Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica

Search for Beyond Standard Model neutral Higgs boson in the $\mu\mu$ channel with the CMS detector at LHC with a multivariate approach

Relatore:

Prof. Daniele Bonacorsi

Correlatori:

Dott. Stefano Marcellini
Dott. Federica Primavera
Dott. Tommaso Diotalevi

Presentata da:
William Korcari

Anno Accademico 2019/2020

Abstract

This thesis describes a study of a multivariate approach in the search for the neutral beyond-standard-model Higgs bosons produced in association with a b-quark pair and decaying into two muons. The study has been made on Monte Carlo samples reproducing the data collected by the CMS experiment during the 2016 proton-proton run of LHC at $\sqrt{s} = 13$ TeV. Two machine learning models, Boosted Decision Trees and Artificial Neural Networks, trained with two different strategies, *inclusive* and *per categories*, have been tested for seven signal mass hypothesis in the range between 140 and 1000 GeV. The statistical significance obtained in the four cases has been compared to the one obtained in a similar search, published by CMS, which adopted a cut-based approach. The multivariate approach has been found to achieve a better performance for all the tested mass hypothesis. The Boosted Decision Trees model trained with the *per categories* strategy has been found to give the best significance.

Sommario

Il lavoro descritto in questa tesi riguarda l'uso di tecniche di analisi multivariata applicate alla ricerca dei bosoni di Higgs neutri, previsti da teorie che estendono il Modello Standard, prodotti in associazione ad una coppia di quark b, che decadono in due muoni. Lo studio è stato svolto sui campioni Monte Carlo che riproducono le condizioni di presa dati del 2016 dell'esperimento CMS durante le collisioni protone-protone di LHC a $\sqrt{s} = 13$ TeV. Sono state esaminate sette ipotesi di massa del suddetto segnale nell'intervallo di massa tra 140 e 1000 GeV. Sono stati utilizzati due diversi modelli di machine learning, Boosted Decision Trees (BDT) e Artificial Neural Networks, seguendo due diversi tipi di training, uno fatto su un campione inclusivo, l'altro fatto su due categorie indipendenti a partire dal campione inclusivo. Le prestazioni dei modelli esaminati sono state valutate in termini di significatività statistica. I risultati sono stati messi a confronto con quelli ottenuti da un'analisi simile pubblicata da CMS basata su tagli di selezione. L'uso di tecniche di analisi multivariata si è rivelato in generale migliore nella discriminazione segnale fondo. In particolare il modello BDT con training per categorie ha raggiunto il miglior risultato in termini di significatività in tutte le ipotesi di massa studiate.

Contents

Introduction

The discovery of the Higgs boson with a mass of approximately 125 GeV in 2012 by ATLAS [1] and CMS [2] collaborations marked a triumph of the Standard Model (SM). However, the SM as it is now cannot explain problems like the matter-antimatter asymmetry, the origin of dark matter and dark energy, as well as the mass of the Higgs boson, which should diverge to infinity according to quantum field theory due to the vacuum radiative corrections to its mass.

Physicists, during the last half century, have come up with many different theories that attempt to overcome these problems, by extending the SM. One example is Supersymmetry which extends the particle spectrum of the SM. In the Higgs sector five physical Higgs boson arise: two neutral CP-even scalars h (whose properties should correspond to the observed 125 GeV Higgs boson) and H, one CP-odd scalar A and a charged-scalar pair $H^{\pm}$. Such particles have masses and couplings to fermions and bosons that depend on the specific model. Their masses, in particular, can be in the TeV range.

In this thesis, my work on the search for the neutral Higgs bosons decaying into two muons is presented. At the LHC center-of-mass energies, the main production mechanism for neutral supersymmetric Higgs bosons are the gluon fusion, in which the Higgs is produced by a virtual loop of top/bottom quarks, and the b-associated production where two real b-quarks are produced together with the Higgs. The latter is the only production mechanism studied in this thesis. The dimuon final state has a very clean signature and it is not so difficult to separate it from the background exploiting the CMS high efficiency on the muon identification.

The CMS collaboration has already published an analysis [3] on the search for neutral BSM Higgs bosons using the 35 fb^{-1} of data collected during the 2016 at $\sqrt{s} = 13$ TeV. That analysis adopts a cut-based categorization based on the presence/absence of b-jets. A further veto on the E_T^{miss} value allows an efficient reduction of the background.

A cut-based analysis, however, does not exploit all of those features that show a partial separation between signal and background. Moreover, a cut-based analysis is not optimized for all the possible mass hypotheses of the searched Higgs bosons. For these reasons, the goal of this thesis is to explore whether a multivariate analysis achieves a better separation between signal and background, increasing the sensitivity for this search.

All the steps of this work are discussed in the next chapters. The first four chapters are meant to give a theoretical and technical description, while the last one contains the description of the work done. Chapter 1 contains a brief introduction to the SM and the Higgs Mechanism. Then Supersymmetry and the Minimal

Supersymmetric SM are presented, along with some relevant scenarios. In Chapter 2 there is a description of the LHC and the CMS apparatus. Chapter 3 contains a description of the event reconstruction in CMS while Chapter 4 gives an overview of some concepts of machine learning. Finally the study performed and the obtained results are presented in Chapter 5.

1

Theoretical introduction

1.1 Brief introduction of the Standard Model

The Standard Model (SM) is a theory that very well describes all the known particles and their interactions. The constituents of matter can be mainly divided into quarks and leptons which can be categorized in two - so called - families.

Quarks and leptons can be categorized in three, so called, families. Concerning the leptons, the three groups are the following: the electron (e) and electron neutrino (ν_e), the muon (μ) and muon neutrino (ν_μ), the tau (τ) and tau neutrino (ν_τ). All the neutrinos are chargeless, while the other particles do not. Quarks are named as up (u) and down (d), charm (c) and strange (s), top (t) and bottom (b). The *up-type* quarks (u, c, t) have electric charge $+2/3$, while *down-type* quarks (d, s, b) have electric charge $-1/3$.

The interactions can be described as the exchange of point-like bosons. The electromagnetic and the strong interactions are mediated by photons (γ) and gluons (g) respectively. Both have mass equal to zero and spin 1. Weak interactions are mediated by $W^\pm$ and Z^0, spin 1 bosons with measured masses of about 80 and 90 GeV respectively [4]. A summary of the SM fundamental fermions and their interactions is shown in Table 1.1.

TABLE OF ELEMENTARY PARTICLES		
QUANTA OF RADIATION		
Strong Interactions	Eight gluons	
Electromagnetic Interactions	Photon (γ)	
Weak Interactions	Bosons W^+, W^-, Z^0	
Gravitational Interactions	Graviton (?)	
MATTER PARTICLES		
	Leptons	Quarks
1st Family	ν_e, e^-	$u_a, d_a, a = 1, 2, 3$
2nd Family	ν_μ, μ^-	$c_a, s_a, a = 1, 2, 3$
3rd Family	ν_τ, τ^-	$t_a, b_a, a = 1, 2, 3$
HIGGS BOSON		

Table 1.1: Standard Model as of today.

The Standard Model is a quantum field theory that can be described as a $SU(3) \times SU(2) \times U(1)_Y$ gauge symmetry. Each term pictures the invariance of one or more interactions under a specific transformation. The lagrangian of the SM can be schematically written as

$$\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{EW}} + \mathcal{L}_{\text{QCD}} \tag{1.1}$$

The two components can be obtained separately by requiring the free lagrangian to be locally invariant under a certain group of transformations.

The electroweak lagrangian is obtained by requiring the lagrangian of a free spinor field to be invariant under local $SU(2)_L \times U(1)_Y$ transformations. The subscript L means that the transformation acts on left-chiral fermions. The free lagrangian of a $SU(2)_L$ massless fermion doublet ψ can be written as

$$\mathcal{L}_{\text{free}} = \bar{\psi}\left(i\gamma^\mu \partial_\mu\right)\psi \tag{1.2}$$

This introduces four gauge fields, $\vec{W}_\mu = \left(W^1_\mu, W^2_\mu, W^3_\mu\right)$ for $SU(2)_L$ and B_μ for $U(1)_Y$. These allow to write the photon and weak bosons as a linear combination:

$$\begin{cases} W^\pm_\mu = \frac{W^1_\mu \pm i W^2_\mu}{2} & \text{for } W^\pm \text{ bosons} \\ Z_\mu = -\sin\theta_W B_\mu + \cos\theta_W W^3_\mu & \text{for the } Z^0 \text{ boson} \\ A_\mu = \cos\theta_W B_\mu + \sin\theta_W W^3_\mu & \text{for the photon} \end{cases} \tag{1.3}$$

where θ_W is the Weinberg angle. The gauge invariant lagrangian can be obtained with the substitution of the partial derivative ∂_μ with the covariant derivative

$$D_\mu = \partial_\mu + \frac{ig'}{2} Y B_\mu + \frac{ig}{2}\sigma_a W^a_\mu \tag{1.4}$$

where g and g' are the $SU(2)_L$ and $U(1)_Y$ gauge couplings, Y is the hypercharge and σ^a, with $a = 1, 2, 3$, are the Pauli matrices. Substituting (1.4) into (1.2) leads to

$$\mathcal{L}_{EW} = \bar{\psi} i\gamma^\mu \partial_\mu \psi - \frac{1}{4} W^{\mu\nu} W_{\mu\nu} - \frac{1}{4} B_{\mu\nu} B^{\mu\nu} \tag{1.5}$$

$B_{\mu\nu}$ and $W_{\mu\nu}$ are defined respectively as $B_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu$ and $W_{\mu\nu} = \partial_\mu W_\nu - \partial_\nu W_\mu + ig\left[W_\mu, W_\nu\right]$. The first term in (1.5) contains the free propagation of fermions and their interaction with bosons defined in (1.3). On the other hand, the last two terms describe free propagation and self interactions for W and Z and free propagation for photons. Since these are not invariant under local $SU(2)_L$ transformations, the mass terms for fermions and bosons are not allowed in (1.5). This does not agree with the measured masses of $W^\pm$ and Z^0 bosons, but the problem is fixed by the introduction of the Spontaneous Symmetry Breaking mechanism and of the Higgs boson (see Sec.1.2).

For what concerns the quantum chromodynamic QCD section of (1.1), that describes the interaction between quarks and gluons, the free lagrangian for massless quarks can be written as

$$\mathcal{L}_{free} = \bar{q}_j \left(i\gamma^\mu \partial_\mu\right) q_j \tag{1.6}$$

where q is the quark spinor field and the $j = 1, 2, 3$ index runs over the three color fields. When local invariance under $SU(3)$ transformations is required, eight gauge

fields G_μ^a, with $a = 1, 2, \ldots, 8$, need to be introduced. By substitution of the ∂_μ with the covariant derivative

$$D_\mu = \partial_\mu + i g_s T_a G_\mu^a \tag{1.7}$$

(where g_s is the strong coupling constant) the gauge invariant QCD lagrangian can be obtained:

$$\mathcal{L}_{QCD} = \bar{q}_j \left(i\gamma^\mu \partial_\mu\right) q_j + g_s \left(\bar{q}\gamma^\mu T_a q\right) G_\mu^a - \frac{1}{4} G_{\mu\nu} G^{\mu\nu} \tag{1.8}$$

As for the electroweak part, $G_{\mu\nu} = \partial_\mu G_\nu - \partial_\nu G_\mu + i g_s \left[G_\mu, G_\nu\right]$ is the gluon strength tensor, written as a function of $G_\mu \equiv T_a G_\mu^a$, where T^a are the eight generators of $SU(3)$. In (1.8), the first term is the free quark propagation, the second describes the quark-gluon interaction and the third term describes both gluon free propagation and gluon-gluon interactions.

1.2 Spontaneous Symmetry Breaking and Higgs Mechanism

The principle of local gauge invariance described in Section 1 works perfectly in the case of electromagnetic and strong interactions, but it constraints weak interaction gauge fields to be massless, which disagrees with experimental results.

In the Standard Model, the mass of a particle can be generated through a process called "spontaneous symmetry breaking" (SSB): in this case, a system can be written in a way where some symmetry rule is broken for perturbations around lowest-energy vacuum solutions (see Figure 1.1). When this happens, massless scalars called Goldstone bosons are generated. The process by which the unwanted massless Goldstone bosons are "eaten up" by a massive gauge particle, within the spontaneous symmetry breaking of a local gauge symmetry, is called *Higgs mechanism*. This mechanism is summarized in the following.

The key idea is to formulate the Higgs mechanism for (1.5) so that the $W^\pm$ and the Z^0 become massive and the photon remains massless. To do this, we introduce four real scalar fields ϕ_i arranged in $SU(2)_L \times U(1)_Y$ multiplets with weak hypercharge $Y = 1$ like follows

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} \text{ with } \quad \begin{matrix} \phi^+ \equiv \left(\phi_1 + i\phi_2\right)/\sqrt{2} \\ \phi^0 = \left(\phi_3 + i\phi_4\right)/\sqrt{2} \end{matrix} \tag{1.9}$$

and use them inside the Higgs potential

$$V(\phi) = m^2 \phi^\dagger \phi + \lambda \left(\phi^\dagger \phi\right)^2 \tag{1.10}$$

where, to obtain the "mexican hat" shape, we need $\lambda > 0$ and $m^2 < 0$. By choosing as vacuum expectation value

$$\phi \equiv \sqrt{\frac{1}{2}} \begin{pmatrix} 0 \\ v \end{pmatrix} \tag{1.11}$$

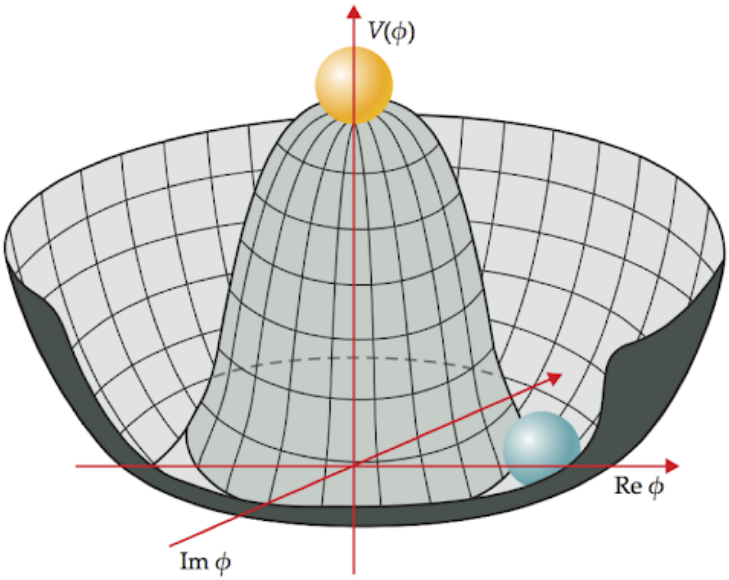


Figure 1.1: Graphical representation of the Higgs potential.

where $v^2 \equiv -m^2/\lambda$, we induce the symmetry breaking. The reason why this choice is suitable for the problem in hand is explained in [5]. As a consequence of this choice, three out of four generators of $SU(2)_L \times U(1)_Y$ are broken, resulting in three massless Goldstone bosons and a massive Higgs field. The Higgs field couples to the $\vec{W}_\mu$ and B_μ gauge fields associated with the $SU(2) \times U(1)_Y$ local symmetry through (1.4) in the kinetic term of the Higgs lagrangian

$$\mathcal{L}_{Higgs} = (D_\mu \phi)^\dagger (D^\mu \phi) - V(\phi) \tag{1.12}$$

As a result, the neutral and the two charged massless Goldstone degrees of freedom mix with the gauge fields corresponding to the broken generators of $SU(2)_L \times U(1)_Y$ and become the longitudinal components of the Z and W physical gauge bosons respectively. The Z and W gauge bosons acquire masses:

$$m_W^2 = \frac{g^2 v^2}{4}, \quad m_Z^2 = \frac{(g^2 + g'^2)\, v^2}{4} \tag{1.13}$$

, where g' and g are the the $U(1)_Y$ and $SU(2)_L$ coupling constants respectively defined as $g = e/\sin\theta_W$ and $g'/g = \tan\theta_W$. The fourth generator, associated with the conserved $U(1)_Y$ gauge symmetry, remains unbroken, and its corresponding gauge field, the photon, remains massless. Also the gluons, associated with eight unbroken generators of the $SU(3)_C$ gauge symmetry, remain massless. Hence, from the initial four degrees of freedom of the Higgs field, two are absorbed by the $W^\pm$ gauge bosons, one by the Z, and the last remaining degree of freedom, H, is the physical Higgs boson.

The fermions of the SM acquire acquire mass through renormalisable interactions between the Higgs field and the fermions, the Yukawa interactions, which respect the symmetries of the SM but generate fermion masses once SSB occurs. Considering

for example an $SU(2)_L$ lepton doublet $\begin{pmatrix} \nu_l \\ l \end{pmatrix}_L$, the Yukawa term can be written as:

$$\mathcal{L}_{Yukawa} = -\lambda_l \left[\left(\bar{\nu}_l, \bar{l}\right)_L \begin{pmatrix} \phi^+ \\ \phi^o \end{pmatrix} l_R + \bar{l}_R \left(\phi^-, \bar{\phi}^0\right) \begin{pmatrix} \nu_l \\ l \end{pmatrix}_L \right] \tag{1.14}$$

where $\bar{\phi}^0 \equiv \phi^{0^*}$, $\phi^- \equiv \phi^{+*}$ and λ_l is the Yukawa coupling constant. We then spontaneously break the symmetry by picking as lowest-energy vacuum solution:

$$\phi = \frac{1}{\sqrt{2}} e^{i\frac{\sigma_a a^a}{2\nu}} \begin{pmatrix} 0 \\ v + h \end{pmatrix} \tag{1.15}$$

where θ^a, for $a = 1, 2, 3$, and the Higgs field are the four Goldstone bosons. These last ones can be removed with the appropriate gauge transformation, and only the Higgs field remains. The (1.14) becomes:

$$\mathcal{L}_{Yukawa} = -\frac{\lambda_l}{\sqrt{2}} v \left(\bar{l}_L l_R + \bar{l}_R l_L\right) - \frac{\lambda_l}{\sqrt{2}} \left(\bar{l}_L l_R + \bar{l}_R l_L\right) h \tag{1.16}$$

A similar procedure can be applied to quarks in order to generate a Yukawa term of the same form. It is worth pointing out that, for a fermion f, the generated mass term is:

$$m_f = \frac{\lambda_f v}{\sqrt{2}} \tag{1.17}$$

The consequence is that Higgs boson couplings to fermions are set by their masses, and more specifically the strength of the coupling is linearly proportional to the fermion mass.

1.3 Supersymmetry

The SM provides an accurate description of the phenomenology of the particle world at the energies currently investigated. However, the SM as it is now cannot explain problems like the matter-antimatter asymmetry, the origin of dark matter and dark energy, as well as the mass of the Higgs boson, which should diverge to infinity according to quantum field theory due to the vacuum radiative corrections to its mass. Physicists, during the last half century, have come up with many different theories that attempt to overcome these problems, by extending the SM. One example is Supersymmetry (SUSY). A supersymmetric version of the SM basically means that the lagrangian is now not only gauge invariant under SM gauge group G_{SM}, but also supersymmetric invariant. SUSY needs to be renormalizable and anomaly free, just like SM is. This extension of the SM is built by replacing every standard model matter field by a chiral superfield and every vector field by a vector superfield. Thus the existing particle spectrum of the SM is doubled. The particle spectrum of SUSY and their transformation properties under G_{SM} is reported in Fig.1.2

The scalar partners of the quarks and leptons are typically named as 's'quarks and 's'leptons and together are called sfermions. These are typically represented by a $\sim$ on their SM counterparts. The index i stands for the generation index.

$$Q_i \equiv \begin{pmatrix} u_{L_i} & \tilde{u}_{L_i} \\ d_{L_i} & \tilde{d}_{L_i} \end{pmatrix} \sim \left(3,\ 2,\ \frac{1}{6}\right) \qquad U_i^c \equiv \begin{pmatrix} u_i^c & \tilde{u}_i^c \end{pmatrix} \sim \left(\bar{3},\ 1,\ -\frac{2}{3}\right)$$

$$D_i \equiv \begin{pmatrix} d_i^c & \tilde{d}_i^c \end{pmatrix} \sim \left(\bar{3},\ 1,\ \frac{1}{3}\right)$$

$$L_i \equiv \begin{pmatrix} \nu_{L_i} & \tilde{\nu}_{L_i} \\ e_{L_i} & \tilde{e}_{L_i} \end{pmatrix} \sim \left(1,\ 2,\ -\frac{1}{2}\right) \qquad E_i \equiv \begin{pmatrix} e_i^c & \tilde{e}_i^c \end{pmatrix} \sim (1,\ 1,\ 1)$$

Figure 1.2: Particle Spectrum of SUSY.

In SUSY a vector superfield has to be added that transforms as the adjoint under the gauge group action. Each vector superfield contains the gauge boson and its corresponding super partner, called gaugino. Thus in SUSY there are the vector superfields reported in Fig. 1.3.

$$\begin{aligned} V_s^A &: \begin{pmatrix} G^{\mu A} & \tilde{G}^A \end{pmatrix} \sim (8,1,0) \\ V_W^I &: \begin{pmatrix} W^{\mu I} & \tilde{W}^I \end{pmatrix} \sim (1,3,0) \\ V_Y &: \begin{pmatrix} B^{\mu} & \tilde{B} \end{pmatrix} \sim (1,1,0) \end{aligned}$$

Figure 1.3: Vector superfields of SUSY.

The G and $\tilde{G}$ represent the gluonic fields and their superpartners galled gluinos (A runs from 1 to 8). The W are the $SU(2)_L$ gauge bosons and their superpartners "Winos" (index I taking values from 1 to 3) and the B represent the $U(1)_Y$ gauge boson and its superpartner "Bino". Together are called "gauginos".

At least two Higgs superfields are required to complete the spectrum: one giving masses to up-like quarks and the other giving masses to down-type and charged leptons. These two Higgs superfields have the transformation properties reported in Fig. 1.4.

1.4 The MSSM Higgs boson

The Minimal Supersymmetric Standard Model (MSSM) is one of the most studied model of SUSY. Such model is called "Minimal" indicating minimal choice of article spectrum required to make it work[6]. The MSSM's Higgs sector consists of two $SU(2)_L$ doublets, H_1 and H_2, whose relative contribution to electroweak symmetry breaking (EWSB) is determined by the ratio of vacuum expectation values (vevs) of their neutral components, $tan\beta \equiv v_2/v_1$. At tree level, CP is conserved in the Higgs sector of the MSSM, and the spectrum of physical Higgs

$$H_1 \equiv \begin{pmatrix} H_1^0 & \tilde{H}_1^0 \\ H_1^- & \tilde{H}_1^- \end{pmatrix} \sim \left(1,\ 2,\ -\frac{1}{2}\right)$$

$$H_2 \equiv \begin{pmatrix} H_2^+ & \tilde{H}_2^+ \\ H_2^0 & \tilde{H}_2^0 \end{pmatrix} \sim \left(1,\ 2,\ \frac{1}{2}\right)$$

Figure 1.4: The Higgs field has a fermionic partner, higgsino(represented with a $\sim$ on them), which contributes to the anomalies of the SM. At least two such fields with opposite hypercharges are required to cancel out the anomalies of the SM.

bosons consists of two neutral CP-even scalars, of which we denote the lighter as h and the heavier as H, one CP-odd scalar, A, and a charged-scalar pair, $H\pm$. Supersymmetry (SUSY) imposes relations between the quartic Higgs couplings and the gauge couplings, ensuring that the tree-level masses of all Higgs bosons can be expressed in terms of the gauge-boson masses, M_Z and M_W , plus two additional parameters which can be chosen as the CP-odd scalar mass, M_A (or alternatively the charged Higgs boson mass, $M_{H\pm}$), and $tan\beta$. In particular, the tree-level mass of the lighter CP-even scalar h is bounded from above by $M_Z|cos2\beta|$. However, radiative corrections – especially those involving top and bottom quarks and their scalar partners, the stops and the sbottoms – can significantly alter the tree-level predictions for the Higgs masses, allowing for $M_h \simeq 125$ GeV but bringing along a dependence on many free parameters of the MSSM. Moreover, for specific choices of those parameters, radiative corrections to the mixing between the scalars can also allow for scenarios in which the heavier mass eigenstate, H, is the one with $M_H \simeq 125$ GeV and roughly SM-like couplings. In the presence of complex parameters in the MSSM Lagrangian, radiative corrections can break CP in the Higgs sector and induce a mixing among the two CP-even scalars, h and H, and the CP-odd scalar, A, such that beyond tree-level they combine into three neutral mass eigenstates which we denote as h_a (with a = 1, 2, 3). The large number of free parameters complicates the task of interpreting within the MSSM both the properties of the observed Higgs boson and the results of the ongoing searches for additional, non-standard Higgs bosons [7].

At the LHC center-of-mass energies, the main production mechanism for neutral MSSM Higgs boson is the gluon fusion (GF). Here the Higgs is produced by a virtual loop of top/bottom quarks. The other main mechanism is in association with real b-quarks, so called b-associated (BA). Diagrams for this processes are showed in Fig. 1.5.

The GF mechanism has more relevance for small to medium values of $tan\beta$, while at large values ($tan\beta > 30$) it is predicted that the Higgs coupling with down-type fermions is enhanced. That leads the BA production mechanism to be dominant under these conditions.

For the same reason, the coupling of the neutral Higgs boson to charged leptons

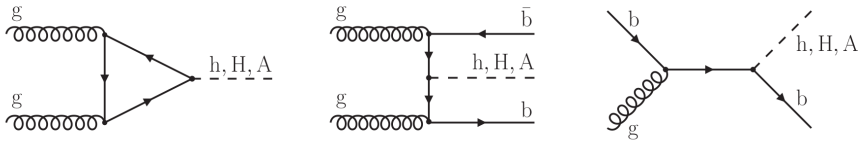


Figure 1.5: Leading order Feynman diagrams for the production of the MSSM Higgs boson: gluon fusion production (left) and b-associated production (middle and right).

is enhanced. Although the cross section times the branching ratio for the dimuon final state is smaller of a factor 10^3 than the $\tau^+\tau^-$ one, the $\mu^+\mu^-$ final state is fully reconstructed and the invariant mass can be measured taking advantage of the excellent muon momentum resolution of the CMS detector. This makes the dimuon final state an interesting probe for the MSSM.

The common experimental signature of the GF and BA processes is a pair of muon with opposite charge and high transverse momentum (p_T), and a small imbalance of the vectorial sum of the total p_T of the event. On top of that, the b associated processes are distinguished by the presence of additional jets originating from b quarks, while the events with only so called *light jets* (meaning jets originated from lighter quarks or gluons) are sensitive to the GF production mechanism.

In the light of the latest LHC data on the discovered Higgs-like boson, and given the fact that the MSSM contains many independent parameters which makes it a difficult task to perform a full scan, there have been many studies which lead to several benchmarks that could fit the observed Higgs boson as well as be tested at the future LHC with higher luminosity [8].

1.4.1 The m_h^{mod+} Scenario

The m_h^{mod+} scenario is a modification of the time-honored m_h^{max} scenario (also called maximal mixing scenario), which was originally defined to give conservative exclusion bounds on $tan\beta$ in the context of Higgs boson searches at LEP. The m_h^{max} scenario was introduced in order to maximize the value of M_h by incorporating large radiative correction effects for a large $M_A \gg M_Z$ mass, fixed value of $tan\beta < 8$ and large SUSY scale of the order 1 TeV. However, this scenario predicts M_h to be much higher than the observed Higgs boson mass, due to the large mixing in the scalar top sector. Hence, the maximal mixing scenario has been modified, by reducing the amount of scalar top mixing, such that the mass of the lightest Higgs state, M_h, is compatible with the mass of the observed Higgs boson within ± 3 GeV in a large fraction of the considered parameter space. In fact, modifications of the m_h^{max} scenario can be done in two ways depending on the sign of $(A_t - \mu \cot\beta)/M_{SUSY}$, leading to an m_h^{mod-} and m_h^{mod+}. It has been demonstrated that when m_h^{mod+} is confronted with LHC data, there is a substantial region in the plane (M_A, $tan\beta$) with $tan\beta > 7$ for which the light CP-even Higgs mass is in a good agreement with the measured one at the LHC, hence our choice of this scenario.

1.4.2 The hMSSM Scenario

In the previous scenario, one need to input $tan\beta$, M_A and also the other SUSY parameters to get the Higgs and SUSY (mass and coupling) spectrum. Taking into account the theoretical uncertainty of the order 3 GeV on the Higgs mass, which could originate from unknown high order loop effects, a light CP-even Higgs boson with a mass in the range [122, 128] GeV would be an MSSM candidate for the observed Higgs-like particle. However, plenty of points on the (M_A, $tan\beta$) plane would correspond to one configuration of M_h mass. To avoid this situation, the hMSSM benchmark was introduced. In this scenario, the light CP-even Higgs state is enforced to be 125 GeV while setting the SUSY mass scale M_{SUSY} to be rather high (i.e., $>$ 1 TeV) in order to explain the non-observation of any SUSY particle at colliders. The hMSSM setup thus describes the MSSM Higgs sector in terms of just M_A and $tan\beta$, exactly like for tree-level predictions, given the experimental knowledge of M_Z and M_h. In this scenario, therefore, the dominant radiative corrections would be fixed by the measured experimental value of M_h which in turn fixes the SUSY scale. It defines a largely model-independent scenario, because the predictions of the properties of the MSSM Higgs bosons do not depend on the details of the SUSY sector, somewhat unlike the previous case, wherein squark masses are fine-tuned to obtain $M_h \simeq 125$ GeV [8].

1.4.3 Model independent search

In the model-independent search no constraint is put neither on the cross section of the signal nor on the width of the signal. We select a production mechanism (e.g. b-associated or gluon fusion) and decay channel ($\mu\mu$) and then the neutral ϕ boson is searched as a single resonance with mass M_ϕ assuming a narrow width (i.e. dominated by the experimental width). So, differently from the model-dependent search in which limits in the (M_A, $tan\beta$) plane are set assuming cross sections and Branching Ratios (B.R.) foreseen by the MSSM model, in the model-independent search we put a limit on the $\sigma_x \times B.R.$ ($M_\phi \to \mu\mu$).

2

The Large Hadron Collider and the CMS experiment

2.1 An introduction to the LHC

The Large Hadron Collider (LHC) is the biggest and most powerful particle accelerator in the world. It is part of the CERN accelerator complex (see Figure 2.1) in Geneva and consists of a circular 27 km circumference ring designed to accelerate protons and heavy ions, divided into eight independent sectors. In the accelerator, the particles travel in two beams on opposite directions and in extreme vacuum conditions (see Section 2.1.1). Their trajectory is well defined by superconductive electromagnets (see Section 2.1.2), keeping them in their trajectory.
In Table 2.1 are reported some of the features of the LHC.

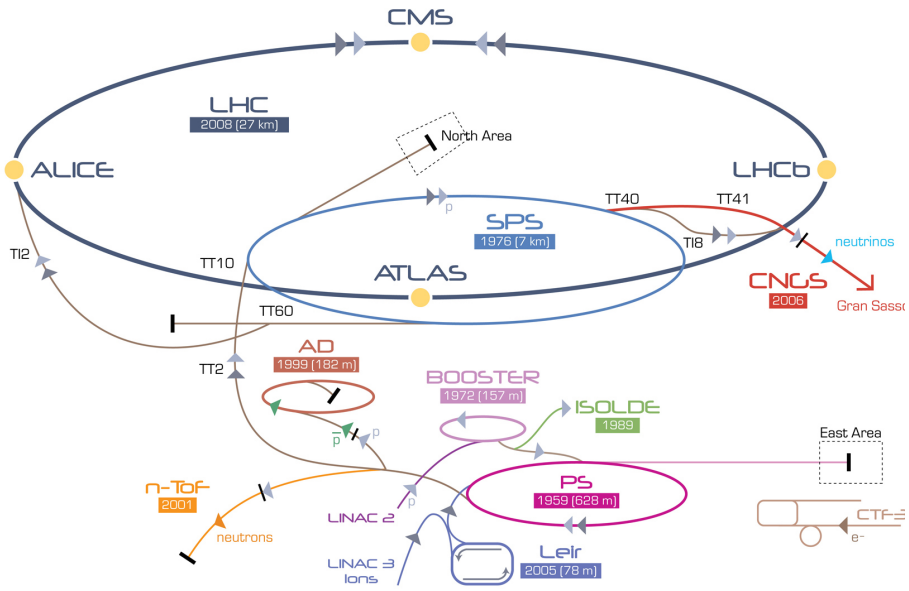


Figure 2.1: The accelerators chain at CERN.

Table 2.1: Main technical parameters of LHC.

Quantity	value
Circumference (m)	26 659
Magnets working temperature (K)	1.9
Number of magnets	9593
Number of principal dipoles	1232
Number of principal quadrupoles	392
Number of radio-frequency cavities per beam	16
Nominal energy, protons (TeV)	6.5
Nominal energy, ions (TeV/nucleon)	2.76
Magnetic field maximum intensity (T)	8.33
Project luminosity ($cm^{-2}\,s^{-1}$)	2.06×10^{34}
Number of proton packages per beam	2808
Number of proton per package (outgoing)	1.1×10^{11}
Minimum distance between packages (m)	∼7
Number of rotations per second	11 245
Number of collisions per crossing (nominal)	∼20
Number of collisions per second (millions)	600

2.1.1 The vacuum system

The LHC vacuum system [9] has a double purpose: the first is to make sure that collisions between beam particles and air molecules inside the ducts are avoided, creating an extreme vacuum condition (10^{-13}atm), as empty as interstellar space; the other is to deny heat exchange between components that need extremely low temperatures in order to perform properly and maximize the efficiency.
The vacuum system is composed by three independent parts:

- an isolated vacuum system for cryomagnets;
- an isolated vacuum system for Helium distribution line;
- a vacuum system for beams.

2.1.2 Electromagnets

The electromagnets [10] are designed to drive beams along the pipes, modifying single particles trajectories as well as aligning them so to maximise collision probability. More than fifty distinct kind of magnets compose the LHC, totalling approximately 9600 magnets. The main dipoles generate a magnetic field with a maximum intensity of 8.3 T. In order to reach such intensities, a current of 11 850 A is required. To minimize power dissipation, superconducting magnets are employed. A system of liquid helium distribution allows to keep the magnets at a temperature of approximately 1.9 K. At this incredibly low temperatures, below what required to operate in conditions of superconductivity, helium becomes also super-fluid: this translates in a high thermal conductivity, thus an efficient refrigeration system for the magnets.

2.1.3 Radiofrequency cavities

Radiofrequency (RF) cavities [11] [12] [13] are metallic chambers where an electromagnetic field is applied. Their primary purpose is to divide protons in packages and to focus them at the collision points, in order to guarantee an high luminosity and thus a large number of collisions.
Particles passing through the cavity, feel the overall force due to electromagnetic fields and are pushed forward through the accelerator. In this scenario, the ideally timed proton with exactly the right energy, will see zero accelerating voltage when LHC is at nominal energy while protons with slightly different energies will be accelerated or decelerated ending up sorting particle beams into "bunches". LHC has eight RF cavities per beam, each of which provides 2 MV at 400 MHz. The RF cavities work at 4.5 K and are grouped into four cryomodules.
At regime conditions, each proton beam is divided into 2808 bunches, containing about 10^{11} protons each. When far from the collision point, the bunches are a few cm long and 1 mm wide, and are compressed down to 16 nm near the latter. At maximum luminosity, packages are divided in time by 25 ns, resulting in about 600 million collisions/second.

2.2 The CMS experiment

CMS [14, 15] is a general-purpose detector at LHC. It is built around a huge solenoid magnet with a cylindrical shape, able to reach a 3.8 T magnetic field. Its main characteristics are illustrated in Table 2.2.

The CMS main purpose is to explore the p-p physics at the TeV scale including precision measurements of the Standard Model and search for New Physics (NP). Its cylindrical design is built on multiple layers. Each one of them is dedicated to the detection of a specific kind of particle.
The CMS collaboration consists in over 4000 particle physicists, engineers, computer scientists, technicians and students from around 200 institutes and universities coming from more than 40 countries. Figure 2.2 shows an image of the CMS experiment.

Table 2.2: CMS detector specifications:

Dimensions	length: 21 m, height: 15 m, width: 15 m
weight	12 500 tons
Design	barrel plus end caps
Cost of materials	500 MCHF
Location	Cessy, France

2.2.1 The CMS detector: concept and structure

Ahe CMS detector is composed by multiple layers, as showed in Figure 2.3. Each of them is designed to trace and measure the physical properties of different

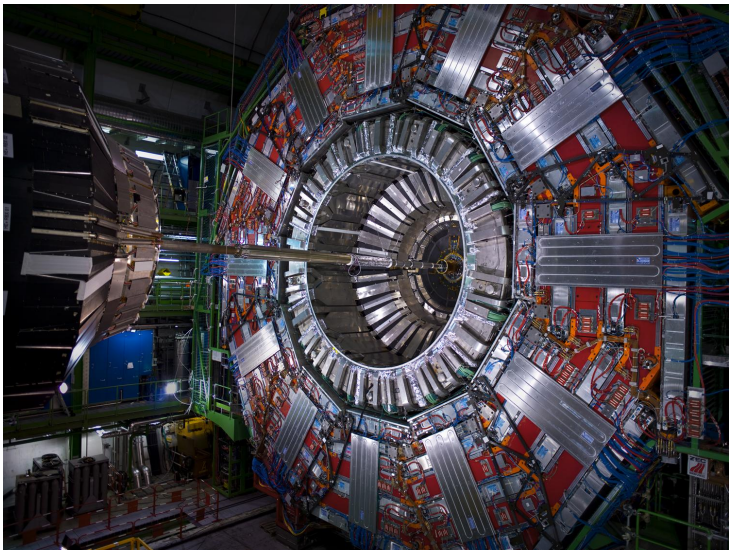


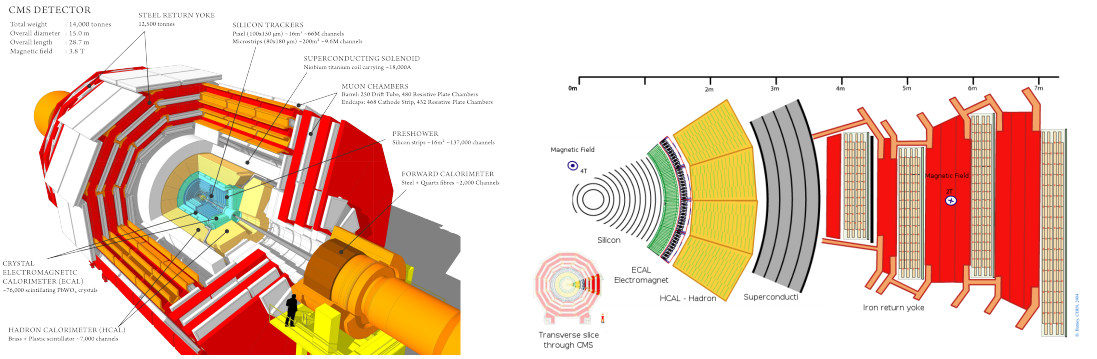
Figure 2.2: A view of the CMS experiment, with one of the endcaps removed.

kinds of subatomic particles. The entire structure is surrounded by a huge solenoid based on superconductive technologies, operating at 4.4 K that generates a 3.8 T magnetic field.
In CMS a right handed coordinate system is defined centred at the nominal collision point: the *x-axis* points radially inward to the center of the accelerator ring, the *y-axis* points upward and the *z-axis* is parallel to the beam1 pipe (in direction of the Jura mountains). The polar angle θ is measured from the *z-axis* using a $0 \leq \theta \leq \pi$ range, while the azimuthal angle ϕ is measured in the *x-y* plane from the *x-axis* in a $0 \leq \phi \leq 2\pi$ range. The *pseudorapidity* η defined as:

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

is usually preferred to the polar angle.

The first and inner layer of the CMS detector is called Tracker [15, pp. 26–89]: it is made entirely of silicon and it is able to reconstruct the paths of high-energy muons, electrons and hadrons as well as reconstruct secondary vertices from the decay of very short-lived particles, All of that, with a resolution of 10 nm.
The second layer consists of two calorimeters: the Electromagnetic Calorimeter (ECAL) [15, pp. 90–121] and the Hadron Calorimeter (HCAL) [15, pp. 122–155] arranged serially. The first one measures the energy deposited by photons and electrons, while the HCAL measures the energy deposited by hadrons.
Unlike the Tracker, which scarcely interacts with passing particles, the calorimeters are designed to absorb them.
Finally, after the superconducting magnet, a series of muon detectors [15, pp. 162–246] are able to track muon particles, escaped from calorimeters. The lack of energy and momentum from collisions is assigned to the electrically neutral and weakly-interacting neutrinos.



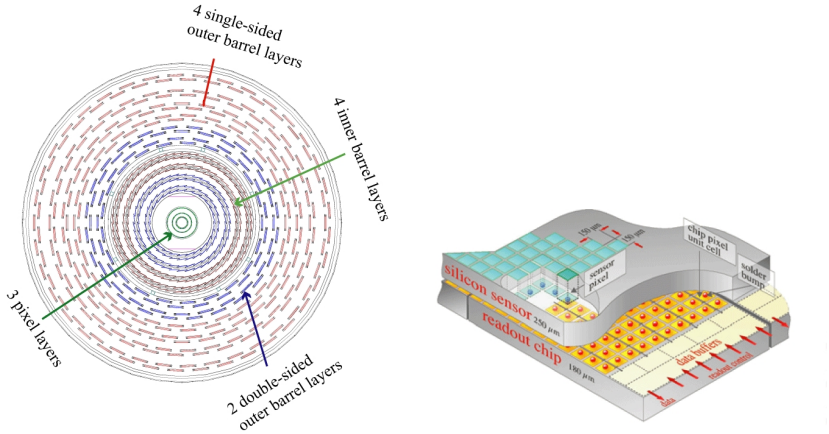
(a) Section view.

(b) View along the plane perpendicular to beam direction.

Figure 2.3: Compact Muon Solenoid.

Tracker

The Tracker is a crucial component in the CMS detector as it measures particles momentum through their path: the greater the curvature radius across the magnetic field the greater their momentum. As said previously, the Tracker can reconstruct muons, electrons and hadrons tracks as well as the ones produced by short-lived particles decay, like the quark beauty. It has a low degree of interference with particles and it is highly resistant to radiations. Being located in the inner part of the detector, it is highly irradiated.
This detector (Figure 2.4a) is entirely made of silicon: internally there are three levels of pixel detectors, after that particles pass through 10 layer of strip detectors, until a 130 cm radius from the beam pipe.
The pixel detector, Figure 2.4b, is constituted of about 120 millions of pixels, with three levels of respectively 4, 7 e 11 cm radius. The flux of particles at this radius is maximum: at 8 cm is about 10^6 particles/$cm^2 \cdot s$. Each level is divided into small units containing a silicon sensor of 150 µm × 100 µm. When a charged particle passes through one of this units, the amount of energy emits an electron with the consequent creation of an hole. This signal is than received by a chip which amplifies it. In the end, it is possible to reconstruct a 3-D image using bi-dimensional layers for each level.
The power absorption must be kept at minimum, because each pixel absorbs about 50 µW with an amount of power not irrelevant; for this reason pixels are installed in low temperature pipes.
Strip detectors, instead, consist of ten layers. This section of the Tracker contains 10 million detector strips divided into 15 200 modules, scanned by 80 000 chips. Each module is made up of three elements: a set of sensors, a support structure and the electronics necessary to acquire data. The sensors have a high response and a good spatial resolution that allows to receive many particles in a restricted space; they can detect electrical currents generated by interacting particles and send collected data. Even this section of the detector is maintained at low temperature (−20 °C), in order to "freeze" silicon structure damages from radiations and prevent it from perpetuating.



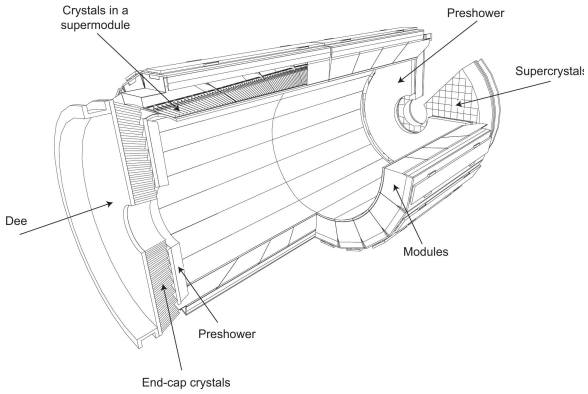
(a) Tracker layers from a point of view perpendicular to beams.

(b) CMS Silicon pixel detector.

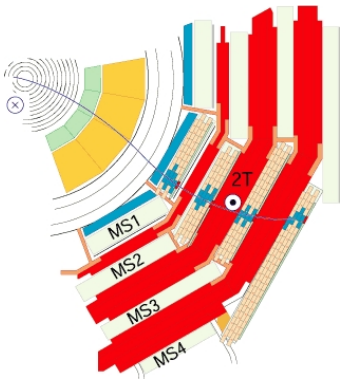
Figure 2.4: Graphical depiction of the CMS Tracker subdetector.

Calorimeters

In the CMS detector there are two types of calorimeters that measure the energy of electrons, photons and hadrons.
Electrons and photons are stopped and detected by the electromagnetic calorimeter (ECAL). The measurement is performed inside the strong magnetic field, with an high level of radiation and in 25 ns from one collision to another. Lead tungstate crystal ($PbWO_4$) is made primarily of metal and is heavier than stainless steel, but with a touch of oxygen in this crystalline form it is highly transparent and scintillates when electrons and photons pass through it, i.e. it produces light in proportion to the particle's energy. These high-density crystals produce light in fast, short and well-defined photon bursts that allow for a precise, fast and fairly compact detector. Photodetectors designed to work within the high magnetic field are glued onto the back of each of the crystals to detect the scintillation light and convert it to an electrical signal that is amplified and sent for analysis. The ECAL is divided into a cylindrical body called "barrel" and the two ends called "endcaps" (see Figure 2.5a) creating a layer between the Tracker and the other calorimeter.
Hadrons, instead, are detected by the hadron calorimeter (HCAL) specifically built for measuring strong-interacting particles. The HCAL is made up with a series of highly absorbent layers and each time a particle produced from various decays crosses a layer, a blue-violet light is emitted. This light is than absorbed by optic cables of about 1 mm diameter, shifting wavelength into the green region of the electromagnetic spectrum and finally converted to digital data by photodetectors. The optic sum of light produced along the path of the particle corresponds to a measure of its own energy, and is performed by specifically designed photo-diodes. The HCAL also provides tools for an indirect measurement of non-interacting particles like neutrinos. During hadrons' decay, new particles can be produced which may leave no sign in any detector at CMS. To avoid that, HCAL is hermetic i.e. it captures each particle coming from the collision point. That allows the detection of "invisible" particles, through the violation of momentum and energy conservation.



(a) ECAL layout showing crystal's modules.



(b) A muon, in the plane perpendicular to beams, leaving a curve trajectory into the four muon stations.

Figure 2.5: The *Electromagnetic CALorimeter* and the *muons detectors*.

Unlike ECAL, HCAL has two more sections called "forward sections", located at the ends of CMS. These are designed to detect particles moving with a low scattering angle. These sections are built with different materials, with the purpose of making them more resistant to radiations since they receive much of the beam energy.

Muon detectors

The muon detectors of CMS are placed outside the solenoid since muons are highly penetrant and most of them are not stopped by the previous calorimeters. Their momentum is measured by recreating trajectories along four muons stations interspersed with the iron of the ferromagnets return yoke, shown in red in Figure 2.5b, and then synchronizing data with those obtained from the Tracker. There are 1400 muons chambers: 250 "drift tubes" (DTs) and 540 "cathode strip chambers" (CSCs) tracking particles and acting at the same time as a trigger, while 610 "resistive plate chambers" (RPCs) form a further trigger system.

2.2.2 Muon trigger

At LHC proton bunches collide every 25 ns. At the design luminosity of $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ the proton-proton interaction rate exceeds 1 GHz. Only a small fraction of these collisions contain events of interest to the CMS physics program, and moreover only a small fraction of those can be stored for later offline analysis. This selection task is performed by the trigger system. The trigger system is expected to be able to select only the interesting events for offline storage from the bulk of the inelastic collision events.

The CMS trigger system is organized in two levels [16]: the first level, also referred to as Level-1 (L1), is implemented in custom hardware and selects events containing candidate objects; the second level is based on an array of commercially available computers running high-level physics algorithms and it is called High-Level

Trigger (HLT). The L1 trigger selection cuts allow to restrict the output event rate down to 100 KHz, the upper limit imposed by the CMS readout electronics. The HLT, implemented in software, further refines the purity of the physics objects, and selects an average rate of 400 Hz for offline storage.

The L1 trigger system is hardware-based and has a fixed latency. Within 4 μs of a collision, the system must decide if an event should be selected to undergo the next selection steps or if it has to be rejected, using information from the calorimeter and muon detectors. The trigger primitives (TP) from ECAL and HCAL and from muon detectors are processed in several steps. The combined event information is used by the Global Trigger (GT) to make the final decision. In the GT stage a menu of triggers is available, i.e. a set of selection requirements applied to the final list of objects. These selection requirements are tuned in order to meet the physics data-taking goals.

The Level-1 muon trigger system

A local trigger system processes the information coming from DTs, CSCs and RPCs to build full muon tracks in each system. The information is then combined at the Global Muon Trigger (GMT) stage, where final muon trigger candidates are provided for the HLT final step.

Local trigger track segments are formed within DT and CSC and then transmitted to respective track finders.

In the DT case, the reconstruction of track segments is performed by electronics installed on the detector. The DT Track Finder (DTTF) reconstructs the muon track candidates and assigns a transverse momentum value to the track candidates. The track reconstruction process exploits both information on segment parameters, e.g. position and bending angle, and reconstruction quality parameters, such as the number of DT layers used to build a track segment. The reconstructed local segments are then stored for each DT sector.

In the CSC case, segments are built separately from the cathode and anode hits of a detector. Because of the CSC structure, cathode-based local segments provide information on the azimuthal position, while anode-based segments provide information on the radial distance from the beam line and precise timing information. These segments are then correlated if both of them exist in the same chamber. The reconstructed local segments are then stored and finally transmitted via optical fiber to the CSC track finder.

The hits from the RPC chambers are used directly for muon trigger candidate recognition. The RPC trigger is based on the spatial and temporal coincidence of hits in different layers. The trigger logic compares the pattern of signals from all RPC chamber layers to predefined hit patterns in order to find muon candidates. After this procedure the muon candidate is provided with an estimate of the p_T, charge, η, and ϕ.

Muon candidates built by the track finder systems from DTs and CSCs and from the RPC trigger system are sent to the GMT. Synchronization, merging, duplicate canceling and p_T assignment of the muon candidates are the main tasks performed by the GMT, as well as sorting muon candidates according to a programmable rank and quality assignment to outgoing candidates. Candidates with the highest rank

are finally sent to the global CMS trigger.

High Level Trigger

The event selection at the HLT stage is based on software algorithms mirroring the offline ones, although less sophisticated to match the time and CPU constraint of the trigger selection. For each event, information coming from all CMS subdetectors is exploited in order to establish if the event must be rejected or kept for further analysis. The algorithm for the muon identification and the determination of its transverse momentum is composed of two main steps: Level-2 (L2), which uses information from the muon system only, and Level-3 (L3), which adds the information from the inner tracker.

At the L2 stage, muon tracks are reconstructed in the muon spectrometer. The reconstruction starts from seeds built from patterns of DT and CSC segments and confirmed by the L1 decision. Once seeds are determined, tracks are then built using a Kalman filter technique. The set of all reconstructed tracks is then checked to remove all possible duplicates of the same muon candidate. This operation looks for the presence of shared hits in different muon track candidates. The interaction point position is used in order to constrain the track parameters and to improve the transverse momentum. The number of successfully reconstructed muons and their transverse momentum are then used to filter the event.

The L3 muon trigger consists of three main steps: seeding, track reconstruction in the tracker and a final combined fit in both tracker and muon system. The seeding operation starts from the L2 information. Three different algorithms are available for the seeding task. In order to reduce the CPU time the faster algorithm of the three is executed first. Slower algorithms are only called if the faster ones fail to reconstruct a L3 muon.

The second step of the L3 is accomplished using a Kalman filter technique starting from seeds obtained in the previous step. Final combined track fit is performed by a comparison of the reconstructed track in the tracker and the L2 muon track propagated to a common surface. Several parameters are used in order to evaluate the compatibility of the two tracks, such as their separation, directions or relative goodness-of-fit χ^2. The number of successfully reconstructed L3 muons and their transverse momentum is then used to filter the event.

Each set of algorithmic processes, as well as their specific execution order, that is used both to reconstruct the muon and to select the event constitutes what is called an *HLT path*.

3

Event reconstruction in CMS

The reconstruction of *physics objects* (electrons, muons, photons, charged and neutral hadrons, etc.) is based on correlating signals coming from the various subdetectors. In CMS this task is performed by the so-called Particle Flow (PF) algorithm [17]. A particle traveling through the detector is expected to produce signals in the various CMS subdetectors. These could give a track generated by a charged particle, energy clusters in the calorimeters and/or a muon track. The algorithm can test the association between any pair of such elements in the event. The computing time of the algorithm is proportional to the square of the number of particles, so the linking procedure is restricted to the nearest neighbors in the (η, ϕ) plane. The algorithm eventually provides a distance between each linked element that can be used in order to quantify the quality of the association. A *PF block* is typically formed by a maximum of three elements linked together directly or indirectly, and it constitutes the input of the identification algorithm.

3.1 Track reconstruction

Track reconstruction in the inner tracker starts from the reconstructed hits. Multiple iterations of the combinatorial Kalman filter algorithm [18] are used. The basic idea of iterative tracking is that the initial iterations search for tracks that are easiest to find (e. g. of relatively large p_T and produced near the interaction region). After each iteration, hits associated with tracks are removed, hence reducing the combinatorial complexity, and simplifying subsequent iterations for more difficult classes of tracks (e. g. low-p_T, or displaced tracks).

Each tracking iteration consists of into four steps: seed generation, pattern recognition, ambiguity resolution and final track fit.

The seeds define the starting trajectory parameters and associated uncertainties of potential tracks. In the quasi-uniform magnetic field of the tracker, charged particles follow helical paths and therefore five parameters are needed to define a trajectory. Extraction of these five parameters requires either three 3-D hits, or two 3-D hits and a constraint on the origin of the trajectory based on the assumption that the particle originated near the beam spot. The seeds are constructed in the inner part of the tracker and the track candidates are built outwards.

Starting form a coarse estimate of the track parameters provided by the seed, at

each iteration the track parameters are updated combining the extrapolated track state to the next layer and the actual recorded hit. Several hits on the new layer may be compatible with the track, so a number of new trajectories are created and propagated. The maximum number of propagated tracks is a tunable parameter of the algorithm that allows to control the computing time.

Ambiguities in track finding arise because a given track may be reconstructed from different seeds, or because a given seed may result in more than one trajectory candidate. These ambiguities must be resolved in order to avoid double counting of tracks. The ambiguity resolution is based on the fraction of hits which are shared between two trajectories. If this fraction exceeds a value of 0.5 the track with the least number of hits is discarded. If the number of hits is the same, the track with the highest χ^2 value is discarded.

For each trajectory, the building stage results in a collection of hits and in an estimate of the track parameters. In the final stage, the trajectory is refitted using a least square approach, implemented as a combination of a standard Kalman filter and smoother.

The CMS tracking code is continuously evolving to make its computing load compatible with the increasing instantaneous luminosity of LHC, resulting in an increasing number of primary vertices and tracks per bunch crossing. Considering the proton-proton inelastic cross section to be $\sigma^{pp}_{inel} = 69.2$ mb, the mean number of interactions per bunch crossing was 23 for the 2016 data taking period, and 32 for the 2017 and 2018 data taking periods. The number of overlapping proton-proton collisions per bunch crossing is usually referred to as *pile-up*.

3.2 Primary vertex reconstruction

The goal of primary vertex reconstruction is to measure the location, and the associated uncertainty, of proton-proton interaction vertices in each event using the available reconstructed tracks. It consists of three steps:

- selection of the tracks;
- clustering of the tracks that appear to originate from the same interaction vertex;
- fitting for the position of each vertex using its associated tracks.

The selection involves choosing tracks consistent with being produced promptly in the primary interaction region. This is done by imposing requirements on the maximum value of significance of the impact parameter relative to the center of the beam spot, the number of strip and pixel hits associated with a track and the normalized χ^2 from a fit to the trajectory.

The selected tracks are then clustered on the basis of their z-coordinates at their point of closest approach to the center of the beam spot. The clustering algorithm must balance the efficiency for resolving nearby vertices in cases of high pileup against the possibility of accidentally splitting a single interaction into more than one cluster of tracks.

After identifying the candidate vertices based on the clustering algorithm, only candidates containing at least two tracks are fitted using an *adaptive vertex fitter* to compute the best estimate of the vertex parameters: the x, y and z position and covariance matrix, as well as the indicators for the success of the fit, such as the number of degrees of freedom for the vertex, and weights of the tracks used in the vertex.

In the adaptive vertex fit, each track in the vertex is assigned a weight between 0 and 1, which reflects the likelihood that it genuinely belongs to the vertex. Tracks that are consistent with the position of the reconstructed vertex have a weight close to 1, whereas tracks that lie more than a few standard deviations from the vertex have small weights. The number of degrees of freedom in the fit is defined as

$$n_{\text{dof}} = -3 + 2 \sum_{i=1}^{N_{\text{tracks}}} w_i, \tag{3.1}$$

where w_i is the weight of the ith track, and the sum runs over all tracks associated with the vertex. The number of degrees of freedom can be used to select true proton-proton interactions.

3.3 Muon reconstruction

The muon reconstruction starts from the so-called "local reconstruction". At this level the information from only a single muon chamber (DT, RPC or RPC) is used to determine the passage of the muon through the chambers [19]. The electric signal produced in a wire or strip is read out by front-end electronics. Each signal is associated with a position in space-time and it is usually referred to as "hit". In a DT, the position of the crossing particle in the wire plane is determined by measuring the particle arrival time using a Time-to-Digital Converter (TDC). The position x of the particle with respect to the wire can be computed as:

$$x = (t_{TDC} - t_{ped}) \times v, \tag{3.2}$$

where t_{TDC} is the arrival time registered by the TDC, t_{ped} is the time pedestal correction and v is the drift velocity. The main contributions to t_{ped} are:

- the time of flight needed for the muon to reach the chamber;
- the propagation time of the signal along the anode wire;
- Level-1 trigger latency.

The electric field in a DT cell is designed in such a way that the drift velocity can be considered constant along the drift path.

A hit in a CSC chamber is reconstructed by a combination of the position of the signal from cathode strips and anode wires. Anode wires in a CSC are bundled forming a wire group of 1-2 cm width. This results in a less precise measurement in the radial direction. The measured time in a CSC is calibrated such that hits from

muons produced promptly in the triggering bunch crossing have a time distribution centered at zero.

From the reconstructed hits, straight-line track segments are built within each CSC or DT chamber.

Standalone muon tracks are built by reconstructing muon trajectories, from DT and CSC segments and from RPC hits, using a Kalman-filter technique. Segment groups coming from DT and CSC are used for the seeding procedure of the algorithm. To improve the momentum resolution, a beam-spot constraint can be applied in the fit.

Tracker muon tracks are built by propagating the track reconstructed in the inner tracker to the muon system. Each track with transverse momentum $p_T > 0.5\,\text{GeV}$ and a total momentum of $p > 2.5\,\text{GeV}$ is extrapolated to the muon system. The track-to-segment matching is evaluated by measuring the distance of the extrapolated track and the segment in a local (x, y) coordinate system. The extrapolated track and the segment are matched either if the absolute value of the difference between their position in the x coordinate is smaller than 3 cm, or if the ratio of this distance to its uncertainty is smaller than 4. A track built in the tracker can be qualified as a tracker muon track if at least one matching segment is found.

Global muon tracks are built by performing a matching between the standalone and the tracker tracks. The matching is performed by a comparison of the track parameters of the tracks propagated onto a common surface. Also in this case, the track fitting procedure is based on a Kalman filter technique that exploits the information both from tracker and standalone tracks.

3.3.1 Muon identification

Reconstructed muons are given in input to the PF algorithm. For muons, a set of selection criteria are applied to reconstructed candidates with a standalone, global or tracker associated track. The selection cuts rely both on quality parameters of the reconstructed track and on information from the other subdetectors. These criteria were defined in order to allow the analyses to tune the muon selection to reach the desired balance between efficiency and purity. Variables coming from the fitting procedure, such as χ^2, the number of hits per track, or the degree of tracker-standalone track matching, are used as selection criteria. The muon segment compatibility is computed by propagating the tracker track to the muon system, and evaluating both the number of matched segments in all stations and the closeness of the matching in position and direction. Other used variables exploit features not coming from the track parameters, such as the compatibility with the primary vertex. Different sets of cuts correspond to different identification types, often referred to as "muon ID". The one used in this analysis is the *Tight muon ID*. It aims to suppress muons from hadronic punch-through, muons from decays in flight and cosmic muons. Tight muon ID selects PF muons which are both tracker and global muons. The inner track must have hits on at least six layers, including at least one pixel hit. The tracker muon must have matched segments in at least two muon stations. The global muon fit must have $\chi^2/dof < 10$ and include at least one hit from the muon system. Furthermore, a tight muon must be compatible

with the primary vertex, so additional requirements on the impact parameter are set: the distance from the primary vertex must be less than 2 mm on the xy plane and smaller than 5 mm along z.

3.3.2 Muon isolation

To distinguish between prompt muons and those coming from the weak decay of light or heavy quarks, the isolation of a muon is evaluated relative to its p_T summing up the energy in geometrical cones, $\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2}$, surrounding the muon. Different definitions of isolation are possible.

The PF relative isolation (I_{PF}) is based on objects reconstructed by the PF algorithm. It is defined as the sum of the transverse energy of all charged hadrons, neutral hadrons and photons contained in a cone around the muon direction and divided by the muon p_T. The cone width $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ is set to 0.4. A correction is applied in order to take into account for the pile up energy from photon and neutral hadron. A "loose" working point is used corresponding to $I_{PF} < 0.25$.

3.4 Electron, hadrons and photons

Electron reconstruction is based on combined information from the inner tracker and the calorimeters. Due to the large amount of material in the tracker, electrons often emit bremsstrahlung photons and photons often convert to e^+e^- pairs. For this reason, the basic properties and the technical issues to be solved for tracking and the energy deposition patterns of electrons and photons are similar. Isolated photon reconstruction is therefore conducted together with electron reconstruction.

Once muons, electrons, and isolated photons are identified and removed from the PF blocks, the remaining particles to be identified are hadrons from jet fragmentation and hadronization. These particles may be detected as charged hadrons ($\pi^\pm$, $K^\pm$, or protons), neutral hadrons (e. g. K^0_L), non-isolated photons (e. g. from π^0 decays), and more rarely additional muons (e. g. from early decays of charged hadrons).

3.5 Jets

Jets can be reconstructed in two different ways: using the standard jet reconstruction algorithm, which employs the sum of the energies deposited in the calorimeters (Calo jets), or by clustering particles reconstructed by the PF algorithm (PF jets).

In sequential clustering algorithms, such as the anti-k_T algorithm [20], the distance between entities (particles, pseudojets) and the entity-beam distance are defined and computed. If the distance between two entities is found to be the smallest then the two entities are combined into a single object whose momentum is the vector sum of the two entities momenta. If instead the smallest distance is the one between an entity and the beam, then one removes the entity from the list of particles/pseudojets and declares it to be a jet. The procedure is then repeated until no entries are left.

3.6 Missing transverse energy

Neutral weakly interacting particles, such as neutrinos, escape from the detector without producing any signal. The presence of such particles must be inferred from the imbalance of the total momentum. The vector momentum imbalance in the plane perpendicular to the beam axis is called missing transverse momentum and it is defined as the negative of the vector sum of the transverse momenta of all final-state particles reconstructed in the detector:

$$\vec{E}_T^{miss} = -\sum \vec{p}_T. \tag{3.3}$$

The magnitude of the missing transverse momentum is called missing transverse energy, and is denoted E_T^{miss}.

4

Machine Learning in High Energy Physics

4.1 Brief introduction to Machine Learning

The expression *machine learning* was coined in 1959 by Arthur Samuel [21] and, since the ending of the last century, Machine Learning (ML) has become one of the pillars of information technology and ubiquitous used in daily technologies. A Google search or a video on YouTube or Netflix are information that a complex structure of learning algorithms records in order to build a customised environment. Spam filters are another example of application of ML algorithms. An increasing improvement of *pattern recognition* techniques allows not only social networks (Facebook, Instagram etc...) to identify different people in photos, but also to create 3D images of human tissues and organs for medical purposes.
The advent of "Big Data", that is data not only big in size but also in variety, velocity, veracity, etc., has led to the blossom of learning algorithms in many fields. That is to be combined with a technological progress in storage and computational power (CPUs and GPUs), together with lower maintenance and material costs.

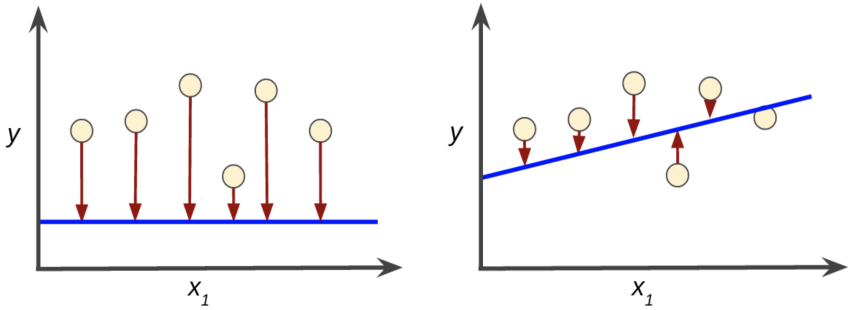
Among all types of ML, three main categories can be distinguished: *supervised* learning, *unsupervised* learning and *reinforcement* learning. Since only the first one of this categories is useful for the purposes of this work, we will go on describing just that one.

4.1.1 Supervised Learning

In this specific type of learning algorithms the machine extrapolates a pattern from labelled data. In other words, along with the input *features* also a *target* variable is provided containing the label of the input data.
Using this particular approach, a model is *trained* on a predefined set of *training examples*, which allows future predictions using new and unseen data.
Suppose we have a set of known inputs (also called *features*) x_i each with a certain weight w_i. In order to infer a quantity y, we define a *predictor* $y' = h(x_i)$ defined as:

$$y' = b + w_i \cdot x_i$$

with b as a given bias. Training a model simply means adjusting the weights and the bias from labelled examples. The idea is that through some iterations, the model attempts to assign a category to the input it receives and then, after comparing its prediction to the actual label, calibrates its own weights correspondingly to the mistake it has done.
The *loss* is a number indicating how far the model prediction was from the label. In Figure 4.1, two different models are shown: Figure 4.1a shows a model with high loss and Figure 4.1b one with low loss. The arrows are a visualisation of the computed loss: the right model show a lower loss and thus a better prediction capability.



(a) Model example with high loss. **(b)** Model example with low loss.

Figure 4.1: Different models for a supervised Machine Learning algorithm. In the left model, an high loss is obtained while in the right model, a lower loss is achieved. Red arrows represent loss while the blue line represents predictions.

The **squared loss**, an example of loss function, is defined as follows:

$$(\text{observation} - \text{prediction}(x))^2 = (y - y')^2$$

An example is the *mean square error* (MSE): the averaged squared loss per example over the whole dataset. To calculate the MSE, all the squared losses for individual examples are summed up and then divided by the number of examples used in the training:

$$\text{MSE} = \frac{1}{N} \sum_{(x,y)\in D} (y - \text{prediction}(x))^2$$

where:

- (x, y) represents an example in which:
  - x is the set of features that the model uses to make predictions.
  - y is the example's label (the quantity to predict).
- prediction(x) is a function of the weights and bias in combination with the set of features x.
- D is a dataset containing many labelled examples, which are (x, y) pairs.

Mean squared error is not the only metric used in Machine Learning, but it is commonly-used especially in regression problems. For instance, the *mean absolute error* (MAE) is another loss function defined as follows:

$$MAE = \frac{1}{N} \sum_{(x,y) \in D} |y - \text{prediction}(x)|$$

4.1.2 Training a ML model

The model takes one or more features as input and return one prediction (y') as output. Initially, all the weights are randomly set. After each iteration, the value of the weights are modified taking into account the computed loss. This is the learning procedure and it continues until the algorithm discovers values of weights with the lowest loss: that is convergence of the model.
Calculating the loss function for every value of w_i is an inefficient way of finding the convergence point. One common approach to this problem is called *gradient descent*. After choosing a starting value for the weights, typically random and non-zero, the gradient descent algorithm computes the gradient of the loss curve at the starting point. Mathematically speaking, a vector of partial derivatives with respect to the weights, is computed and in case of a negative gradient, the algorithm performs the next step (shown in Figure 4.2), thus approaching the minimum.
The gradient descent algorithm multiply the gradient by a scalar value known as *learning rate* (or *step size*: one of the model's *hyperparameters*) to determine the next point. Tuning the learning rate is not trivial: if a learning rate too small is selected, the learning procedure will take too long; conversely, if it is too large, the next point will exceed the minimum weight value and the correct value will never be reached.
When large volumes of data come into play, other parameters in the gradient descent strategy become very relevant for a fast convergence of the minimisation algorithm towards finding the global minimum, e.g. the gradient descent *batch size*: it corresponds to the number of examples you use to calculate the gradient in a single iteration. By default, the batch matches the entire dataset. However, it is not rare for datasets to contain millions and millions of examples as well as a huge number of features. That may affect the computation so, in order to avoid such problems, smaller batch sizes might be better: in the *stochastic gradient descent* (SGD), for instance, the batch size is brought to one single example per iteration. Given enough iterations, SGD works but the training phase is very noisy in terms of loss function.
A trade-off in this sense is the so called mini-batch GD, in which the original training set is divided into smaller batches: these are chosen at random and allow the parallelization of the entire process, which might be extremely useful in case of heterogeneous hardware devices available for training, providing GD algorithm convergence more efficiently.

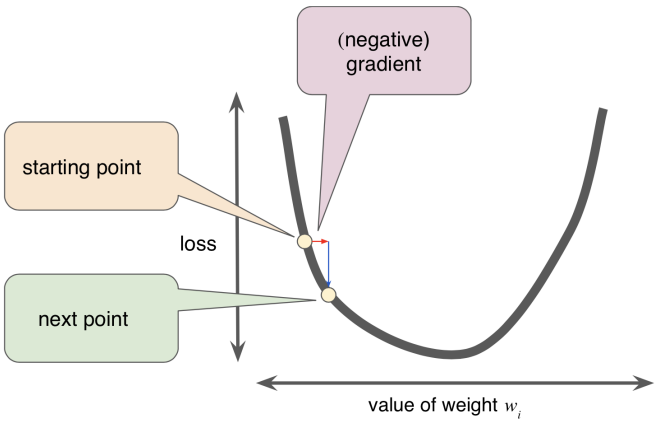


Figure 4.2: Schematic representation of the gradient descent mechanism [22].

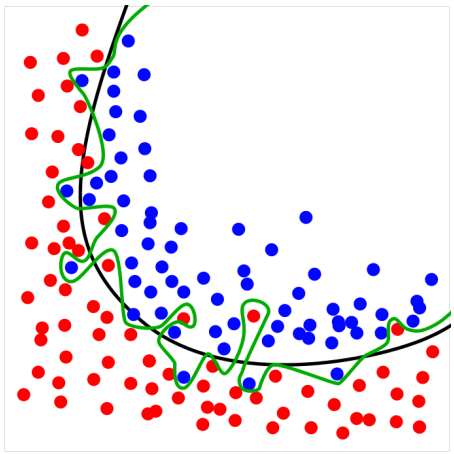


Figure 4.3: Example model that overfits data. In black a model for the data classification is shown. In green an overfitted model is shown.

4.1.3 The problem of Overfitting

A common problem with Machine Learning is *overfitting*. That might occur when a model is trained for too many iterations on a dataset: this will cause the loss computed on the data used for the training to fall very closely to zero, while the loss computed on never seen data to skyrocket, i.e. a lack of generalization capability. As an example in Fig. 4.3, decision boundary for the red vs. blue data points are set by two different models: the black line and the green line. The first model is quite simple (despite some misclassifications) and works quite well for new untrained data. However, during the training, the lowest loss possible might be computed using a complex cut, as shown in the green line. In that case the ability of such model to predict data with an entire new dataset decreases and predictions are no longer reliable.
In conclusion, a model should be trained taking in to account the balance between overfitting (inducing lack of generalization) and underfitting (possible symptom of lack of optimization).

4.1.4 Data Splitting techniques

A Machine Learning model aims to make good predictions on new, previously unseen data. To make sure that tests can be done on data not used in the training, there are different techniques:

1. Split into a Training and Test Set;
2. *k-fold* Cross-Validation;
3. Leave One Out Cross-Validation;
4. Repeated Random Test-Train Splits.

Since only splitting data in to Training and Test set will be used in this work, we will not discuss the other techniques any further.

4.1.5 Training, Validation and Test split

The technique consist in splitting the available data in a training set, validation and a test set. The first one will be used to perform all the training and tuning on the model and will be constantly checked with the validation test (not used in the training), while the test set will be left untouched. When the model has been finally trained to his "best" it can be finally evaluated on a dataset (the test set) it has never seen before. This algorithm evaluation technique is very fast. It has the following pros and cons:

- Pros: It is ideal for large datasets (hundreds of thousands of records). Splitting a large dataset into large sub-datasets allows that, first, each split of the data is not too tiny, and second, both are representative of the underlying problem. Because of the high speed, it is useful to use this approach when the algorithm you are investigating is slow to train.
- Cons: A downside of this technique is that it can have a high variance. This means that differences in the training and test dataset can result in meaningful differences in the estimate of accuracy. Furthermore, in order for this split to work, the test set has to be representative of the dataset as a whole, i.e. the test set should not have different characteristics than the training test.

In Figure 4.4, the model learned from the training data is quite simple. This model is not perfect, since a few predictions are wrong. However, this model does about as well on the test data as it does on the training data. In a way, using this particular model, overfitting does not occur.

4.1.6 ML model validation

After training and checking the model on train and validation set respectively, the best model is evaluated on the test set. In Figure 4.5, an example of workflow is shown: the training set is firstly checked by the validation set, which provides the model's optimal parameters for another cycle of training; then, the best model is tested again using the testing subset to obtain a reliable model.

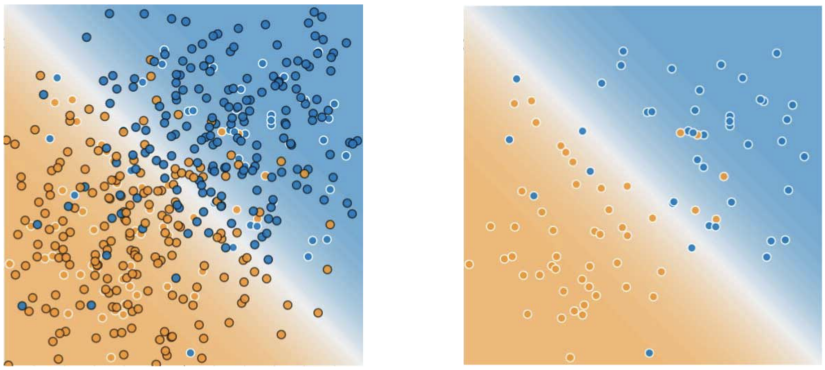


Figure 4.4: Example showing the splitting between training (left) and test (right) datasets (image from [22]).

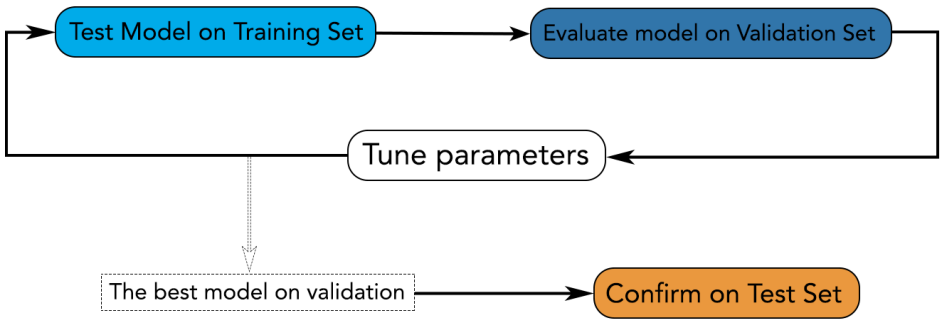


Figure 4.5: Workflow using a validation and test subset.

4.2 ML models

A ML model is a simplified version of the observations. The simplifications are meant to discard the superfluous details that are unlikely to generalize to new instances. To decide what data to discard and what data to keep, we must make *assumptions*. For example, a linear model makes the assumption that the data is fundamentally linear and that the distance between the instances and the straight line is just noise, which can safely be ignored. In a famous 1996 paper [23], David Wolpert demonstrated that if you make absolutely no assumption about the data, then there is no reason to prefer one model over any other. This is called the *No Free Lunch* (NFL) theorem. For some datasets the best model is a linear model, while for other datasets it is a neural network. There is no model that is *a priori* guaranteed to work better (hence the name of the theorem). The only way to know for sure which model is best is to evaluate them all. Since this is not possible, in practice you make some reasonable assumptions about the data and evaluate only a few reasonable models. For example, for simple tasks you may evaluate linear models with various levels of regularization, and for a complex problem you may evaluate various neural networks [24].
In this section some ML algorithms relevant for this thesis work will be described. We will first introduce linear discriminant (*Fisher*), then we will discuss about

Decision Trees and boosting techniques and finally we will move to *Artificial Neural Networks*.

4.2.1 Fisher Discriminants

In the method of Fisher discriminants [25] event selection is performed in a transformed variable space with zero linear correlations, by distinguishing the mean values of the signal and background distributions. The linear discriminant analysis determines an axis in the (correlated) hyperspace of the input variables such that, when projecting the output classes (signal and background) upon this axis, they are pushed as far as possible away from each other, while events of a same class are confined in a close vicinity. The linearity property of this classifier is reflected in the metric with which "far apart" and "close vicinity" are determined: the covariance matrix of the discriminating variable space. The classification of the events in signal and background classes relies on the following characteristics: the overall sample means x_k for each input variable k = $1, \ldots, n_{var}$, the class-specific sample means $\bar{x}_{S(B),k}$, and total covariance matrix C of the sample. The covariance matrix can be decomposed into the sum of a within-(W) and a between-class matrix (B). They respectively describe the dispersion of events relative to the means of their own class (within-class matrix), and relative to the overall sample means (between-class matrix).

4.2.2 Decision Trees

Decision trees are well known classifiers that allow a straightforward interpretation as they can be visualized by a simple two-dimensional tree structure. As in Fig. 4.6, starting from the root node, a sequence of binary splits using the discriminating variables x_i is applied to the data. Each split uses the variable that at this node gives the best separation between signal and background when being cut on. The same variable may thus be used at several nodes, while others might not be used at all. The leaf nodes at the bottom end of the tree are labeled "S" for signal and "B" for background depending on the majority of events that end up in the respective nodes. For regression trees, the node splitting is performed on the variable that gives the maximum decrease in the average squared error when attributing a constant value of the target variable as output of the node, given by the average of the training events in the corresponding (leaf) node.

They are in this respect similar to rectangular cuts. However, whereas a cut-based analysis is able to select only one hypercube as region of phase space, the decision tree is able to split the phase space into a large number of hypercubes, each of which is identified as either "signal-like" or background-like, or attributed a constant event (target) value in case of a regression tree.

For classification trees, the path down the tree to each *leaf node* represents an individual cut sequence that selects signal or background depending on the type of the leaf node. A shortcoming of decision trees is their instability with respect to statistical fluctuations in the training sample from which the tree structure is derived. For example, if two input variables exhibit similar separation power, a fluctuation in the training sample may cause the tree growing algorithm to decide

to split on one variable, while the other variable could have been selected without that fluctuation. In such a case the whole tree structure is altered below this node, possibly resulting also in a substantially different classifier response. This problem is overcome by constructing a forest of decision trees and classifying an event on a majority *vote* of the classifications done by each tree in the forest. All trees in the forest are derived from the same training sample, with the events being subsequently subjected to so-called *boosting*, a procedure which modifies their weights in the sample. Boosting increases the statistical stability of the classifier and is able to drastically improve the separation performance compared to a single decision tree. However, the advantage of the straightforward interpretation of the decision tree is lost. While one can of course still look at a limited number of trees trying to interpret the training result, one will hardly be able to do so for hundreds of trees in a forest. Nevertheless, the general structure of the selection can already be understood by looking at a limited number of individual trees.

In many cases, the boosting performs best if applied to trees (classifiers) that, taken individually, have not much classification power. These so called "weak classifiers" are small trees, *limited in growth* to a typical tree depth of as small as two, depending on the how much interaction there is between the different input variables. By limiting the tree depth during the tree building process (training), the tendency of overtraining for simple decision trees which are typically grown to a large depth and then pruned, is almost completely eliminated [26].

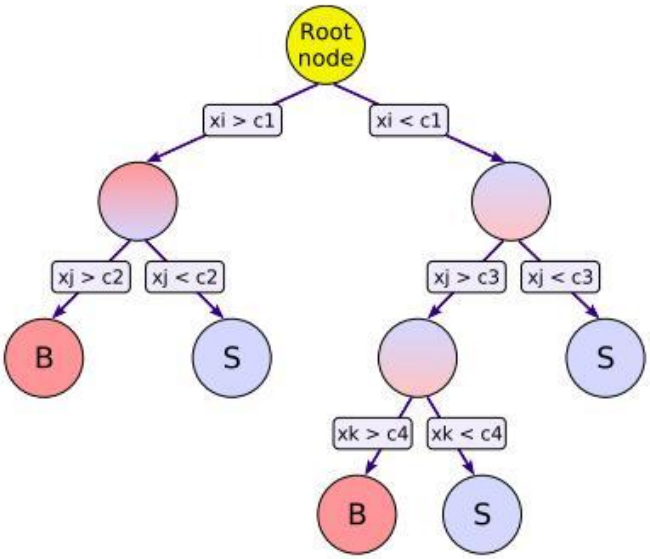


Figure 4.6: Schematic view of a decision tree.

4.2.3 Artificial Neural Network

An Artificial Neural Network (ANN) is any simulated collection of interconnected neurons, with each neuron producing a certain response at a given set of input signals. By applying an external signal to some (input) neurons, which is the passed through to intermediate (hidden) neurons, the network is put into a defined state

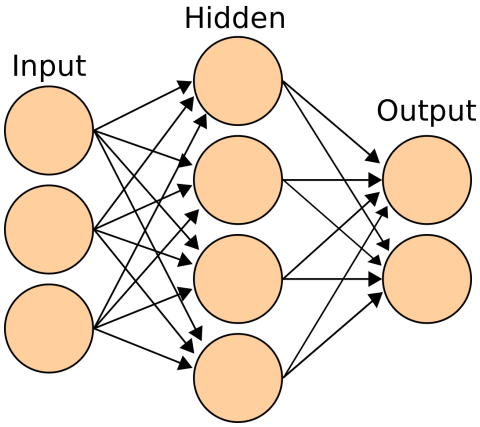


Figure 4.7: Neural Network basic working principle.

that can be measured from the response of one or several (output) neurons. One can therefore view the neural network as a mapping from a space of input variables $x_1, \ldots, x_{n_{var}}$ onto a one-dimensional (e.g. in case of a signal-versus-background discrimination problem) or multi-dimensional space of output variables $y_1, \ldots, y_{m_{var}}$. The mapping is nonlinear if at least one neuron has a nonlinear response to its input [26]. The schematic working principle is described in Figure 4.7.

4.3 Machine Learning Frameworks

Several Machine Learning frameworks have been created allowing developers to build ML models in a very flexible and efficient way.
In this section, a few of these frameworks will be listed with an increasing attention to those involved in this thesis.

4.3.1 TensorFlow

TensorFlow is an open-source software library for data-flow programming across a range of tasks. It is a symbolic math library for numerical computation that uses data flow graphs and is also used for machine learning applications, such as neural networks. It was developed by the Google Brain team [27] for internal use and released under the Apache 2.0 open-source licence on November 2015 [28]. Currently it is one of the most utilised frameworks worldwide for ML thanks to a powerful backend that allows efficient (and relatively transparent) model training over heterogeneous hardware architectures.

Figure 4.8: TensorFlow logo.

The expert-level customisability of TensowrFlow comes with the price of a relative complexity of use and a hard accessibility of a majority of beginners/intermediate-

level users, hence opening the floor to easy and intuitive frontend interfaces, as discussed in the next section.

4.3.2 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano [29]. It has been developed with the purpose of enabling fast experimentation. It has a simple interface, that minimizes the number of user actions required for common use cases and provides a clear and actionable feedback upon user error. Keras is one of the most favourite

Figure 4.9: Keras logo.

tools among ML developers, as shown in the number of mentions in scientific papers uploaded to arXiv.org (Figure 4.10). The core data structure of Keras is a model, a way to organize layers. The simplest type of model is the *Sequential* model, a linear stack of layers.

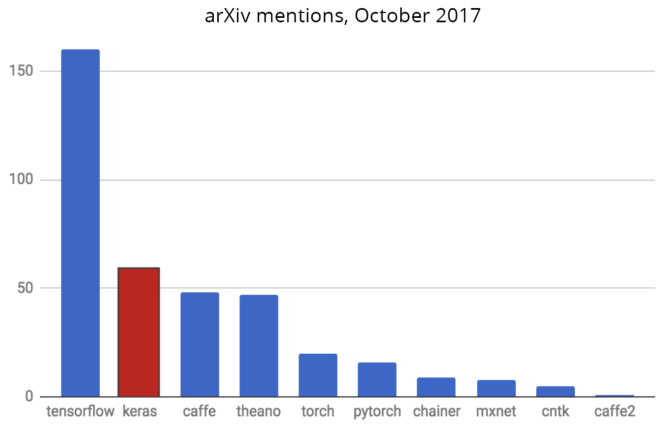


Figure 4.10: Keras popularity in arXiv mentions.

Here is an example of sequential model:

Listing 4.1: Create a sequential model.

```
from keras.models import Sequential

model = Sequential()
```

Stacking NN layers is easily achieved by:

Listing 4.2: Stack layers in Neural Network.

```
from keras.layers import Dense

model.add(Dense(units=64, activation="relu", input_dim=100))

model.add(Dense(units=10, activation="softmax"))
```

In Line 3 a hidden layer is created: *units* corresponds to the number of nodes (or neuron) for that specific layers. The first hidden layer requires also the dimension of input layer which is given in the *input_dim* parameter.
Once the model looks good, the configuration of its learning process is done by:

Listing 4.3: Configure learning.

```
model.compile(loss='mean_squared_error', optimizer='sgd')
```

This method takes as input the *loss* function (mean squared for this example but many more can be used) and the configuration of the *optimizer* (in this case the Stochastic Gradient Descent, introduced in Section 3.2).
Training the model in batches is then performed:

Listing 4.4: Training iteration in batches.

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

At last, the predictions on new data with a simple line:

Listing 4.5: Predictions on new data.

```
classes = model.predict(x_test, batch_size=128)
```

This basic example shows the simplicity in the implementation of a Neural Network.

On the side of flexibility, Keras offers a *functional API*. That is a way to create models that can handle models with non-linear topology, shared layers, and even multiple inputs or outputs [29].

4.3.3 TMVA: a Toolkit for MultiVariate data Analysis

TMVA is a ROOT-integrated environment to structure a Multivariate Analysis provides tools for the processing, parallel evaluation and application of multivariate classification (and in latest release, also for multivariate regression techniques). TMVA is developed for the needs of HEP applications, but may be not limited to these. TMVA implements multivariate techniques belonging to supervised learning algorithms.
The software package consists of abstract, object-oriented implementations in

C++/ROOT for every multivariate analysis (MVA) techniques, as well as auxiliary tools such as parameter fitting. It provides training, testing and performance evaluation algorithms and visualisation scripts. The training and testing is performed with the use of user-supplied data sets in form of ROOT trees or text files, where each event can have an individual weight. To compare the signal-efficiency and background-rejection performance of the classifiers, the analysis job prints tabulated results for some benchmark values [26]. Moreover, in the latest years, it has been added the possibility to implement code in Python and also to import frameworks like Keras/TensorFlow. That allowed for a very easy way to test out different models from both Keras and TMVA built in methods with very few lines of code, as shown hereafter.

Listing 4.6: Build a classifier in Python/ROOT using TMVA and Keras.

```
factory = TMVA.Factory('TMVAClassification', output,
'!V:!Silent:Color:DrawProgressBar:
Transformations=N:AnalysisType=Classification')

dataloader = TMVA.DataLoader('dataset')
for branch in signal.GetListOfBranches():
    dataloader.AddVariable(branch.GetName())

dataloader.AddSignalTree(signal, 1.0)
dataloader.AddBackgroundTree(background, 1.0)
dataloader.PrepareTrainingAndTestTree(TCut(''),
    'nTrain_Signal=4000:nTrain_Background=4000:
    SplitMode=Random:NormMode=NumEvents:!V')

# Define model
model = Sequential()
model.add(Dense(64, activation='relu', input_dim=4))
model.add(Dense(2, activation='softmax'))

factory.BookMethod(dataloader, TMVA.Types.kFisher, 'Fisher',
                   '!H:!V:Fisher:VarTransform=D,G')

factory.BookMethod(dataloader, TMVA.Types.kPyKeras, 'PyKeras',
'H:!V:VarTransform=N:FilenameModel=model.h5:
NumEpochs=20:BatchSize=32')

factory.TrainAllMethods()
factory.TestAllMethods()
factory.EvaluateAllMethods()
```

5

Search for BSM neutral Higgs bosons

This thesis gets its inspiration from an already published CMS analysis [3] searching for neutral non-standard-model Higgs bosons decaying to two muons and aims to improve it. The search has been published using the proton-proton collision data recorded by the CMS experiment at the LHC at a $\sqrt{s} = 13$ TeV, corresponding to an integrated luminosity of 35.9 fb^{-1} and it is sensitive to the neutral Higgs bosons produced via the gluon fusion process or in association with a b-quark pair. A b tag was applied to obtain two exclusive categories: one containing the events with one b-tagged jet and no more than one additional non b-tagged jet and the other containing the events without b-tagged jets. A veto on the E_T^{miss} variable, shown in Fig. 5.1, was applied to each category to achieve a better signal background discrimination. No significant deviations from the standard model expectation are observed. Upper limits at 95% confidence level are set in the context of the m_h^{mod+} and phenomenological hMSSM scenarios on the parameter $tan\beta$ as a function of the mass of the pseudoscalar A boson, in the range from 130 to 600 GeV (see Fig.5.2). The results are also used to set a model-independent limit on the product of the branching fraction for the decay into a muon pair and the cross section for the production of a scalar neutral boson, either via gluon fusion, or in association with b quarks, in the mass range from 130 to 1000 GeV (Fig. 5.3) [3].

Considering the wide range of mass and the dependence on it, even small, of some possible discrimination variables, the purpose of this thesis is to study whether a ML approach can give a better performance in terms of signal background separation than the one obtained in the analysis described earlier.

5.1 Use of a multivariate approach

The analysis aims to perform an effective separation between signal and background by exploiting observables that are differently distributed between signal and background. The goal is to improve the performance of the cut-based analysis, where the applied selection criteria were the same for every Higgs mass assumption. The study presented in this thesis is based on simulation only. The problem is hence framed as a "classification" problem, in particular this is a case of *supervised*

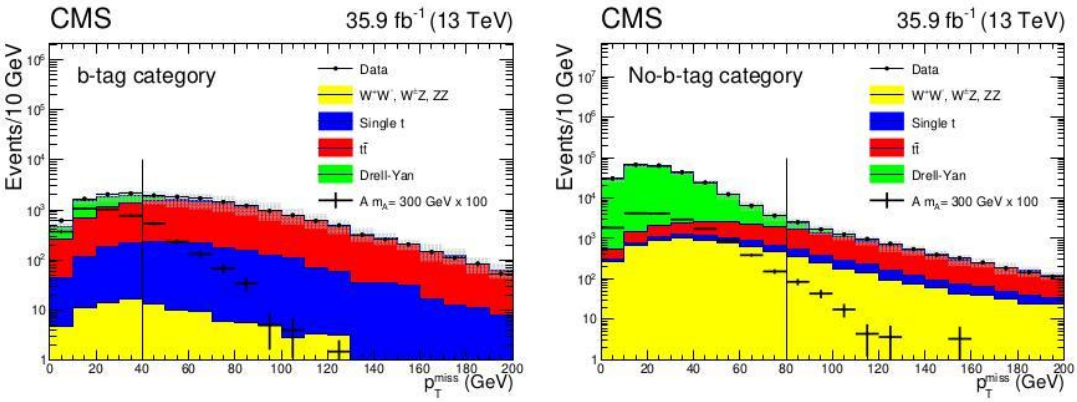


Figure 5.1: Distribution of the missing transverse momentum in (left) b-tag and (right) no-b-tag categories, for events with dimuon invariant mass larger than 130 GeV, as observed in data (dots) and predicted by simulation (colored histograms). The shaded gray band around the total background histogram represents the total uncertainty in the simulated prediction. The contribution of the expected signal for $M_A = 300$ GeV and $tan\beta = 20$, scaled by a factor of 100, is superimposed for illustration. The vertical line represents the upper threshold used to select the events in the two categories.

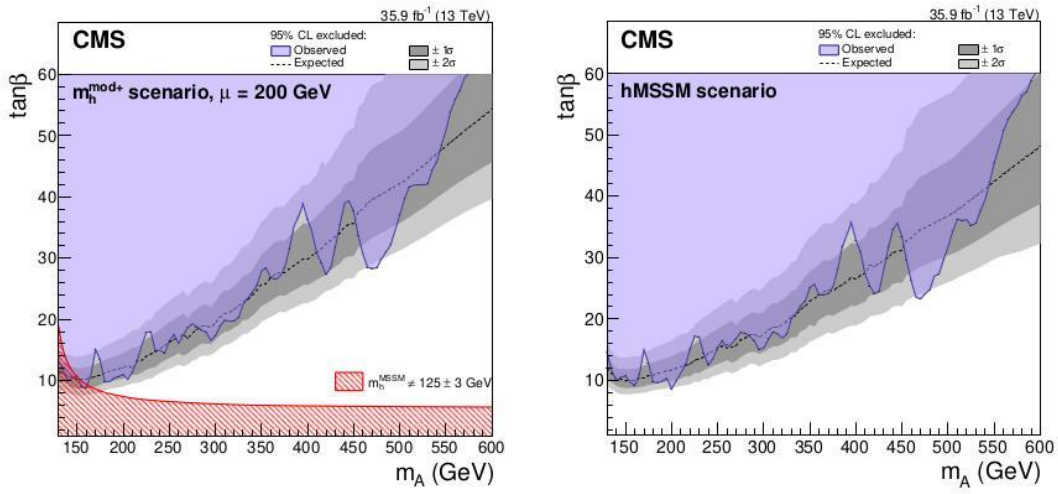


Figure 5.2: The 95% CL expected, including the 68 and 95% CL bands, and observed upper limits, on $tan\beta$ as a function of M_A for the m_h^{mod+} (left) and the hMSSM (right) scenarios of the MSSM. h The observed exclusion contour is indicated by the purple region, while the area under the red curve is excluded by requiring the neutral h boson mass consistent with 125 ± 3 GeV.

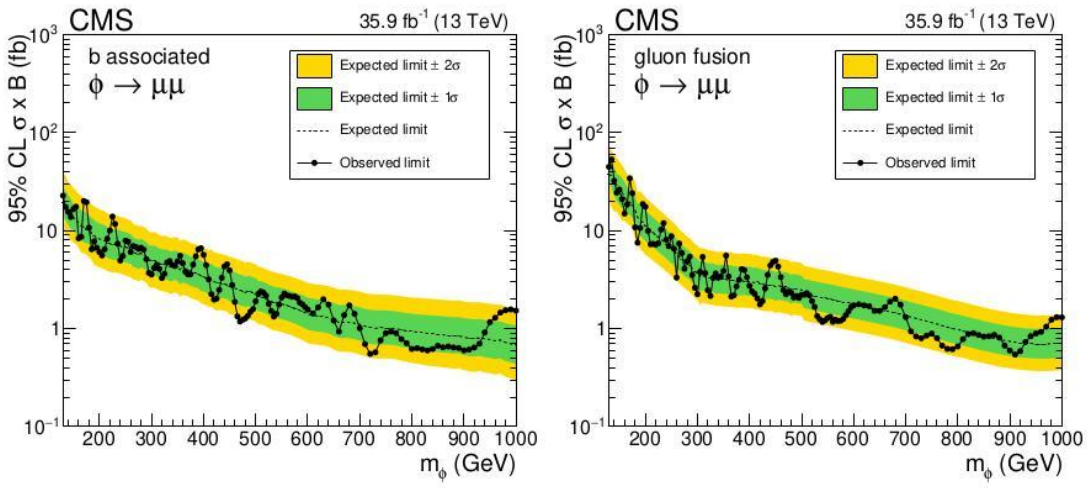


Figure 5.3: The 95% CL expected, including the 68 and 95% CL bands, and observed model- independent upper limits on the production cross section times branching fraction of a generic ϕ boson decaying to a dimuon pair, in the case of b-associated (left) and gluon fusion (right) production. The results are obtained using a signal template with an intrinsic narrow width.

classification in which each event is labeled. In order explore the ML potential two training strategies have been designed:

- *inclusive* in which all the preselected events used to train the classifier;
- *per categories* in which the preselected events are split in categories (see 5.2.1) and a model is trained for each category.

Each model have been trained and evaluated following both the strategies for seven different mass hypotheses: 140, 225, 300, 400, 600, 700 and 1000 GeV.

Choosing the evaluation metric

In order to evaluate the discrimination power of different classification models, *accuracy*, *ROC* curves and *AUC* have been used. The *accuracy* identifies how often the algorithm classifies data points correctly. It is the number of correctly predicted data points out of all the data points. It is defined as the number of true positives and true negatives divided by the total entry number [30]. The ROC curve (receiver operating characteristic curve) represents the performance of a classification model for several working points. This curve relates two parameters:

- the true positive rate (signal efficiency);
- the false positive rate (background rejection).

The AUC (Area under the ROC Curve), measures the area underneath the ROC curve [31].

Unbalanced dataset

The signal and background that have been used in this thesis are strongly unbalanced in terms of number of events contained in each sub-set. This is a common problem in the ML environment and there are different ways to approach that issue:

- get more data: just get more data to balance the dataset;
- oversampling: it consists in taking the minority class and,with different techniques, produce more samples;
- undersampling: takes the majority class and remove samples;
- events re-weight: apply weights to the two categories.

Since generation of massive additional MC samples through CMS-central production mechanism would have required time beyond the budget efforts (and this is hence considered as a next step), and the oversampling/undersampling techniques are very prone to introduce biases in the classification, the choice has fallen on the event re-weight technique. ROOT/TMVA offers the possibility to re-weight the events so that the sum of the weights of the two classes is equal [26].

5.1.1 ML models

The *BDT* model used in this analysis has been built, for all the cases, in the same way: 1000 trees with a maximum dept of 6. The boosting algorithm was AdaBoost [32]. The input variables were normalized following a *normal* distribution.

The *ANN* was built in Keras/TensorFlow. It consists of 3 hidden layers of 150-100-10 nodes respectively.
Following here are some characteristics about the model in detail:

- the chosen activation function is *RELU* [28] for the nodes in all layer except the last one, for which a *sigmoid* activation function has been preferred, following the state-of-art of classification techniques;
- the values of the weights were randomly initialized with a normal distribution;
- the loss function has been computed with the *binary cross-entropy* function;
- the *Adam* has been chosen: it allows to vary the already mentioned step to enhance the performances of the network;
- number of epochs is 15 and batch size is 500.

5.1.2 Monte Carlo Samples

The MC samples used in this thesis are simulated reproducing proton-proton collisions recorded by CMS during the Run2 data taking of the 2016, at $\sqrt{s} = 13$ TeV, with pile-up condition of about 30 collisions per bunch crossing, spaced by 25 ns.

The MC signal samples were simulated for several (M_A, $tan\beta$) points, where M_A goes from 130 to 1000 GeV, and $tan\beta$ goes from 5 to 60. In this thesis the analysis is performed only on the b-associated signal samples.

Many different samples of Monte Carlo simulated events are used to describe the expected background from standard model processes:

- Drell-Yan (DY) processes with two oppositely charged muons in the final state are the dominant background;
- production of $t\bar{t}$ pairs is the next dominant background;
- single top (ST) production;
- WW, WZ and ZZ production, also referred to as "di-boson" production;

The background due to Drell-Yan events is an "irreducible" background, i. e. Drell-Yan events with two oppositely charged muons in the final state are indistinguishable from signal events and in principle they cannot be rejected by applying any selection cut. Instead the other backgrounds can be reduced by applying selection cuts. Two oppositely charged muons can be produced, in association with jets, from the semileptonic decay of both t and $\bar{t}$ in $t\bar{t}$ events. In single top events in addition to a muon from semileptonic top decay there can be another muon from cascade decay of b quarks. This can lead to two oppositely charged muons *plus* jest. In di-boson production, a pair of oppositely charged muons can be produced when both decay to muon final states ($W \to \mu\nu$, $Z \to \mu\mu$). The main sources of background come from Drell-Yan, $t\bar{t}$ and single-top production. Other less relevant backgrounds include WW, WZ and ZZ production.

5.2 Event selection

In this section the event and object selection strategy is presented. The analysis relies primarily on muons for the event selection. Muon reconstruction and identification criteria, as well as requirements on the muon isolation are used to discriminate prompt muons expected from the Higgs decay. Events are selected by the trigger system requiring the presence of at least one isolated muon with $p_T > 24$ GeV (`HLT_IsoMu24` trigger path).

5.2.1 Preselection and tree preparation

Events passing the `HLT_IsoMu24` requirements have to contain at least one well-identified primary vertex, reconstructed with:

- more than four degrees of freedom;
- a $|d_z|$ coordinate within 24 cm from the beam line.

The events must contain at least two well reconstructed and isolated muons within the pseudo-rapidity range of $|\eta| < 2.4$ and $p_T > 26$ GeV. Only muons passing the Tight ID (defined in Sec.3) are selected. A tight cut on the relative PF-isolation

(Sec.3) is also applied to reject non-prompt muons typically coming from weak decays of light and heavy hadrons. The events that pass this preselection are then analyzed using different approaches.

Well reconstructed PF-jets with p_T>20 GeV and within an $|\eta|$ of 2.4 have been selected and their features have been used as potential discrimination variables. In the cut-based analysis they have used to split the preselected events in two independent categories, one containing the events with at least a b-tagged jet, and the other containing the events without any b-jet. In the multivariate analyses, PF-jet variables have been used to prepare the root tree objects used to train the proposed ML models. For the *inclusive* training strategy a tree containing all the preselected events has been prepared with all the following discrimination variables (Fig.5.4):

- the transverse missing energy, E_T^{miss};
- the dimuon ΔR, $\Delta\eta$, $\Delta\phi$;
- the ΔR, Δp_T and $\Delta\eta$ between the dimuon and the highest p_T b-jet;
- the number of b-jets and light-flavour jets;
- the p_T and η of the highest p_T b-jet and highest p_T light-flavour jet.

For the *per categories* training strategy two mutual exclusive trees have been prepared, one, the *cat1* tree, containing the events with at least one b-tagged jet, and the other, the *cat2* tree, containing the events without any b-tagged jet. The *cat1* tree contains the same variables of the tree built for the *inclusive* strategy while while in the *cat2* tree the b-jet related variables have been omitted.

The described trees suffer from a problem of uneven, so-called "jagged", arrays. This means that some of the arrays have empty values, due to the absence of a b-jet or a light-flavour jet which does not allow to build their related variables. To deal with this, all "missing values" have been filled with *zeros*, as appears in Fig. 5.4. This is to avoid filling the arrays with non-physical initialization values corresponding to non-physical ranges which might have led to problems for the classification task.

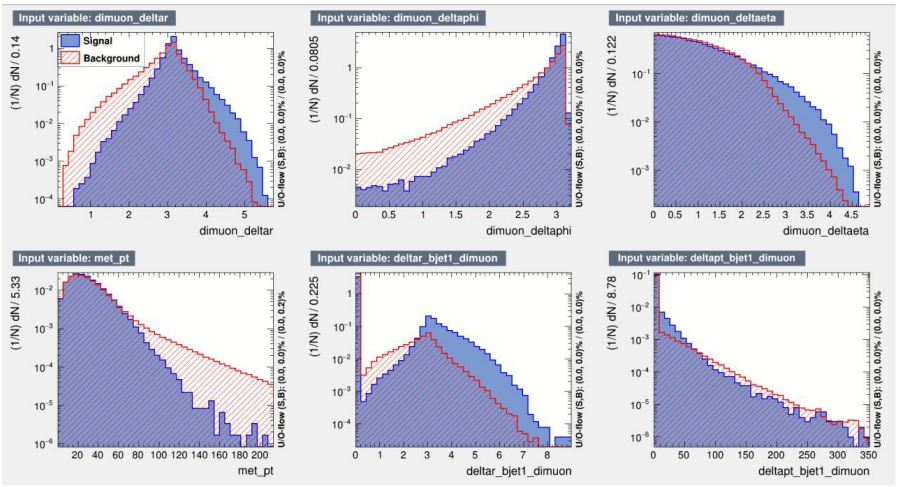
5.2.2 Signal Background discrimination

BDT and ANN, following the inclusive and per categories strategy, are trained using the trees described in Sec.5.2.1. For each training a signal background discriminator is obtained, shown as example in Fig. 5.5.

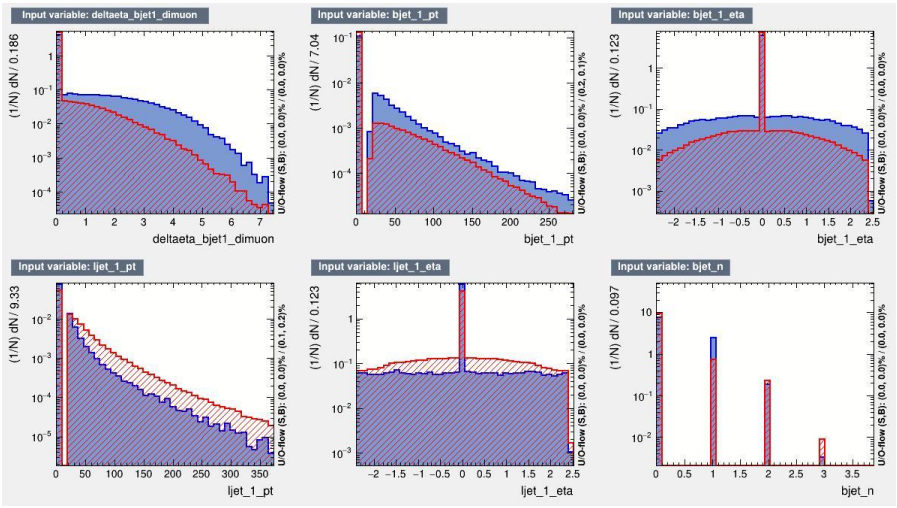
For each strategy, the performance of different ML models is represented using the ROC. In Fig. 5.6 ROC of BDT, ANN and Fisher (described in Sec. 4.2.1) have been compared for three mass hypotheses. Since the Fisher discriminants model performed poorly with respect to the other ML models, it has not been explored any further.

For the other ML classifiers an optimal cut is defined as the discriminator threshold which maximizes both the signal efficiency and the background rejection (Fig.5.7). The *optimal cut* is then used to perform the event selection.

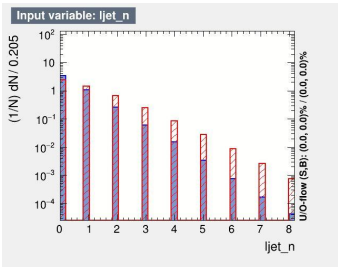
The cut-based analysis, as already mentioned, relies only on the E_T^{miss} veto to perform the final event selection (see Fig. 5.1).



(a)



(b)



(c)

Figure 5.4: Discrimination variables of the *inclusive tree* for the classification task. The signal variables for a mass hypothesis of 300 GeV are represented in blue while background variables are represented in red. The spikes that can be seen in some of the variables are due to the jagged array problem described in Sec.5.2.1

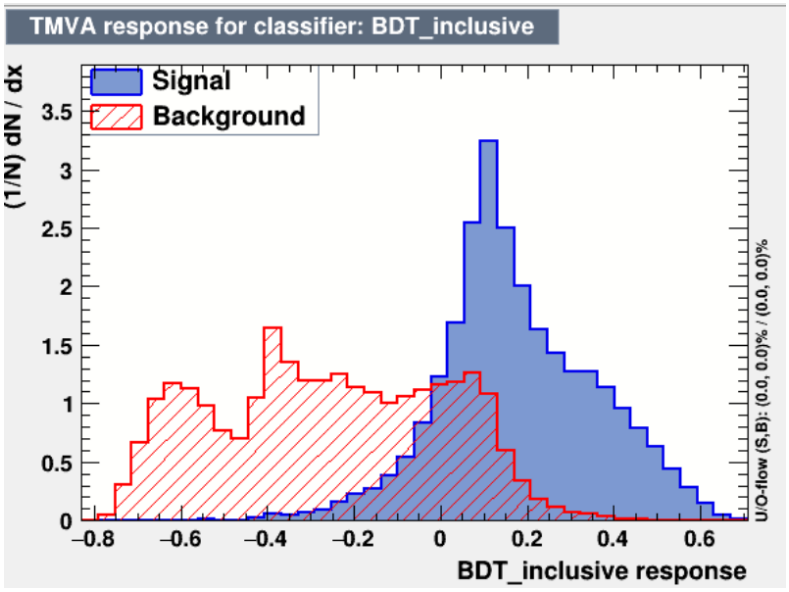
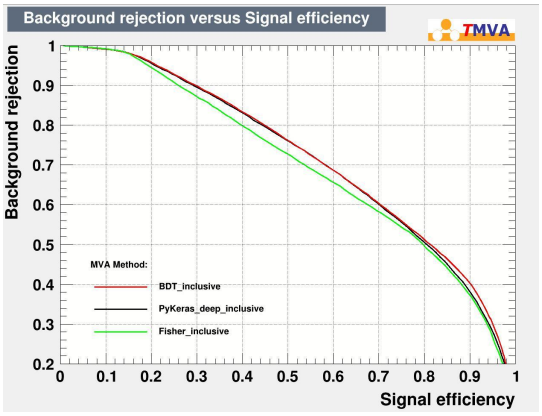
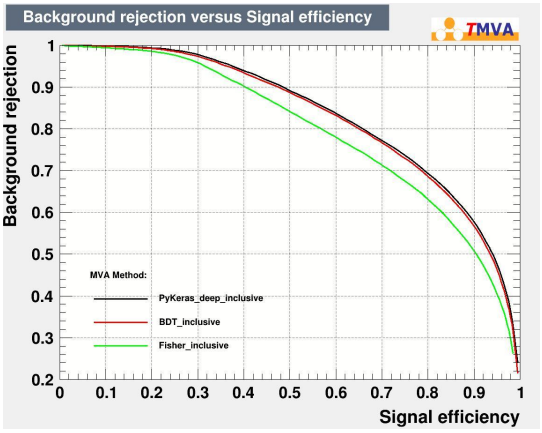


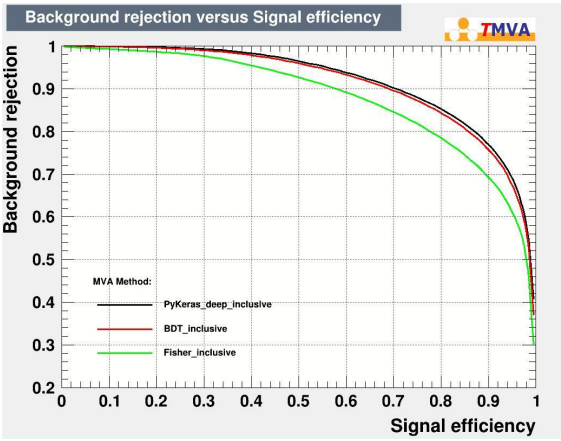
Figure 5.5: Distribution of the classifier after the training has been performed on a signal hypothesis of 1000 GeV with the *inclusive* strategy.



(a) Signal hypothesis at 140 GeV.



(b) Signal hypothesis at 400 GeV.



(c) Signal hypothesis at 1000 GeV.

Figure 5.6: Comparison of ROC curves of different models: BDT (red), ANN (black) and Fisher discriminants (green). This training has been made for three different mass hypothesis (140, 400 and 100 GeV) in order to test the models on signals with different characteristics. Fisher discriminants achieved the worst separation power in comparison with the other tested models.

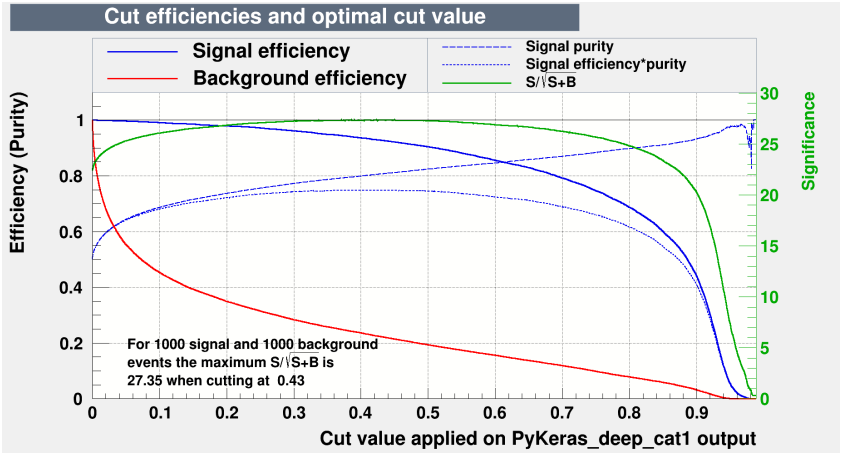


Figure 5.7: In this plot are reported the signal efficiency (blue line) and the background efficiency (red line). The green line represent the trend of the significance (see Sec. 5.3) and the optimal cut is the value for which it reaches its maximum (reported in the lower left corner).

5.3 Results

To evaluate the performance obtained by applying the event selections previously described the statistical significance has been calculated. The statistical significance is defined as:

$$\alpha = \frac{N_s}{\sqrt{N_s + N_b}}, \tag{5.1}$$

where N_s is the number of signal events and N_b is the number of background events. The signal and background yields have been calculated from MC samples within a 10% mass window centered at the Higgs signal mass hypothesis. In the *per categories* the total statistical significance has been computed as root sum squares of the significance obtained for each category. This is possible since the two categories are independent each other.

The significance values obtained for the models trained on the seven inspected mass hypotheses are reported in Tab. 5.1 together with the significance obtained for the cut-based analysis.

The multivariate models provide a better performance with respect to the cut-based analysis for all the mass hypotheses as can be seen in the graph reported in Fig. 5.8. The BDT model trained using the *per categories* strategy is the one giving the best performance for almost all the inspected mass hypotheses with an improvement that ranges from 4%, for lower mass hypotheses, to 25% for higher mass hypotheses. The ANN achieves an improvement with respect to the cut-based analysis that goes from 3% to 19%.

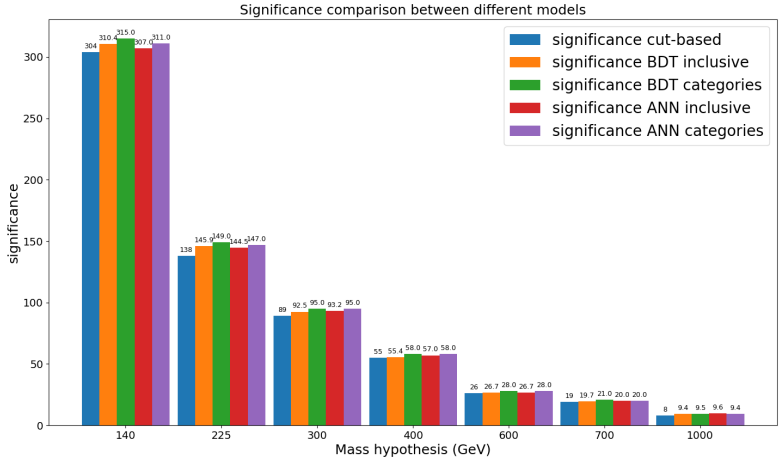


Figure 5.8: Significance vs. mass values obtained for the different analysis models.

mass (GeV)	Analysis models (α)				
	Cut-based	BDT Incl.	BDT Cat.	ANN Incl.	ANN cat.
140	299.0	310.4	310.6	307.0	306.5
225	135.7	145.9	146.9	144.5	145.6
300	87.6	92.5	91.0	93.2	93.7
400	53.8	55.4	59.8	57.0	56.9
600	24.7	26.7	30.2	26.7	27.1
700	18.2	19.7	22.8	20.0	19.9
1000	8.1	9.4	9.4	9.6	9.4

Table 5.1: Significance values obtained for the cut-based analysis and the tested multivariate approaches.

5.4 Next steps

The multivariate approach applied to the search for the neutral BSM Higgs bosons described in this thesis might be improved and extended mainly in two directions:

- increase of statistics of simulated events;
- adoption of parametrized ML algorithms.

Additional statistics of simulated events

Firstly, the availability of larger simulated samples would increase the ability to characterize the various background sources more in details, thus yielding better datasets as input to ML algorithms for the supervised training phase. Pursuing this route would anyway have required the placement of additional request for production of Monte Carlo samples to the CMS central production team. This kind of needs usually requires a careful scrutiny by the central CMS Physics team, and - even if accepted - would necessarily need to go through a lengthy negotiation process about its actual priority with respect to other production needs. Realistically, it has been estimated [33] that this action, despite valuable, would ultimately imply that in the best case scenario the additional samples might have become available too far in the future to be concretely available in the context of this thesis. It was hence decided and agreed to use the available samples for this exploratory study, and securing additional larger samples is suggested as a next step to be pursued to anyone who might be continuing this work.

As a mitigation action, the option of performing some Data Augmentation techniques was also explored. Data augmentation is a techniques often used by ML practitioners to increase the diversity of the training set by applying random (but realistic) transformations to the existing training set. This means that the available training dataset is assumed to be fixed with no option to increase its size, but each entry (or some of the entries) of such dataset are properly transformed into new additional entries, so to artificially produce an increase in the total number of training examples. This is regularly and extensively done on computer vision datasets, by e.g. taking one single labelled picture of an object and transforming it

(through rotation, reflection, etc) into additional labelled images of the same object, to be actually treated like new, distinct objects instead. In the context of this thesis, this action was considered to be a potentially very effective but dangerous route to follow, given the nature of the dataset under study, in terms of possible biases that could have been introduced in the distribution of key observable that characterise and distinguish signal and background events. A proper preliminary study to evaluate the risk connected to such approach on this specific HEP dataset would have been required, which in turn would have brought the work far away from its main goals: hence, it was estimated that the time to be spent in the validation of the transformed data would have been considerable, thus making the overall process not effective and time practical.

Parametric ML

Secondly, a parametric Neural Network (pNN) approach on a HEP use-case is suggested as a promising next step for this work. A full description can be found in [34], and only a brief outline is given in the following.

The current approaches to the classification task in this thesis require the use of a single signal sample in training, thus somehow sacrificing performances at other values. A pNN is based on a simple but clever strategy: a single neural network is relied upon to tackle a full set of related tasks. This is done by simply extending the list of input features to include not only the traditional set of event-level features, but also one or more parameters that describe the larger scope of the problem (e.g. a new particle's mass). Namely, a single parameterized network can replace a set of individual networks trained for specific cases, as well as smoothly interpolate to cases where it has not been trained upon - and the parameter in this case could be the mass of a particle. In other words, in the case of a search for a hypothetical new particle, this approach greatly simplifies the task – by requiring only one NN – as well as making the results more powerful – by allowing them to be interpolated between specific values. In addition, they may outperform isolated networks by generalizing from the full parameter-dependent dataset.

In this sense, one aspect is worth noting - and this is specially relevant to this thesis work: this approach can be applied to *any* ML classification algorithm, however, neural networks provide a smooth interpolation, while tree-based methods (like BDT) may not. This is one of the reasons why ANN are used in this thesis, alongside with other more traditional ML approaches: despite the ANN adoption in this thesis does not show the best performances at all mass values, the use of ANN is nevertheless suggested in this task, as it could be the framework from which to extend the current work towards pNN in the next future.

Conclusions

This thesis describes a study of a multivariate approach in the search for a neutral BSM Higgs boson produced in association with a b-quark pair and decaying into two muons with the CMS detector at LHC. The analysis is based on the search for a peak in the dimuon mass spectrum, scanning several mass hypothesis within the range 140-1000 GeV. Monte Carlo samples, reproducing the 2016 proton-proton LHC Run at $\sqrt{s} = 13$ TeV, have been analyzed.

The purpose of this study is to improve a similar analysis published by CMS [3] which adopts a cut-based approach either for the signal categorization and the background rejection. The categorization is done requiring the presence/absence of a b-tagged jet with an additional veto on the number of jets from light-flavour quarks, while the background rejection is obtained with a veto on the missing transverse energy (E_T^{miss}).

Additional kinematic variables have been used in the proposed multivariate analyses. Two multivariate models have been used, a Boosted Decision Trees (BDT) and a Artificial Neural Networks (ANN), both trained following two different approaches: one exploiting the full preselected dataset, and the other exploiting the dataset split in two mutual exclusive categories.

The achieved statistical significance was used to compare the different approaches. The significance is calculated for each signal hypothesis as the number of signal events over the square root of the number of signal events plus the number of background events in a 10% window centered at the tested Higgs mass. The multivariate approach shows an improved performance with respect to the cut-based analysis in all the tested signal hypothesis. The BDT trained with the *per categories* approach has achieved the best results in terms of statistical significance in almost all the inspected points. Moreover, the goodness of the results obtained with the ANN model in all the mass points is a good starting point for using this kind of approach which can be further improved. Furthermore, ANN would allow the implementation of a *parametrized training* strategy [34]. Given the wide range of this search this is expected to be a promising technique, since it makes use of a unique training over the whole mass range.

Bibliography

[1] ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett. B 716 (2012), doi:10.1016/j.physletb.2012.08.020, arXiv:1207.7214* () (cit. on p. ix).

[2] The CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Phys. Lett. B, vol. 716 (2012),* () (cit. on p. ix).

[3] CMS Collaboration. "Search for MSSM Higgs bosons decaying to $\mu^+\mu^-$ in proton-proton collisions at $\sqrt{s}$ = 13TeV ". In: *Phys. Rev. D 98, 030001* () (cit. on pp. ix, 39, 51).

[4] M. Tanabashi et al. "Particle Data Group". In: *Phys. Rev. D 98, 030001* () (cit. on p. 1).

[5] F Halzen and Martin A.D. *Quarks and Leptons: an Introductory Course in Modern Particle Physics*. 1984 (cit. on p. 4).

[6] P. Fayet. "Supergauge Invariant Extension of the Higgs Mechanism and a Model for the Electron and Its neutrino". In: *Nucl. Phys.* "B90" ("1975"). DOI: 10.1016/0550-3213(75)90636-7 (cit. on p. 6).

[7] Henning Bahl Emanuele Bagnaschi et al. "MSSM Higgs Boson Searches at the LHC: Benchmark Scenarios for Run 2 and Beyond". In: (2019). URL: https://arxiv.org/abs/1808.07542 (cit. on p. 7).

[8] Hicham Harouiz Stefano Moretti Abdesslam Arhrib Rachid Benbrik and Abdessamad Rouchad. "A Guidebook to Hunting Charged Higgs Bosons at the LHC". In: *frontiers in Physics* (). URL: https://doi.org/10.3389/fphy.2020.00039 (cit. on pp. 8, 9).

[9] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/vacuum-empty-interstellar-space (cit. on p. 12).

[10] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/pulling-together-superconducting-electromagnets (cit. on p. 12).

[11] Oliver Sim Brüning et al. *LHC Design Report*. Ed. by CERN library copies. Vol. 1, 2, 3. 2012. URL: http://ab-div.web.cern.ch/ab-div/Publications/LHC-DesignReport.html (cit. on p. 13).

[12] Lyndon Evans and Philip Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008). Ed. by IOPscience, S08001. URL: http://iopscience.iop.org/1748-0221/3/08/S08001 (cit. on p. 13).

[13] *CERN Engineering*. URL: http://home.web.cern.ch/about/engineering/radiofrequency-cavities (cit. on p. 13).

[14] *The CMS Experiment*. URL: http://home.web.cern.ch/about/experiments/cms (cit. on p. 13).

[15] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004. URL: http://stacks.iop.org/1748-0221/3/i=08/a=S08004 (cit. on pp. 13, 14).

[16] The CMS Collaboration. "The CMS trigger system". In: *Journal of Instrumentation* 12 (). Ed. by IOPscience, P01020–P01020 (cit. on p. 17).

[17] A.M. at al. Sirunyan. "Particle-flow reconstruction and global event description with the CMS detector. Particle-flow reconstruction and global event description with the CMS detector". In: *JINST* 12.CMS-PRF-14-001. CMS-PRF-14-001-004. 10 (2017). Replaced with the published version. Added the journal reference and DOI. All the figures and tables can be found at http://cms-results.web.cern.ch/cms-results/public-results/publications/PRF-14-001 (CMS Public Pages), P10003. 82 p. DOI: 10.1088/1748-0221/12/10/P10003. URL: https://cds.cern.ch/record/2270046 (cit. on p. 21).

[18] R. Frühwirth. ""Application of Kalman filtering to track and vertex fitting". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* 262.2 (1987), 444 – 450 (cit. on p. 21).

[19] R. Frühwirth. "Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s}$=13 TeV". In: *Journal of Instrumentation* 13 (), P06015–P06015 (cit. on p. 23).

[20] G. P. Salam M. Cacciari and G. Soyez. "The anti-ktjet clustering algorithm". In: *Journal of High Energy Physics* 2008 (), pp. 063–063 (cit. on p. 25).

[21] A. Samuel. *Some Studies in Machine Learning Using the Game of Checkers*. Ed. by New York Springer, pp. 335–365. DOI: 10.1007/978-1-4613-8716-9_14 (cit. on p. 27).

[22] *Machine Learning Crash Course - Google Developers*. URL: https://developers.google.com/machine-learning/crash-course (cit. on pp. 30, 32).

[23] David H. Wolpert. "The Lack of A Priori Distinctions Between Learning Algorithms". In: *Neural Computation* 8.7 (1996), pp. 1341–1390. DOI: 10.1162/neco.1996.8.7.1341. eprint: https://doi.org/10.1162/neco.1996.8.7.1341. URL: https://doi.org/10.1162/neco.1996.8.7.1341 (cit. on p. 32).

[24] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. y O'Reilly Media, Inc, 2019. URL: https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (cit. on p. 32).

[25] Ronald A Fisher. "The use of multiple measurements in taxonomic problems". In: *Annals of eugenics* 7.2 (1936), pp. 179–188 (cit. on p. 33).

[26] Andreas Hoecker et al. "TMVA: Toolkit for Multivariate Data Analysis". In: *PoS* ACAT (2007), p. 040. arXiv: physics/0703039 (cit. on pp. 34, 35, 38, 42).

[27] *Google Brain Team*. URL: https://research.google.com/teams/brain (cit. on p. 35).

[28] *TensorFlow Documentation*. URL: https://www.tensorflow.org (cit. on pp. 35, 42).

[29] *Keras Documentation*. URL: https://keras.io (cit. on pp. 36, 37).

[30] *Accuracy in Machine Learning*. URL: https://deepai.org/machine-learning-glossary-and-terms/accuracy-error-rate#:~:text=Accuracy\%20in\%20Machine\%20Learning&text=Accuracy\%20is\%20the\%20number\%20of,false\%20positives\%2C\%20and\%20false\%20negatives. (cit. on p. 41).

[31] *Classification: ROC Curve and AUC*. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (cit. on p. 41).

[32] Robert E. Schapire. "A Brief Introduction to Boosting". In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, 1401–1406 (cit. on p. 42).

[33] D.Bonacorsi, private communication (cit. on p. 49).

[34] Pierre Baldi et al. "Parameterized Machine Learning for High-Energy Physics". In: (). DOI: 10.1140/epjc/s10052-016-4099-4. URL: arXiv:1601.07913[hep-ex] (cit. on pp. 50, 51).