for incorrect classification until the number of iterations are reached, and a better model have been made from the majority vote of the previous combinations. Gradient boost is a tree boosting method similar to the AdaBoost, but uses weights for gradients in the loss function giving more optimized fitting. With our large data sets, we use a histogram based gradient boost to decrease computation time and increase accuracy by using bins with integer-valued inputs. The Extreme gradient boosting (XGB) model uses an optimized histogram distributed gradient boosting algorithm for accurate and fast parallel tree boosting. Its scalability makes it much faster than many other classifiers. Uses same framework as GradientBoost, but with distributions of features to make branches. This is a very complex algorithm with many hyperparameters to consider and tune. The Light Gradient Boosting Machine is a distributed gradient boosting framework made to be much faster, better memory efficient and accurate, which is an advantage for large data sets. The LGBM uses information gain to drop features below a threshold to improve accuracy.

Multiclass: We also test some multiclassifiers that transforms the multiclass case down to several binary problems with different methods.

Classification Results: Results of the models with the highest accuracy for both signal types of trained models with validation set after finished tuning. Small tendency to overfit with train bigger, less diff with 450 GeV. LGBM and XGB best with LGBM barely better, but also much faster. The reason for choosing LGBM further.

Light Gradient Boosting Machine: Further test the LGBM algorithm on

the test set without doing anything to the hyperparameters for that model. Test with more evaluation metrics this time to further check the model performance. Look at the confusion matrix for each neutrino signal. Both have the diagonal values between 0.8 and 1.0, which indicates that the model can predict the classes quite good. It is better with the 450 GeV trained model than the 150 GeV. The 132, 231, 312 and 321 seems to be predicted better than the 123 and 213 classes for both models.

Scores: 1) Accuracy scores, Cohen Kappa score and log loss for both signals with test set for LGBM. Similar results for validation and test for accuracy, showing the model give similar results for different data (though both are from the same original data set). 450 GeV still better for predicting higher accuracy and CKS and lower log loss.

2) Computed both ROC and precision-recall curves to further show the prediction performance of the LGBM model on the test set by looking at the area under the curve scores, see bottom Table. A good classifier would normally have an AUC larger than 0.8, and both signals have micro and macro AUC scores basically at 1.0. For the precision-recall we have both micro AUC above 0.9, showing the LGBM is a good model for predicting the vertex permutations on the test set.

Classify Simulated Data: The LGBM classifier seems to be performing well on the neutrino signals with the truth information available. The trained LGBM models are then used to classify similar datasets for simulated background production-mechanisms that may have give a three lepton final state with a given transverse momentum plus MET and two reconstructed neutrino signals with the same neutrino masses as we have already trained on, 150 and 450 GeV.

As we would expect, we have most classified particle vertices where the lepton with the highest  $p_T$  comes from the  $N_1$  production vertex, since we would expect the subsequent decaying particles to loose momentum. This happens for both the backgrounds and signals where the most predicted vertices are 123 and 132. For the backgrounds we also get some or at least enough predictions of the 213 vertex that makes sense to use it further.

Predicted Signal Vertices: After the thesis was done, a closer check on the signal predictions was done. The vertex predictions are seen in this table. The 150 GeV signal is predicted with the 150 signal trained LGBM classifier, and the 450 GeV signal is predicted with the 450 signal trained LGBM classifier. We see the number of events for each vertex of the two signals and the fractions of the total number of events. On the left we see the original truth data we trained the classifiers on. The middle shows the number of events for the reconstructed signals we classify, like the backgrounds. The right shows the number of events for the predicted vertices.

Thought it could be interesting to mention this since we get a few differences. The truth (after Generation step) and reconstructed 150 GeV signal has similar fractions for the vertices, while the 450 GeV does not. A lot of events from the truth disappears after reconstruction. The classified vertices only has events for the 123 and 132 vertices, except for 1 213 with the 450 GeV signal. So the predictions does not seem to fit very well with the truth of the reconstructed signals, which might come from that the classifier is trained on the truth signals right after the Generation step in the data flow simulation, but I am not sure if this is the case or not. Might need further analysis to find the solution to this.

Signal Regions: Study different signal regions to look for differences between same flavor and different flavor (lepton flavor violation) for the leptons classified as leptons from vertex 1 and 2 for the three vertices with predicted events 123, 132 and 213.

Compare with a more standard analysis at  $\sqrt{s}=14$  TeV by Pascoli et al., cuts seen in the table. Missing one cut for a  $p_T^{b\text{-tagged}}$  variable we do not have available to us in the datasets.

Analysis Results: Look at the invariant mass of the three lepton system and the MET for both signal models. We look at distributions of the backgrounds and recon signal events and the significance of the signals quantifying the separation between the backgrounds and the signals. A high significance means we have good sensitivity, and is used to show where to cut in the variable to maximize the significance.

Comparing the number of events with these cuts applied to earlier, we now have much less events for each combination of cuts used. We also have more events in the cuts with SF between lepton 1 and 2 compared to DF for each vertex cut. The difference for SF vs DF mainly comes from Z+jets with the most number of events of the backgrounds.  $Z$  can not decay into electron-muon events and thus reduces the large backgrounds (DF) such as  $WZ$  and  $Z$ +jets. For the simulated signals we have almost the same number of events for SF and DF

For the 213 vertex cuts we do not get any significance since we do not have any signal events predicted with this vertex. There are also very few background events with this cut compared to the other two vertices, 123 and 132. Not much information is gained by looking at the 213 vertex cut distributions for either features.

For the 450 GeV sim signals we found it easier to differentiate against the backgrounds for higher masses than 400-500 GeV for invariant three lepton mass for all cuts (excluding 213 vertex), including the benchmark cut. This is also somewhat visible for the MET with DF between 100 and 400 GeV.

The significance was found to be higher with the 450 GeV compared to the

150 GeV signal, and higher in general for the three lepton system invariant mass compared to the MET. The standard analysis gave similar results for signal significance for the SF, OS and 123 vertex cuts.

Summary and Outlook: We have shown that we can use multiclass classification and ML to predict and classify from where simulations of subsequent decay leptons come from in the decay process from two protons colliding, yielding a specific final state model with three leptons and a neutrino/MET.

Training the models on proper data is a big challenge to not get overfitting and bad predictions. That is why preprocessing of the data is necessary to ensure good datasets, and hyperparameter tuning is important for model performance.

There is a difference between the number of events regarding flavor (SF and DF) for lepton 1 and 2 in the vertices we have predicted, yielding high significance in some signal regions.

We found that our trained ML models performed better than the simple benchmark analysis. The benchmark model was however not optimized on the signal models under study.

Future: Modern ML techniques are rapidly modified and developed in the field of particle physics and high energy physics analysis. Other and maybe better suited ML models for such analysis we have done, may be further tested. Other particle physics aspects are also under consideration of the future regarding other neutrino mass signals, look at same sign versus opposite sign or include the detector data of real collisions at CERN.