Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Application of Machine Learning for the Charged Higgs Boson Search Using ATLAS Data

Jiří Pospíšil

Supervisor: doc. Dr. Andre Sopczak
May 2022

Acknowledgements

I would like to thank my supervisor, doc. Dr. André Sopczak for his expert advice and support during the work. Also thanks to Prof. Dr. Ing. Jan Kybic for his valuable feedback and ideas for improvement. And above all, I would like to thank my family for their trust and support throughout my studies.

Declaration

I declare that the presented work was made independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of the university thesis.

In Prague, date ....

Signature ....

Abstract

The discovery of the Higgs boson (2012) motivated scientists searching for charged Higgs bosons. The presence of a charged Higgs boson is predicted by many theories that describe an extended Standard Model, with several different Higgs bosons, called the "extended Higgs sector". Neural networks (NN) have recently been a big trend for solving classification, detection, and segmentation tasks. The advantage of NN is their ability to learn complex relationships hidden in data without any restrictions on the input data. The aim of this thesis is to separate the Signal process tbH^+ from the Background processes. In this thesis, two NN architectures were tested: Multi-Layer Perceptron (MLP) and TabNet. A good separation of Signal and Background was obtained as a function of the charged Higgs boson mass.

Keywords: ATLAS, CERN, classification, cross section, machine learning, neural networks, PyTorch, particle physics, ROOT, tbH^+

Supervisor: doc. Dr. Andre Sopczak

Abstrakt

Objevení Higgsova bosonu (2012) motivovalo vědce k hledání nabitého Higgsova bosonu. Přítomnost nabitého Higgsova bosonu je předpovídána mnoha teoriemi, které popisují rozšířený Standardní Model s několika různými Higgsovými bosony, nazývaný „rozšířený Higgsův sektor“. Neuronové sítě (NN) jsou v poslední době velkým trendem pro řešení klasifikačních, detekčních a segmentačních úloh. Výhodou NN je jejich schopnost naučit se složité vztahy skryté v datech bez jakýchkoli omezení vstupních dat. Cílem této práce je oddělit proces Signal tbH^+ od Background procesů. V Práci byly otestovány dvě NN architektury: vícevrstvý perceptron (MLP) a TabNet. Bylo dosaženo dobré separace Signálu od Background, jako funkce hmotnosti nabitého Higgsova bosonu.

Klíčová slova: ATLAS, CERN, klasifikace, cross section, strojové učení, neuronové sítě, PyTorch, částicová fyzika, ROOT, tbH^+

Překlad názvu: Aplikace strojového učení pro hledání nabitého Higgsova bosonu z ATLAS dat

Contents

Figures

Tables

Chapter 1

Introduction

This thesis focuses on NN techniques for the classification task. For comparison of applied methods, several approximations of significance were tested. NN output is then used in Trex-fitter [Tf] to estimate the Signal cross section for a 95% CL detection sensitivity.

The thesis first describes particle physics as a short introduction to the problem and then discusses neural network principles used techniques. This is followed by the results with a detailed explanation of all experiments. The conclusion follows with a summary and an outlook.

Chapter 2

Particle physics

2.1 Introduction

The Standard Model (SM) of particle physics describes a relationship between particles and the three fundamental forces (electromagnetic, weak, and strong interactions). Nevertheless, some things remain unexplained. The SM omits the fourth fundamental force (gravity) and does not explain dark matter. Last but not least is the gauge hierarchy problem, which is associated with the presence of elementary scalars (Higgs) in the SM [Csa96]. New theories that extend the SM try to explain these mysteries. The two doublet Higgs Model (2DHM) predicts five physical Higgs bosons [EP16]:

- two neutral CP-even Higgs bosons: h (SM Higgs) and H (heavy Higgs),
- two charged Higgs bosons $H^{\pm}$,
- one neutral CP-odd Higgs boson A.

The Minimal Supersymmetric Standard Model (MSSM) is a type II 2DHM theory. Supersymmetry theory explains presence of light Higgs boson by predicting new particles that would cancel out the contributions to the Higgs mass from their Standard Model partners. In addition, Supersymmetry predicts a partner particle with a spin that differs by half a unit for each of the particles in the standard [CER], in other words, Supersymmetry links fermions and bosons together.

2.2 Signal

Every particle has a property called spin. In the Standard model, for spin-1 and spin-1/2, particles have charged and neutral states [Col21]. The neutral Higgs boson is the first know particle with spin-0, discovered in 2012. Since a charged particle with spin-0 has not yet been discovered, a charged Higgs boson could complete this table of charged and neutral pairs.

In proton-proton collisions, charged Higgs boson can be produced associated with the top and bottom quarks. The decay of the charged Higgs boson depends mainly on its mass. In this analysis, the search of charged Higgs boson focuses on decay channel $H^+ \to hW$, specifically the channel 2lSS1tau, which requires the occurrence of two leptons of the same sign and one hadronically decaying tau. The decay channel of interest is in particle physics called signal, while other processes are called background. The Feynman diagram for tbH^+ decay is shown in Figure 2.1. Other important decays, especially in the low mass region, are $H^+ \to cb$ and $H^+ \to cs$ and the most dominant decay channels in the high mass region are $H^+ \to \tau\mu$ and $H^+ \to tb$ [EP16].

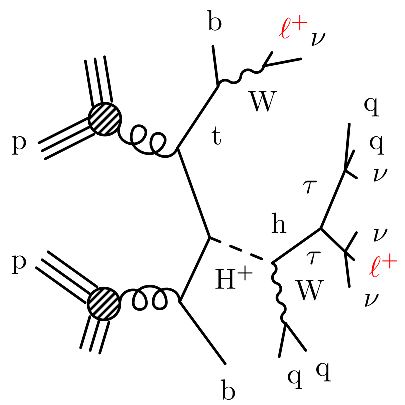


Figure 2.1: tbH^+ Feynman diagram, leading to the 2lSS1tau final state.

2.3 Background

Unfortunately, detectors cannot capture complete process information. For example, neutrinos are undetectable. Higgs boson as a particle transforms into lighter particles almost immediately after being produced in proton-proton collisions [CER20]. The ATLAS and CMS detectors can detect these lighter particles. However, the final state of many processes can be the same, making them difficult to distinguish.

To simulate representative conditions for classification several backgrounds need to be considered. This analysis considers $t\bar{t}h$, $t\bar{t}W$, $t\bar{t}Z$, $t\bar{t}$, VV, and Others processes as the Background.

The charged Higgs decays into neutral Higgs. Therefore, $t\bar{t}h$ is one of the most similar processes to tbH^+. Despite the rare production of $t\bar{t}h$, about 1% of all neutral Higgs production, tbH^+ is expected to be less frequent. The similarities of $t\bar{t}h$ and tbH^+ Feynman diagrams are shown in Figures 2.1 and 2.2.

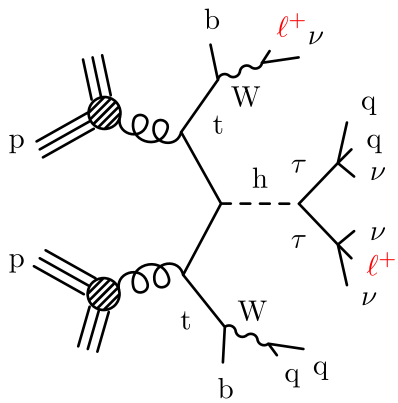
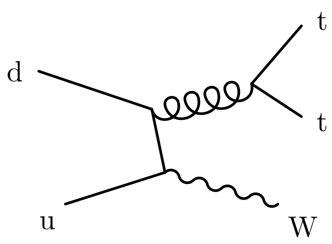


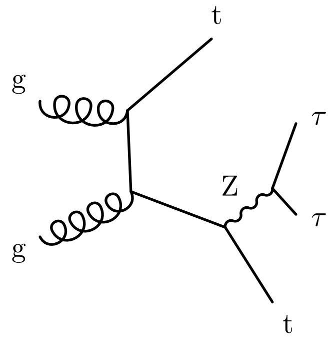
Figure 2.2: $t\bar{t}h$ Feynman diagram.

Another decay channel leading to the 2lSS1tau final state is $t\bar{t}W$. The W boson decays either leptonically (into one of the three charged lepton and a neutrino $W \rightarrow \ell^+\mu$) or hadronically (into quark-antiquark pair $W \rightarrow q\bar{q}$). The $t\bar{t}W$ Feynman diagram is shown in Figure 2.3a.

The Z boson decays in three different ways. The decay into charged lepton-antilepton pairs (electron-positron, muon-antimuon, and tau-antitau pairs) leads to the same final state as tbH^+. However, Z boson decaying to neutrino-antineutrino pair or a quark-antiquark pair[Col] is more common. Figure 2.3b shows the Feynman diagram of Z boson decaying to pair of tau-antitau.



(a) : $t\bar{t}W$.



(b) : $t\bar{t}Z \rightarrow t\bar{t}\tau\bar{\tau}$.

Figure 2.3: $t\bar{t}W$ and $t\bar{t}Z \rightarrow t\bar{t}\tau\bar{\tau}$ Feynman diagrams.

2.4 Simulated Data

2.4.1 Ntuple format

The Provided simulated data are stored in ROOT n-tuples format (*.root*). ROOT is a data processing framework developed by CERN that enables fast and efficient work with large files [ROO]. The provided data were generated by the program PYTHIA, which is a general-purpose Monte Carlo event generator [PYT]. In Ntuple creation, Simulated data are preprocessed, and some of the events are removed. After that, a preselection filter is applied to keep only events with 2lSS1tau final state. The simulation is always based on a particular detector configuration to simulate events precisely. The Background data were simulated for all three settings: mc16a, mc16d, and mc16e corresponding to 2015/2016, 2017, and 2018 recorded data. The summary of Background data is given in Table 2.1. However, Signal events were simulated only for the latest version. Because the H^+ mass is unknown, the data were produced for four discrete mass points. The requested masses with the number of simulated events are shown in Table 2.2.

Each file contains the events of only one process, but several files can belong to the same process. One can distinguish files by their names, which uniquely identify the process they contain. Table 2.3 shows the list of the file names for the given processes. ROOT files have a tree structure with 790 features that characterize an event. Some are low-level particle properties, such as mass, energy, momentum, or more complex precomputed high-level features. The files also include supporting information for the simulation that is not useful for training NN and the true information, which is used as input for

Process	**Ntuple**	**Preselected**	**Weight**
ttH	6 060 601	29 670	22.843
ttZ	8 592 621	27 700	18.420
ttW	2 010 387	10 294	24.857
tt	19 677 014	186	22.166
VV	59 206 801	2 306	6.291
Others	1 781 273	3 192	11.471
Total	97 328 697	73 348	106.049

Table 2.1: Overview of Background events. **Ntuple** - number of events after prepossessing and ntuple creation, **Preselected** - number of events after selecting the 2lSS1tau channel, **Weight** is expected number of events.

Mass [GeV]	**Created**	**Ntuple**	**Preselected**	**Ratio [‰]**
300	1 200 000	252 981	4 401	3.668
800	800 000	198 834	5 334	6.668
1500	600 000	138 866	3 283	5.472
2000	400 000	82 449	1 339	3.348
Total	3 000 000	673 130	14 357	4.786

Table 2.2: Overview of Signal events counts. **Created** - number of events originally produced by the simulation, **Ntuple** - number of events after prepossessing and ntuple creation, **Preselected** - number of events after selecting the 2lSS1tau channel, **Ratio = Preselected / Created** is the preselection efficiency.

the simulation and cannot be used to train with it.

Process	**File IDs**
tbH^{+}	510374, 510375, 510376, 510377
ttH	346343, 346344, 346345
tt	410470
ttW	700168
ttZ	413023
VV	364250, 364253, 364254, 364255, 364283, 364284, 364285, 364286, 364287, 363355, 363356, 363357, 363358, 363359, 363360, 363489
Others	410397, 410398, 410399, 410408, 410560, 410080, 410081, 304014, 342284, 342285, 364242, 364243, 364244, 364245, 364246, 364247, 364248, 364249

Table 2.3: The list of file data set identifiers for each process.

2.4.2 Event weight

The probability of an event's presence in the detector represents the event's weight which is calculated as:

$$w = \frac{\prod_{i=0} w_i}{w_t}, \tag{2.1}$$

$$w_0 = \begin{cases} 58450.1, & \text{if RunYear} = 2018 \\ 44307.4, & \text{if RunYear} = 2017 \\ 36207.66 & \text{otherwise} \end{cases} \tag{2.2}$$

where w_0 is luminosity scaling for the different detector configurations, and w_i is the event value for each feature listed inTable 2.4.

Variable	Name
w_1	custTrigSF_LooseID_FCLooseIso_DLT
w_2	weight_pileup
w_3	jvtSF_customOR
w_4	bTagSF_weight_DL1r_70
w_5	weight_mc
w_6	xs
w_7	lep_SF_CombinedTight_0
w_8	lep_SF_CombinedTight_1
w_t	totalEventsWeighted

Table 2.4: List of events parameters used in weight formula.

A few issues need to be mentioned. The theoretical cross section σ_{H^+} of the charged Higgs production is not known, and therefore the Signal weights serve only as an initial estimate and cannot be directly compared with the Background weights. The weight formula from the Eq. 2.1 assumes that data are generated for all three configurations. With an assumption that the Signal events will be generated similarly for two other detector configurations, Signal weights are scaled as if all three parts were simulated. Table 2.5 summarizes the Signal weights. The last problem is that the Monte Carlo simulation often creates events with negative weight due to features weight_pileup and weight_mc, which occasionally have a negative value. These events cannot be easily deleted because they are already used in the preprocessing part when creating Ntuples. However, negative weighted events would cause problems in training, and therefore such events are excluded from the training phase but are included in the validation phase.

Mass [GeV]	Original	Normalised	Scaled
300	$4.662 \cdot 10^{-2}$	204.24	485.57
800	$3.004 \cdot 10^{-3}$	351.89	836.61
1500	$1.039 \cdot 10^{-4}$	300.46	714.34
2000	$1.087 \cdot 10^{-5}$	204.38	485.91
Total	$4.974 \cdot 10^{-2}$	1060.96	2522.42

Table 2.5: Overview of Signal events weights, **Original** - initial weights estimate, **Normalised** - weights are computed with cross section of one picobarn, **Scaled** - normalised weights additionally scaled to compensate for the missing Signal mc16a, mc16d data sets.

2.5 Significance

In addition to traditional machine learning metrics used in classification tasks such as accuracy, ROC curve, F1 score, recall, or precision, statistical significance is often used in particle physics. The concept of statistical significance is based on hypothesis testing. Hypothesis testing is a method for comparing null hypothesis H_0 and alternative hypothesis H_1. In particle physics, the Standard Model is considered a null hypothesis, and in our case, the alternate hypothesis predicts the presence of charged Higgs boson.

To establish a discovery [Gro18] defines statistical test q_0 as follows:

$$q_0 = \begin{cases} -2\frac{L(0,\ \hat{\hat{\theta}})}{L(\hat{\mu},\ \hat{\theta})} & \text{if } \hat{\mu} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where L is profile like-hood function, $\hat{\mu}$ is the parameter of interest, $\hat{\hat{\theta}}$ and $\hat{\theta}$ represent the nuisance parameters. Significance Z and p-value p_0 for a one-sided test are derived from 2.3 as:

$$Z = \Phi^{-1}(1 - p_0) = \sqrt{q_0}, \qquad p_0 = 1 - \Phi(\sqrt{q_0}). \tag{2.4}$$

A hypothesis is excluded if the p_0 is below the threshold α. The α determines the quality and consistency of the test. Therefore a small α should be chosen as it corresponds to the probability that the null hypothesis is correct and the observation is an arbitrary fluctuation [Sin02]. In particle physics, a significance of 5σ is needed to confirm the discovery of a new particle. Table 2.6 shows the relationship of p-value p_0 and significance Z for a one-sided hypothesis test.

Significance [σ]	1	2	3	4	5
p-value [%]	16	2.3	0.14	$3 \cdot 10^{-5}$	$3 \cdot 10^{-7}$

Table 2.6: Relationship of significance and p-value.

The formula from Eq. 2.3 is complex and assumes that the observations are integers; however, this does not apply to simulated data because event weights are fractions. With an assumption that the observations are from the Poison distribution and they are independent, significance can be approximated as follows:

$$Z_0 = \frac{S}{\sqrt{B}} \tag{2.5}$$

$$Z_1 = \frac{S}{\sqrt{S + B}} \tag{2.6}$$

$$Z_2 = \frac{S}{\sqrt{B} + 1.5} \tag{2.7}$$

$$Z_3 = \sqrt{2 \cdot ((S + B) \cdot \log(1 + \frac{S}{B}) - S)} \tag{2.8}$$

where S is the expected predicted Signal (true positive) and B is the expected misclassified Background (false positive). Not to be confused with the previous definition of Signal, Background, in this section Signal/Background refers to the NN output while previously, Signal denotes the class of interest. Eq. 2.8 is derived as a median approximation of significance Z. Eq. 2.8 can be further simplified to Eq. 2.5 assuming $S \ll B$. Eq. 2.6 and Eq. 2.7 are special modification for a better estimate of significance when the B is very small.

Chapter 3

Neural Networks overview

3.1 Introduction

Machine learning is a set of computational algorithms that can learn the various characteristics of a given data. Neural networks are wildly used for classification, pattern recognition, and segmentation tasks. NN training can be supervised (data true-ground information is included) or unsupervised (where the true-ground information is unknown). This work focuses on supervised binary classification.

The Given data set is divided into training and validation parts. The validation data set serves only for the model's performance evaluation and cannot be used for NN optimization. NN model is trained for a certain number of epochs, in which the model classifies all events in the training data set, and after each batch of events, based on the model output, updates its parameters.

3.2 Cost function

The cost function, which is also referred to as the loss function or simply the loss, is used to measure NN performance, and in the training phase, NN weights are updated to minimize the loss. Several iterative algorithms based on gradient descent can be used to find the local minimum of the cost function. This work used Stochastic gradient descent (SDG) with momentum and an exponentially decaying learning rate. According to [HRS16], SDG can generalize better and is more stable compared to the Adam algorithm, although fine-tuned adaptive methods can converge faster and find a better solution.

Cross-entropy (CE) loss is wildly used in classification tasks. This cost function builds upon the idea of information theory entropy introduced by Claude Shannon in his 1948 paper „A Mathematical Theory of Communication“. For binary classification, the cross-entropy is defined as:

$$\mathrm{CE}(y,\, l) = -l \cdot \log(\sigma(y)) + (l-1) \cdot \log(1-\sigma(y)), \tag{3.1}$$

where $l \in \{0,1\}$ is event label and $y \in \mathbb{R}$ is network output.

3.2.1 Focal loss

Cross-entropy loss can be extended to focal loss (FL). FL was initially designed for image object detection with a very imbalanced data set [aa17]. FL is defined as:

$$\mathrm{FL}(y,\, l) = \mathrm{L1(y,\, l)}^{\gamma} \cdot \mathrm{CE}(y,\, l), \qquad \mathrm{L1(y,\, l)} = |l - \sigma(y)|,$$

where CE, defined in Eq. 3.1, is multiplied by modular scalar with additional hyper-parameter $\gamma \geq 0$. The modular scalar can be written for binary classification as the L1 loss to the power of γ. Parameter γ controls the impact of well-classified events. For example, with $\gamma = 2$, a well-classified event with $\mathrm{L1} = 0.1$ has a 100-fold smaller loss compared to the original CE. On the other hand, in the case of a misclassified event with $\mathrm{L1} = 0.5$, the loss is reduced only four times, and such events have a much bigger impact on the NN training. When $\gamma = 0$, FL is reduced to CE. Visualisation of the FL for $\gamma \in \{0, 0.5, 1, 2, 5\}$ is in Figure 3.1.

3.2.2 Imbalanced data set methods

An imbalanced data set affects the training of NN significantly since a model will be overwhelmed with the majority class. Another aspect is that a model will learn the distribution of simulated data that differs from the expected real data.

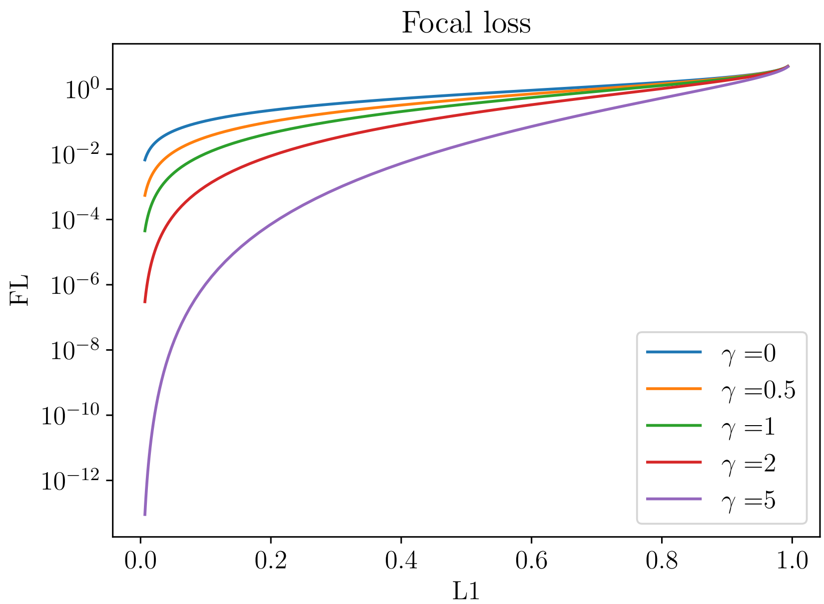


Figure 3.1: Focal loss characteristic for different γ factors. On the left part of the figure are well-classified events.

Cost-sensitive methods

Cost-sensitive methods take the costs of predictions into account when computing the loss. The weighted loss function is defined as:

$$\mathrm{WL}(y,\, l,\, \bar{w},\, \alpha,\, \gamma) = \alpha\bar{w} \cdot \mathrm{FL}(y,\, l,\, \gamma), \tag{3.2}$$

where $\bar{w}$ are event weights, α is the normalization factor that scales loss to the same range to compare different methods and FL is defined in Eq. 3.2.1. The factor $\bar{w}$ does not need to be the same as weights from Eq. 2.1. Moreover, $\bar{w}$ can be implemented into the NN model structure and be optimized during the training. However, the thesis tested only $\bar{w}$ as scaled event weights for different classes and total class weights.

Data set sampling methods

Another approach is data set sampling methods. For each epoch, events are sampled to match the desired distribution. Event weights can be used as probabilities of a random variable with a multinomial distribution random variable with an exception that $\sum_i p_i \neq 1$.

3.3 Network Architectures

3.3.1 Multi-Layer Perception

Multi-Layer Perception (MLP) is one of the first NN architectures. MLP is a feed-forward network with three primary layer types: a linear layer, an activation function, and a dropout layer. The linear layer is an affine transformation. The activation function can be any nonlinear function allowing a network to learn complex problems. The dropout layer sets a portion of input data to zero. Figure 3.2 shows schema of MLP architecture.

MLPs shows the concept of feed-forward shortcuts introduced by ResNet and FishNet. Although both architectures are CNN, same principle can be applied to MLP. [ea19b] states that shortcuts enable the gradient from the very deep layer to be directly propagated to shallow layers.

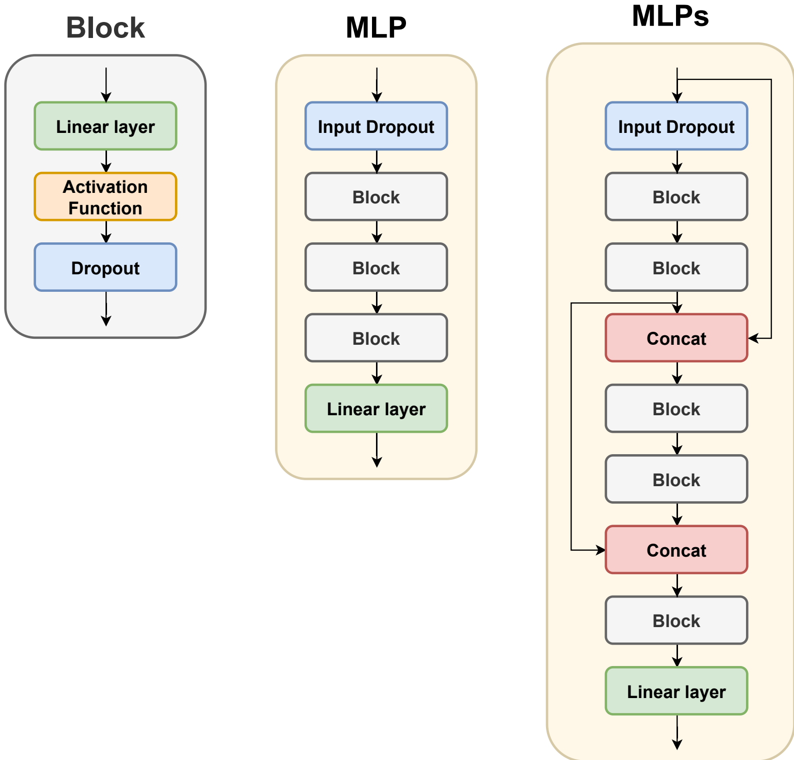


Figure 3.2: MLP architecture schema.

3.3.2 TabNet

TabNet is a feature attention type architecture explicitly designed for tabular data [AP19]. The network is composed of decision blocks with feature and attentive transformers. Each decision block outputs transformed masked input features that are summed and fed to the final linear layer. Attentive transformer creates feature mask from feature transformer's output, and as an activation function is used sparse max. Part of the feature transformers is shared in all decision blocks, which helps the model to be more general and saves memory. Figure 3.3 shows the TabNet model schema.

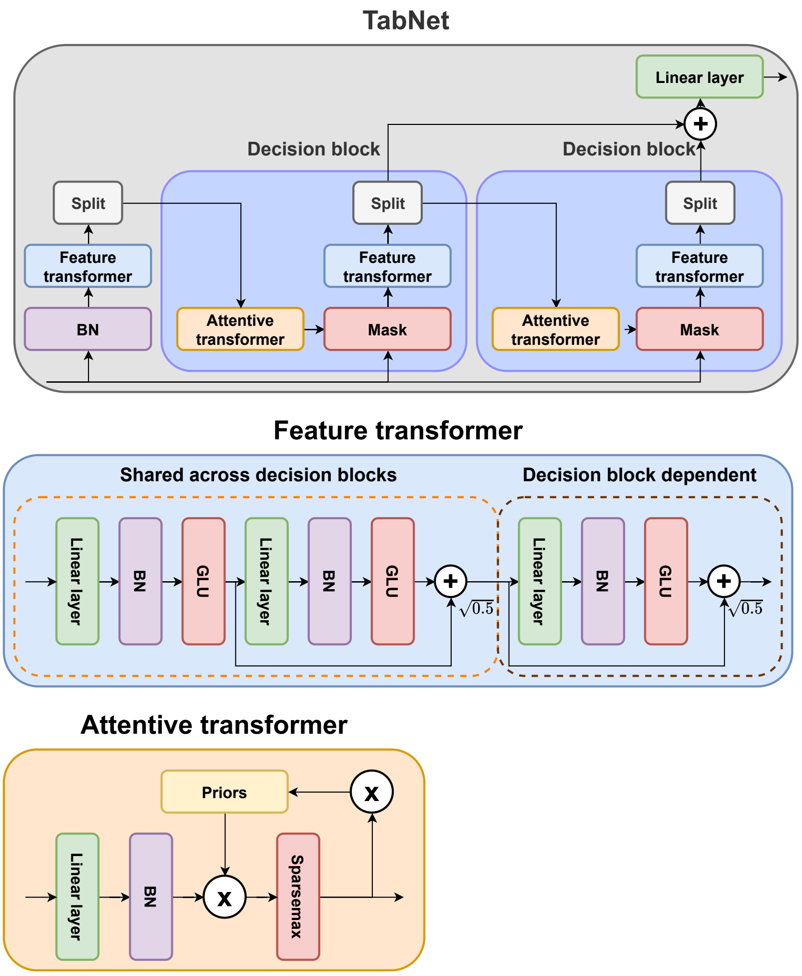


Figure 3.3: Tabnet architecture schema.

3.4 Neural network output

The NN output is used to estimate the probability p that an event belongs to the positive class. The event is classified as positive whether the $p > t$, where t is the chosen working point to maximize significance for a given data set.

Models can be compared via many metrics, for example, accuracy, precision, or recall. Many of them can be computed from the confusion matrix. A weighted confusion matrix, which contains the event's weights (expected number of events) instead of the number of events, is used to adopt metrics to real data expectations. If only part of the data set is used, the event's weights must be scaled to represent all the data.

3.4.1 Significance computation

Below is an example of a significance computation for a given working point. In the binary classification task the expected Signal S and Background B are computed as:

$$S = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \cdot \bar{S}, \qquad B = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \cdot \bar{B}, \tag{3.3}$$

where $\bar{S}$ is total weight of Signal, $\bar{B}$ is total weight of Background, TP is true positive, FP is false positive, TN is true negative and FN is false negative from weighted confusion matrix.

Figure 3.4 shows the Eq. 2.8 dependence on Signal and Background. A right-top corner is a special case when the NN fails and classifies every event as a positive class. There is also a minimal significance value that can be achieved for the given data set.

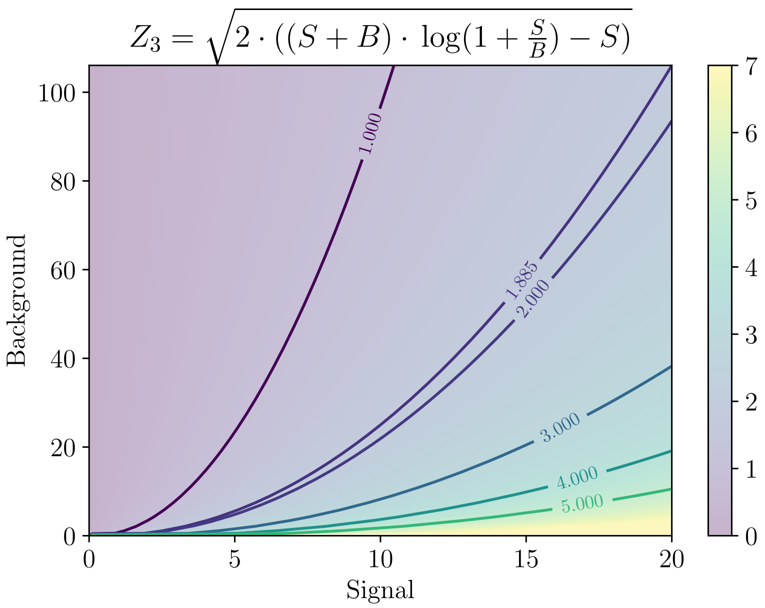


Figure 3.4: The Significance Z_3 dependence on the Signal and the Background.

Chapter 4

Implementation

4.1 Environment and libraries

The implementation was tested in the following environment:

- Debian GNU/Linux 10
- Python 3.8.2
- PyTorch 1.11.0
- Ray Tune 1.12.0
- TensorBoard 2.8.0
- Uproot 4.2.2
- tqdm 4.64.0
- sparsemax 0.1.9

PyTorch is a deep learning library mainly written in C++ with GPU computation support. PyTorch enables the implementation of effective training pipelines and NN models from predefined building blocks with automatic gradient computation. [ea19a] shows that PyTorch outperforms Tensorflow in all tested models.

Ray Tune is a library for hyper-parameter optimization that allows running and managing multiple training experiments. Library offers several scheduling algorithms, such as Population Based Training or HyperBand/ASHA, and a highly customizable system for model checkpointing and logging [ea18].

Uproot library provides faster and easier handling of ROOT Ntuples than PyROOT, the official ROOT key binding for Python [Piv].

4.2 Code description

Project files are divided into utility files, stored in the *utils* directory, and user scripts. User scripts serve as a simple user interface for common tasks, while each utility file is dedicated to the key part of the training pipeline.

4.2.1 Utility files

- dataset_utils.py contains definitions of the Dataset, Process classes with methods for creating and manipulating the data sets.
- model_utils.py provides the MLP and the TabNet definitions.
- metric_utils.py implements metric with the working point evaluation and significance computation.
- loss_utils.py defines the weighted focal loss implementation.
- trainer_utils.py implements training pipleline as a Ray Tune Trainable class.
- root_utils.py provides utility functions for dataset creation related to the ROOT Ntuples.
- file_utils.py implements file handling.
- utils.py contains common functions.

4.2.2 User scripts

User scripts include a help option that displays a list of all available arguments with a description.

create_dataset.py

The script creates one or two datasets from the given Ntuples. It is intended to create a multi-class dataset and select the class of interest at the beginning of the training. The input directory can contain subdirectories. However, the script will only parse files whose names are listed in the process text file. Each line in the file represents one class. The syntax is as follows:

$$\text{Name} : \text{ID}_1, \text{ID}_2, \text{ID}_3;$$

where ID_i is the file name without the *.root* extension, the class name and the file IDs are separated by a colon, and each line must end semicolon. Do not use spaces in ID_i or the process names because the script removes all whitespaces. A preselection cut can be specified using uproot or ROOT logical syntax in a text file. The script does not support vector type features, and it is preserved to the user to provide a text file with a list of feature names (Names can be separated with any whitespace).

train_config.py

The script provides an interface to the NN training procedure. The script can run multiple experiments defined in the file config.py and automatically select the best model for each experiment. Addition evaluation can be specified for the best-performing models.

The script requires two variables defined as a Python dictionary or a list of dictionaries in the config.py. The *configs* variable contains the configuration of training parameters for each experiment. The *eval_configs* variable overwrites the best model *configs* configuration for additional evaluation.

eval_config.py

The script evaluates trained NN using the *eval_configs* variable in the file config.py.

config.py

The file serves as a configuration file for train_config.py and eval_config.py. Ray Tune hyper-parameter search functions are used to create multiple training configurations in the experiment. An explanation of all available parameters for *configs* and *eval_configs* variables is given in Appendix D.

Chapter 5

Results

This chapter summarises the performed experiments. The table headers in this section use the following abbreviations:

- Z_i - significance approximation defined in Eq. 2.5- 2.8,
- S_i - number of Signal events (TP) for W_i,
- B_i - is the number of Background events (FP) for W_i,
- S_i acc. - Signal accuracy for W_i,
- B_i acc. - Background accuracy for W_i,

where W_i is the working point with the maximum significance Z_i. Significance approximations are accurate only when sufficient numbers of S_i and B_i are observed; thus, an additional working point requirement of $S_i > 1$ and $B_i > 1$ has been introduced.

Section 2.4.2 introduces the concept of event weights. Since the Signal production cross section is unknown, these event weights serve only as an initial estimate. The scaling of the Signal weights creates new estimates of the expected number of events that are more similar to the Background numbers; moreover, the Signal weight scaling factor η is used as another model hyper-parameter. In addition, different scaling factors for different Signal masses control the importance of the mass dependence in the NN training.

The models were trained in two phases. The first phase performs a hyper-parameters search and selects the best model based on the significance Z_1. In the second phase, selected model configurations are trained with fixed training hyper-parameters for several repetitions. The second training phase shows the dependence of the model on input data because new unique training and validation data sets are created from simulated data for each NN training.

5.1 Methods comparison

In the first experiment, the data set contained H^+ events of all four masses. In the training, the total weight of the Signal S_{all} was scaled to $S_{all} \in \{5, 20, 80\}$ expected events and different scaling factors η were used to ensure that all masses were equally represented. Different methods are comparable by significance only whether the experiments use the same number of expected Signal and Background events; therefore, in the validation evaluation, the total Signal weight is set such that five events of each weight are expected.

Table 5.1 lists the mean values of significance and the Signal/Background accuracy with standard error from the first training phase, in which one hundred trials were trained for each experiment. The suffix in the model's name describes the used training method: e stands for weighted loss function with event weights, c stands for weighted loss function with class weights, s stands for data set sampling method. Surprisingly, the Linear model, the simplest MLP model with only one linear layer, achieved the best mean significance and most of the hyper-parameters result in a good model performance. Table 5.2 lists the best results selected based on the significance Z_1 for each experiment. Cost-sensitive methods outperformed traditional NN training and data set sampling methods. The TabNet architecture is more complex, with a significantly longer training time than the MLP, and due to the simulation data production delays, the MLP optimization was prioritized. Only a few experiments were performed with the TabNet architecture. That might be the reason why the TabNet achieved the worst results of the tested models, even though it is state of the art for the classification of tabular data.

Model	Z_0	Z_1	S_1 acc.	B_1 acc.
Linear	13.428 ± 0.159	3.812 ± 0.016	0.830 ± 0.006	0.977 ± 0.001
MLP	9.411 ± 0.292	3.195 ± 0.047	0.669 ± 0.014	0.962 ± 0.002
MLPc$_{80}$	9.201 ± 0.295	3.176 ± 0.047	0.670 ± 0.014	0.959 ± 0.002
MLPe$_{20}$	8.893 ± 0.273	3.094 ± 0.039	0.623 ± 0.012	0.965 ± 0.002
MLPe$_{80}$	8.569 ± 0.225	3.044 ± 0.033	0.615 ± 0.010	0.962 ± 0.002
MLPe$_5$	8.414 ± 0.222	3.007 ± 0.033	0.604 ± 0.010	0.962 ± 0.002
MLPc$_{20}$	7.597 ± 0.176	2.892 ± 0.026	0.589 ± 0.008	0.953 ± 0.002
MLPc$_5$	6.670 ± 0.121	2.764 ± 0.018	0.566 ± 0.006	0.947 ± 0.002
MLPs$_{80}$	6.527 ± 0.099	2.756 ± 0.014	0.570 ± 0.006	0.943 ± 0.003
MLPs$_{20}$	6.515 ± 0.138	2.753 ± 0.019	0.563 ± 0.007	0.946 ± 0.003
MLPs$_5$	6.197 ± 0.093	2.709 ± 0.013	0.563 ± 0.006	0.941 ± 0.003
TabNete$_{20}$	4.192 ± 0.081	2.426 ± 0.020	0.542 ± 0.007	0.905 ± 0.007
TabNet	4.325 ± 0.093	2.420 ± 0.020	0.537 ± 0.008	0.909 ± 0.006

Table 5.1: Significance and the standard error for Z_0 and Z_1 approximations from the first training phase. Signal and Background accuracies with uncertainties are also listed.

Model	Z_0	Z_1	S_1 acc.	B_1 acc.
MLPe$_{20}$	18.37	4.22	0.94	0.99
MLPe$_{80}$	17.06	4.21	0.94	0.99
MLPe$_5$	16.53	4.18	0.93	0.99
MLPc$_{80}$	17.33	4.16	0.92	0.99
MLP	18.06	4.15	0.91	0.99
Linear	16.70	4.13	0.91	0.99
MLPc$_{20}$	16.56	4.10	0.93	0.98
MLPs$_{20}$	14.34	4.01	0.92	0.97
TabNet	9.20	3.40	0.71	0.97
MLPc$_5$	10.92	3.36	0.69	0.97
MLPs$_{80}$	8.61	3.12	0.61	0.97
MLPs$_5$	8.57	3.05	0.61	0.97
TabNete$_{20}$	5.82	2.85	0.63	0.93

Table 5.2: Significance for Z_0 and Z_1 approximations of the best performing model in terms of Z_1 significance from the first training phase. Signal and Background accuracies with uncertainties are also listed.

Table 5.3 lists the mean values of significance and Signal/Background accuracy with standard error from the second training phase. Twenty trials were trained for each method. The data set sampling method did not perform well, although it significantly reduced training time and in the second training phase was almost as good as traditional MLP.

Model	Z_0	Z_1	S_1 acc.	B_1 acc.
MLPe$_5$	15.765 ± 0.303	4.134 ± 0.016	0.930 ± 0.006	0.984 ± 0.001
MLPc$_{20}$	16.301 ± 0.368	4.123 ± 0.020	0.919 ± 0.007	0.986 ± 0.001
MLPe$_{20}$	16.001 ± 0.337	4.121 ± 0.018	0.931 ± 0.006	0.983 ± 0.002
MLPe$_{80}$	14.941 ± 0.420	4.108 ± 0.016	0.927 ± 0.004	0.983 ± 0.001
MLPc$_{80}$	14.922 ± 0.220	4.033 ± 0.018	0.904 ± 0.010	0.981 ± 0.002
MLP	14.970 ± 0.347	4.017 ± 0.025	0.900 ± 0.012	0.980 ± 0.002
MLPs$_{20}$	14.813 ± 0.357	4.000 ± 0.019	0.896 ± 0.009	0.979 ± 0.002
Linear	14.094 ± 0.409	3.900 ± 0.020	0.860 ± 0.007	0.979 ± 0.002
MLPc$_5$	10.527 ± 0.216	3.337 ± 0.024	0.664 ± 0.010	0.976 ± 0.002
TabNet	6.751 ± 0.392	3.137 ± 0.068	0.769 ± 0.020	0.912 ± 0.012
MLPs$_5$	8.105 ± 0.157	2.982 ± 0.023	0.575 ± 0.009	0.968 ± 0.002
MLPs$_{80}$	7.906 ± 0.118	2.949 ± 0.015	0.571 ± 0.009	0.966 ± 0.002
TabNete$_{20}$	4.684 ± 0.152	2.532 ± 0.032	0.552 ± 0.021	0.921 ± 0.009

Table 5.3: Significance and the standard error for Z_0 and Z_1 approximations from the second training phase. Signal and Background accuracies with uncertainties are also listed.

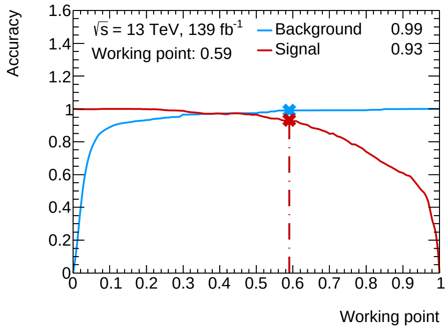
5.2 The best model analysis

The highest significance Z_1 was achieved using the event weighted loss function with Signal total weight scaled to twenty expected events. This section analyzes the model in more detail by performing further experiments. Figure 5.1 shows the used metrics and the normalized NN output distributions.

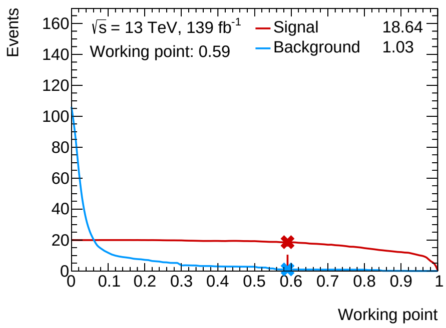
5.2.1 The 300 GeV mass charged Higgs boson

Figure 5.1d shows the NN output distributions. The Signal distribution contains two local maxima, the smaller and broader peak around working point 0.8 represents hard-to-classify events, whereas well-classified events create the steeper peak around point one. Predominantly events of the 300 GeV mass charged Higgs boson were difficult to classify, as shown in Figure 5.2.

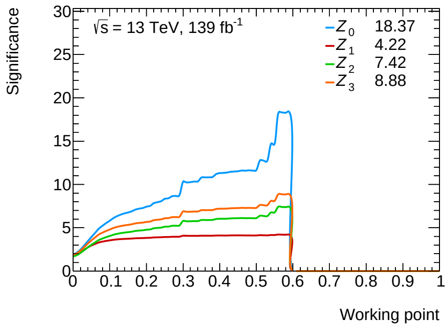
The following experiment uses scaling factors η_{300} and η_{800} as hyper-parameters to improve the separation of the 300 GeV mass charged Higgs boson. All Signal weights were initially scaled so that there were five expected events for each mass, and the scaling factors η_{300}, η_{800} were additionally multiplied by $\alpha \in (1, 2)$ in the training procedure. Table 5.4 lists the



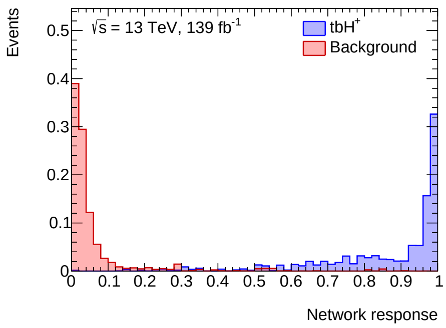
(a) : Accuracy working point characteristic.



(b) : Signal/Background working point characteristic.



(c) : Significance working point characteristic.



(d) : Normalized distributions of the NN output.

Figure 5.1: Working point characteristics of the model with the highest significance Z_1.

significance Z_1 and Z_1^{300}. These results were for the best performing model in terms of the significance from the one hundred trials for each test. The Significance Z_1^{300} was computed for validation data set containing only 300 GeV mass charged Higgs boson as the Signal. Scaling factors η_{300}, η_{800} are also listed. The Signal of the validation set was scaled up to twenty expected events with equally distributed masses. The scaling factors optimization did not improved the significance at all. Although the models focus more on the 300 GeV mass charged Higgs boson, the efficiency of classifying the Background is reduced, resulting in less separation power, as shown in Figure 5.3.

Model	$\mathbf{Z_1^{300}}$	$\mathbf{Z_1}$	η_{300}	η_{800}
MLPe$_{20}$	4.02	4.22	428.97	6657.58
optimized η_{300}	3.68	4.17	1247.3	6657.58
optimized η_{300}, η_{800}	3.61	4.10	484.24	8388.88

Table 5.4: Significance Z_1 and Z_1^{300} for optimized scaling factors η_{300} and η_{800}.

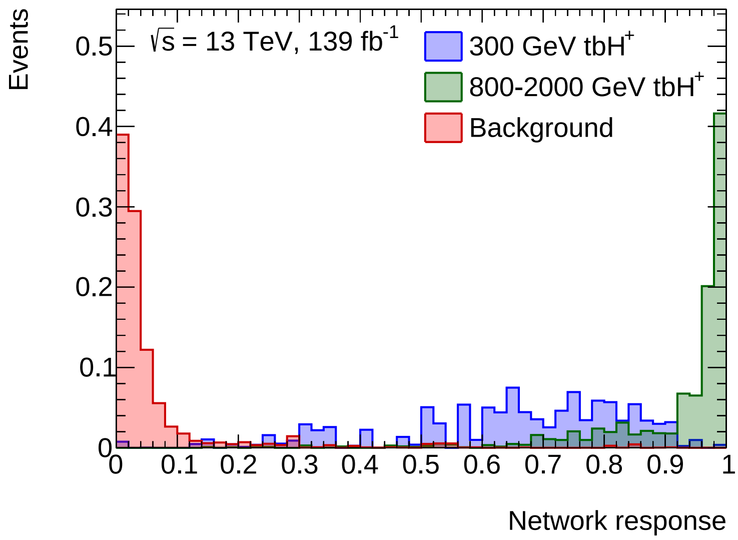
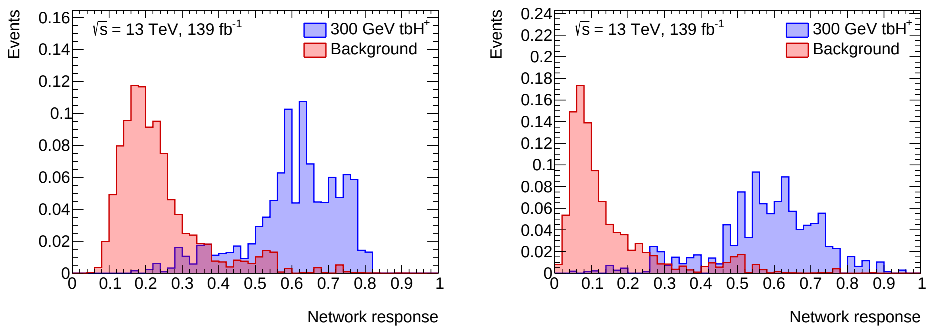


Figure 5.2: The comparison of the normalized NN output distributions of the hard-to-classify 300 GeV charged Higgs boson evens (blue) and the well-separated, higher mass charged Higgs boson events (green). The Background normalized NN output distribution (red) is also shown.



(a) : Normalized distributions of the NN output for optimized η_{300}.

(b) : Normalized distributions of the NN output for optimized η_{300} and η_{800}.

Figure 5.3: The comparison of the normalized NN output distributions..

5.2.2 Cross section estimation

The cross section is a measure of the probability that a proton-proton collision results in a specific process [Gra]. The unit of reaction cross section is the barn. The main advantage of the cross section is its independence from the particle beam's intensity and the accelerator's power; therefore, two experiments from different accelerators can be directly compared. The number of expected events of the process is a product of the cross section and the integrated luminosity.

The classifier can detect a new particle only if its sensitivity is less than the theoretical cross section of the particle production. The classifier's sensitivity is computed as a cross section exclusion limit at the 95% confidence level which corresponds to significance of 2σ. In this thesis, two methods were used to determine the classifier's sensitivity, both of which calculate the cross sections for each signal mass individually. The first method changes the cross section so that a certain significance approximation from Eq. 2.5-2.8 is equal to 2σ and the estimated cross section is shown in Figure 5.5b.

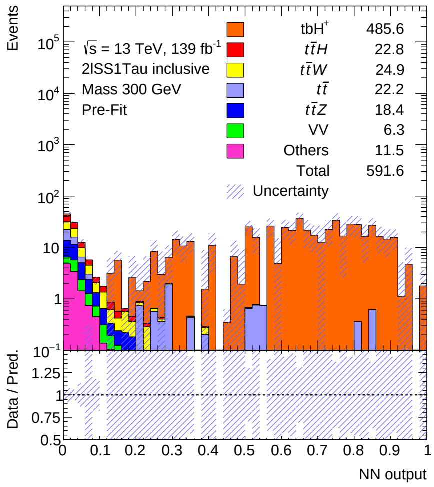
In the second method, the Trex-fitter program [Tf], which is widely used in the ATLAS Collaboration, calculates the cross section exclusion limits at 95% CL. Trex-fitter takes a distribution of a random variable defined as a Region as an input and performs the CLs method. In addition to the shape of the random variable distribution, the sensitivity mainly affects the binning option of the Region histogram and the number of expected Signal events. Figure 5.4 shows the NN output with the Signal weights normalized to the cross section of one picobarn and with scaled luminosity to compensate for missing mc16a, mc16d data sets which was used as a Trex-fitter input. Figure 5.5a shows the cross section computed by the Trex-fitter for 95% CL.

Despite the very good separation of Signal and Background and working point with a small number of Background events for which the approximations of significance are less accurate, the sensitivity computed with the Trex-fitter and the Z_0, Z_3 significance approximations is almost identical, as shown in Figure 5.5. The Z_1, Z_2 significance approximations led to lower sensitivity close to the upper bound of the Trex-fitter's two-sigma confidence interval.

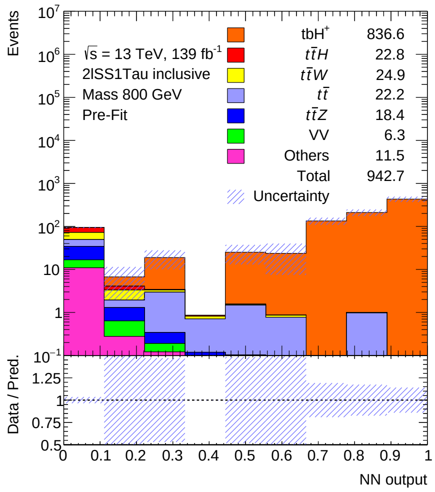
Figure 5.6 shows the expected and observed upper limits at 95% CL on the product of cross section and branching fraction $\sigma_{\mathrm{H}^\pm}(\mathrm{H}^\pm \to \mathrm{HW}^\pm, \mathrm{H} \to \tau\tau)$ from the CMS charged Higgs boson decaying into a heavy neutral Higgs boson and a W boson analysis [Col22]. The methods proposed in this thesis obtained better sensitivity, as shown in Figure 5.5a, than the CMS analysis; however, the CMS analysis includes systematic uncertainties which reduce the sensitivity and were not modeled in this analysis.

5.2.3 Feature importance

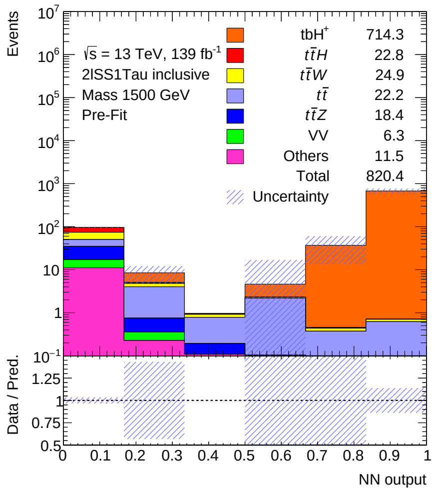
Feature importance is a valuable measurement providing an insight into the model's decision-making. Although the NN can classify data without any theoretical background, scientists can use this information to improve the measurement or the reconstruction of essential features.



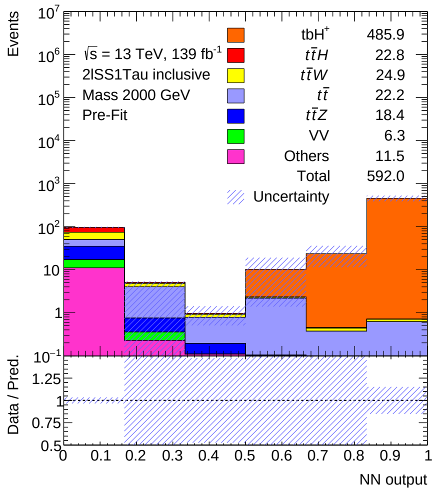
(a) : The NN output for the 300 GeV mass charged Higgs boson and the Background.



(b) : The NN output for the 800 GeV mass charged Higgs boson and the Background.



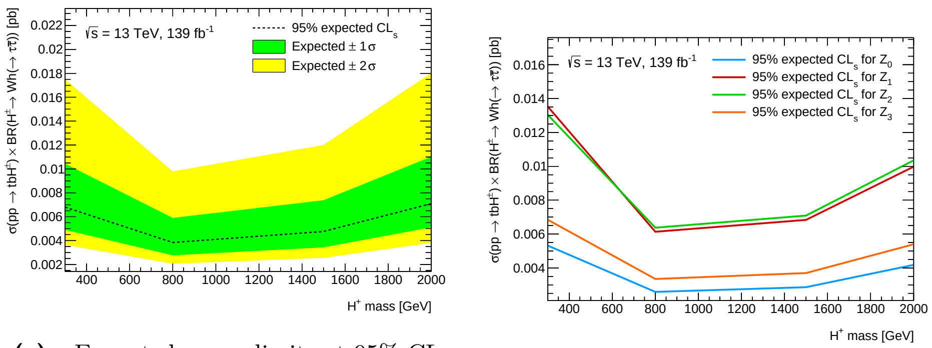
(c) : The NN output for the 1500 GeV mass charged Higgs boson and the Background.



(d) : The NN output for the 2000 GeV mass charged Higgs boson and the Background.

Figure 5.4: The NN output distributions. The number of expected Signal events is normalized to the cross section of one picobarn and weights are scaled to compensate for the missing Signal mc16a, mc16d data sets.

This thesis uses the permutation feature importance method. The importance is defined as the absolute difference of significance ΔZ_1 obtained for the default validation data set and the modified validation data set and is computed for each feature separately. The modified data set has a changed order of the values for the currently calculated feature, which can be interpreted as if the feature values were measured with noise. One of the method's advantages is the computation speed because the model does not need to be retrained. Figure 5.7 shows the top twenty most important features; however,



(a) : Expected upper limits at 95% CL on the cross section using the Trex-fitter. The 68% (inner green band), and 95% (outer yellow band) confidence intervals are also shown.

(b) : The cross section estimate using signal scaling to obtain significance approximation of 2 σ.

Figure 5.5: Expected cross section for 95% CL.

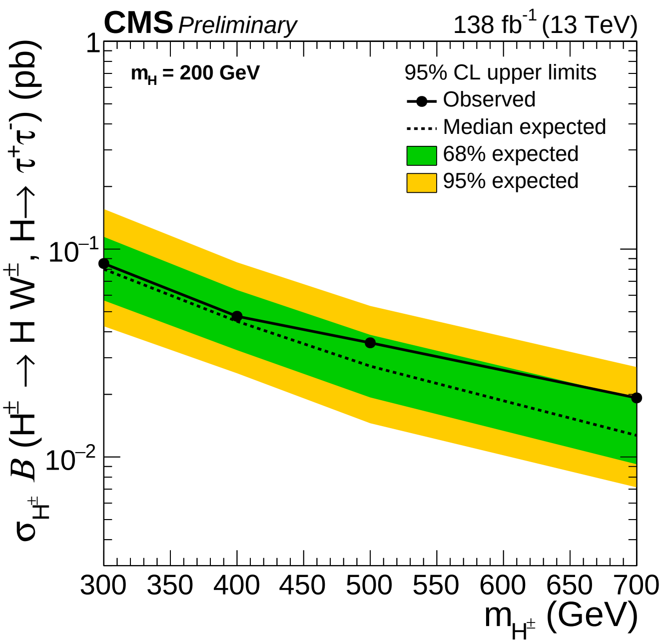
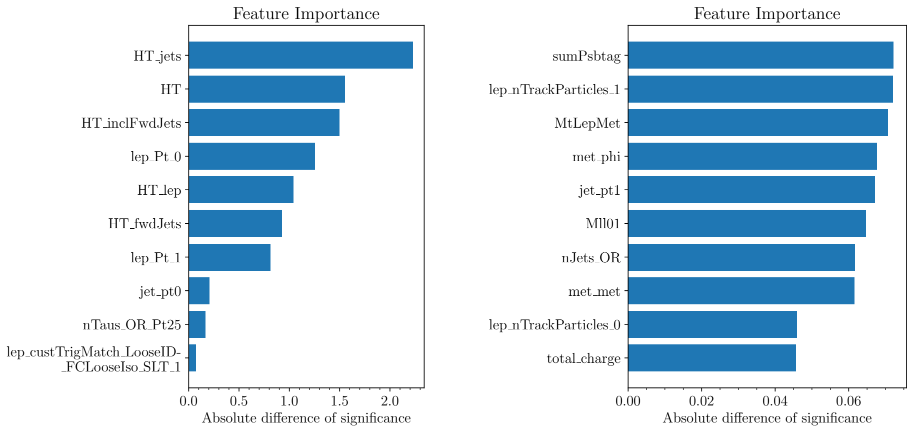


Figure 5.6: Expected and observed upper limits at 95% CL on the product of cross section and branching fraction $\sigma_{\mathrm{H}^\pm}(\mathrm{H}^\pm \to \mathrm{HW}^\pm, \mathrm{H} \to \tau\tau)$ as a function of $m_{\mathrm{H}^\pm}$ and assuming $m_{\mathrm{H}} = 200$ GeV for the combination of all final states considered. The observed upper limits are represented by a solid black line and circle markers. The median expected limit (dashed line), 68% (inner green band), and 95% (outer yellow band) confidence intervals are also shown. Taken from [Col22].

the top ninth most important feature already affected the resulting significance negligibly compared to the top five most important features. The importance of each feature was calculated as the average of ten different permutations of values. The most important feature was the sum of transverse momentum of jets HT_jets with the significance difference $\Delta Z_1 = 2.229$. Table B.3 lists the importance of all used features.



(a) : Top 10 most important features. **(b) :** Top 10-20 most important features.

Figure 5.7: Feature importance of the model with the highest significance Z_1.

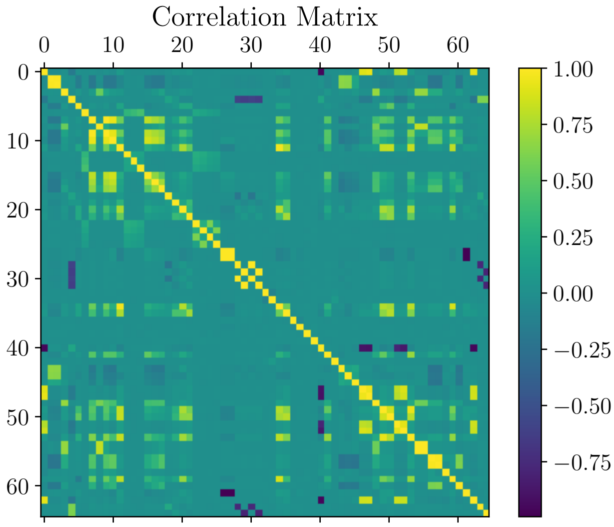


Figure 5.8: Correlation matrix of the simulated data set.

The following experiment was performed to verify the proposed method. The NN was optimized using a data set with all 65 features or only the five, ten, and twenty most important features. One hundred trials in the first phase of training and twenty trials in the second phase of training were trained for each set of functions, Table 5.5 lists the mean significance and Signal and Background accuracies with standard error from the second stage of the training. As expected, even for only the top five most important features, the model retains decent separation power. It is also worth mentioning that the model trained with the data set containing only the top ten most important features outperformed the original model trained with all features.

Table B.1 lists the definitions of the ten top most important features and their distributions are shown in Figure C.1.

Features	$\mathbf{Z}_0$	$\mathbf{Z}_1$	$\mathbf{S}_1$ **acc.**	$\mathbf{B}_1$ **acc.**
top_{10}	16.635 ± 0.221	4.143 ± 0.017	0.923 ± 0.007	0.987 ± 0.001
all	16.001 ± 0.337	4.121 ± 0.018	0.931 ± 0.006	0.983 ± 0.002
top_{20}	15.050 ± 0.285	4.061 ± 0.013	0.937 ± 0.005	0.976 ± 0.001
top_{5}	14.242 ± 0.479	4.025 ± 0.028	0.923 ± 0.009	0.975 ± 0.004

Table 5.5: Significance and the standard error for Z_0 and Z_1 approximations for different data set feature sets from the second training phase. Signal and Background accuracies with uncertainties are also listed.

Chapter 6

Conclusion

The goal of this thesis was to separate the Signal tbH^+ from the Background using optimized NN and estimate the sensitivity of the classifier. The thesis was dealing with the simulated data with preselected events of the 2lSS1tau channel, that means two same-sign light leptons and one hadronically decaying τ.

Two NN architectures, MLP and TabNet, were implemented, which were optimized using data set sampling and cost-sensitive methods to overcome imbalanced data sets. It was discovered that smaller shallow MLP results in better significance than deeper networks. The best performing model in terms of the Signal separation was achieved with the MLP and event weighted loss function. Due to the delayed data set production, only a few experiments were done with the TabNet architecture. That might be the reason why the performance of the TabNet was much worse compared to MLP.

The Trex-fitter program was used to estimate the cross section exclusion limit at the 95% CL of tbH^+ production using the output distribution of the best performing model in terms of significance. The results indicate a higher sensitivity for tbH^+ low masses compared to a recent CMS Collaboration tbH^+ analysis.

The permutation feature importance method was used to determine the importance of individual features of the best MLP model, which ranked the sum of the transverse momentum of all jets as the most important. Several experiments were performed with a data set containing a reduced set of the important features to verify this method. The experiment with the top ten most important features reached better significance than the experiment with all 65 features. The reason could be that the dimensionality of the problem was reduced, and the remaining features have very good separation power.

Appendix A

The Tight lepton l2SS1tau channel preselection formula

While the preselection was applied as given below, consistent with a previous $t\bar{t}H$ analysis of December 2021, it is noted here that instead of the variable nTaus_OR, the variable nTaus_OR_Pt25 should be used.

$$
\begin{aligned}
preselection = {} & nJets\_OR\_TauOR > 2 \wedge nJets\_OR\_DL1r\_70 > 0 \wedge \\
& (lep\_Pt\_0 \geq 10e3 \wedge lep\_Pt\_1 \geq 10e3) \wedge ((\mathrm{abs}(lep\_ID\_0) = 13 \wedge \\
& lep\_isMedium\_0 \wedge passPLIVVeryTight\_0 \wedge \\
& lep\_isolationFCLoose\_0) \vee (\mathrm{abs}(lep\_ID\_0) = 11 \wedge \\
& lep\_isolationFCLoose\_0 \wedge lep\_isTightLH\_0 \wedge \\
& lep\_ambiguityType\_0 = 0 \wedge \mathrm{fabs}(lep\_Eta\_0) \leq 2.5 \wedge \\
& lep\_chargeIDBDTResult\_recalc\_rel207\_tight\_0 > 0.7 \wedge \\
& passPLIVVeryTight\_0 \wedge \\
& ((\neg(\neg(lep\_Mtrktrk\_atConvV\_CO\_0 < 0.1 \wedge \\
& lep\_Mtrktrk\_atConvV\_CO\_0 \geq 0 \wedge lep\_RadiusCO\_0 > 20) \wedge \\
& (lep\_Mtrktrk\_atPV\_CO\_0 < 0.1 \wedge \\
& lep\_Mtrktrk\_atPV\_CO\_0 \geq 0))) \wedge \\
& (\neg(lep\_Mtrktrk\_atConvV\_CO\_0 < 0.1 \wedge \\
& lep\_Mtrktrk\_atConvV\_CO\_0 \geq 0 \wedge \\
& lep\_RadiusCO\_0 > 20))))) \wedge ((\mathrm{abs}(lep\_ID\_1) = 13 \wedge \\
& lep\_isMedium\_1 \wedge passPLIVVeryTight\_1 \wedge \\
& lep\_isolationFCLoose\_1) \vee (\mathrm{abs}(lep\_ID\_1) = 11 \wedge \\
& lep\_isolationFCLoose\_1 \wedge lep\_isTightLH\_1 \wedge \\
& lep\_ambiguityType\_1 = 0 \wedge \mathrm{fabs}(lep\_Eta\_1) \leq 2.5 \wedge
\end{aligned}
$$

$$lep\_chargeIDBDTResult\_recalc\_rel207\_tight\_1 > 0.7 \wedge$$
$$passPLIVVeryTight\_1 \wedge$$
$$((\neg(\neg(lep\_Mtrktrk\_atConvV\_CO\_1 < 0.1 \wedge$$
$$lep\_Mtrktrk\_atConvV\_CO\_1 \geq 0 \wedge lep\_RadiusCO\_1 > 20) \wedge$$
$$(lep\_Mtrktrk\_atPV\_CO\_1 < 0.1 \wedge$$
$$lep\_Mtrktrk\_atPV\_CO\_1 \geq 0))) \wedge$$
$$(\neg(lep\_Mtrktrk\_atConvV\_CO\_1 < 0.1 \wedge$$
$$lep\_Mtrktrk\_atConvV\_CO\_1 \geq 0 \wedge$$
$$lep\_RadiusCO\_1 > 20))))) \wedge nTaus\_OR = 1 \wedge$$
$$lep\_ID\_0 \cdot lep\_ID\_1 > 0$$

Appendix B

Tables

Feature Name	Definition
HT_jets	The sum of the transverse momentum of jets.
HT	The sum of the transverse momentum of all objects.
HT_inclFwdJets	The sum of the transverse momentum including forward jets.
lep_Pt_0	Transverse momentum of the leading light lepton.
HT_lep	The sum of the transverse momentum of light leptons.
HT_fwdJets	The sum of the transverse momentum of forward jets.
lep_Pt_1	Transverse momentum of the subleading light lepton.
jet_pt0	Transverse momentum of the leading jet.
nTaus_OR_Pt25	Number of taus with at least 25 GeV transfers momentum.
lep_custTrigMatch_LooseID_FCLooseIso_SLT_1	The matching of light lepton trigger condition.

Table B.1: The Definition of the top ten most important features.

ID	Name	ID	Name
1	best_Z_Mll	34	lep_Phi_1
2	DeltaR_min_lep_jet	35	lep_Pt_0
3	DeltaR_min_lep_jet_fwd	36	lep_Pt_1
4	dEta_maxMjj_frwdjet	37	lep_sigd0PV_0
5	dilep_type	38	lep_sigd0PV_1
6	DRll01	39	lep_Z0SinTheta_0
7	eta_frwdjet	40	lep_Z0SinTheta_1
8	HT	41	max_eta
9	HT_fwdJets	42	met_met
10	HT_inclFwdJets	43	met_phi
11	HT_jets	44	minDeltaR_LJ_0
12	HT_lep	45	minDeltaR_LJ_1
13	jet_eta0	46	minDeltaR_LJ_2
14	jet_eta1	47	minOSMll
15	jet_eta2	48	minOSSFMll
16	jet_pt0	49	mjjMax_frwdJet
17	jet_pt1	50	MLepMet
18	jet_pt2	51	Mll01
19	lep_custTrigMatch_Loose--ID_FCLooseIso_SLT_0	52	Mlll012
20	lep_custTrigMatch_Loose--ID_FCLooseIso_SLT_1	53	Mllll0123
21	lep_E_0	54	MtLepMet
22	lep_E_1	55	nFwdJets_OR
23	lep_Eta_0	56	nFwdJets_OR_TauOR
24	lep_Eta_1	57	nJets_OR
25	lep_EtaBE2_0	58	nJets_OR_TauOR
26	lep_EtaBE2_1	59	nTaus_OR_Pt25
27	lep_ID_0	60	Ptll01
28	lep_ID_1	61	sumPsbtag
29	lep_Mtrktrk_atConvV_CO_0	62	total_charge
30	lep_Mtrktrk_atConvV_CO_1	63	total_leptons
31	lep_Mtrktrk_atPV_CO_0	64	lep_nTrackParticles_0
32	lep_Mtrktrk_atPV_CO_1	65	lep_nTrackParticles_1
33	lep_Phi_0		

Table B.2: List of used features.

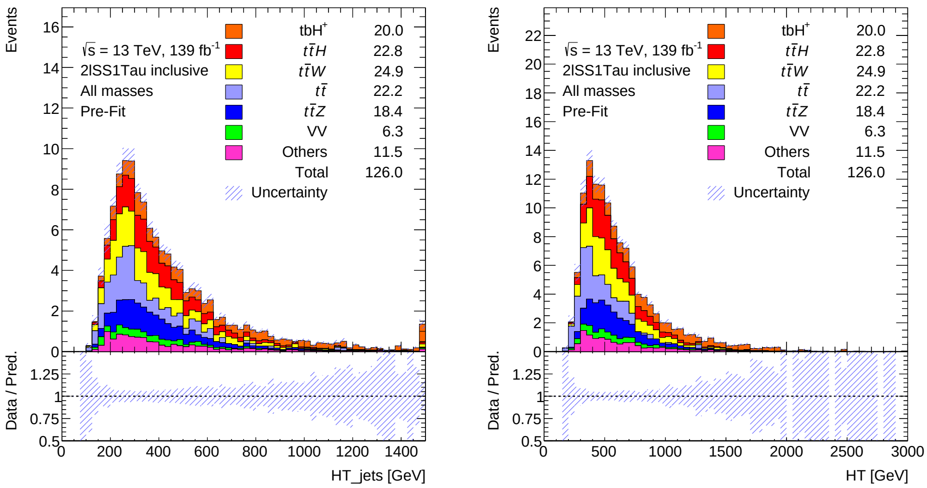
Feature Name	ΔZ_1	Feature Name	ΔZ_1
HT_jets	2.229	lep_E_0	0.028
HT	1.552	lep_Z0SinTheta_0	0.027
HT_inclFwdJets	1.498	MLepMet	0.027
lep_Pt_0	1.257	lep_E_1	0.024
HT_lep	1.039	lep_ID_1	0.023
HT_fwdJets	0.927	eta_frwdjet	0.022
lep_Pt_1	0.812	jet_eta2	0.021
jet_pt0	0.206	mjjMax_frwdJet	0.019
nTaus_OR_Pt25	0.167	nJets_OR_TauOR	0.018
lep_custTrigMatch_Loose- -ID_FCLooseIso_SLT_1	0.074	Mllll0123	0.018
sumPsbtag	0.072	lep_Z0SinTheta_1	0.017
lep_nTrackParticles_1	0.072	lep_Phi_1	0.017
MtLepMet	0.071	lep_Eta_1	0.017
met_phi	0.068	DeltaR_min_lep_jet_fwd	0.016
jet_pt1	0.067	minDeltaR_LJ_2	0.015
Mll01	0.065	DeltaR_min_lep_jet	0.013
nJets_OR	0.062	minDeltaR_LJ_1	0.013
met_met	0.062	lep_Eta_0	0.012
lep_nTrackParticles_0	0.046	lep_Mtrktrk_atPV_CO_0	0.011
total_charge	0.046	lep_Phi_0	0.011
jet_pt2	0.045	max_eta	0.01
lep_EtaBE2_0	0.044	minOSSFMll	0.01
lep_sigd0PV_0	0.041	lep_ID_0	0.009
DRll01	0.041	minDeltaR_LJ_0	0.009
jet_eta0	0.039	lep_Mtrktrk_atConvV_CO_1	0.008
nFwdJets_OR_TauOR	0.037	minOSMll	0.008
Ptll01	0.035	total_leptons	0.008
dEta_maxMjj_frwdjet	0.035	lep_Mtrktrk_atConvV_CO_0	0.006
lep_custTrigMatch_Loose- -ID_FCLooseIso_SLT_0	0.034	lep_Mtrktrk_atPV_CO_1	0.006
lep_EtaBE2_1	0.031	lep_sigd0PV_1	0.006
nFwdJets_OR	0.03	best_Z_Mll	0.005
dilep_type	0.029	Mlll012	0.002
jet_eta1	0.029		

Table B.3: Feature importance ΔZ_1.

Appendix C

Figures

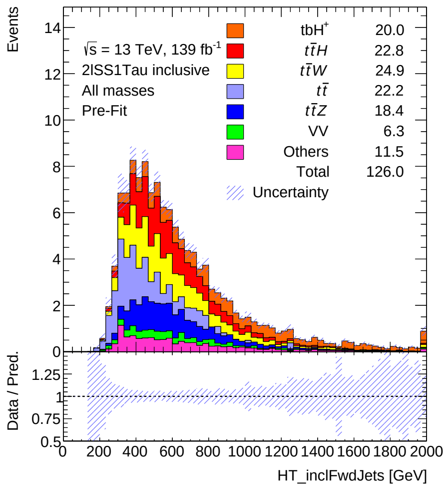
C.1 Feature distributions



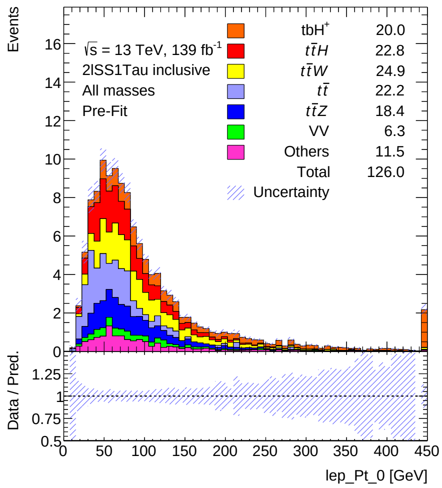
(a) : Distribution of HT_jets variable.

(b) : Distribution of HT variable.

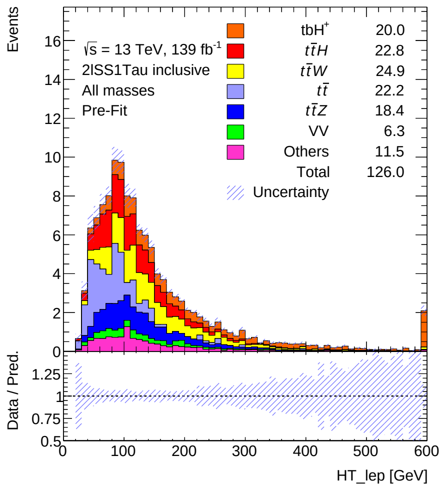
Figure C.1: The top ten most important features for the MLPe$_{20}$ model.



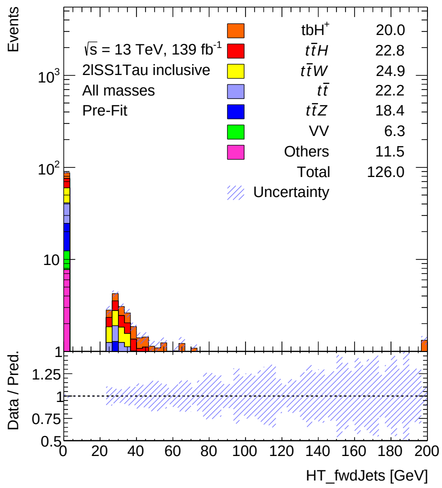
(c) : Distribution of HT_inclFwdJets variable.



(d) : Distribution of lep_Pt_0 variable.

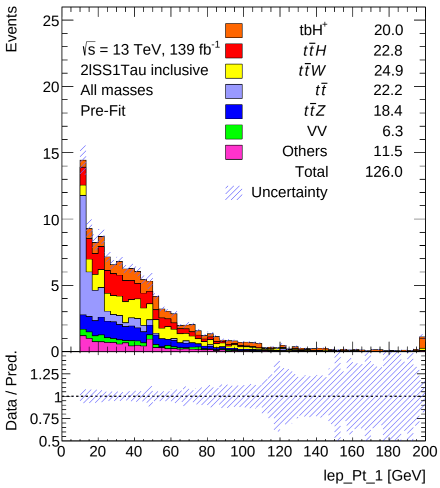


(e) : Distribution of HT_lep variable.

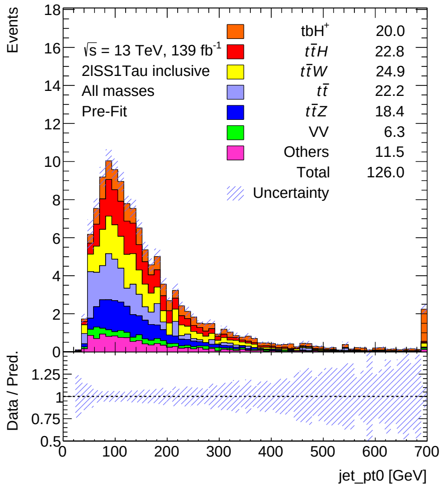


(f) : Distribution of HT_fwdJets variable.

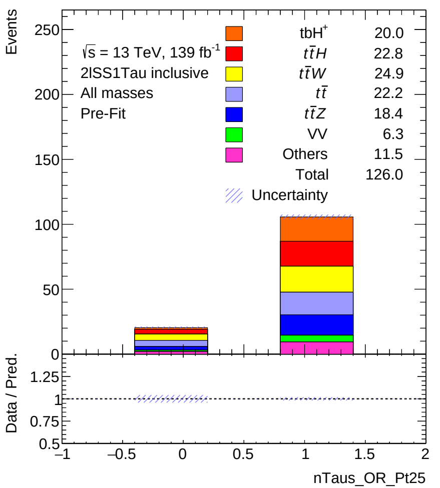
Figure C.1: The top ten most important features for the MLPe$_{20}$ model.



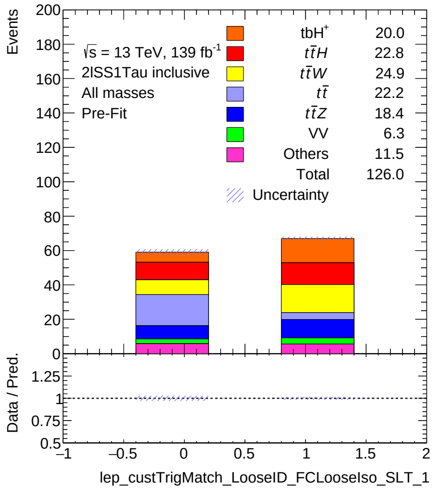
(g) : Distribution of lep_Pt_1 variable.



(h) : Distribution of jet_pt0 variable.



(i) : Distribution of nTaus_OR_Pt25 variable.



(j) : Distribution of lep_custTrigMatch_LooseID_FCLooseIso_SLT_1 variable.

Figure C.1: The top ten most important features for the MLPe$_{20}$ model.

Appendix D

Documentation

```
Available parameters of variable "configs"
Variable type: dict, list of dict
Usage: Variable has to be defined in the file config.py and is
↪  used by the train_config.py to train and optimize NN.
Parameters:

  epochs_num - int > 0, number of training epochs
  samples_num - int > 0, number of ray tune run samples

  ### Dataset

  dataset_name - str, name of the data set, data set will be
   ↪  split on training and validation set using split_divider
  dataset_trn_name - str, name of the training data set
  dataset_val_name - str, name of the validation data set, data
   ↪  set directory path can be specified letter when running
   ↪  the scripts, which allows to save the configuration as a
   ↪  pickle file and use it on multiple machines that have
   ↪  data sets stored on different paths
  dataset_path - str, absolute path to the data set, data set
   ↪  will be split on training and validation set using
   ↪  split_divider
  dataset_trn_path - str, absolute path to the training data
   ↪  set
  dataset_val_path - str, absolute path to the validation data
   ↪  set, ray tune require absolute paths

  split_divider - float (0, 1), ratio of the training data set,
   ↪  split_divider=0.7 means 70% training set 30% validation
   ↪  set
  batch_size - int > 0, number of events per one iteration,
   ↪  epoch has len(data set) / batch_size iterations
  use_sampler - bool, use data sampling method for training,
   ↪  multinomial distribution of event weights
```

```
  samples_num - int > 0, number of sampled events per epoch if
  ↪   use_sampler=True

  signal_weights - dict, each key is process name, value is
  ↪   weights scaling factor
  signal_weight - float,  factor that scales all signal
  ↪   weights, can be used in a combination with
  ↪   signal_weights, to scale the signal differently in
  ↪   training and validation epoch use signal_weight_trn,
  ↪   signal_weight_val or signal_weights_trn,
  ↪   signal_weights_val which overwrites signal_weight,
  ↪   signal_weights
  signal_processes - str, list, names of signal processes, the
  ↪   name format is "NAME_MASS" (select as the Signal process
  ↪   with specific mass) or "NAME" (select as the Signal all
  ↪   of process mass
variants), do not use process names with an underscore
  remove_processes - str, list, names of processes to remove,
  ↪   process is removed only if is not in signal_processes,
  ↪   the name format is the same as for signal_processes

  Example:
    config = {
        signal_processes : ["tbH_300", "tbH_800"],
        remove_processes : "tbH"
    }
    will select process tbH_300, tbH_800 as signal and remove
    ↪   all other tbH masses from the data set

  ### Metric

  use_single_threshold - bool, whether to use fixed working
  ↪   point or tunable working point maximizing significance
  threshold_step - float > 0, resolution of tunable working
  ↪   point
  threshold - float (0, 1), fixed working point that is used if
  ↪   use_single_threshold=True

  ### Model

  model_name - "mlp" or "tabnet", whether to se MLP or TabNet
  ↪   architecture

  ## MLP

  layer_sizes - list of int > 0, sizes of hidden layers
```

```
  input_dropout - float (0, 1), dropout applied on input data
  dropout - float (0, 1), dropout between layers
  shortcut_freq - int > 0, frequency of forward pass shortcuts

  ## TabNet

  feature_size - int > 0, output size of feature transformer
  attention_size - int > 0, input size of attentive transformer
  layers_num - int > 0, number of non-shared feature
  ↪  transformer layers
  shared_layers_num - int > 0, number of shared feature
  ↪  transformer layers
  block_num - int > 0, number of decision steps
  sparse_gamma - float > 0 | coefficient of TabNet sparse loss
  batchnorm_momentum - float > 0, momentum of batch-norm layers
  relaxation - float > 0, priors relaxation parameter

  ### SDG Optimizer

  lr - float > 0, learning rate
  momentum - float > 0, SDG optimizer momentum
  weight_decay - float > 0, model weight regularization
  use_lr_scheduler - bool, use exponentially decaying learning
  ↪  rate
  lr_scheduler_gamma - float > 0, epoch learning rate decay

  ### Loss

  use_weights - None or "event" or "class", type of
  ↪  cost-sensitive method
  focal_gamma - float > 0, focal loss gamma

  ### Utility

  log_debug - bool, if occurs the error (NaN values of loss,
  ↪  weights, metric) log additional training information
  debug_log_len - int > 0, number of iterations to log before
  ↪  the error occurrence

Available parameters of variable "eval_configs"
Variable type: dict, list of dict
Usage: Variable has to be defined in the file config.py and is
↪  used by the train_config.py, eval_config.py files to
↪  evaluate trained models. The eval_configs variable
↪  overwrites parameters of the original configs variable.
```

```
Parameters:

    ### Same as for configs variable

    dataset_name - str
    dataset_val_name - str
    dataset_path - str
    dataset_val_path - str

    signal_weights - dict
    signal_weight - float

    signal_processes - str, list
    remove_processes - str, list

    use_single_threshold - bool
    threshold_step - float > 0
    threshold - float (0, 1)

    ### Additional parameters for train_config.py:
    # The script eval_config.py modifies these parameters using
    ↪  command line arguments

    features_importance - bool, if yes, computes feature
    ↪  importance
    fi_repetitions - int > 0, number of permutation repetitions

    store_nn_output - bool, save the network output as pickle
    ↪  file
    store_metric - bool, save the metric as pickle file
    store_root - bool, save the data set and network output as
    ↪  root file
    store_log - bool, save the working point results as text
    ↪  file
```

Appendix E

Bibliography

[aa17] Lin at al., *Focal Loss for Dense Object Detection*, https://arxiv.org/abs/1708.02002, 2017.

[AP19] Sercan O. Arik and Tomas Pfister, *TabNet: Attentive Interpretable Tabular Learning*, https://arxiv.org/abs/1908.07442, 2019.

[CER] CERN, *Supersymmetry*, https://home.cern/science/physics/supersymmetry.

[CER20] ———, *Exploring new ways to see the Higgs boson*, https://home.cern/news/news/physics/exploring-new-ways-see-higgs-boson, 2020, Accessed: February 6, 2021.

[Col] ATLAS Collaboration, *Decay of Z bosons*, https://atlas.physicsmasterclasses.org/en/zpath_lhcphysics2.htm, Accessed: February 6, 2021.

[Col21] ———, *ATLAS pushes forward the search for a charged Higgs boson*, https://atlas.cern/updates/blog/charged-higgs-workshop, 2021, Accessed: February 7, 2021.

[Col22] CMS Collaboration, *Search for a charged Higgs boson decaying into a heavy neutral Higgs boson and a W boson in proton-proton collisions at* $\sqrt{s} = 13$ TeV, https://cds.cern.ch/record/2803735, 2022.

[Csa96] Csaba Csaki, *The Minimal supersymmetric standard model (MSSM)*, https://arxiv.org/abs/hep-ph/9606414, 1996, p. 599.

[ea18] Liaw et al., *Tune: A Research Platform for Distributed Model Selection and Training*, https://arxiv.org/abs/1807.05118, 2018.

[ea19a] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, https://arxiv.org/abs/1912.01703, 2019.

[ea19b] Sun et al., *FishNet: A Versatile Backbone for Image, Region, and Pixel Level Prediction*, https://arxiv.org/abs/1901.03495, 2019.

[EP16] Jan Eysermans and Isabel Pedraza, *Charged Higgs Analysis in CMS*, https://iopscience.iop.org/article/10.1088/1742-6596/761/1/012030/pdf, 10 2016, p. 012030.

[Gra] H. Gray, *The Higgs boson: The Hunt, the discovery, the study and some future perspectives*, https://atlas.cern/updates/feature/higgs-boson.

[Gro18] E. Gross, *Practical Statistics for High Energy Physics*, https://e-publishing.cern.ch/index.php/CYRSP/article/view/303/405, 2018, pp. 199–221.

[HRS16] Moritz Hardt, Ben Recht, and Yoram Singer, *Train faster, generalize better: Stability of stochastic gradient descent*, http://proceedings.mlr.press/v48/hardt16.pdf, 20–22 Jun 2016, pp. 1225–1234.

[Piv] J. Pivarski, *Rapidly moving data from ROOT to Numpy and Pandas*, https://indico.cern.ch/event/686641/contributions/2894906/attachments/1606247/2548596/pivarski-uproot.pdf.

[PYT] PYTHIA, *Welcome to pythia*, https://pythia.org/.

[ROO] ROOT, *Analyzing petabytes of data, scientifically.*, https://root.cern/.

[Sin02] Pekka K. Sinervo, *Signal significance in particle physics*, https://arxiv.org/abs/hep-ex/0208005, 6 2002, pp. 64–76.

[Tf] Trex-fitter, *Trex-fitter documentation*, https://trexfitter-docs.web.cern.ch/trexfitter-docs/.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Pospíšil Jiří** Personal ID number: **492372**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Application of Machine Learning for the Charged Higgs Boson Search Using ATLAS Data

Bachelor's thesis title in Czech:

Aplikace strojového učení pro hledání nabitého Higgsova bosonu z ATLAS dat

Guidelines:

The neutral Standard Model Higgs boson was discovered in 2012 at CERN, and the search for further Higgs bosons of extended models continues. In particular, the search for charged Higgs bosons. A charged Higgs boson can be produced in association with a top quark and a bottom quark (tbH+). There are various production and decay modes of the charged Higgs boson and the top quark. Using machine learning technology, this analysis addresses the charged Higgs boson which decays into a neutral Higgs boson and a charged W boson. The focus of this search is the channel with two same-sign light leptons and a hadronically decaying tau lepton, in addition to multiple jets from quark production. There are three analysis levels: generator level, full ATLAS detector simulation, and real recorded data. In this project, the data from the full ATLAS detector simulation shall be used and the performance of the machine learning algorithms be optimized in the search for charged Higgs boson, separating the charged Higgs boson signal from unwanted background events. The specific tasks are:

1) Study the provided ROOT framework and the data storage format.
2) Study the provided features for the observed objects (electrons, muons, taus, jets)
3) Design and implement a deep-learning based classifier to separate signal and background events based on simulated data.
4) Optimize the separation of signal and background and evaluate the influence of individual features.
5) Determine the significance measure as a function of the charged Higgs boson mass and compare your results with previous work.

Bibliography / sources:

[1] https://home.cern
[2] https://atlas.cern
[3] Charged Higgs boson searches, https://cppp.web.cern.ch
[4] Heather Gray, Bruno Mansoulié, The Higgs boson: the hunt, the discovery, the study and some future perspectives, https://atlas.cern/updates/feature/higgs-boson
[5] Guest et al., Deep Learning and Its Application to LHC Physics article in Annu. Rev. Nucl. Part. Sci. 2018. 68:1–22 https://arxiv.org/pdf/1806.11484.pdf
[6] M. Andrews et al., End-to-End Event Classification of High-Energy Physics Data http://www.sergeigleyzer.com/wp-content/uploads/2017/12/end-end-event.pdf
[7] Melanie Weber, Bachelor thesis, University Leipzig, 2016, Event Classification with Convolutional Neural Networks for Diboson Channels in the ATLAS Experiment at the Large Hadron Collider https://web.math.princeton.edu/~mw25/project/cnn/
[8] Jakub Maly, Master Thesis, CTU in Prague, 2020, and references therein https://cds.cern.ch/record/2722145

Name and workplace of bachelor’s thesis supervisor:

doc. Dr. André Sopczak Institute of Experimental and Applied Physics CTU Prague

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **27.01.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

doc. Dr. André Sopczak	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature