LIP-STUDENTS-22-30

Distinguishing decay processes for Heavy Neutral Leptons with Machine Learning at SHIP

Rui Santos[1,a]

[1] *Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal*

Project supervisor: N. Leonardo, G. Soares *January 30, 2023*

Abstract. Despite the success of the Standard Model, it has certain limitations that may be solved by particles belonging to the so called Hidden Sector, among them being Heavy Neutral Leptons. SHIP is an intensity frontier experiment at CERN that aims to explore these feeble interacting particles. Using previously simulated data relative to this experiment, this project intends to develop machine learning algorithms to classify decays of Heavy Neutral Leptons based on their kinematic properties.

KEYWORDS: SHIP, Heavy Neutral Leptons, Hidden Sector, Machine Learning

1 Introduction

Although the Standard Model (SM) gives us a good description of nature, it fails to explain certain known phenomena, such as dark matter, the low masses of neutrinos and the baryon asymmetry of the universe. To solve these problems, it is theorized that there exists more particles than the ones in the SM. There are two main reasons why these particles have not yet been detected. One possibility is that the theorized particles are too massive and so, to produce them, higher energy collisions are required. This is known as the energy frontier, a problem that is being tackled at the LHC. Another possibility is that interactions between these hypothetical particles and the SM particles may be too feeble. To solve this, experiments need to have higher luminosity and account for longer lived particles. This intensity frontier will be explored by the SHIP experiment.

2 The SHIP experiment

The Search for Hidden Particles experiment is a general-purpose fixed-target experiment that will be set at the SPS accelerator at CERN. It aims to explore the intensity frontier in the search for feebly interacting particles (FIPs), with Heavy Neutral Leptons (HNLs) being one of them and the focus of this project. To detect the Standard Model decay products (daughter particles) originated from the decay of FIPs (mother particle) the set-up presented in Fig. 1 will be used.

A high intensity beam of protons (400 GeV) from SPS collides with a target that is followed by a hadron absorber and an active Muon shield, where muons produced in the collision are deflected by strong magnetic fields. After this, there is a scattering and neutrino detector and a muon identification system that works as a veto for the upcoming decay vessel. The Hidden sector particles will decay in the decay vessel, a pyramidal structure with 50 m of length. Products from the interaction of neutrinos along the decay vessel can be mistaken as those produced in the decays of particles of interest, constituting our main background. To reduce the number of neutrino interactions, the decay vessel is maintained at a constant 1mbar pressure.

The decay products are detected in a spectrometer that follows the decay vessel. Through the schematic shown in Fig. 2, we observe that the spectrometer is initially made up of straw trackers, which are separated by a magnet with a defined magnetic field. These straw trackers combined with the electromagnetic calorimeter allow the reconstruction of different kinematic properties of particles such as decay vertex, mass, among others. Between the straw trackers and the calorimeter is the timing detector. The timing detector reconstructs the arrival times of decay products and matches these times, within certain limits, to form vertex candidates which reduces the background. Finally, there is a muon identification system.

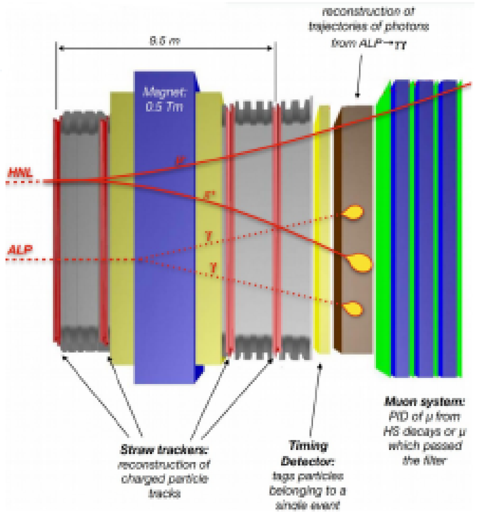
reconstruction of trajectories of photons from ALP→γγ
8.5 m
Magnet: 0.5 Tm
HNL
ALP
μ
π
γ
γ
Straw trackers: reconstruction of charged particle tracks
Timing Detector: tags particles belonging to a single event
Muon system: PID of μ from HS decays or μ which passed the filter

Figure 2: Schematic of the Decay Spectrometer at SHIP, adapted from [1].

[a] e-mail: fc54741@alunos.fc.ul.pt

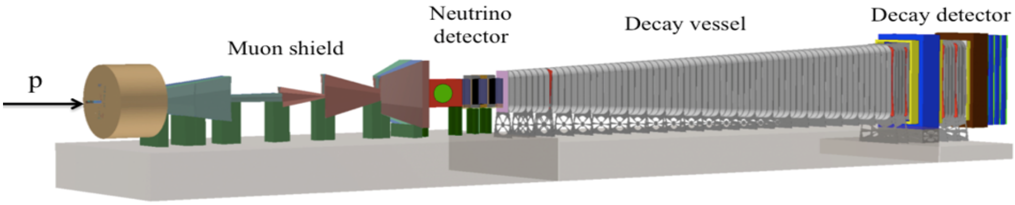

Figure 1: Schematic of SHIP's experimental set-up.

3 Heavy Neutral Leptons and data simulation

First proposed by Minkowsky in 1977, Heavy Neutral Leptons are right-handed massive neutrino-like particles that do not couple to the weak force, and consequently to no force in the SM. Even so, they can couple to the SM through neutrino oscillations. These hypothetical particles could solve a lot of the current problems in the SM. Through seesaw mechanisms they may explain the low masses of SM neutrinos, and through leptogenesis they may explain the Baryonic Asymmetry of the Universe. Other than this, they are also candidates for Dark Matter.

Because data acquisition has not started at SHIP, since it has not already been built, we used simulated data throughout this project. The data was generated using the FairShip software, based on the FairRoot framework. The initial interaction of the proton beam and the target was simulated with Pythia 8, while propagation and interaction of produced particles was done with GEANT4. Neutrino interactions were simulated with GENIE, and heavy flavour production and inelastic muon interactions were done with a combination of Pythia 6 and GEANT4.

The HNL data used in this project is relative to two different decay processes, the decay into a Pion-Muon pair (PiMu) and the decay into a Rho-Muon pair (RhoMu). In the second case, the Rho particle will eventually decay into a charged pion and a neutral pion, the latter one decaying into two photons. HNL data was generated from the parameter spaces shown in Figs. 3 and 4. Finally, as aforementioned, the background in our data corresponds to interactions of neutrinos along the decay vessel.

4 Motivation

Exploration of the intensity frontier requires high intensity beams, which in turn leads to significant production of background. Since the main objective of the SHIP experiment is to detect a total of two to three decays of Hidden Sector particles over the whole duration of the experiment, SHIP needs to be a 0-background experiment. Background is already significantly reduced by the experimental set-up, but even so, correct data analysis needs to be applied to distinguish between signal and background. Another problem that the SHIP experiment may face is that in some cases different decay processes may lead to the same daughter particles being detected. For example, there's a possibility that in the RhoMu decay process the two photons are not detected and, in that case, there is loss in the reconstruction of the energy of the mother particle, and thus the daughter particles detected in the PiMu and RhoMu decay processes will be the same.

To tackle these problems, this project aims to develop Machine Learning algorithms, more precisely binary classifiers, to distinguish between the two decay processes of HNL particles (RhoMu-PiMu) and separately distinguish each signal from background (PiMu-Back, RhoMu-Back). In the following, the first process in the classifications presented before will always be referred to as the first sample, being assigned a target of 1, and the other as the second sample, with target 0.

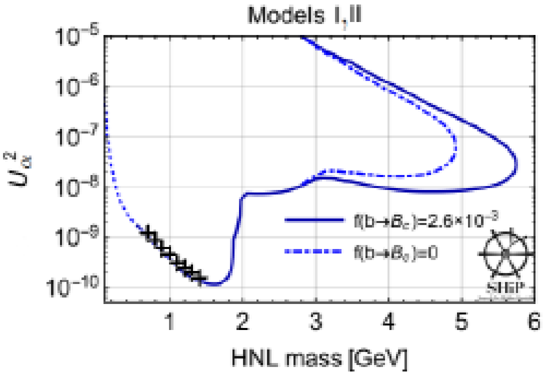

Figure 3: Parameter space for the HNL effective coupling to muon (Model I) and electron neutrinos (Model II), according to Ref. [2].

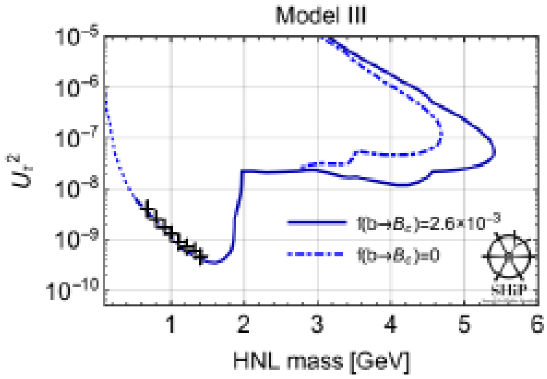

Figure 4: Parameter space for the HNL effective coupling to tau neutrinos (Model III), according to Ref. [2].

5 Machine Learning

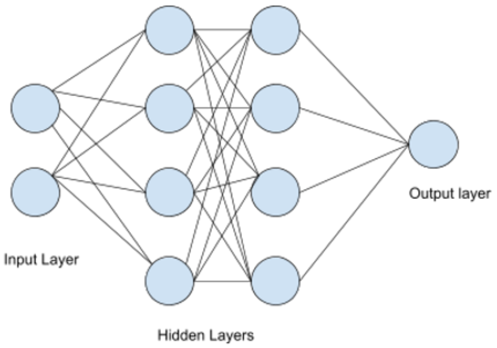

Figure 5: Schematic representation of the typical structure of a Neural Network, in this case with W=4, S=0 and D=2.

The binary classifiers developed in this project were Neural Networks (NN), more precisely Multilayer Perceptrons (MLP). This was done using Keras and TensorFlow, both open-source libraries for Python.

A common NN, as in Fig. 5, starts with an input layer that receives data features and so has as many nodes as there are selected features. The last layer, the output layer, outputs the NN response and is composed, in the case of this project and binary classifiers, by only one node. In between these layers there are hidden layers that will be described by a depth (D), width (W) and step (S), as well as other hyperparameters. In this way, the NN will have D hidden layers, having the first one a total of W nodes and subsequent layers having fewer nodes, reduced by S per layer. Each node in the NN is connected to every other node in adjacent layers, with the connections defined by weights, and each node may have a threshold or bias. The information processed in a node is passed to the next layer with an activation function, that in the case of a MLP is a nonlinear one.

The values of weights and biases are learned in the training phase that, in the case of a MLP, corresponds to supervised learning. Supervised learning means that the data used for training has defined inputs (features) and results (targets). This way, parameters are learned by minimizing the loss function, this is, by minimizing the mean square error between the output and the target over all the data in the training sample. To find the minima and the optimal weights and biases an optimization function is used and the rate at which parameters are updated is referred to as the learning rate. The number of epochs corresponds to the number of times the Neural Network will pass through the whole training dataset, while batch size corresponds to the number of events the NN sees before updating.

Finally, because MLPs are NNs with supervised learning, they will eventually overfit the training data, becoming obsolete when faced with data which they never trained on. To avoid this, a separate data set that is not used for training (validation set) is used for validation. The loss function value for the validation set will decrease in the beginning, but eventually will start increasing. After a number of epochs where the loss value increases (Patience) the training is stopped and the NN parameters are the ones relative to the loss function minima for the validation set. Other than this, it's possible to reduce codependencies between neurons, and so reduce overfitting, with dropout. With this technique, for each batch, there is a set probability of blinding, or dropping, information from the nodes (dropout rate).

6 Feature Selection

The simulated data has a total of 15 variables:

- **Total Momentum** (mother particle and both daughter particles);
- **Transverse momentum** (mother particle and both daughter particles): component of new particle's momentum perpendicular to the beam line;
- **Fraction of transverse momentum** (mother particle and both daughter particles): ratio between transverse and total momentum;
- **Opening angle**: angle between daughter particles momentum.
- **Impact parameter**: Closest distance between the beam intersection with the target and the intersection of the propagation of the mother's particle trajectory with the target;
- **Decay vertex position (x, y, z)**: Space coordinates of the point where the mother particle decayed. Z coordinate is along the decay vessel, while X and Y are orthogonal;
- **DOCA (Distance Of Closest Approach)**: Minimal distance between the tracks of both daughter particles;
- **Invariant mass**: Energy at the centre of mass of the hidden sector particle candidate.

Although we have all these variables, only some will be used in the NN. Firstly, given that the dimension of the input layer is equal to the number of features selected, using fewer variables simplifies our NN and reduces computational cost. Secondly, features that are linearly correlated to each other or have similar distributions for different classes do not greatly impact the training of the NN. Knowing this, feature selection was then done based on how much a feature was linearly correlated to other features and mainly on how discernible were the differences between classes for that feature. Because of this, selection was done based on the visualization of correlation matrices, see Fig. 7 as an example, and normalized plots of all classes for each feature (in Fig. 6 are presented some features of interest for each classification case).

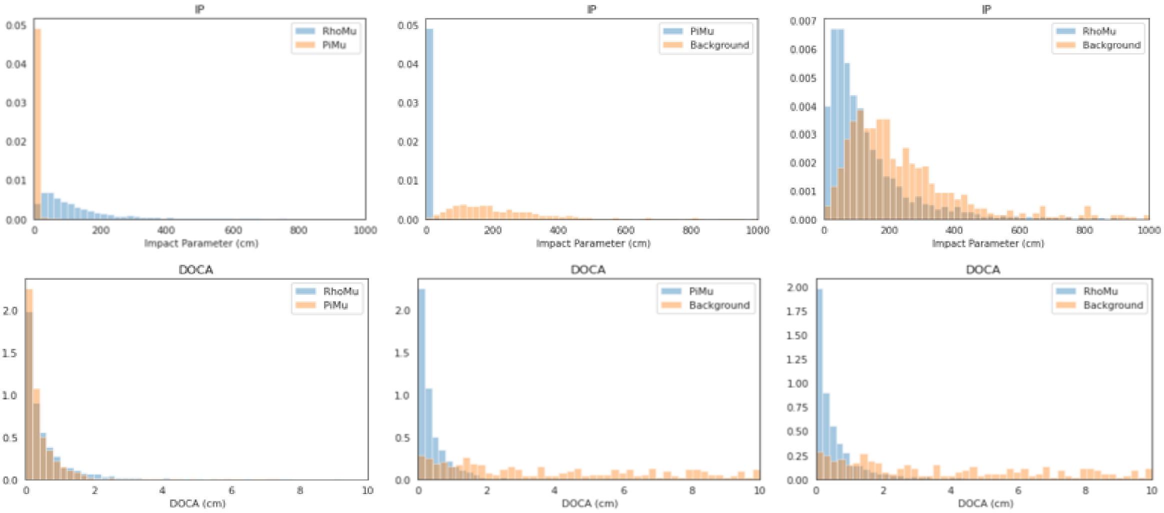

Figure 6: Normalized distributions of Impact Parameter and DOCA for RhoMu-PiMu (left), PiMu-Back (middle) and RhoMu-Back (right) classification cases

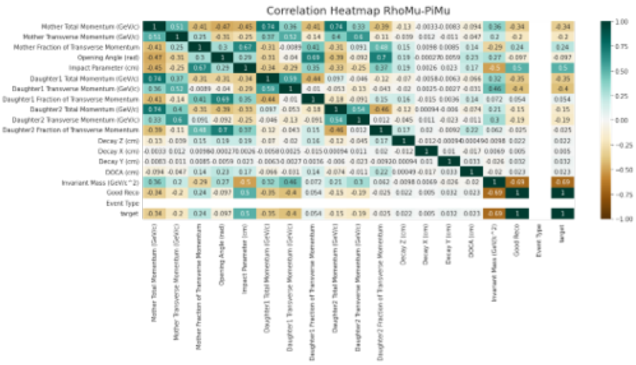

Figure 7: Correlation matrix for the RhoMu-PiMu classification case.

6.1 RhoMu-Pimu

For this case, it's important to remember that in the RhoMu process certain decays cannot be fully reconstructed because photons may not be detected. The error in the reconstruction of the mother particle will be propagated to the target, which explains the differences observed in the Impact Parameter distributions for this classification case. Other than the Impact Parameter, the following features were also chosen: Mother Total Momentum, Daughter1 Transverse Momentum, Daughter2 Transverse Momentum, Daughter1 Total Momentum, Daughter2 Total Momentum.

6.2 PiMu-Back

In this case, in addition to all the features used in the previous case, Decay Z and DOCA were also used. The background originates from violent interactions, so energy loss should occur, which justifies the differences in Impact Parameter distributions for PiMu and Background. In addition, DOCA distributions are also quite distinct, since, due to energy losses, the reconstructions of the daughter particle trajectories do not lead to such exact vertexes. On the other hand, Decay Z becomes a relevant parameter because the decay vessel is in a vacuum, and therefore most neutrino interactions will occur at its beginning. In contrast, hidden sector particles, having longer lifetimes, will have a higher probability of having their decay vertices to be along the decay vessel.

6.3 RhoMu-Back

The parameters used in this case are the same as in the previous case. Even so, it should be noted that for Impact Parameter the differences between RhoMu and Background are not as pronounced as in the case of PiMu-Back, something that would be expected since this parameter was already relevant for the RhoMu-PiMu case. The justification for this difference not being so great is the same as presented in Sec. 6.1.

7 Pre-Processing

Firstly, due to the losses in the energy reconstruction of the mother particle in the RhoMu process, in order to avoid some form of bias, only events with a reconstructed invariant mass below 1,2 GeV/C^2 were used. On the other hand, the chosen features have different units and some have much higher ranges than others. As such, in order to prevent the neural network from giving more relevance to some features than others due to these aspects, the data was normalized and standardized to mean 0 and standard deviation of 1 and Principal Component Analysis (PCA) was applied to change the parameter basis and so decorrelate the parameters as much as possible. Then the data was divided into development and validation sets. Since for the PiMu-Back and RhoMu-Back cases the number of background events was much smaller than the ones for the signal, and as in the RhoMu-PiMu case the PiMu events are a little smaller, the split was 70-30 for the first sample and $70 \times backratio-(100-70 \times backratio)$ for the second, where *backratio* corresponds to the ratio of the dimension of the first sample relative to the dimension of the second sample. After that, the development set was divided into training and test sets with a 70-30 split.

8 Hyperparameters

After trial and error, the hyperparameters that led to the best results were the following:

Table 1: List of hyperparameters that led to the best results

Hyperparameter	RhoMu- -PiMu	PiMu- -Back	RhoMu- -Back
Depth	2	2	5
Width	30	10	60
Step	10	0	10
Activation function	Hidden Layers- Relu Output layer- Sigmoid		
Batch size	100	30	30
Optimizer	ADAM		
Number of epochs	2000		
Patience	500		
Dropout rate	0,05		
Learning rate	0,001		

The differences in batch-sizes are due to the pre-processing of the data. As explained in Sec. 7, given that the background had a much smaller number of events, the way the data pre-processing was done left the training sets with fewer data than in the RhoMu-PiMu case. Therefore, it seemed sensible to also reduce the batch-size.

9 Results

Firstly, as shown in Fig. 9, the loss function evolves as explained in Sec. 5.

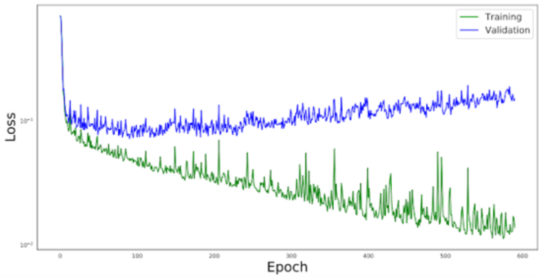

Figure 9: Evolution of the loss function for the training and validation sets, relative to the RhoMu-PiMu classification case

In order to evaluate the performance of the NN, ROC curves and Confusion Matrices were used. Receiver Operator Characteristic (ROC) curve corresponds to a plot of the true positive rate, the ratio of correct positive predictions (true positive-TP) against all positive events (sum of TP with false negative-FN), against the false positive rate, the ratio of correct negative predictions (true negative-TN) against all negative events (sum of TN with false positive-FP), for different classification thresholds. In order to compare different classifiers, the Area Under the Curve (AUC) was used, which provides us with a measure of the performance of the NN for different classification thresholds. In a way, this area indicates the probability of the model attributing a greater output to a positive event than to a negative event. Therefore, the closer the AUC is to 1, the greater the tendency for correct predictions. As such, what we want is AUCs close to 1 and ROC curves that look squarer, this is, that come closer to the top left corner of the graph. On the other hand, in the case of a confusion matrix, the classification threshold is defined as 0,5 and then TP, TN, FP and FN are counted based on this threshold and the result is presented in the form of a matrix, as shown in Fig. 8. Finally, given that SHiP should be a zero-background experiment, then for the PiMu-Back and RhoMu-Back classifications, the classification threshold that would lead to such a situation was also sought. For this, the threshold was defined as the largest output that the model gave to a background event. In this way, the background is reduced to zero, but certain signal events are classified as background, as shown in Table 2.

Table 2: Results relative to the 0-background study.

	PiMu-Back	RhoMu-Back
TP	135	80
TN	138	138
FP	0	0
FN	3	58
Selection efficiency	0,978	0,580

10 Conclusions

From the results presented in Fig. 8, it is possible to conclude that machine learning was able to provide very satisfactory results, only getting sub-optimal results in the case of the RhoMu-Back classification. In the case of the last method presented in the results, we observed that, as expected, we obtained 0-background, but at the expense of some signal being classified as background. This can be worrying in the case of an experiment where only two to three decays are expected to be detected, as is the case of SHIP. Furthermore, in this 0-background study we observed the same trend that was presented at the beginning of this conclusion, with the selection efficiency being much lower in the RhoMu-Back case. A possible explanation for the results not being so good in this last classification case could be missing energy in the reconstructions that happens in the RhoMu decay process, which led to the smearing of certain features, such as the Impact Parameter (as was presented in Sec. 6.1 and 6.3).

Despite the effects that the lack of efficiency in the reconstruction of the RhoMu decay had in the previous

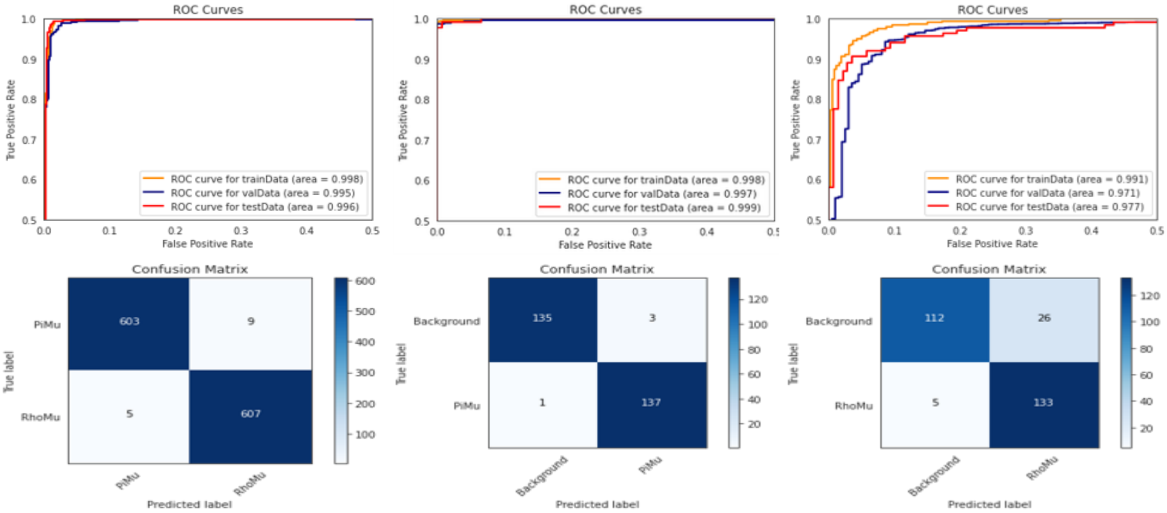

Figure 8: ROC curves and confusion matrices relative to RhoMu-PiMu (left), PiMu-Back (midle) and RhoMu-Back (right) classification cases.

analysis, it did not prevent the distinction between the two decay modes (RhoMu-PiMu), since the results are quite satisfactory. By observing the confusion matrix for this classification, we can see that there is a slight bias to classify the event as coming from a decay to RHO mesons. This way, in one hand, the developed algorithm was able to solve one of the problems presented in Sec. 4. On the other hand, this can reinforce the idea that the Impact Parameter corresponds to a very relevant parameter for the performance of the classifier, given that in this case, contrary to what happened in RhoMu-Back, there are discernible differences between classes for this feature.

However, certain changes and progress still have to be done. Firstly, it would be interesting to develop a multi-classifier that performs all classifications simultaneously and compare its performance with that of the binary classifiers developed throughout this project. Finally, it is still necessary to apply the developed methods to real data taken from experiment and to study the problems and adversities that could arise.

Even so, we can say that Machine Learning algorithms correspond to very promising tools in the exploration of New Physics in Particle Physics, and show an interesting potential to be applied in the SHIP experiment.

Acknowledgements

I would like to thank LIP for the opportunity to participate in this internship, which not only allowed me to stimulate my curiosity about particle physics, but also allowed me to develop and deepen my knowledge in physics and related areas, such as machine learning. I would like to thank my supervisors in particular, as without them the path taken would have been much harder, and for the support, patience, knowledge and experience they have given me throughout this project.

References

[1] C. Franco, A. Blanco, N. Leonardo, P. Bordalo, S. Ramos, G. Soares, B. Araújo, M. Pinto, *The SHiP experiment at CERN and the role of the LIP group within the collaboration* (2020), [Conference Session]

[2] SHiP Collaboration, *Sensitivity of the SHiP experiment to heavy neutral leptons*, https://arxiv.org/abs/1811.00930 (2018)

[3] H. Santos, A. Branco, F. Safara, R. Santos, *Student reports*, LIP STUDENTS/21-06, STUDENTS-20-17

[4] M. Cristinziani, *The SHiP experiment at CERN*, https://arxiv.org/abs/2009.06003 (2020)

[5] SHiP Collaboration, *SHiP Experiment - Comprehensive Design Study report*, https://cds.cern.ch/record/2704147 (2019)

[6] P. Bhat, *Advanced analysis methods in particle physics*, https://www.annualreviews.org/doi/10.1146/annurev.nucl.012809.104427

[7] G. Soares, *Optimization of the selection of hidden particles in the SHiP experiment*, https://cds.cern.ch/record/2765979 (2021), CERN-THESIS-2021-038