Finding Higgs to charm decays in electron-proton collisions

by Izzy Harris

Finding Higgs to charm decays in electron-proton collisions

Izzy Harris

201027340

A Thesis Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Science

Under the supervision of Uta Klein

At the

Department of Physics

[May 2017]

Table of Contents

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Date: 12/05/2017

Signature:

Abstract

This report details the ways in which an existing multivariate analysis framework can be optimised for the purpose of measuring Higgs-to-charm decays. Previous studies have been undertaken to predict the precision with which Higgs-to-charm signal events would be measured in a Large Hadron electron Collider operating at an accumulated luminosity of 1000 fb^{-1}. This study presents ways to improve on the results of these predictions and the analysis framework itself by reducing the number of variables used in the framework and adjusting the configuration options of the chosen analysis method to optimise it for this specific task. With this refined multivariate analysis framework using 14 variables, assuming all backgrounds to 2 %, the expected number of Higgs-to-charm signal events which would be detected is 440, with a coupling error of 3.7 % and a signal significance of 16.0.

1 Introduction

The Higgs boson interacts with all the massive elementary particles in the Standard Model, and can therefore decay by many different processes. The focus of this research was a rarer decay mode in which the Higgs couples to a charm-anticharm quark pair. The primary decay mode of the Higgs, with a branching fraction of 60%, is into a bottom–antibottom quark pair. In the standard model, the branching fraction of Higgs-to-charm pairs is only 3%. This means it is difficult to make measurements relating to Higgs-to-charm coupling with a high degree of precision, due to the large multi-jet background [1].

The existence of the Higgs boson was confirmed by its detection at the Large Hadron Collider in July 2012. Further research has been made at the LHC investigating the decay modes of the Higgs, however the experimental conditions at the LHC do not permit for precise measurements relating to Higgs-to-charm couplings to be made. This study investigates the precision with which these measurements could be made at a Large Hadron electron Collider, where the Higgs would be produced during electron-proton collisions. The LHeC is being designed to employ a 60 TeV electron beam which will be used in conjunction with the already existing 7 TeV proton beam at the LHC to instigate high energy electron-proton collisions.

The aim of this project was to improve the precision with which Higgs to charm decays can be identified by a multivariate analysis framework. This is largely an extension of a study undertaken at the University of Liverpool by Dan Hampson, under the supervision of Dr. Uta Klein, where datasets were created modelling the events which would occur during electron-proton collisions at the LHeC. In the same study, these datasets were supplied to a multivariate analysis framework using 26 variables to estimate the precision with which measurements pertaining to Higgs-to-charm coupling events could be made. The objectives of this project were to reduce the number of variables used in the framework and to optimise the analysis framework to obtain higher precision measurements.

2 Electron-Proton Collisions

Collisions between electrons and protons are instances of the electroweak interaction, and can be described as the exchange of either a virtual photon with imaginary mass, or a Z/W boson. Electron-proton collisions proceed via three kinds of scattering processes which can be categorised using the kinematic variables of the collision. In order to describe the kinematics of electron-proton scattering processes it is necessary to define the Lorentz invariant variable Q^2, which is the negative square of the four-momentum transfer from the electron to the virtual photon and is given by:

$$Q^2 = -q^2 = (k - k')^2 \tag{1}$$

Where k is the four-momentum of the incoming electron, and k' is the four-momentum of the outgoing electron.

The energy lost by the electron, v, can be expressed in terms of Q^2 as:

$$v = E - E' = \frac{Q^2}{2m_p} \tag{2}$$

Where E is the energy of the incoming electron, E' is the energy of the outgoing electron, and m_p is the mass of the proton.

The type of scattering which takes place in electron-proton collisions is characterised by the Q^2 value of the collision. As Q is equal to the momentum transferred to the virtual photon, a greater Q^2 value results in a shorter photon wavelength. Figure 1 depicts an idealised spectrum showing the different regions scattering processes fall into with increasing electron energy, denoted here by ω. The greater the initial electron energy, the larger the momentum transfer to the virtual photon. This results in the scattering process having a larger Q^2 value.

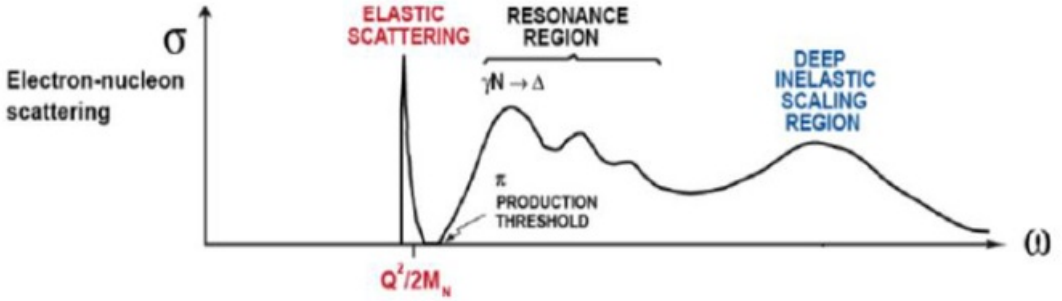


Figure 1: Idealised depiction of variation in scattering cross-section with increasing energy of incident electron [2].

When Q^2 is small, the wavelength of the virtual photon is long in comparison to the size of the proton. The process is effectuated as if the proton were point-like, and the proton structure is not resolved. When the Q^2 value is such that the photon wavelength is comparable to the size of the proton, the proton displays excited states, where particles are temporarily formed within the nucleon [3]. This is referred to as the resonance region, in which the finite size of the proton can be resolved. The existence of these excited states is an initial suggestion that the proton exhibits a composite structure, which can be probed by scattering processes with very high Q^2 values. When the photon wavelength is much less than the size of the proton, the internal structure of the proton may be resolved. In this regime, known as deep inelastic scattering (DIS), the proton separates to give a many particle final state, largely comprising of quarks and gluons, which are referred to as partons [4]. The collision then proceeds via the weak interaction, rather than the electromagnetic interaction. In the case of charged current interactions, DIS processes are mediated by the exchange of W bosons. The Z boson mediates the neutral current interaction.

Electron-proton DIS collisions may be expressed using the generalised equation:

$$e^- + p \rightarrow e^- + X \tag{3}$$

Where X is a hadronic jet which constitutes a 'missing mass', as only the outgoing electron is observed. A further two kinematic variables are introduced when describing deep inelastic scattering processes. Bjorken x is a ratio used to describe the fraction of the proton four-momentum transferred to the struck parton, and is given by:

$$x = \frac{Q^2}{2(p \cdot q)} \tag{4}$$

Where p is the longitudinal four-momentum of the proton. In DIS collisions, it is observed that the cross-sections of the collisions become independent of Q^2 for fixed values of x, a property known as Bjorken scaling. This is explained in more detail in the following section.

The other additional variable is the fractional energy loss of the incoming electron, denoted by y, and expressed as:

$$y = \frac{(p \cdot q)}{p \cdot k} \tag{5}$$

Furthermore, the invariant mass of the final state, W, must be given deeper consideration in inelastic scattering processes. In an elastic collision, the proton remains intact and W is equal to the proton mass. This is not the case in deep inelastic scattering, where the final state hadronic system must contain at least one baryon. This implies that the invariant mass of the final state is greater than the proton mass. Rather than being equal to the square of the proton mass, the square of the centre of mass energy of the hadronic system, W^2, is given by:

$$W^2 = (p + q)^2 = \frac{Q^2(1 - x)}{x} + m_p^2 \tag{6}$$

Where m_p is the mass of the proton [5]. If the proton splits up into its constituent parts, then $W > m_p$, so $Q^2 < 2(p \cdot q)$ and therefore $0 < x < 1$.

These kinematic variables become particularly important when applying multivariate analysis techniques to data obtained during DIS processes, as they are used to define the discriminating variables required to classify data. In addition, the pseudorapidity, η, and the transverse momentum, p_T, of the hadronic jets play a significant role in defining the variables to be used in the MVA framework.

The pseudorapidity is a measure of the angle of the jet relative to the beam axis, and is given by:

$$\eta = -ln\left[\tan\left(\frac{\theta}{2}\right)\right] \tag{7}$$

Where θ is the polar angle between the jet three-momentum and the positive direction of the beam axis. The transverse momentum of the jet is defined as the sine component of the momentum vector p, and is given by:

$$p_T = psin\theta \tag{8}$$

Where θ is again the polar angle with respect to the beam axis. [6]

2.1 The Parton Model

The parton model was proposed by Richard Feynman in 1969 to describe deep inelastic scattering by modelling the nucleon as a collection of point-like constituents (partons) with an effective mass less than the mass of the proton. The premise of this model is that every object with a finite size must have at least one form factor and so scattering cross-sections involving these objects must experience a dependence on Q^2 [7]. Point-like objects do not experience this dependence, and so the parton model can be used to explain Bjorken scaling- the scaling behaviour referred to in the previous section. The applicability of this model can be shown by considering the cross-sections of electron-proton scattering collisions in the cases of elastic, inelastic, and deep inelastic collisions.

For elastic electron-proton collisions where the electron is relativistic and the proton kinetic energy is negligible, the differential cross section is given by the Mott formula:

$$\left(\frac{d\theta}{d\Omega}\right)_{Mott} = \frac{\alpha^2}{4E_K^2 \sin^4\left(\frac{\theta}{2}\right)} \cos^2\left(\frac{\theta}{2}\right) \tag{9}$$

Where $\alpha = e^2/2\pi$ in which e is the charge carried by a single electron, E_K is the kinetic energy of the electron, and θ is the scattering angle.

When the electron is non-relativistic, this reduced to the Rutherford scattering formula:

$$\left(\frac{d\theta}{d\Omega}\right)_{Rutherford} = \frac{\alpha^2}{16E_K^2 \sin^4\left(\frac{\theta}{2}\right)} \tag{10}$$

However, these formulae can only be used to describe elastic electron-proton collisions and encounter a limitation in treating protons as point-like objects. In inelastic collisions, the recoil of the proton is accounted for by the following correction to the Mott formula:

$$\left(\frac{d\theta}{d\Omega}\right)_{Mott} = \frac{\alpha^2}{4E_K^2 \sin^4\left(\frac{\theta}{2}\right)}\left(\cos^2\left(\frac{\theta}{2}\right) + \frac{Q^2}{2m_p^2}\sin^2\left(\frac{\theta}{2}\right)\right) \tag{11}$$

In addition, protons are extended objects and so have a matter density $\rho(r)$, the Fourier transform of which is the form factor, $F(q)$. The cross-section of a collision involving an extended object is modified by the form factor to account for irregularities in the spatial distribution of certain properties of the object. In the case of electron-proton collisions, two form factors are required: one to describe the electric charge distribution of the proton and the other to describe its magnetic distribution [8]. Both form factors are functions of Q^2 and v, therefore the cross-section of the collision is also dependent on Q^2 and v.

The general expression for the differential cross-section of inelastic electron-proton collisions in which only the electron is detected is given by:

$$\frac{d^2\sigma}{d\Omega dE'} = \left(\frac{d\theta}{d\Omega}\right)_{Mott}\left[F_2(Q^2, v) + 2F_1(Q^2, v)\tan^2\left(\frac{\theta}{2}\right)\right] \tag{12}$$

Where $F_1(Q^2, v)$ and $F_2(Q^2, v)$ are the electric and magnetic proton form factors (also referred to as Structure Functions). [9]

Figure 2 shows a graph of the ratio of the differential cross-section, $\frac{d^2\sigma}{d\Omega dE'}$, (for inelastic and deep inelastic collisions) to the Mott cross-section, $\left(\frac{d\theta}{d\Omega}\right)_{Mott}$, as a function of Q^2. This graph was plotted using experimental data taken during the SLAC-MIT program, in which a series of electron-proton scattering experiments were conducted to investigate the proton structure.

In this particular experiment, results were collected for increasing values of W, where a value of $W > m_p$ is attributed to a collision in which the proton splits into its constituent partons. Inelastic collision measurements were taken with $W = 2$ GeV and DIS collision measurements were taken with $W = 3$ GeV and $W = 3.5$ GeV. Two separate sets of measurements with scattering angles of 6° and 10° were made. This data demonstrates a decreasing dependence on Q^2 of the differential cross-section as the value of W is increased. Also included in the graph is the ratio of the elastic scattering cross-section to the Mott cross-section for $\theta = 10°$, accentuating the difference in behaviour of inelastic scattering cross-sections.

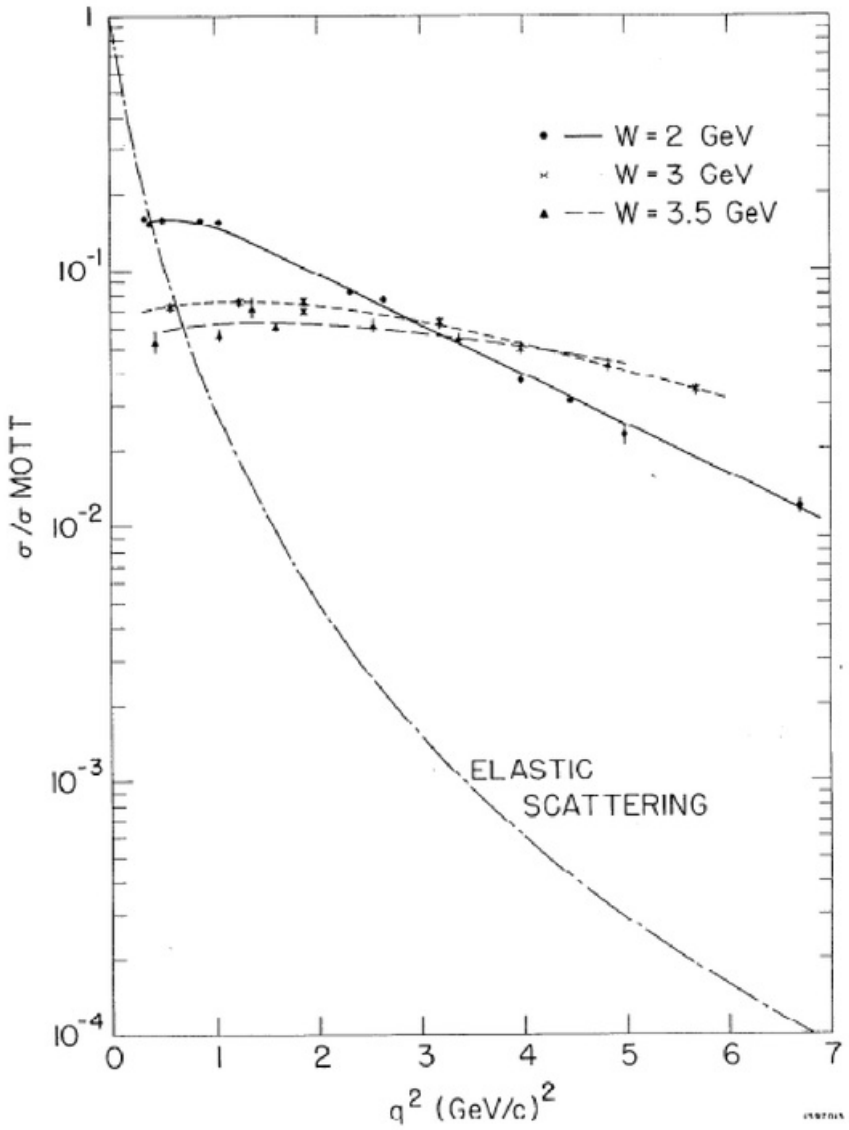


Figure 2: Graph showing $\frac{d^2\sigma}{d\Omega dE'} / \left(\frac{d\theta}{d\Omega}\right)_{Mott}$ as a function of Q^2 for $W = 2$, 3, 3.5 GeV, and $\frac{d\sigma}{d\Omega} / \left(\frac{d\theta}{d\Omega}\right)_{Mott}$ as a function of Q^2. $\frac{d\sigma}{d\Omega} / \left(\frac{d\theta}{d\Omega}\right)_{Mott}$ is the cross-section for elastic electron-proton scattering calculated for $\theta = 10°$, using the electric dipole form factor [9].

The rapid decrease in the elastic scattering cross-section as Q^2 increases is attributed to the strong dependence of $\frac{d\sigma}{d\Omega}$ on the proton form factors, and hence on Q^2, due to the collision occurring as if the proton were an extended object. The inelastic cross-sections are more weakly dependent on Q^2, and the deep inelastic scattering cross-sections are almost entirely independent of Q^2. In the deep inelastic limit, the cross-section depends only on the ratio x rather than depending on Q^2 and ν separately- the previously mentioned phenomenon of

Bjorken scaling. This is evidence to suggest that the incident electron is scattered by point-like objects rather than extended objects in high energy inelastic collisions.

The experiments conducted during the SLAC-MIT program validated Feynman's predictions and provided evidence that the proton is not a fundamental object, but a composite object consisting of partons. The partons which the proton comprises of have since been recognised as quarks and gluons, verifying that the parton model is compatible with the theory of quantum chromodynamics. The mechanism for Higgs production during DIS collisions can be explained in terms of the parton model.

2.2 Higgs Production in Deep Inelastic Scattering

Higgs boson production can be accessed in DIS collisions between electrons and protons via either the charged current process or the neutral current process, both of which allow for vector boson fusion, during which two bosons fuse together and couple to the Higgs boson.

In the neutral current process, a Z boson is exchanged, and the flavour and charge of the electron and quark involved in the collision are preserved. In the charged current process, either a W^+ or W^- boson is exchanged, and conservation laws demand a change in the flavour of both the electron and quark. In a charged current interaction between an electron and a quark which exists within a proton, the electron is converted into an electron neutrino, and the quark is converted from a down quark to an up quark, or vice versa. In this type of interaction the neutrino is undetected, however its presence is inferred from the apparent breaking of conservation laws.

Both types of interaction give rise to Higgs production via vector boson fusion. In this process, the two bosons radiated by the particles involved in the collision couple to the Higgs. The charged current interaction gives rise to a WWH coupling, whilst the neutral current interaction gives rise to a ZZH coupling. The Feynman diagram for such processes is shown in figure 3:

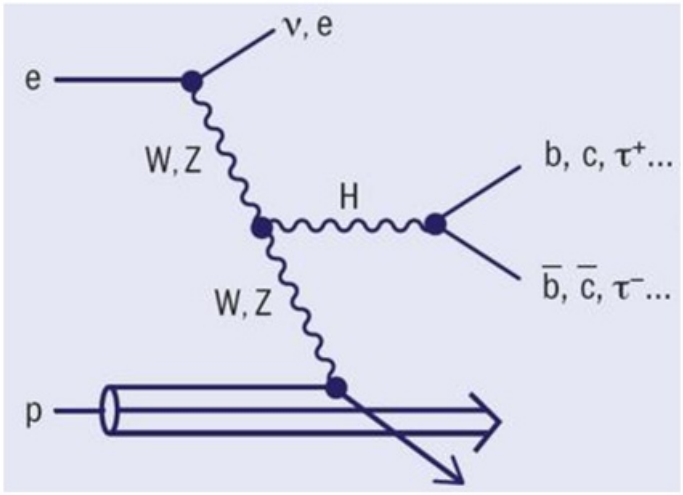


Figure 3: Feynman diagram showing Higgs production via W/Z vector boson fusion in a collision between an electron and one of the quarks within a proton [10].

3 The Large Hadron Electron Collider

The Large Hadron electron Collider (LHeC) is a proposed upgrade to the current LHC. The aim of this upgrade is to complement the already existing proton-proton collider with an electron-proton collider which will operate in the same TeV energy range. This presents the opportunity for further DIS measurements to be taken, with the potential to reveal completely the partonic structure of the proton and provide information on physics beyond the Standard Model.

The electron-proton collider is intended to operate synchronously with the proton-proton collider, making most efficient use of both the proton and electron beams and allowing for a high integrated luminosity. It has been shown that it is feasible for the LHeC to achieve an electron-proton luminosity of 10^{33} cm^2s^{-1}. The formula for the luminosity of the anticipated LHeC configuration is given by:

$$L = \frac{N_e N_p f \gamma_P}{4\pi \varepsilon_p \beta^*} \tag{13}$$

Where L is the luminosity of the proton beam, N_e is the number of electrons per bunch, N_p is the number of protons per bunch, f is the bunch frequency, ε_p is the normalised proton transverse beam emittance, and β^* is the value of the proton beta function at the interaction point, assumed to be equal in x and y [11].

The 7 TeV LHC proton beam consists of bunches of 1.67×10^{11} protons with an inter-bunch spacing, Δ, of 25 ns. f is given by the reciprocal of Δ and hence is equal to 40 MHz. The normalised transverse emittance of the proton beam is 3.75 µm. The LHeC electron beam would contain $N_e = 10^9$ electrons per bunch. Given that the electron beam current, I_e can be expressed as:

$$I_e = eN_e f = \frac{P}{E_e} \tag{14}$$

Where e is the charge on a single electron, P is the electron beam power, and E_e is the electron beam energy in GeV, it can be demonstrated that a luminosity of 10^{33} cm^{-2} s^{-1} can be achieved by the LHeC using only a 60 GeV, 6.4 mA electron beam. This would require an electron beam power of 384 MW, which introduces a constraint to the LHeC design. With a 100 MW power limit, the design must implement an energy recovery system if the beam power needed to reach the projected luminosity of 10^{33} cm^{-2} s^{-1} is to be achieved.

3.1 Electron Accelerator

The LHeC is being designed to operate using a Linac Ring (LR) configuration, with two energy recovery linear accelerators (ERLs) arranged in a racetrack shape positioned tangential to the LHC. The LHeC would exist in a tunnel separated from the LHC operation, other than at the collision point and in the surrounding interaction region [12]. A schematic diagram of the current accelerator design is shown below in figure 4:

Figure 4: Diagram showing the proposed Linac Ring structure of the LHeC, with two 10 GeV energy recovery linacs arranged parallel to one another and tangential to the LHC [13].

The accelerator design employs two approximately 10 GeV ERLs, both of which the electrons pass through three times. With this arrangement, it is intended that electrons will be accelerated to 60 TeV when they reach the interaction point and collide with the 7 TeV protons circulating the LHC. The energy recovery system allows for an electron beam power of 384 MW to be achieved by recuperating the energy of the spent beam. This is done by returning the beam 180° out of phase through the radio frequency power system used to accelerate the electrons [11]. The double-ERL configuration allows for the projected luminosity of 10^{33} cm^{-2} s^{-1} to be realised despite the 100 MW wall-plug power limit for the lepton beam.

3.2 Detector Simulation

To predict the Higgs-to-charm coupling measurements which could be taken at the LHeC, a simulated electron-proton collision dataset was needed for the chosen multivariate analysis technique to be applied to. The work involved in simulating this data had been undertaken previously by Dr Uta Klein, Dan Hampson, and Ellis Kay. The programs MadGraph5, Delphes, and Pythia were used to simulate Higgs production in electron-proton collisions and the subsequent decay modes of the Higgs. The simulated production mechanism was Higgs formation by W vector boson fusion, i.e. the charged current process described in section 2.2. The software used to do this is discussed in more details in the following sub-sections.

3.3.1 MadGraph

MadGraph is a fully automatised tool which can generate both cross sections and events for a collision process when supplied with only the process and the physical model to which the process belongs. Collision processes adhering to not only the Standard Model, but models with additional particles and interactions may be used as input in MadGraph5.

To begin the exercise of event simulation, the user need only input the initial and final state particles, from which MadGraph generates all the Feynman diagrams for the process and the code required to evaluate the matrix element at a given phase space point [14]. The process-dependent information, such as sub-process amplitudes and phase space mappings, is then passed to MadGraph's own event generator: MadEvent

This package uses a technique referred to as 'Single-diagram-enhanced multichannel integration' to produce weighted and unweighted events by integrating the squared amplitude over the phase space of the final state particles [15]. This procedure allows for easy determination of the appropriate mapping to give an efficient integration over the phase-space and can be expressed as:

$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{total}|^2 \tag{15}$$

Where A_i is the amplitude corresponding to a single Feynman diagram and $A_{total} = \sum_i A_i$ is the total amplitude [16]. f_i is one of n functions which form a complete basis to give the function to be integrated over the set of phase-space variables. This approach works by expressing the phase space as a basis of amplitude functions, where the peak of each amplitude can be efficiently mapped by a single channel, and has an advantage in giving high unweighting efficiencies for multi-particle final states.

MadGraph5 was used to separately simulate both the signal and background events produced by electron-proton collisions. These events were generated in the form of four vectors and passed to Pythia, a particle shower simulator which simulates the hadronisation and subsequent decays of the events generated by MadEvent.

3.3.2 Pythia

Pythia is an event generator which simulates the perturbative evolution and eventual hadronisation, or fragmentation, of partons, whereby the coloured partons form colourless hadrons due to colour confinement [17]. Pythia describes the parton distributions, the initial and final state parton showers and the interactions between the partons in the shower. In creating the model dataset for electron-proton collisions, Pythia was used to simulate the hadronisation and decay of the events generated by MadGraph5. The events were then passed in the form of four-vectors to Delphes, a detector simulation package.

3.3.3 Delphes

The Delphes detector response simulation includes a tracking system, calorimeters, and a muon system, which are embedded into a magnetic field. The software simulates the direction of the final state particles by the magnetic field to the calorimeters, allowing for analysis of the events generated by Pythia in terms of calorimeter response, whilst accounting for the effect of the magnetic field [18]. Delphes was used to generate two datasets, where a labelled dataset was to be used in the first stage of the analysis, and a dataset of unknown signal and background composition was to be used in the final stage.

4 Multivariate Analysis

4.1 TMVA

The TMVA (Toolkit for Multivariate Analysis) is a package available within ROOT, which is an object oriented, C++ based analysis framework developed by CERN. The TMVA provides a range of algorithms capable of large-scale data analysis using variables supplied by the user. This tool was designed for high energy physics purposes, making it suitable for the intended use of identifying Higgs-to-charm coupling events.

The TMVA analysis is performed in several separate stages, the reasons for which are explained in the following section. The first stage is the training stage, which is initiated by the creation of an object belonging to the Factory class and the booking of the input variables. Once the chosen MVA method has been trained, it is then tested on a dataset separate to the training data to ensure a statistically independent evaluation of the performance of the MVA method. This stage is also governed by the Factory object.

The Factory class generates the MVA methods and provides member functions to specify the training and test datasets. This class also calculates the linear correlation coefficients of the input variables by performing pre-analysis and pre-processing of the training dataset to assess the basic properties of the input variables [19]. The Factory object performs the interactions between the user and the TMVA modules. A flow diagram of the training stage is shown on the left in figure 5:

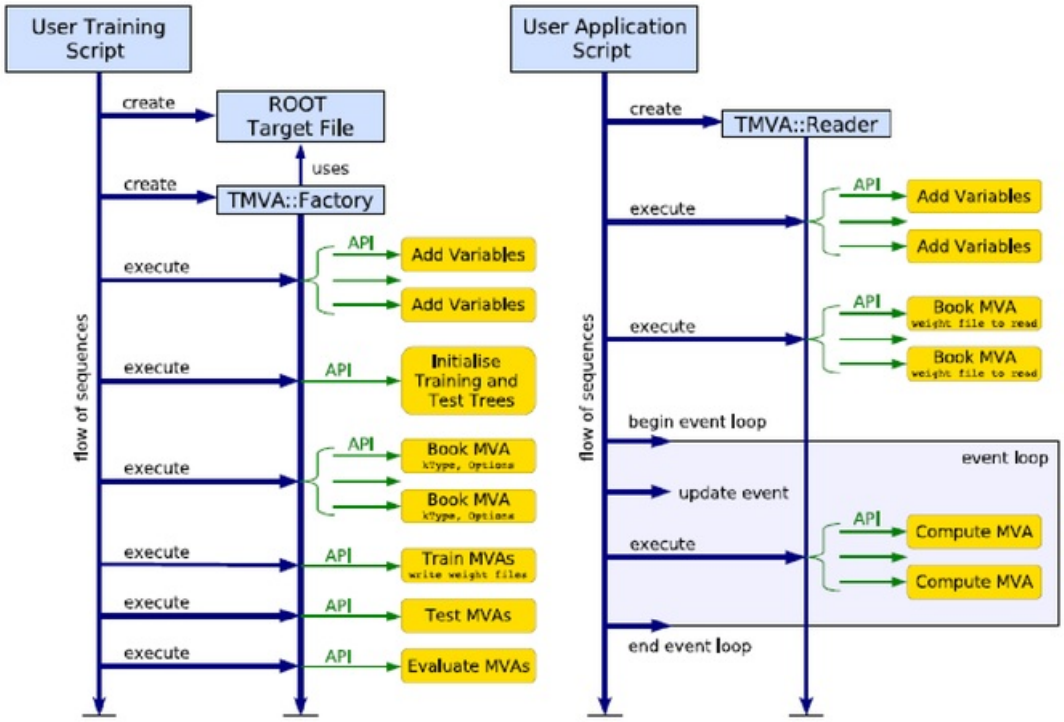


Figure 5: Shown on the left is the flow of a typical TMVA training application, and on the right flow of a typical TMVA training application [19].

The final stage is the application stage, shown on the right in figure 5. The application stage is handled by the Reader class. To begin the application stage, the user creates a Reader object and registers the input variables with the Reader in the same way as with the Factory. The discriminating variables must be identical in both cases. The reader object interfaces the communication between the user and the MVA methods, analogously to the Factory object in the training stage. In the application stage, the user runs the event loop, which fills a vector corresponding to the input variables used to train the chosen MVA method. The vector is then passed to the Reader object, which computes the MVA response value for each event in the application dataset.

4.2 MVA Methods

The multivariate analysis techniques used within the TMVA all belong to a group of machine learning methods known as supervised learning algorithms. These analysis techniques can be used for classification problems, where discrete values are predicted (which class a data point belongs to), or regression problems, where continuous values are predicted. The TMVA supports the use of most of the available supervised learning methods for both classification and regression problems. However, the analysis type needs to be specified when instantiating a new Factory object, and so it is important to correctly identify the type of problem before beginning analysis. As the aim of using the TMVA in this project was to classify an event as either a signal or background event, a classification algorithm was required and was specified as such within the TMVA.

Supervised learning algorithms require training data from which they can induce a model that can then be applied to classify unlabelled data [20]. This is achieved by using the training data to determine the mapping function which, in the case of classification, describes a decision boundary, and applying this mapping function to the unlabelled dataset.

This necessitates that any MVA process utilising a supervised learning technique is split into at least two stages: a training phase and an application phase. In the case of using a method within the TMVA, testing and evaluation phases are conducted automatically after the training stage. This serves as an assessment of the performance of the MVA method on labelled data and allows for an estimation of how the method may perform on an unlabelled dataset with the same attributes.

Once the chosen MVA method has been trained and tested, the performance of the method is evaluated in terms of signal efficiency and background rejection. These two quantities respectively describe the proportion of events correctly classified as signal events and the proportion of background events correctly classified as such. The signal efficiency of the classifier, for both the training and test data, at 3 benchmark values of background efficiency, B, (given by $B = 1 - background\ rejection$) is displayed to the user at the end of the evaluation phase, along with the separation and discrimination significance of the classifier. Along with the information printed to standard output, the user can also view graphical illustrations showing the classifier performance via the TMVA GUI.

The separation, $\langle S^2 \rangle$, of a classifier, y, describes the separation between the signal and background distributions and is defined by the integral:

$$\langle S^2 \rangle = \frac{1}{2} \int \frac{\left(\hat{y}_s(y) - \hat{y}_B(y)\right)^2}{\left(\hat{y}_s(y) + \hat{y}_B(y)\right)} \, dy \tag{16}$$

Where $\hat{y}_s$ is the signal probability density function of y and $\hat{y}_B$ is the background probability density function of y [19]. A separation of 1 corresponds to absolutely no overlap between the signal and background distributions, and therefore it is beneficial to maximise $\langle S^2 \rangle$. A greater separation between the signal and background distributions allows for an increased signal purity whilst keeping the partial loss of the signal sample to a minimum. This can be easily visualised by referring to figure 6, which is an example classifier output distribution. A classifier output of 1 is a certain signal classification, and a classifier output of -1 is a certain background classification. The signal events with a value greater than 0 are referred to as 'true positives', as they have been correctly classified, and the signal events with a value less than 0 are referred to as 'false negatives', as they have been misclassified as background events. Equivalently, background events with a value greater than 0 are referred to as 'false positives', and background events with a value less than 0 are referred to as 'true negatives'.

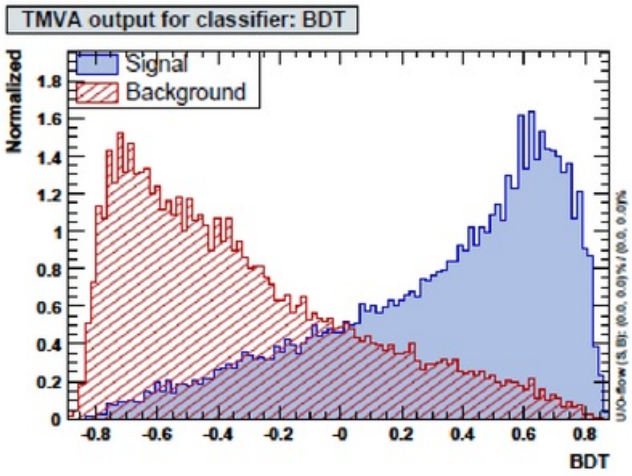


Figure 17: Example of a classifier output distribution for a Boosted Decision Tree [19].

It can be seen that by cutting on the classifier output where the signal sample peaks, a small proportion of the signal sample will be lost, and a small proportion of misclassified background events will be included along with the signal sample. A greater separation between the signal and background distributions would decrease these proportions and therefore improve the signal purity.

A plot of signal efficiency against background rejection (referred to as a 'Receiver Operating Characteristic' or 'ROC' curve) can be automatically generated using the TMVA GUI. The integral of this function is also given in the evaluation results displayed to the user when the training stage is complete. A greater area under the ROC curve indicates better performance of the classifier, as this equates to a larger proportion of true positive classifications. This is demonstrated by the graph in figure 18.

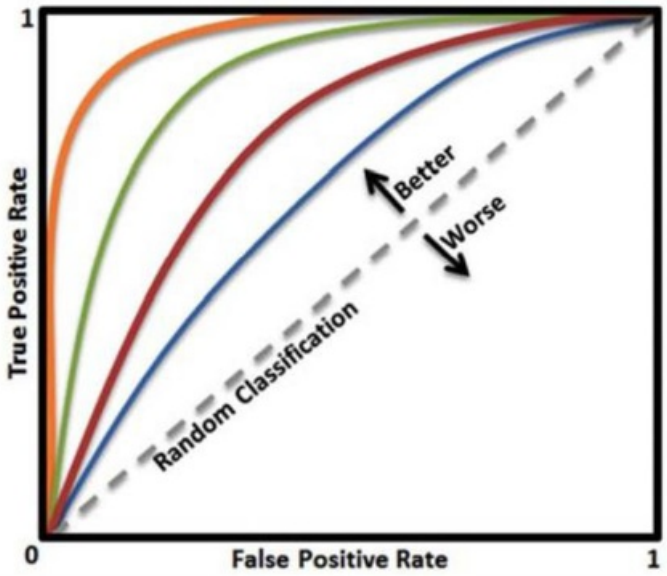


Figure 18: Graph showing different ROC curve shapes and how shape corresponds to classifier performance. In this example, the most powerful classifier is represented by the orange curve [21].

The results of the test and evaluation phases can therefore be used to assess several aspects of the classifier performance. In terms of identifying Higgs-to-charm coupling events, this translates to being able to quantify the error on the number of signal events identified by the chosen MVA method when it is passed an unlabelled dataset. This means it is possible to predict the number of signal events and the associated error which would be seen when applying the method to real, experimental data. This prediction is of course reliant on the assumptions that the simulated data accurately represents experimental data, and that the MVA method performs approximately as well on unlabelled data as it does on labelled data. The latter assumption, however, does not hold true if the classifier has been overtrained.

4.3 Overtraining

Overtraining of supervised learning methods occurs when the classification model becomes too complex due to containing an excessive amount of rule sets for the given problem. Instead of learning a general mapping function which can be used to classify events in similar datasets, the method learns features which are specific to the training dataset. To train supervised learning methods correctly, they must be trained on a dataset which is truly representative of any unlabelled datasets they may be used to analyse. Overtraining is typically caused by either training sample overtraining or Data/Monte Carlo overtraining [22].

In training sample overtraining, the MVA method is tailored to perform extremely well when classifying events in the training dataset by using features belong to the training data, but not necessarily an unlabelled dataset. This form of overtraining can be detected by comparison of the results when the classifier is given the training data with the results when the same classifier is given a statistically independent test dataset. If the classifier performs significantly

better when given the training data rather than the test data, it has been overtrained on the training dataset.

Data/Monte Carlo overtraining occurs when the training data uses real data for background events, and a Monte Carlo simulation for signal events. Because the simulated data isn't completely accurate, some differences between the features of the background and signal events may be due to imperfections in the simulation instead of physical differences. The training algorithm can then be overtrained by learning to separate signal-like background events from simulated signal events, based on differences between real and simulated data. This becomes a problem when the algorithm is given a dataset containing real signal and background events, as the classifier will be trained to identify simulated signal events. Because every signal and background process was simulated separately in MadGraph5, this form of overtraining could be ruled out as a source of error in this project. It was clear that any disparity between the performance of the chosen classifier with the training and test datasets was due to training sample overtraining.

5 Initial Data

An initial estimation for the number of Higgs-to-charm coupling events which could be detected at the LHeC was made by Dan Hampson under the supervision of Dr. Uta Klein. Dan used the TMVA to analyse the same model dataset used in this project, with his chosen MVA method being the default BDT algorithm.

Using a BDT cut of 0.2 and assuming a luminosity of 1000 fb^{-1}, and that all backgrounds were known to 1% (2%), Dan predicted that the number of Higgs-to-charm signal events detected would be 474, with a coupling error of 3.9 % (5 %) and a $\frac{S}{\sqrt{N}}$ value of 15.6 (12.5) [23].

5.1 Initial Variables

This estimate was obtained using the TMVA BDT method to analyse the simulated detector data with 26 variables, many of which were defined using the kinematic variables of electron-proton deep inelastic scattering introduced in section 2. Each of the 26 variables, with their respective names and definitions, is shown in figure 19.

Variable Number	Variable Name	Definition
1	jmet	Missing Energy
2	jtrack	Number of tracks
3	jpt[0]	p_T of highest p_T jet
4	jpt[1]	p_T of second highest p_T jet
5	jpt[2]	p_T of third highest p_T jet
6	jeta[0]	Pseudorapidity of highest p_T jet
7	jeta[0] - jeta[2]	Pseudorapidity difference between highest and third highest p_T jets
8	jeta[1] - jeta[2]	Pseudorapidity difference between second and third highest p_T jets
9	jsiptrack[0]	SIP of highest p_T jet
10	jsiptrack[1]	SIP of second highest p_T jet
11	jsiptrack[2]	SIP of third highest p_T jet
12	pjetNp[0]	Positive jet lifetime probability of highest p_T jet
13	pjetNp[1]	Positive jet lifetime probability of second highest p_T jet
14	pjetNp[2]	Positive jet lifetime probability of third highest p_T jet
15	jphi[0] - jmetphi	Azimuthal angle between highest p_T jet and missing energy
16	jmass[0] + jmass[1] + jmass[2]	Sum of 3 highest p_T jet masses
17	jdij	Mass of two lowest η jets
18	jtri	Mass of three highest p_T jets
19	Min pjetNm	Minimum negative jet lifetime probability
20	pjetNm[0] + pjetNm[1] + pjetNm[2]	Sum of negative jet lifetime probabilities of 3 highest p_T jets
21	pjetNm[2]	Negative jet lifetime probability of third highest p_T jet
22	njet	Number of jets
23	jdij12	Mass of second and third lowest η jet
24	Max jeta	Most forward η jet
25	Min jeta	Lowest η jet
26	jtri31	Mass of 3 lowest η jets

Figure 19: Names and definitions of the original 26 variables used in the MVA framework.

As stated previously, one of the principle aims of this project was to reduce the number of the variables used in analysing the simulated data. This is because using a larger number of variables increases the potential for there to be a source of error in the measurement estimations which is not accounted for in the coupling error. Since the variables passed to the TMVA are defined using simulated kinematic variables, a poor simulation of the reaction kinematics will result in an inaccurate estimation the number of signal events which can be detected, and of the precision of the measurements which can be achieved.

5.2 Training Performance

Passing the default BDT method these variables and training the algorithm on the simulated dataset produced a number of plots and results which could be used to measure the initial performance and set a standard for the classification results upon which to improve. The data provided by the TMVA evaluation is shown in figure 20.

Signal Efficiency			Area under ROC curve	Separation	Significance
B = 0.01	B = 0.1	B = 0.3			
0.121	0.524	0.818	0.840	0.348	0.580

Figure 20: Table displaying results from the TMVA evaluation phase when using the default BDT method to analyse the model dataset with the initial 26 variables.

The area under the ROC curve was within a smaller range of 1 than of 0.5, implying that this analysis framework was already producing relatively good classification results. In addition, the separation between the signal and background distributions was 0.348, so the signal and background distributions were at least partially separated, as a separation of 0 indicates complete overlap of the distributions. The ROC curve and classifier output distribution corresponding to these results are shown in figures 21 and 22, respectively.

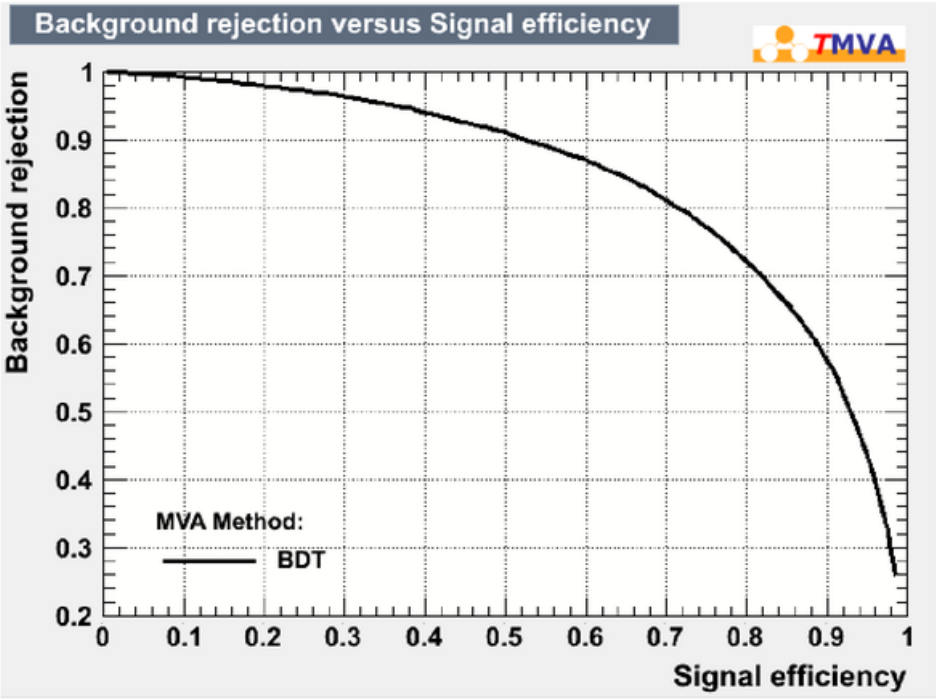


Figure 21: ROC curve produced when using the default BDT method to analyse the model dataset with the initial 26 variables. The area under the curve is 0.840.

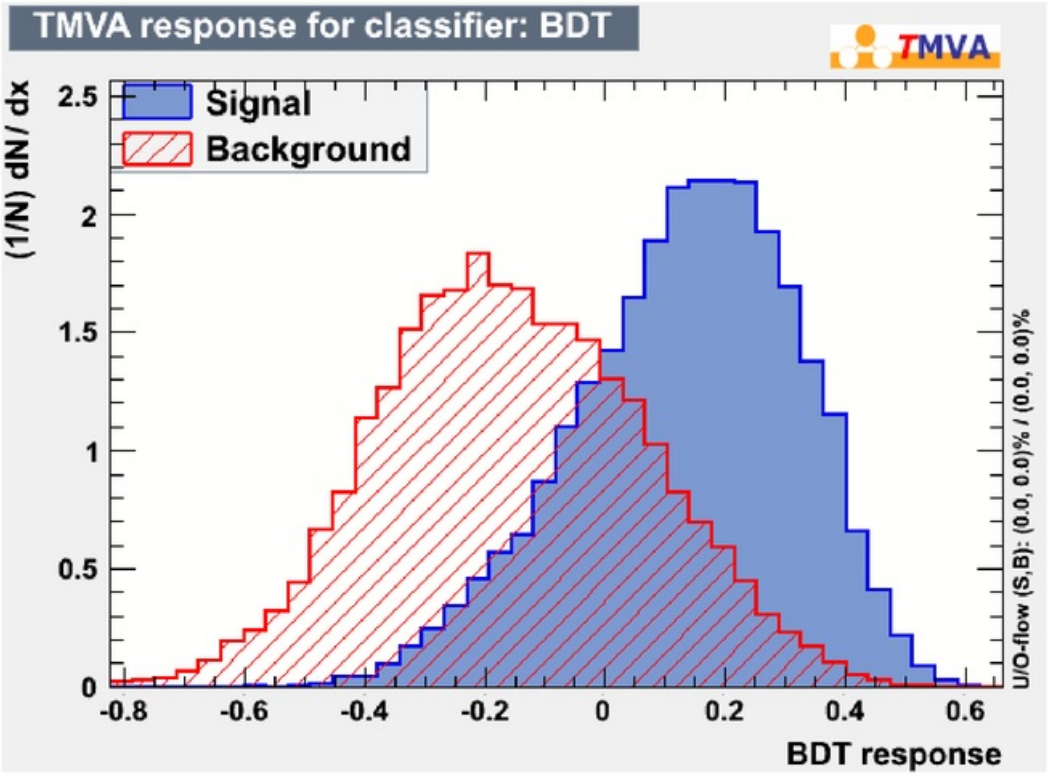


Figure 22: Classifier output distribution produced when using the default BDT method to analyse the model dataset with the initial 26 variables.

This stage of the TMVA analysis also generates superpositions of the signal and background distributions for each of the discriminating variables supplied to the MVA framework. These plots are helpful in identifying which variables may have the greatest use in classification, and are shown in figures 23 and 24.

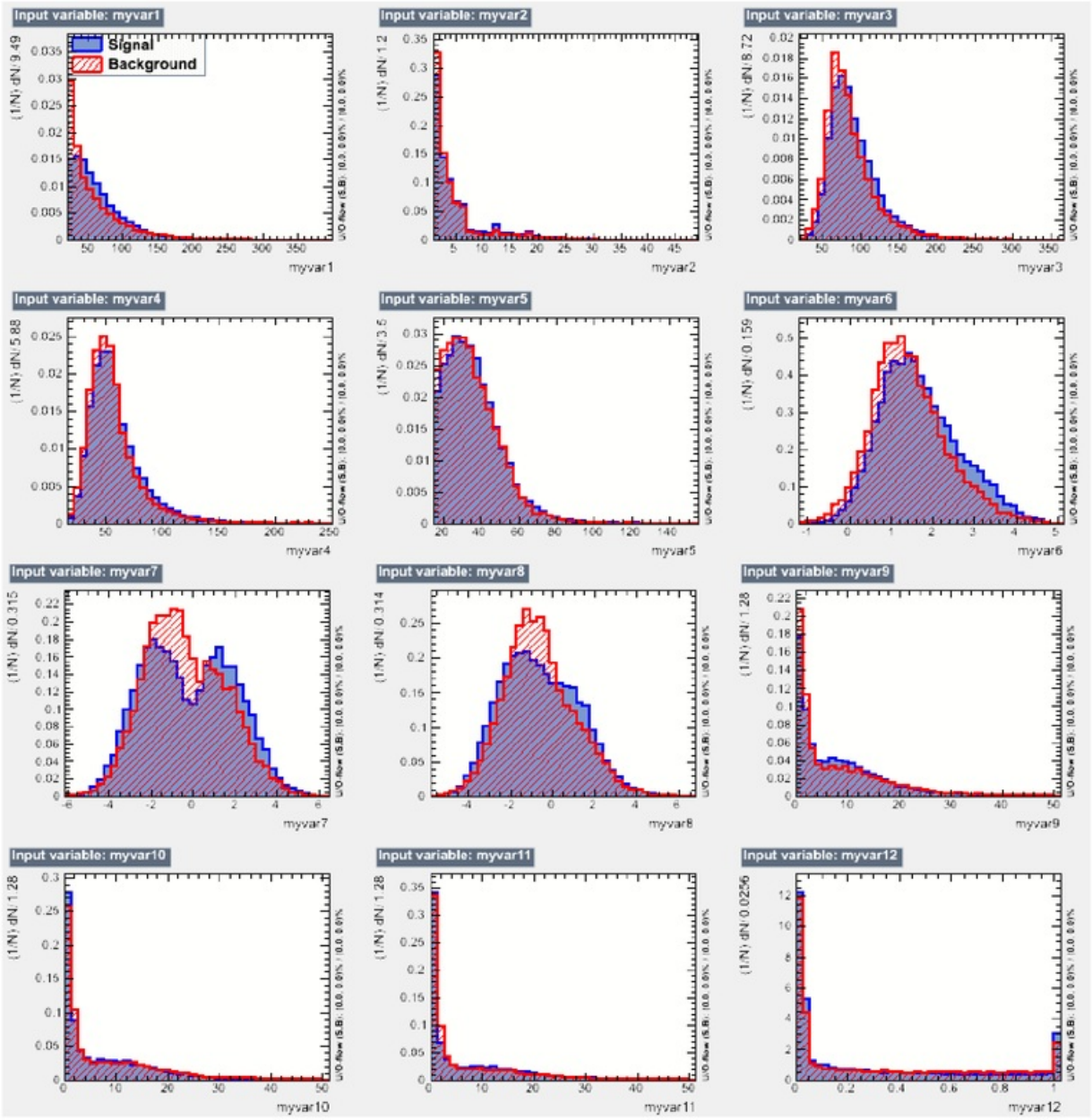


Figure 23: Plots showing superposition of signal and background distributions for each of the variables 1-12 of the original 26 variables, where the units are omitted for technical reasons.

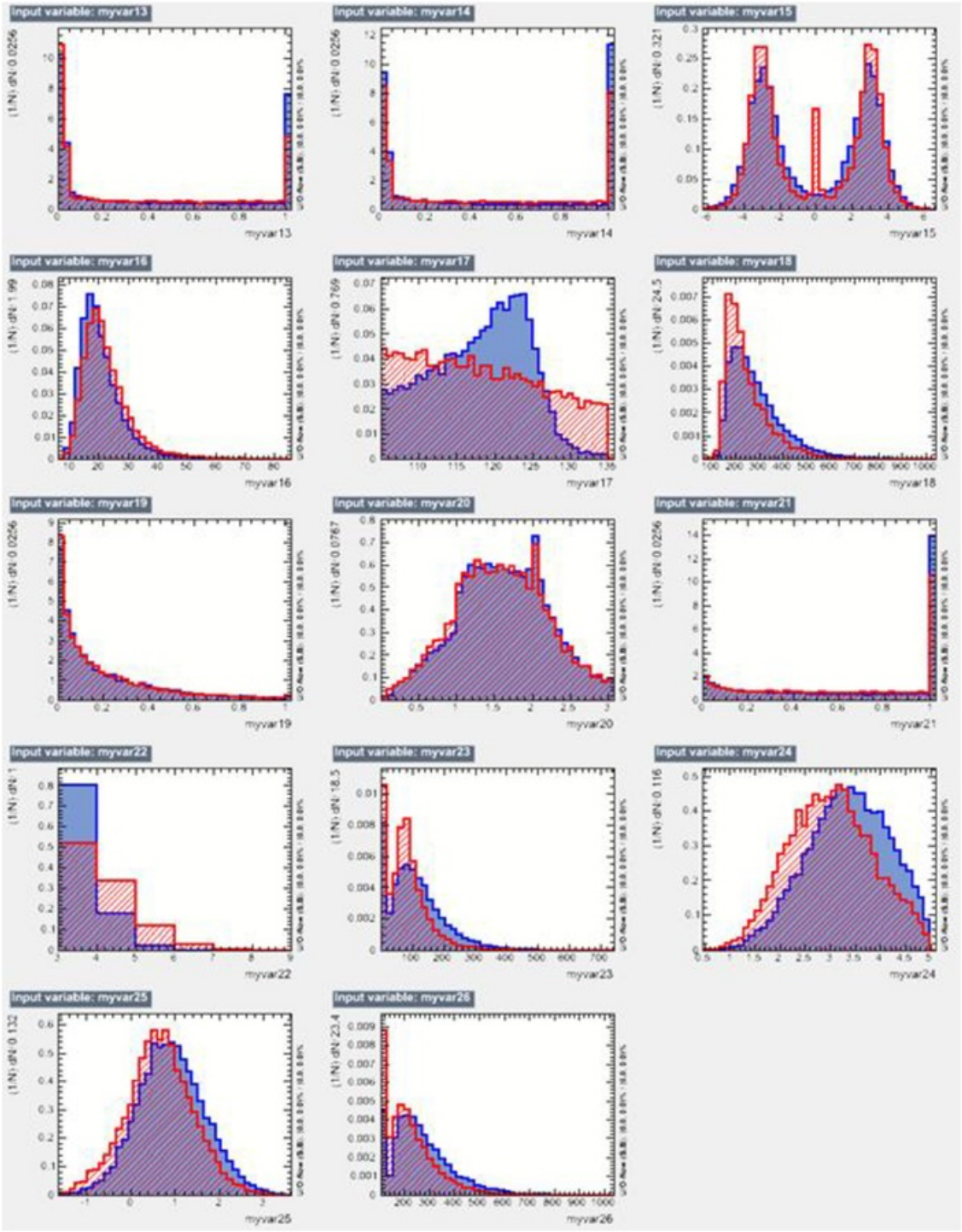


Figure 24: Plots showing superposition of signal and background distributions for each of the variables 13-26 of the original 26 variables, where the units are omitted for technical reasons.

Where a significant proportion of the signal distribution is not overlaid by the background distribution, or vice versa, for a particular variable, it is likely that the variable will be comparatively useful in classifying signal events. This is because each variable contributes to the overall signal and background distributions for the classifier, and less overlap between these distributions allows for a greater signal purity. Examples of this can be seen with variables 7, 8, 20, 22, and 24, where the difference between the signal and background distributions is particularly discernible.

Figures 23 and 24 would later be used to cross-check conclusions made when collecting data to determine the discriminating power of each variable. The results from the TMVA evaluation phase, shown in figure 20 would serve as a reference point for the performance of the chosen classifier. Improving upon these results would ultimately improve the final results for the number of Higgs-to-charm coupling events identified in the application phase. The initial estimation for the number of Higgs-to-charm decays which could be measured was obtained by applying the default BDT method to an unlabelled dataset and is discussed in the following sub-section.

5.3 Application Performance

The BDT response in the TMVA application stage can be used to generate a plot showing the distribution of h→ $c\bar{c}$ and background events over a range of BDT output values. The response of the default BDT method using the aforementioned 26 variables is shown in figure 25:

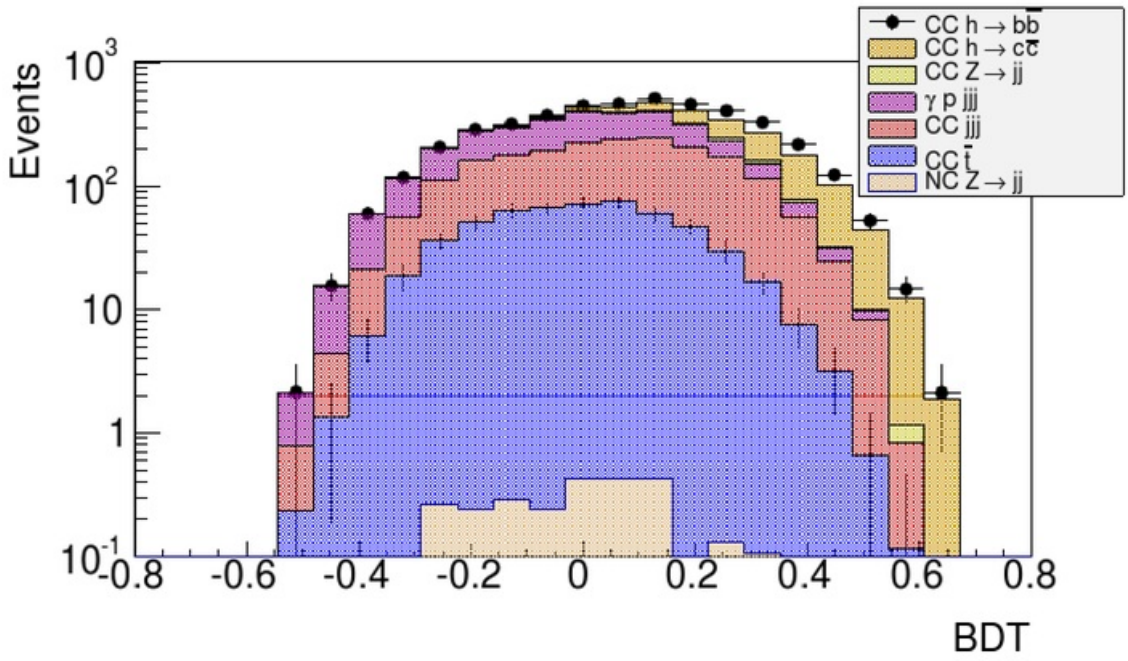


Figure 25: BDT output distribution when using default BDT method with original 26 variables.

The results which would be obtained when cutting along the x-axis of this distribution are also computed by the TMVA, and are shown in figure 26:

BDT Cut	Number of Signal Events	Number of Background Events	Coupling Error (%)	Significance
-0.40	657	3761	5.1	10.4
-0.35	657	3710	5.0	10.5
-0.30	657	3624	5.0	10.0
-0.25	656	3499	4.9	10.8
-0.20	654	3317	4.8	11.0
-0.15	651	3101	4.7	11.3
-0.10	645	2874	4.6	11.6
-0.05	638	2605	4.5	12.0
0.00	621	2216	4.3	12.6
0.05	601	1891	4.2	13.2
0.10	571	1564	4.0	13.7
0.15	530	1209	3.9	14.3
0.20	475	901	3.9	14.7
0.25	405	638	4.0	14.7
0.30	326	416	4.2	14.4
0.35	243	241	4.5	13.7
0.40	140	102	5.6	11.8

Figure 26: Table of results showing measured number of Higgs-to-charm events and background events when using default BDT method with original 26 variables.

Figure 26 shows the expected number of Higgs-to-charm signal events measured at each BDT cut, along with the percentage error on this value and the signal significance, Z. Z is a statistical measure of the reproducibility of a measurement, and is approximated by:

$$Z \approx \frac{S}{\sqrt{B}} \tag{16}$$

Where S is the number of signal events and B the number of background events [24].

Using the default BDT method with the 26 variables defined in figure 19 and assuming all backgrounds to 1 %, it is predicted that the most precise measurement of the number of Higgs-to-charm coupling events would be made at a BDT cut of 0.2, giving 475 events with a coupling error of 3.9 % and a signal significance of 14.7. The intention was to improve on this result by investigating how to best analyse the model dataset using the TMVA, with the aim of decreasing the coupling error and increasing the signal significance.

6 Methodology

Improving the analysis framework began with selecting the MVA method most suited to this classification problem, and reducing the number of discriminating variables passed to the method. Once the classifier had been chosen and the number of variables reduced, the classifier configuration options would be explored to determine whether the performance of the method could be improved upon.

6.1 MVA Method Selection

It was necessary to determine which supervised learning algorithm was most appropriate for the model datasets. The TMVA supports numerous machine learning algorithms, many of which are variations on other algorithms using the same principle, and fall under the broader categories of Cut Optimisation, 1-Dimensional Likelihood Estimators, Multidimensional Likelihood Estimators, Linear Discriminant Analysis, Function Discriminant Analysis, Neural Networks and Boosted Decision Trees (BDT). The TMVA also includes a Support Vector Machine (SVM) algorithm and Friedman's RuleFit method.

To narrow down the selection of possible supervised algorithms to choose from, they were each researched to understand how they operate and in which contexts they are most useful. Part of this research involved consulting the 'Which MVA method should I use for my problem?' section of the TMVA manual, which provided the following table as a general assessment of each MVA method's properties:

		MVA METHOD									
	CRITERIA	Cuts	Likeli-hood	PDE-RS / k-NN	PDE-Foam	H-Matrix	Fisher / LD	MLP	BDT	Rule-Fit	SVM
Perfor-mance	No or linear correlations	★	★★	★	★	★	★★	★★	★	★★	★
	Nonlinear correlations	○	○	★★	★★	○	○	★★	★★	★★	★★
Speed	Training	○	★★	★★	★★	★★	★★	★	★	★	○
	Response	★★	★★	○	★	★★	★★	★★	★	★★	★
Robust-ness	Overtraining	★★	★	★	★	★★	★★	★	★[39]	★	★★
	Weak variables	★★	★	○	○	★★	★★	★	★★	★	★
Curse of dimensionality		○	★★	○	○	★★	★★	★	★	★	
Transparency		★★	★★	★	★	★★	★★	○	○	○	○

Figure 27: Table showing an assessment of MVA method properties. A 'good' performance in any respect is denoted by the symbol '★★', a 'fair' performance by '★', and a poor performance by 'o'. The 'curse of dimensionality' property refers to the phenomenon whereby some MVA methods require larger populations of training statistics when the number of input variables is increased. This inevitably elongates the processing time [19].

The 'likelihood' method refers to the 1-Dimensional Likelihood Estimator. PDE-RS, k-NN, and PDE-Foam are Multidimensional Likelihood Estimators, and are referred to as the range search method, the k-Nearest Neighbour method, and the self-adapting phase-space binning method, respectively. The multilayer perceptron (MLP) is a feed-forward artificial neural network and is recommended by the TMVA developers as the neural network to use with the TMVA due to its speed and flexibility [19]. The H-Matrix and Fisher classifiers are both linear discriminant analysis methods.

Examining figure 27 with the input variables in mind, it is immediately obvious that the group of potentially suitable MVA methods need be narrowed down to PDE-RS, PDE-Foam, MLP, BDT, RuleFit and SVM, due to the nonlinear correlations between the discriminating variables. Further considering the requirements of the analysis framework, the Support Vector Machine was ruled out as a consequence of the algorithm's slow response and training speeds.

A concerning observation regarding boosted decision trees and neural networks was their susceptibility to overtraining, and the reduced performance of neural networks when passed weak input variables. If either of these methods were to be used, it would be necessary to perform checks for overtraining and consider ways in which overtraining could be reduced. Additionally, ensuring that only the most useful discriminating variables are passed to the framework would be required.

Both the PDE-RS and PDE-Foam methods suffer greatly from the 'curse of dimensionality' and require exceedingly large datasets to be properly trained. This wasn't an issue in this project, as the training datasets were sufficiently large to populate the phase space. However, it meant that the processing time of these algorithms would likely be very slow. Both methods also perform poorly when given weak input variables and are consequently subject to the same considerations as MLPs in this respect. In addition, the PDE-RS approach has a poor response speed, meaning that applying this classifier to unlabelled data would be a slow process.

It therefore seemed that coming to a conclusion regarding which MVA method was most suitable for analysing the model data would involve compromising performance in some respects in favour of others. For example, the extent to which a quicker training speed of one MVA method is advantageous is dependent on how the training speed compares to other methods and how well the method performs in other areas compared to other methods. It is impossible to quantify the importance of factors which aren't directly linked to the quality of the results and choosing the most appropriate MVA method therefore becomes subjective. Ultimately, it would be necessary to simply compare the evaluation results of each MVA method when trained on the simulated dataset and take qualitative factors such as speed and potential for optimisation into account as secondary considerations. The initial conclusion drawn from consulting the TMVA manual was that further research into Boosted Decision Trees, Multidimensional Likelihood Estimators, and Neural Networks would be beneficial.

Each of the MVA methods in this selection was then researched more deeply to ascertain how data classification was achieved, to understand more specifically the features a dataset must have for the method to perform optimally, and to explore the capacity each method had for

optimisation and being tailored for a certain classification problem. These methods are described in more detail in the following sub-sections.

6.1.1 Decision Trees

Decision trees utilise the features of the supplied dataset to learn decision rules which can then be applied to classify unlabelled data. The decision rules are the functions which map the input to the expected output by means of a linear logic flow which, for decision trees, is equivalent to a series of 'if' conditions being met to give the classification outcome. Unlabelled data can then be scrutinised against these conditions and will be classified as the outcome of the decision rule for which every condition evaluates to true [25].

A decision tree structure consists of 'nodes' and 'branches' where each node represents a binary condition for an event in the input data to be tested against, and each branch represents the outcome of this test. At each node, the input variable which maximises the splitting of the dataset is used to generate the criterion for each event to be tested against. This way, a large dataset can be successively split into different classes until a terminal node is reached where the data can be assigned a class label. The terminal nodes of the decision tree are referred to as leaf nodes, and in the case of the simulated detector data, represent either one of two class labels: signal or background. Every event in the given dataset can therefore be labelled as either a signal event or a background event, depending on which type of leaf node the event is assigned to. A diagram of a simple decision tree structure is shown below in figure 28:

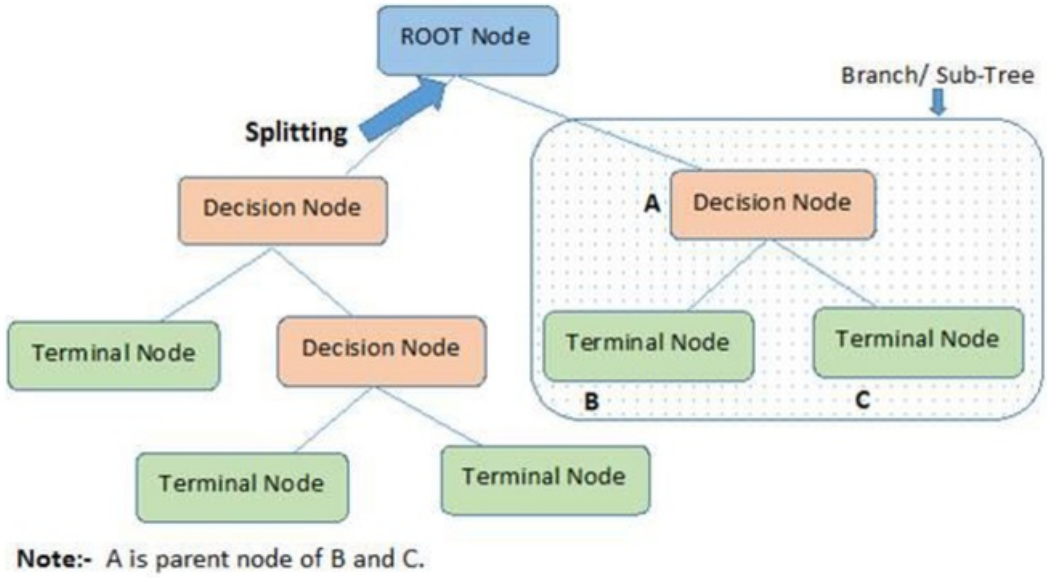


Figure 28: Schematic of a decision tree structure, consisting of a root node, decision nodes and several terminal or 'leaf' nodes [26].

The tree begins with a root node which branches into two decision nodes representing different conditions, followed by further branching into more decision nodes until the variable fulfils a stop condition at a leaf node. The maximum depth of the decision tree is the greatest number of queries allowed between a root node and a leaf node, in other words the maximum number of queries before classification is obtained.

6.1.2 Boosted Decision Trees

Decision trees alone are considered to be 'weak' supervised learning algorithms. An individual decision tree would not be effective when used to solve a complex problem requiring a powerful classifier, such as the challenge of correctly identifying Higgs to charm decays. Single decision trees are prone to bias in event classification as a result of making a large amount of approximations when determining mapping functions. Decision trees also respond poorly to statistical fluctuations in the training data, increasing the variance of the results.

Boosting was introduced to improve the performance of machine learning algorithms by combining numerous weaker learning algorithms into one, with the effect of reducing bias and stabilising the resulting ensemble of algorithms. In the case of decision trees, this means creating a 'forest' comprising of many trees. This begins with training a single tree on a labelled dataset where all events are equally weighted. Events which are classified as signal are given a score of +1 and events which are classified as background are given a score of -1. The tree is then boosted by increasing the weights of events which were misclassified by the tree. A new tree is created which is trained on the re-weighted dataset, and this process is repeated until a BDT forest is formed [27]. The scores for each event are renormalised and the average taken to give every event a value in the range of -1 to +1, where +1 is a certain signal classification and -1 is a certain background classification.

For the BDT forest, there are many parameters which can be adjusted to optimise the technique, such as the number of trees, the maximum depth, the node size, and variables pertaining to the boosting algorithm. This meant that the BDT method had great optimisation potential.

6.1.3 Probability Density Functions

For the infinite range of values that a continuous variable may take, the probability of the variable taking the value of any exact number is 0. If the probability were greater than 0, then the total probability for all values would be greater than 1, as the variable would have a finite probability of equalling each number in the infinite set of possible values. Probability density functions take a comparative approach to solve this problem by giving the relative probability of the variable taking any value within a specific range. If the probability density around a particular point is large compared to other points, there is a greater likelihood of the variable taking a value within a small range of that point [28].

6.1.4 Projective Likelihood Estimator (PDE approach)

The projective likelihood estimator approach uses probability density functions to create a model which reproduces the signal and background input variables. For each event, the

product of all the signal probability densities for every input variable is taken and normalised by the sum of the signal and background likelihoods [19]. The result of this gives the probability of the event being a signal event. Due to the speed of this process, this analysis method is suitable for large sets of data, however it is rarely used for experimental purposes due to the nature of the likelihood model. An inherent feature of this model is that the input variables are assumed to have no correlations between each other. In practice, this assumption often leads to reduced performance of the projective likelihood estimator method, due to experimentally observed correlations between input variables.

This method had no optimisation potential, with the only option in the TMVA allowing for a transformation of the likelihood output, which would not improve classification performance.

6.1.5 Multidimensional Likelihood Estimator (PDE range-search approach)

The multidimensional likelihood estimator extends the PDE approach so that the phase space in which the data is sampled has the same number of dimensions as there are input variables. In this approach, the input data for the signal and background events are stored in binary-search trees and a range-searching algorithm is used to sample the signal and background densities around the phase-space points chosen for classification [29]. The range of this method is decided by the size of the multidimensional 'box' which surrounds the chosen point in the phase-space. The volume of this box is user-defined, presenting a single option for refining the method for a specific problem.

6.1.6 Artificial Neural Networks

Artificial neural networks (ANNs) are connectionist systems intended to replicate the way human brains process information, with many units working simultaneously to solve a specific problem. All neural networks consist of a large array of interconnected processing elements, referred to as 'neurons', and may be either feed-forward networks or recurrent networks. Feed-forward networks allow signals to travel in one direction only: from input to output, whereas recurrent networks allow signals to travel in both directions by introducing loops into the network [30].

The recommended neural network to be used with the TMVA is the Multilayer Perceptron (MLP). Multilayer Perceptrons are a distinct kind of feed-forward ANN in which the neurons are organised into layers. The TMVA offers various configuration options for the MLP method, however very few directly correspond with classifier performance as with the BDT options, and therefore optimising the MLP method would be a more complex task with less potential for improving the classifier performance.

Each neuron takes numerous input signals with weighting coefficients from the given dataset to produce a single output. The input signals are individually weighted according to their usefulness for classification of the given dataset. The weightings can be greater than or less than 1, so can effectively increase or decrease the input signal. The product of each input and its corresponding weight is taken by the net input function, and the sum of all these products is passed to the activation function. The activation function then determines the extent to which the signal produced will continue through the network to affect the final classification

result. Shown in figure 29 is a diagram of a single neuron with an unspecified activation function:

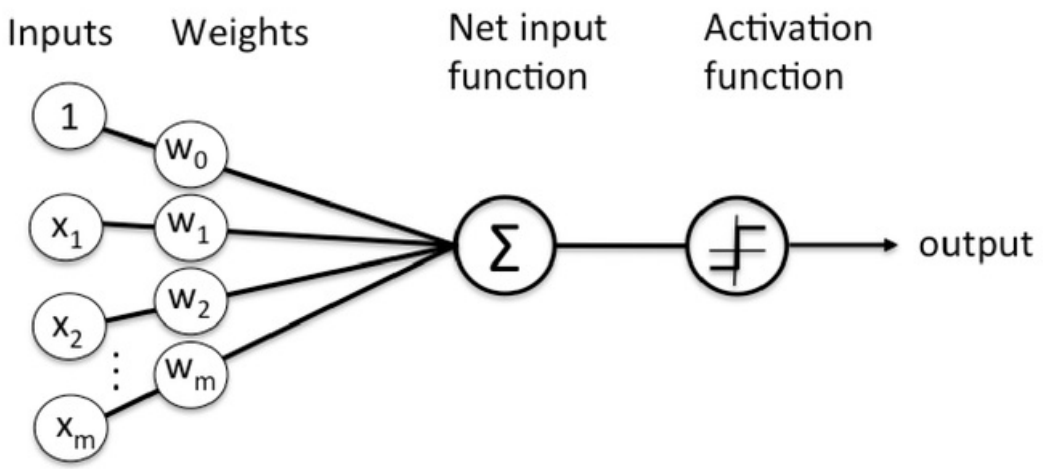


Figure 29: Single neuron diagram showing how multiple weighted inputs are passed to the net input function, which calculates the net input signal and sends it to the activation function [31].

The perceptron itself is a specific kind of neuron which uses the Heaviside step function to produce a binary output. The Heaviside step function returns 1 if the input is 0 or positive, and 0 if the input is negative. A group of neurons arranged in a row is referred to as a layer, where the output of the neurons in one layer is directed to at least one node in the next layer. A diagram of how the topology of a Multilayer Perceptron network might look is shown in figure 30.

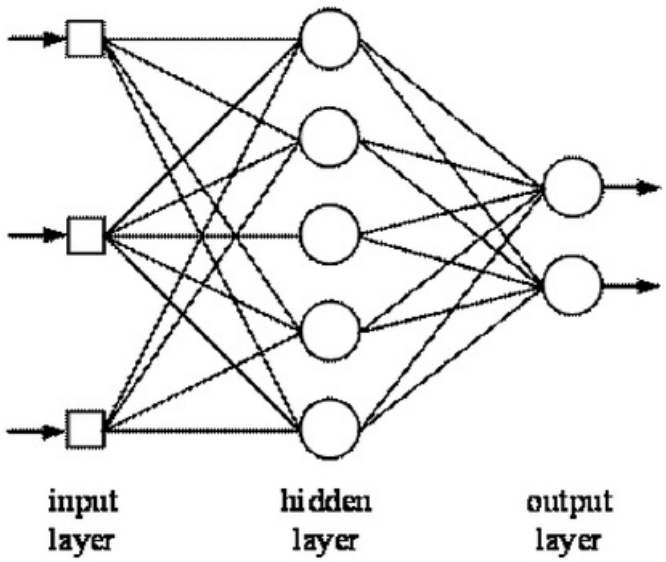


Figure 30: Schematic of a simple 3-layer feed-forward artificial neural network, with two 'exposed' input and output layers and an intermediate hidden layer [32].

In this kind of network, raw data fed into the ANN is represented by the behaviour of the neurons in the input layer. The activity of these units determines the behaviour of the neurons in the hidden layer, which in turn determine the activity of the neurons in the output layer. In each layer, the input signal is reweighted, determining the activity of the neurons in the subsequent layers. Multilayer networks may be far more complex than this, but operate on the same principles.

Whilst the neural network is being trained, all neurons are in training mode and map input data to an expected output to develop 'firing rules'. The firing rules are used to determine whether the neuron fires (i.e. continues processing the inputs) for a given combination of inputs and can be applied to additional input patterns for which the node was not trained. In the case that a node is receives an input pattern it was not supplied with in the training stage, the node compares the pattern to the most similar pattern which caused the node to fire in the training stage. The node also compares the input pattern with the most similar pattern in the training stage which didn't result in the node firing. This allows the node to establish which of the two training patterns the unknown pattern has more elements in common with. If the unknown pattern is more similar to the former, then the firing rule is satisfied and the input signal will continue to the activation function. It is this ability of ANNs to extend their pattern recognition capabilities past data patterns they have been trained on to which artificial neural networks owe their high flexibility [30].

6.1.7 MVA Method Results

Researching this group of MVA methods had provided some useful insight into the types of classification problems they are most suited to, and into how they can be optimised for a particular problem. Although all three methods were powerful enough classifiers to perform well when being used to solve a problem as complex as identifying charm signal events in

Higgs decay data, the BDT method appeared to offer the most potential for being tailored to this problem. Within the context of the TMVA, optimisation of the BDT method seemed to be most easily attained due to the vast number of configuration options and the capacity to adjust each option. Additionally, from comparison of the ROC curves for each classifier in the example diagram taken from the TMVA manual and shown in figure 31, it was anticipated that the BDT method may also provide the strongest out-of-the-box classification performance.

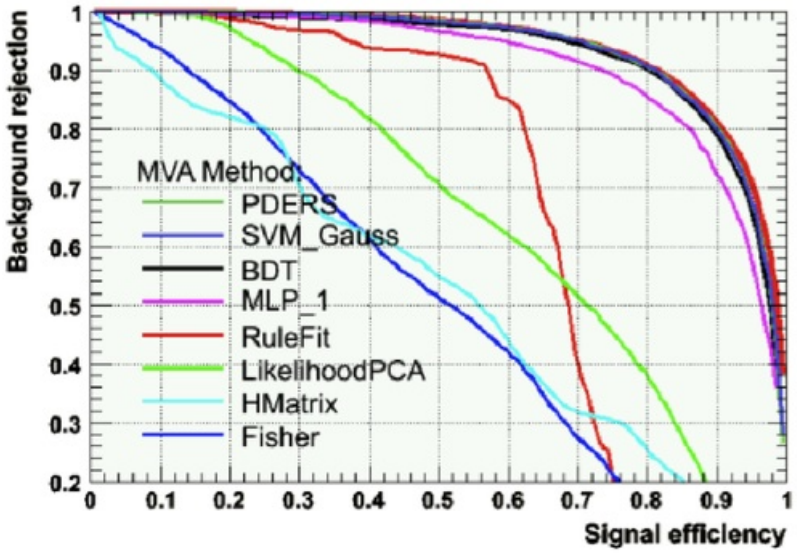


Figure 31: Comparison of ROC curves produced by each MVA method supported by the TMVA [19].

To test this prediction, each algorithm was trained on the simulated detector data, to compare how each performed and conclude which was most appropriate for analysing this dataset, and by extension any unlabelled dataset with the same attributes. The methods were tested in their default configuration so that the performance prior to optimisation could be seen, and the potential for optimisation could then be taken into consideration.

The performance measures of the area under the ROC curve and the signal-background separation were used to asses each method, where larger values of either quantity were preferable in terms of the classifier performance. The significance and relative speed were also taken into account when deciding on the MVA method most suited to analysing the simulated dataset. This data was inserted into the table shown in figure 32.

MVA Method	Area under ROC curve	Separation	Relative Speed
BDT	0.840	0.348	Quick
PDE-RS	0.831	0.336	Slow
k-NN	0.747	0.184	Slow
MLP	0.843	0.356	Medium

Figure 32: Table of data comparing performance during training stage of the BDT, PDE-RS, k-NN, and MLP MVA methods in their default configurations.

It is clear from this data that the MLP method performed better than both the BDT and Multidimensional Likelihood Estimator algorithms in terms of both the area under the ROC curve and the separation between the signal and background distributions. This refutes the prediction that the BDT method would provide the best performance in its default configuration. However, the BDT performance comes close to rivalling the MLP performance and the BDT method is advantageous with respect to the speed with which the method was trained. The PDE-RS algorithm performed worse than the BDT algorithm with respect to the area under the ROC curve and the signal-background separation, and took considerably longer to train than the BDT method, as did the k-NN algorithm. The PDE-RS and k-NN algorithms are immediately seen to give worse classification performance than the other methods in all respects, and hence were given no further consideration.

In light of this information, it was decided that the BDT method was to be used for the identification of Higgs-to-charm decays in the model dataset. Although the MLP algorithm is seen quantitatively to be the strongest classifier in this set of methods, all methods were tested in their default configuration, and the combination of the BDT algorithm running comparably faster and having many configuration options made it a practical choice as the most suitable MVA method for this classification problem.

6.2 Variable Reduction

The process of reducing the number of variables began with creating a ranking showing the effect of the variables on the performance of the analysis framework. This was done by repeatedly running the TMVA training phase with the default BDT method whilst taking out a single variable each time. After each variable had been removed, the evaluation results were saved and the variable was replaced. The consecutive variable was then removed and the process was repeated until results had been obtained showing the effect of the removal of every variable. The values recorded to measure the BDT performance were the area under the ROC curve and the separation between the signal and background distributions. This data is shown in figure 33.

Variable Number	Area under ROC curve
1	0.833
2	0.826
3	0.841
4	0.842
5	0.839
6	0.841
7	0.838
8	0.842
9	0.824
10	0.839
11	0.839
12	0.843
13	0.841
14	0.841
15	0.833
16	0.831
17	0.826
18	0.839
19	0.841
20	0.841
21	0.840
22	0.826
23	0.838
24	0.841
25	0.842
26	0.839

Variable Number	Separation
1	0.344
2	0.320
3	0.352
4	0.352
5	0.347
6	0.351
7	0.346
8	0.353
9	0.352
10	0.348
11	0.347
12	0.354
13	0.350
14	0.350
15	0.334
16	0.330
17	0.322
18	0.348
19	0.351
20	0.351
21	0.348
22	0.319
23	0.345
24	0.349
25	0.352
26	0.344

Figure 33: Data taken when each variable was independently removed during the training stage. The table on the left shows the area under the ROC curve upon the removal of each variable number, and the table on the right shows the separation between the signal and background distributions upon the removal of each variable number.

The effect that the removal of a particular variable had on the area under the ROC curve and the separation between the signal and background data could be used to determine the variable's effectiveness in discriminating between signal and background events. A reduction in either of these values meant that the BDT performance had suffered as a result of a variable being removed, with a greater reduction indicating a more significant impact on the classifier performance. When these values saw the greatest decrease upon removal of a variable, it was clear that this variable was one of the most significant contributors to the analysis framework.

The variables could then be ordered, with a lower ranking number corresponding to a more useful discriminating variable. Due to using two measures of classifier performance, two separate rankings were created: one showing the variable importance in terms of the ROC curve area and the other in terms of the signal-background separation. The 15 variables with the greatest significance for both the ROC curve and the signal-background separation were selected and placed in two new tables for additional testing. These tables display the variables in order of their rank and are shown below in figure 34:

Variable Number	Area under ROC curve
9	0.824
2	0.826
17	0.826
22	0.826
16	0.831
1	0.833
15	0.833
7	0.838
23	0.838
5	0.839
10	0.839
11	0.839
18	0.839
26	0.839
21	0.840

Variable Number	Separation
22	0.319
2	0.320
17	0.322
16	0.330
15	0.334
1	0.344
26	0.344
23	0.345
7	0.346
5	0.347
11	0.347
10	0.348
18	0.348
21	0.348
24	0.349

Figure 34: The top 15 variables taken from the two rankings of variable importance in terms of the area under the ROC curve and the signal-background separation.

It is interesting to note that although the order of the variable ranking in these two tables is different, the same group of variables appears in both tables, with the exception of variable 9 being included in the 15 greatest contributors to the ROC curve area but not to the signal-background separation, and the converse being true for variable 24. This suggests that this variable ranking method is an effective way of assessing the separate contribution of each variable, since the rankings were made using two measures of classifier performance, yet are largely in agreement with one another. In addition, checking these rankings against the signal and background distributions superimposed for each variable in figures in 23 and 24 verifies the prediction that the most useful variables would display the least overlap in these distributions. This is particularly visible in the case of variable 17, which is ranked 3rd in both tables.

The next step was to retrain the BDT algorithm with a selection of variables taken from these rankings to determine the effect of a significant variable reduction on the BDT performance. The aim was to decrease the number of variables by as many as possible whilst retaining the values of the performance measures to within a certain range. It was decided that the area under the ROC curve and the signal-background separation should be kept to within 1% and

5%, respectively, of their original values, as these reductions were effectively negligible and the loss in classifier performance would be compensated for by the increased robustness of the analysis framework.
Initially, only the first ten variables from each ranking (a total of 11 variables) were used to test the BDT analysis, however this proved to be too aggressive a reduction. The performance of the analysis was dramatically reduced, giving a ROC curve area of 0.815 and a signal-background separation of 0.299. This was an approximately 3.0% decrease in the area under the ROC curve and a 14.1% decrease in the separation between the signal and background distributions

The next variable from each of the rankings was then included, which was variable 10 in the ROC curve ranking and variable 11 in the separation ranking. This combination of 13 variables produced better results than the top 11 variables only, giving a ROC curve area of 0.821 and a signal-background separation of 0.311. This time, the area under the ROC curve was decreased by approximately 2.3% and the signal-background separation by 10.7%.

To increase the performance further, the variable ranked 13^{th} in both rankings, which was variable 18, was again included in the framework, giving 14 variables in total. This gave an ROC curve area of 0.826 (a reduction of 1.7% from the original value) and a signal-background separation of 0.321 (a reduction of 7.8% from the original value).

A 15^{th} variable (variable 21) was then added to the combination in attempt to restore these quantities to within the desired range of their original values. This was still unsuccessful, as the ROC curve area only increased to 0.827 and the signal-background separation to 0.322, suffering undesirable reductions of around 1.5% and 7.5%, respectively.

Whilst this method had proven beneficial in identifying the most useful variables out of the 26 used, it was unsatisfactory in sustaining the performance of the classifier when using significantly fewer variables, and so a different approach needed to be taken.

It was important to explore whether substituting certain variables with other potentially more useful variables may allow for a significant reduction in the number of discriminating variables used within the framework without an appreciable reduction in the classification performance. Several of the variables identified as comparatively small contributors to the framework were substituted for newly defined variables which were expected to have more use in classification, and some variables were combined, giving an alternative set of 25 variables shown in figure 35.

Variable Number	Variable Name	Definition
1	jmet	Missing Energy
2	jtri - jtri31	Difference between mass of three highest p_T jets and mass of 3 lowest η jets
3	jphi[1] - jmetphi	Azimuthal angle between second p_T jet and missing energy
4	jphi[2] - jmetphi	Azimuthal angle between third highest p_T jet and missing energy
5	jpt[2]	p_T of third highest p_T jet
6	jeta[0]	Pseudorapidity of highest p_T jet
7	0.5*(jeta[1] + jeta[2]) - jeta[0]	Pseudorapidity difference between average of second and third highest p_T jets and highest p_T jet
8	0.5*(jeta[0] + jeta[1]) - jeta[2]	Pseudorapidity difference between average of highest and second highest p_T jets and third highest p_T jet
9	jsiptrack[0] - jsiptrack[1]	SIP difference between highest and second highest p_T jets
10	jsiptrack[1]	SIP of second highest p_T jet
11	jsiptrack[2]	SIP of third highest p_T jet
12	pjetNp[0]	Positive jet lifetime probability of highest p_T jet
13	pjetNp[1]	Positive jet lifetime probability of second highest p_T jet
14	pjetNp[2]	Positive jet lifetime probability of third highest p_T jet
15	jphi[0] - jmetphi	Azimuthal angle between highest p_T jet and missing energy
16	jmass[0] + jmass[1] + jmass[2]	Sum of 3 highest p_T jet masses
17	jdij	Mass of two lowest η jets
18	jtri	Mass of three highest p_T jets
19	Min pjetNm	Minimum negative jet lifetime probability
20	pjetNm[0] + pjetNm[1] + pjetNm[2]	Sum of negative jet lifetime proabilities of 3 highest p_T jets
21	pjetNm[0]	Negative jet lifetime probability of highest p_T jet
22	njet	Number of jets
23	jdij - jdij12	Difference in mass of two lowest η jets and mass of second and third lowest η jets
24	Max jeta	Most forward η jet
25	Min jeta	Lowest η jet

Figure 35: Alternative set of 25 variables created after replacing and combining several variables in the original set of 26.

This combination of variables resulted in an immediate improvement in classification performance, producing an ROC curve with an area of 0.845 and a signal-background separation of 0.359. Although this was a step forward in improving the classifier performance and removing a single variable, the large number of variables was still a problem. To resolve

this issue, the same process of eliminating each variable independently and recording the evaluation results was then repeated for this set of 25 variables and two rankings based on the area under the ROC curve and the separation between the signal and background distributions were again created. The recorded values for these quantities upon removal of each variable are shown in figure 36, and the two rankings generated using this data are shown in figure 37.

Variable Number	Area under ROC curve
1	0.837
2	0.843
3	0.833
4	0.841
5	0.843
6	0.842
7	0.841
8	0.844
9	0.843
10	0.844
11	0.841
12	0.840
13	0.842
14	0.842
15	0.840
16	0.832
17	0.820
18	0.840
19	0.842
20	0.843
21	0.842
22	0.830
23	0.841
24	0.844
25	0.844

Variable Number	Separation
1	0.343
2	0.357
3	0.335
4	0.351
5	0.355
6	0.353
7	0.353
8	0.357
9	0.355
10	0.357
11	0.352
12	0.349
13	0.354
14	0.355
15	0.349
16	0.333
17	0.308
18	0.349
19	0.354
20	0.356
21	0.354
22	0.328
23	0.350
24	0.359
25	0.357

Figure 36: Data taken when each variable was independently removed from the alternative set of 25 variables. The table on the left shows the area under the ROC curve upon the removal of each variable number, and the table on the right shows the separation between the signal and background distributions upon the removal of each variable number.

Variable Number	Area under ROC curve
17	0.820
22	0.830
16	0.832
3	0.833
1	0.837
12	0.840
15	0.840
18	0.840
4	0.841
7	0.841
11	0.841
23	0.841
6	0.842
13	0.842
14	0.842

Variable Number	Separation
17	0.308
22	0.328
16	0.333
3	0.335
1	0.343
12	0.349
15	0.349
18	0.349
23	0.350
4	0.351
11	0.352
6	0.353
7	0.353
13	0.354
19	0.354

Figure 37: The top 15 variables taken from the two rankings of variable importance in terms of the area under the ROC curve and the signal-background separation, for the alternative set of 25 variables.

As with the initial set of 26 variables, these two rankings largely contain the same variables, again implying that this is indeed a useful method for determining the effectiveness of each variable in relation to the others. This ranking was again used to retrain the BDT method with a smaller set of variables, with aim of reducing the number to as few as possible whilst retaining the area under the ROC curve and the signal-background separation to within 1% and 5%, respectively, of their values when the original set of 26 variables was used.

To begin with, the first 10 variables from each ranking in figure ... (a total of 11 variables) were included in the framework, resulting in a 2.3 % reduction (0.821) from the original ROC curve area and a 10.6 % reduction (0.311) from the original signal-background separation.

Clearly, these variables alone were inadequate, so it was decided that the most useful variables from both sets would be combined in the final set. Variable 2 (jtrack) from the original set of 26 was added to the framework, as it was the second most highly ranked in both tables in figure ... Including this variable resulted in a reduction of 1.2 % (0.830) in the ROC curve area and a 5.5 % reduction (0.329) in the signal-background separation.

In an attempt to keep these quantities within the desired range, both variable 9 (jsiptrack[0]), from the set of 26 variables, and variable 11 (jsiptrack[2]), from the set of 25 variables, were separately added to the framework, as they were the next most highly ranked from each set.

Including variable 9 resulted in a reduction of 1.1 % (0.831) in the ROC curve area and a 5.2 % reduction (0.330) in the signal-background separation. Including variable 11 resulted in a reduction of 1.1 % (0.831) in the ROC curve area and a 4.9 % reduction (0.331) in the signal-background separation.

Whilst adding variable 11 to the framework had been successful in keeping the signal-background separation to within the desired range, the loss in the area under the ROC curve was still greater than 1 %. However, including both variables 9 and 11 reduced the area under the ROC curve by only 0.7 % (0.834) and resulted in a loss in signal-background separation of only 3.4 % (0.336). As both these values represented only minimal losses in the classifier performance, a final set of 14 variables comprising of jmet, jtrack, jsiptrack[0], jsiptrack[2], (jphi[1] – jmetphi), (jphi[2] – jmetphi), (0.5*(jeta[1] + jeta[2]) - jeta[0]), pjetNp[0], (jphi[0] – jmetphi), (jmass[0] + jmass[1] + jmass[2]), jdij, jtri, (jdij - jdij12) and njet was decided upon.
The signal and background distributions superimposed for each variable in the final combination of 14 are shown in figure 38. Again, many of these variables display prominent sections where the signal and background distributions do not overlap, explaining their usefulness in discriminating between signal and background events. Additionally, the variables found to be most effective in this classification problem are associated with the particle tracking, which is to be expected as identification of the events in question is related to inspecting the tracks and displaced vertices arising from the decays of the particles.

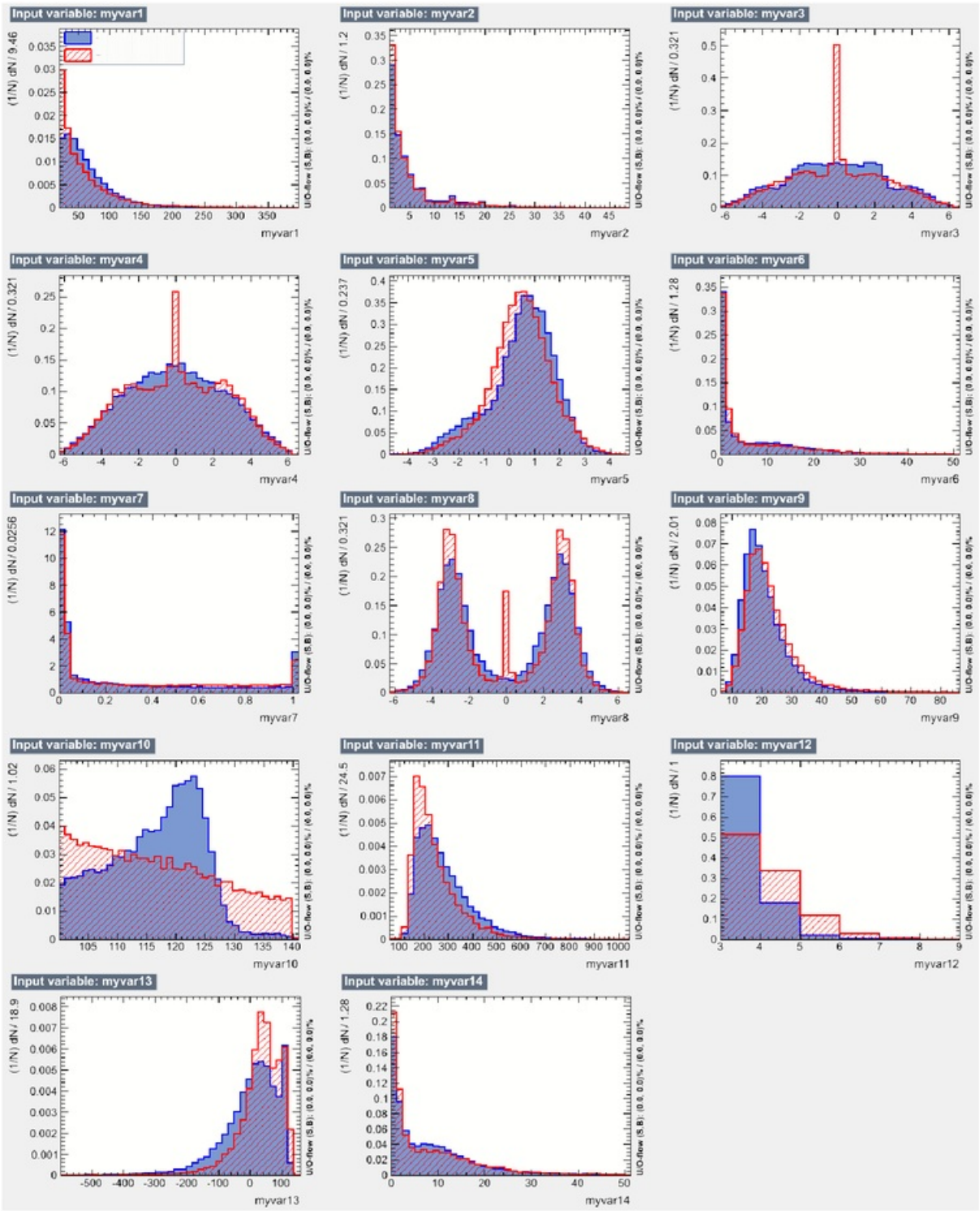


Figure 38: Plots showing superposition of signal and background distributions for each of the final 14 variables, where the units are omitted for technical reasons.

By using this variable ranking method, it was possible to identify the greatest contributors to the classification framework and eliminate the remaining variables, while minimising the loss in classifier performance. Although the performance was marginally reduced, the robustness of the analysis framework had been improved by using a combination of only 14 variables instead of 26. The application performance of the default BDT method using this combination of variables was also assessed, and the results of this are shown in discussed in appendix A.
The next aim of this project was to then optimise the chosen analysis method with the intention of restoring the classifier performance to achieve the same results with less variables, or possibly improve the performance even further.

6.3 BDT Optimisation

The TMVA offers a variety of booking options, serving as adjustable parameters which can be altered to optimise the BDT performance for a given dataset. To find the best combination of configuration options for identifying Higgs-to-charm signal events, the adjustable parameters were incremented a certain range above and below their default values. For each different number tested, the area under the ROC curve and the signal-background separation were recorded. The intention was to maximise these two measures of performance whilst preserving the speed of the BDT analysis.

The first parameter which was varied was the number of trees, N. As the default value for this was N = 500, the range initially chosen to be tested was from 100 to 1000, increasing the number of trees in increments of 100. However, as the area under the ROC curve and the signal-background separation still exhibited variation with each increment in N as N was increased to 1000, it was clear that this range of values was insufficient in deciding in upon the optimum number of trees for this classification problem.

To ensure that the effect of increasing N was measured across a suitable range, the number of trees was increased further to 1600 until the area under the ROC curve and the signal-background separation showed no significant increase. The results taken were used to plot separate graphs showing the variation in the area under the ROC curve with increasing tree number and variation in the signal-background separation with increasing tree number. These graphs (shown in figures 39 and 40) would be used to decide upon the value of N which was most appropriate for this classification problem.

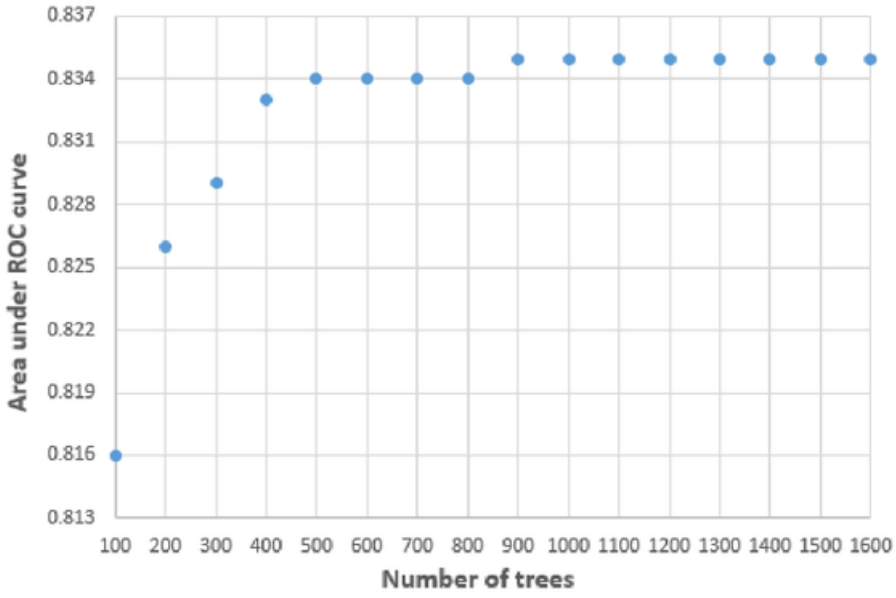


Figure 39: Graph showing variation in the area under the ROC curve as the number of the trees in the BDT forest was increased.

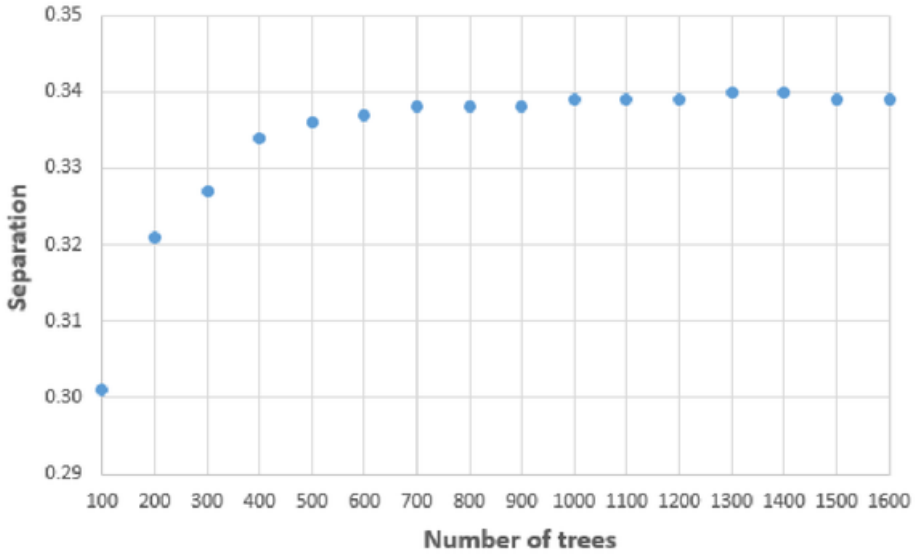


Figure 40: Graph showing variation in the separation between the signal and background distributions as the number of the trees in the BDT forest was increased.

Whilst increasing the number of trees improved the classifier performance up to a certain point, it also had the negative consequence of increasing the time taken in training the BDT forest and in classifying unlabelled data in the application phase. This meant that both the performance and speed needed to be taken into consideration when deciding upon the number of trees to be used.

From inspection of both figures 39 and 40, it can be seen that increasing N from 100 to 500 led to a substantial improvement in the classifier performance in terms of both the area under the ROC curve and the signal-background separation, and that the signal-background separation continues to increase until the point where $N = 700$. Subsequently, the value of both quantities approximately plateaued until $N = 900$, after which point they saw a slight increase before levelling off again. As the increase in these performance measures was marginal past $N = 900$, but there was a considerable decrease in the BDT speed due to the increase in N, it was decided that $N = 700$ was the most suitable number of trees for this classification problem.

The next parameter tested for optimisation potential was the maximum tree depth, D, which had a default value of 3. This was reduced to $D = 1$ and again increased until there was no further effect on the BDT performance, up to $D = 25$. Because the maximum depth is the maximum number of conditions an event can be tested against before classification, this parameter had to be incremented in integers, and was increased by 1 each time. The area under the ROC curve and the signal-background separation were recorded for each value of D, as done previously, and used to plot the graphs in figures 41 and 42:

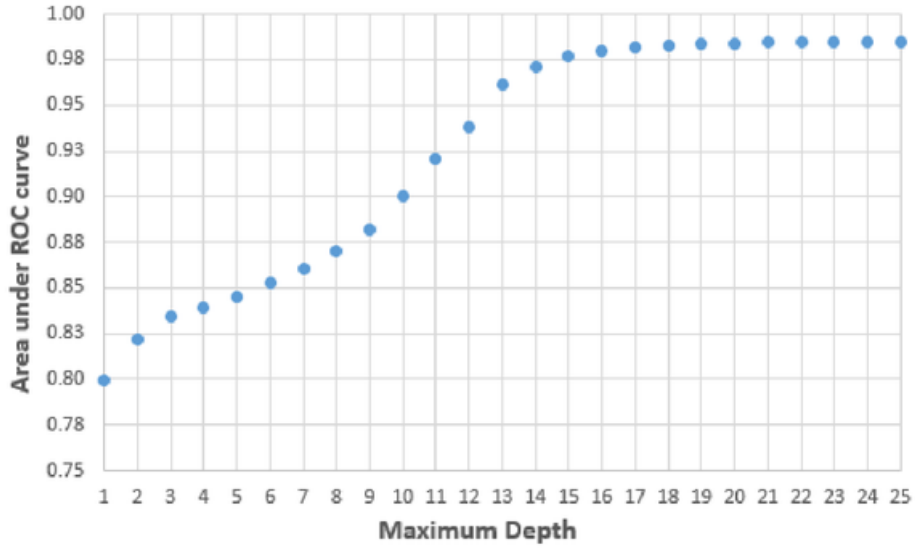


Figure 41: Graph showing variation in the area under the ROC curve as the maximum depth was increased.

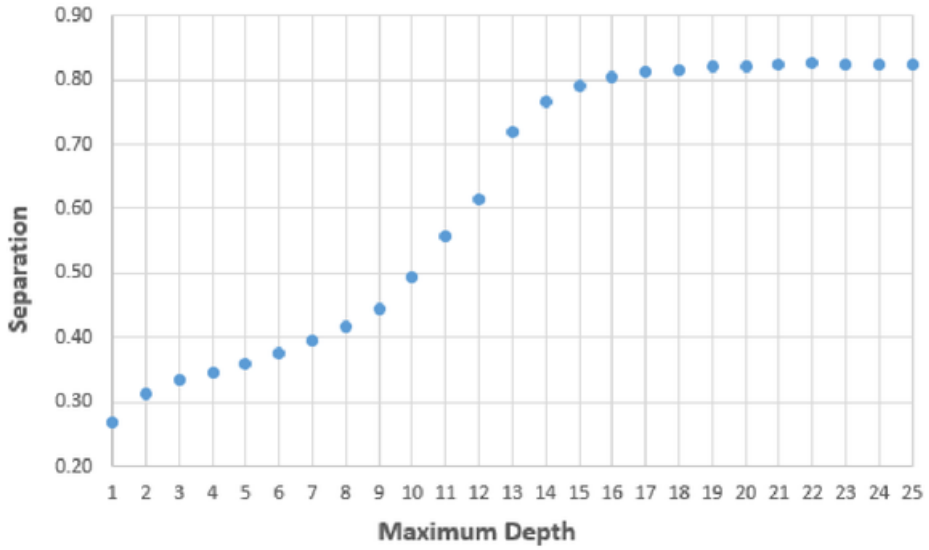


Figure 42: Graph showing variation in the separation between the signal and background distributions as the maximum depth was increased.

Increasing the maximum depth decreased the speed of the of the BDT forest in the same way as increasing the number of trees, and therefore deciding upon the optimum depth was subject to the same caveat whereby the most suitable value should improve the classifier performance without significantly slowing the algorithm. This value was again decided upon by inspection of the graphs showing the variation in the performance measures to determine the point where increasing the value of D no longer had a significant enough effect to compensate for the decreased BDT speed.

Both curves have similar shapes, which is another reassuring indication that both performance measures, although describing different quantities, correlate with each other and can be expected to indicate the overall classifier performance. The sharpest increase in both performance measures was between $D = 10$ and $D = 13$, where the gradient of the curve was steepest. Increasing the maximum depth past $D = 13$ resulted in slight improvements in the classifier performance, however the training stage took significantly longer than when the default BDT configuration was used. For this reason, $D = 13$ was decided to be the optimum value for the maximum depth, as the BDT performance was greatly increased with only a slight decrease in speed.

Another configuration option, which is denoted by 'nEventsMin' allows the user to determine the minimum number of events a leaf node can contain. This is comparable to requiring that a leaf node must be a certain size, and limits the total number of leaf nodes. This parameter isn't included in the default BDT configuration and so there is no restriction on the minimum node size, however, including this constraint can prevent highly complex trees being formed, reducing the risk of overtraining.

The convention when applying this restriction to classification trees is to require that a minimum of 5% of the training events end up at each leaf node. Since 168,793 events were used to train the BDT forest, the conventional minimum number of events required by a leaf node would be approximately 8440. The minimum number of events was varied between ~2.5-7.5% in intervals of 500 events, with a central value of 8500, to explore whether changing the minimum node size would influence the BDT performance. This was important to investigate because if the BDT performance did not suffer with increasing node size, it was likely that a larger node size would be preferable as a preventative measure to stop the classifier becoming overtrained. The BDT performance was measured using the area under the ROC curve and signal-background separation, as done previously, and is shown in the graphs in figures 43 and 44.

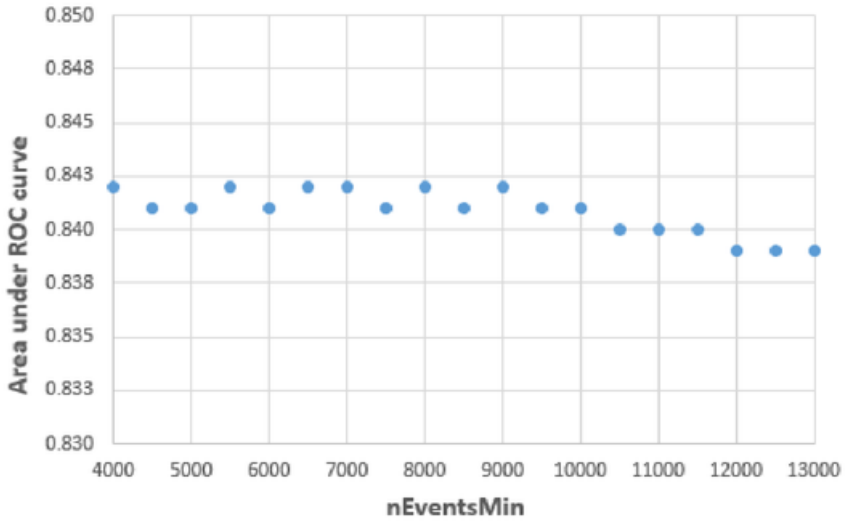


Figure 43: Graph showing variation in the area under the ROC curve as the minimum leaf node size was increased.

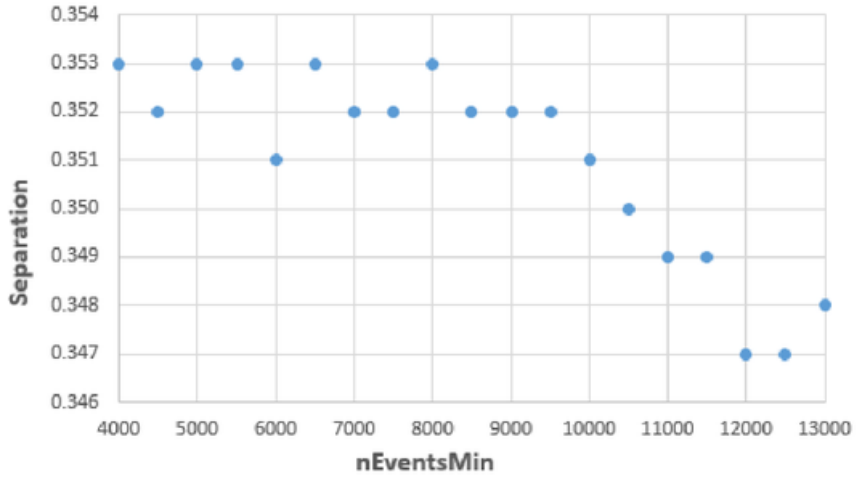


Figure 44: Graph showing variation in the separation between the signal and background distributions as the minimum leaf node size was increased.

Evidently, increasing the minimum node size doesn't have a significant impact on the area under the ROC curve, other than a slight decrease when the minimum number of events becomes greater than around 6%. However, increasing nEventsMin to a value above this threshold has a considerably more pronounced effect on the signal-background separation. The separation decreased from a maximum of 0.353 at nEventsMin = 8000 to 0.347 at nEventsMin = 12000 – a reduction of approximately 1.7 %. As discussed in the variable reduction section, this is an unfavourable decrease in terms of signal-background separation, and so it was concluded that increasing the minimum node size would not be a suitable preventative measure with regards to overtraining, as the loss in the BDT performance would be too appreciable. This booking option was thus omitted from the chosen BDT configuration.

Adaptive boosting is the boosting algorithm used to create the ensemble of decision trees by fitting each tree on differently weighted training data. The parameter 'AdaBoostBeta' defines the learning rate of the adaptive boosting algorithm. Decreasing this variable has the effect of increasing the number of iterations performed before the boosting algorithm terminates. To see how this might affect the BDT performance, this parameter was varied across the inclusive range 0.1-2.5 in increments of 0.1. It was expected that increasing the AdaBoostBeta value past the default value of 0.2 would decrease the BDT performance, however, there was a possibility of improving the BDT performance at lower AdaBoostBeta values, as the boosting algorithm would be allowed to perform more iterations. The recorded values of the area under the ROC curve and the signal-background separation for each AdaBoostBeta value in this range were used to plot the graphs shown in figures 45 and 46.

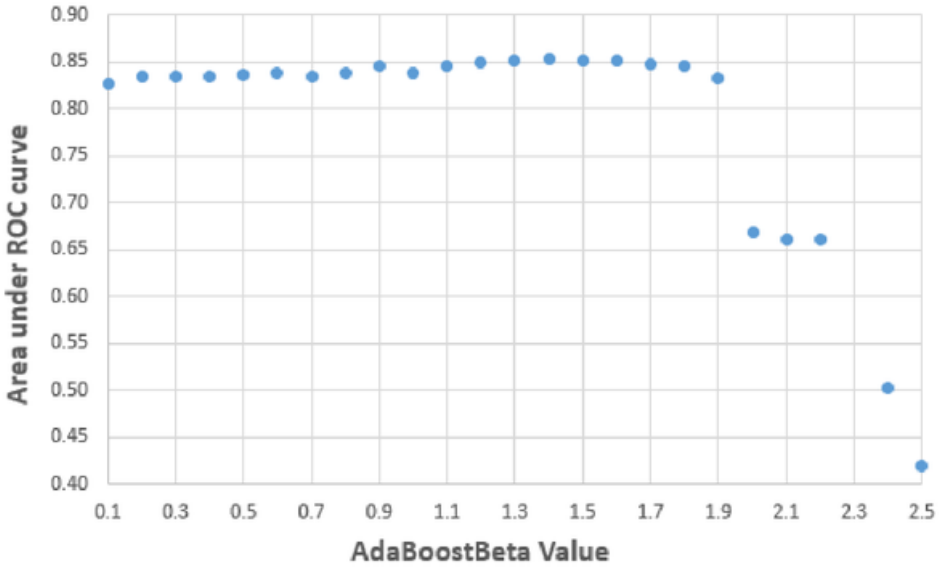


Figure 45: Graph showing variation in the area under the ROC curve as the AdaBoostBeta value was increased.

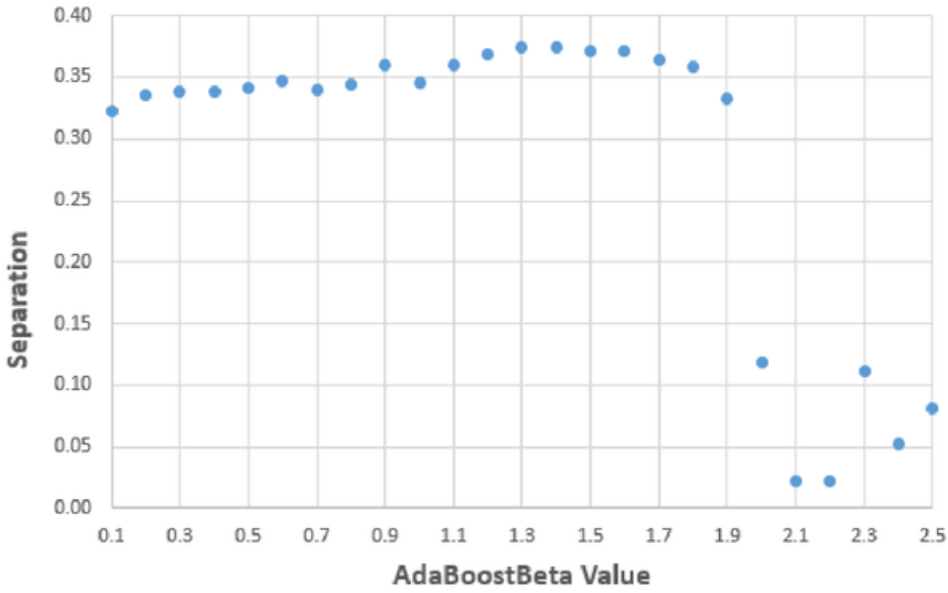


Figure 46: Graph showing variation in the separation between the signal and background distributions as the AdaBoostBeta value was increased.

Both the greatest area under the ROC curve and signal-background separation were achieved at $AdaBoostBeta = 1.4$, indicating this to be the optimum value in terms of the BDT performance. This was surprising, as it meant that the boosting algorithm was performing less iterations than in the default BDT configuration. However, as the data suggested that this

allowed for the strongest classification power of the BDT forest, the value of $AdaBoostBeta = 1.4$ was used in what was expected to be the optimum BDT configuration. However, using this configuration led to an interesting observation, which is discussed in the next section.

6.4 Experimental Overtraining

From the data collected whilst investigating how the BDT algorithm could be tailored to the problem of classifying Higgs-to-charm coupling events, it seemed that the optimum parameter values for the BDT forest when presented with this classification problem would be $N = 700$, $D = 13$, $AdaBoostBeta = 1.4$, where N is the number of trees in the forest, D is the maximum depth of each tree, and $AdaBoostBeta$ is the learning rate of the adaptive boosting algorithm.

When the BDT method was trained with these parameter values, it was noted that although the performance appeared to be greatly improved, there was a striking disparity between the performance when supplying the method with the test dataset as opposed to the training dataset. The evaluation results are shown in figure 47 along with a comparison of the test and training signal efficiencies at the benchmark background efficiency values of $B = 0.01$, $B = 0.1$, and $B = 0.3$.

Signal Efficiency (test)			Signal Efficiency (test)			Area under ROC Curve	Separation
B = 0.01	B = 0.1	B = 0.3	B = 0.01	B = 0.1	B = 0.3		
0.816	0.947	0.977	0.968	0.977	0.984	0.975	0.791

Figure 47: Evaluation results when training the BDT method with 14 variables on the model dataset, with a BDT configuration of $N = 700$, $D = 13$, $AdaBoostBeta = 1.4$.

At all three background efficiencies, there is a significant difference between the test and training signal efficiencies, which is a clear indicator that the BDT algorithm had been overtrained. This was clearly a direct consequence of the optimisation attempt, as prior to this, when the BDT method had been trained in default configuration, there had only been slight deviations between the test and training signal efficiencies.

As overtraining of machine learning algorithms causes the classifier performance to appear better than it truly is, it prevents reasonable error margins from being derived, as it becomes impossible to tell how well the classifier would perform on an unlabelled dataset. To prevent this, minimising BDT overtraining became an additional project aim. Preferably, overtraining of the BDT method would be entirely avoided, whilst still attaining an improvement in classification power.

6.5 Minimising Overtraining

With the aim of reducing the overtraining of the BDT forest, further analysis was performed on the data which had been collected whilst separately varying each parameter. It was pragmatic to first devise a method to quantify the extent of the overtraining before

proceeding with this analysis. As overtraining is detected when there is a disparity between the performance on training and test data, a comparative measure of the BDT performance on training and test data was used.

The results generated during the TMVA evaluation phase were used to calculate a ratio of test performance to training performance by dividing the test signal efficiency by the training signal efficiency for each of the three benchmark background efficiency values provided. As the BDT forest performs better on the training dataset relative to the test dataset when it has been overtrained, it was evident that the smaller the ratio, the more the BDT forest had been overtrained. The closer the ratio to 1, the better.

Analysing the results in this way provided a method of determining which parameter had the most significant effect on the extent of the overtraining when varied. Perhaps more importantly, it allowed for an estimation of the range of values each parameter could take for which the overtraining was minimised.

To begin, the ratios of signal efficiency to test efficiency for each background efficiency were calculated from the data collected when the only variable was the number of trees in the BDT forest. This was done for all the data collected in the range N = 100 to N = 1600 and used to plot the graph shown in figure 48:

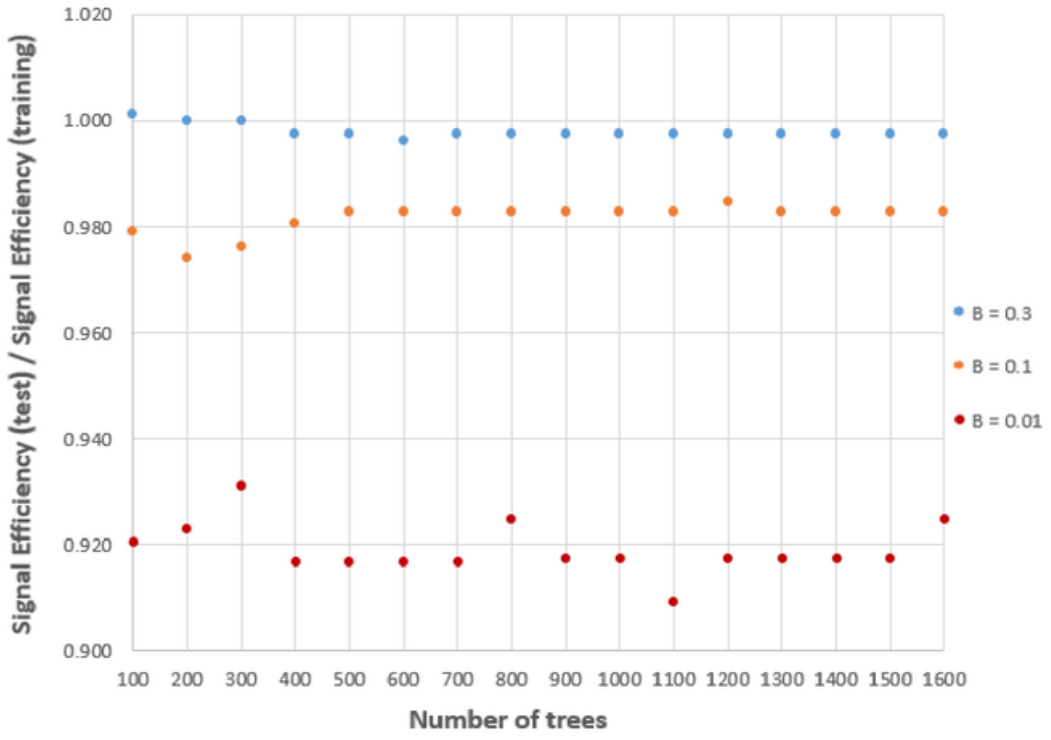


Figure 48: Graph showing BDT performance when trained on the test sample divided by BDT performance when trained on the training sample for 3 benchmark values of background efficiency, B = 0.01, B = 0.1 and B = 0.3. The variation in this ratio with increasing number of trees is demonstrated.

Given that the smaller the ratio of test/training signal efficiency, the more prominent the BDT overtraining, it would appear that the BDT method suffered the greatest overtraining at N = 1100, as there is a sharp dip in the test/training signal efficiency ratio for this number of trees at B = 0.01. However, due to having only 3 benchmark values as a comparison of test and training signal efficiencies, it is difficult to assess the extent of the overtraining in an overall sense. Whilst the ratios for B = 0.1 and B = 0.3 don't display a minimum at N = 1100, they display less fluctuation throughout the entire range than for B = 0.01, therefore this ratio largely determines the extent to which overtraining is observed, in this case. There is a noticeable decline in the ratio for B = 0.1 around the minimum N = 200, however the ratios for B = 0.01 and B = 0.3 are both above their average values at this point, which is assumed to reduce the overall amount by which the BDT was overtrained.

Whilst investigating the optimum BDT configuration, it was decided that a value of N = 700 would allow for the best classification performance, disregarding all other BDT parameters. This data would suggest that actually N = 800 would be a more suitable value, as this number of trees lessens the overtraining in comparison to N = 700, with the same classifier performance. Increasing the number of trees from N = 700 to N = 800 would decrease the speed of the BDT algorithm, however the difference in speed between these two values is marginal, and so N = 800 was chosen to be the final value for the number of trees in the BDT forest.

Next, the test/training signal efficiency ratios were calculated for all values of maximum depth in the measured range of 3-25. This data was used to plot the graph shown in figure 49.

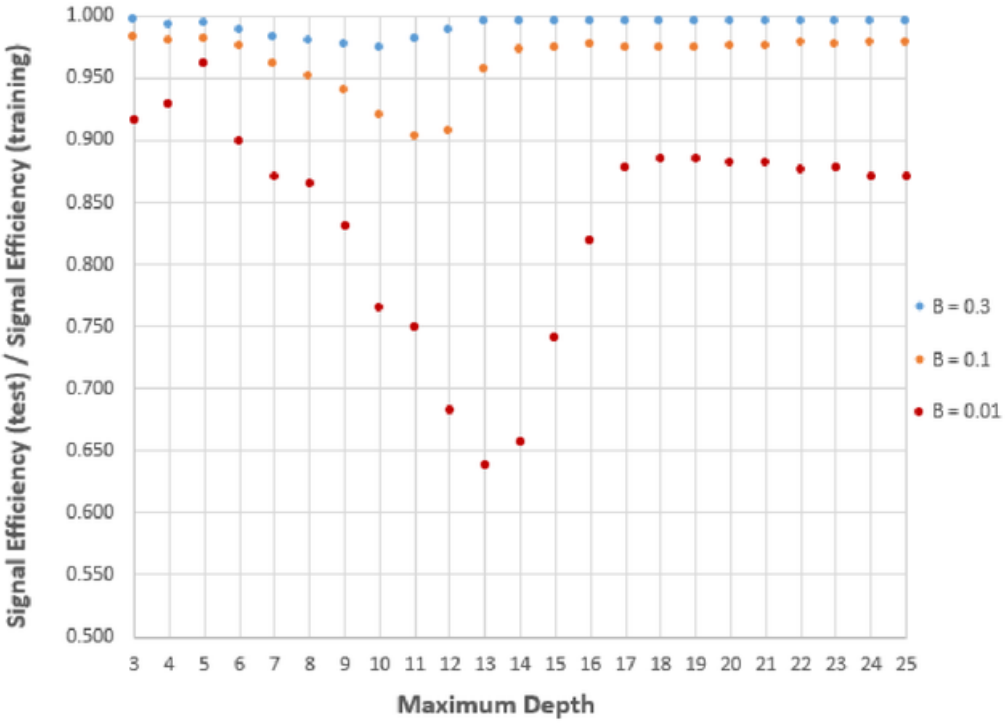


Figure 49: Graph showing BDT performance when trained on the test sample divided by BDT performance when trained on the training sample for 3 benchmark values of background efficiency, $B = 0.01$, $B = 0.1$ and $B = 0.3$. The variation in this ratio with increasing maximum tree depth is demonstrated.

The variation in the test/training signal efficiency ratios was considerably more apparent with increasing values of D than for N, with a clear trend in the way each ratio changed as D increased. This graph shows distinct minima in the test/training signal efficiency ratios, corresponding to maximum overtraining of the BDT forest. However, these minima occur at different maximum depth values and appear to become more pronounced at lower background efficiencies.

There is a marginal decrease in the test/training signal efficiency ratios for $B = 0.3$ and $B = 0.1$ when the maximum depth is increased from $D = 3$ to $D = 5$. Conversely, the test/training signal efficiency ratio for $B = 0.01$ increases more dramatically when the maximum depth is increased from $D = 3$ to $D = 5$. This suggests that $D = 5$ may be the optimum value for the maximum depth, as the BDT performance would be improved with a maximum depth of $D = 5$ as opposed to the default value of $D = 3$, without a significant increase in BDT overtraining.

Past a maximum depth of $D = 5$, the test/training signal efficiency ratios for all 3 background efficiency values decrease until they reach their respective minima, before increasing again and levelling out. Whilst the test/training signal efficiency ratio for $B = 0.3$ increases to approximately equal the ratio at a maximum depth of $D = 3$, the ratio for $B = 0.1$ is still slightly lower than its original level, and the ratio for $B = 0.01$ is significantly lower than in the default configuration. Although the ratio values increase and stabilise at $B = 0.01$,

$B = 0.1$ and $B = 0.3$, the dramatic difference between the $B = 0.01$ test/training ratios at a maximum depth of D = 3 compared to maximum depths past D = 17 mean that any depths greater than D = 5 would be unsuitable. For this reason, it was decided that $D = 5$ was the optimum maximum tree depth.

The only other parameter varied during the optimisation process was the AdaBoostBeta value, and so the process of calculating test/training signal efficiencies was repeated one final time for the data collected whilst incrementing this parameter. As demonstrated by the graphs in figures 45 and 46, the BDT performance decreased massively once the AdaBoostBeta value was increased past 1.9, therefore only data in the range of 0.1-1.9 was considered when assessing the extent of the BDT overtraining, as values outside this range would not be considered in the final BDT configuration. The ratios calculated for this range of values are shown on the graph in figure 50:

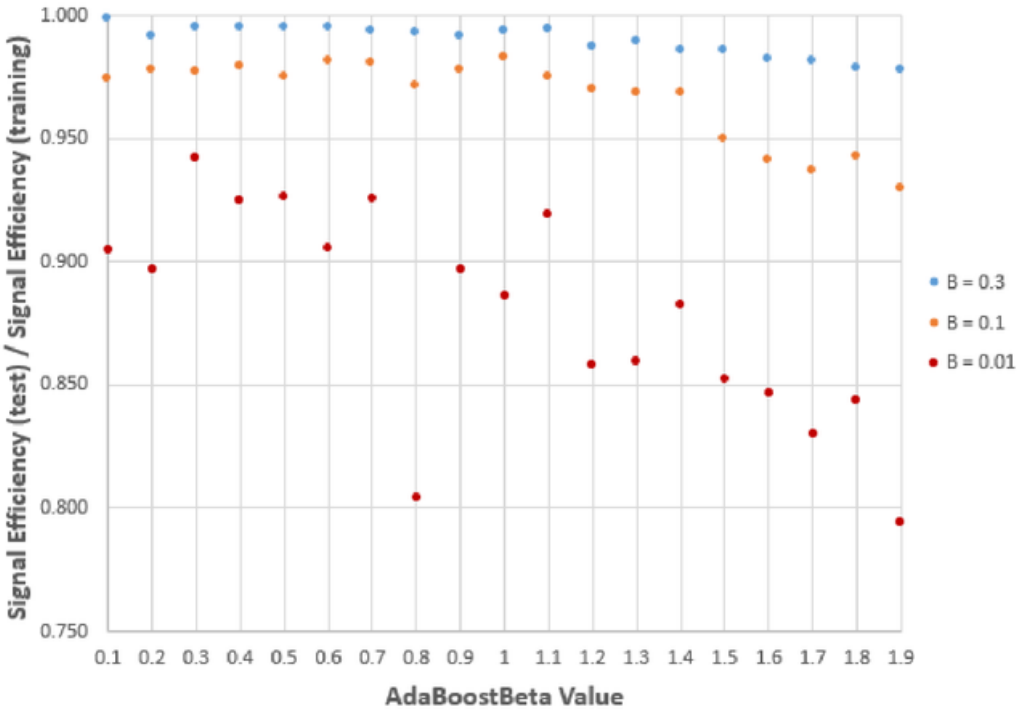


Figure 50: Graph showing BDT performance when trained on the test sample divided by BDT performance when trained on the training sample for 3 benchmark values of background efficiency, $B = 0.01$, $B = 0.1$ and $B = 0.3$. The variation in this ratio with increasing learning rate of the adaptive boosting algorithm is shown.

For all 3 background efficiencies, the BDT overtraining showed a tendency to increase as the AdaBoostBeta value increased. The BDT algorithm appeared to show the least overtraining at $AdaBoostBeta = 0.3$, with the signal efficiency ratios tending to decrease past this point. Therefore, an AdaBoostBeta value of 0.3 was chosen for the final BDT configuration.

This gave a final BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3, which was expected to perform optimally for the simulated detector dataset. The correction to the maximum depth would appear to be the most important change here, as increasing this parameter produced the greatest variation in the test/training signal efficiency ratios, and the pattern in this variation was most perceptible compared to the trends seen when incrementing N and $AdaBoostBeta$. This shows that increasing D had a distinct, measurable effect on the BDT overtraining, and using this data to select the value D = 5 was imperative in avoiding overtraining the BDT method.

During this analysis, it was noted that the BDT method tended to show greater overtraining at the lower background efficiencies of B = 0.01 and B = 0.1 than at B = 0.3. This suggests that overtraining may become more apparent when cutting on samples with lower signal purities. Therefore, increasing the separation between the signal and background distributions and maximising the signal purity of the cut sample may be an effective way to prevent overtraining. Unfortunately, there was not enough time to explore this option during this project, so it was ensured that the BDT configuration options were chosen as such that they did not allow for significant overtraining of the algorithm. Hence, the configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3 was used in attaining the final results, which are discussed in the next section.

7 Results and Discussion

Training the BDT algorithm in the final configuration on the model dataset created a powerful classifier for identifying Higgs-to-charm coupling events. The evaluation results demonstrate this, with an ROC curve area of 0.851 and signal-background separation of 0.373, both of which are improvements upon the original values. The plots for the ROC curve and classifier output distribution can be seen in figures 51 and 52. The classifier was then passed the unlabelled dataset for the application stage, the results of which are shown in figures 53 and 54. The test and training signal efficiencies of the classifier at the benchmark background efficiency values are compared in figure 55 as a final check for overtraining.

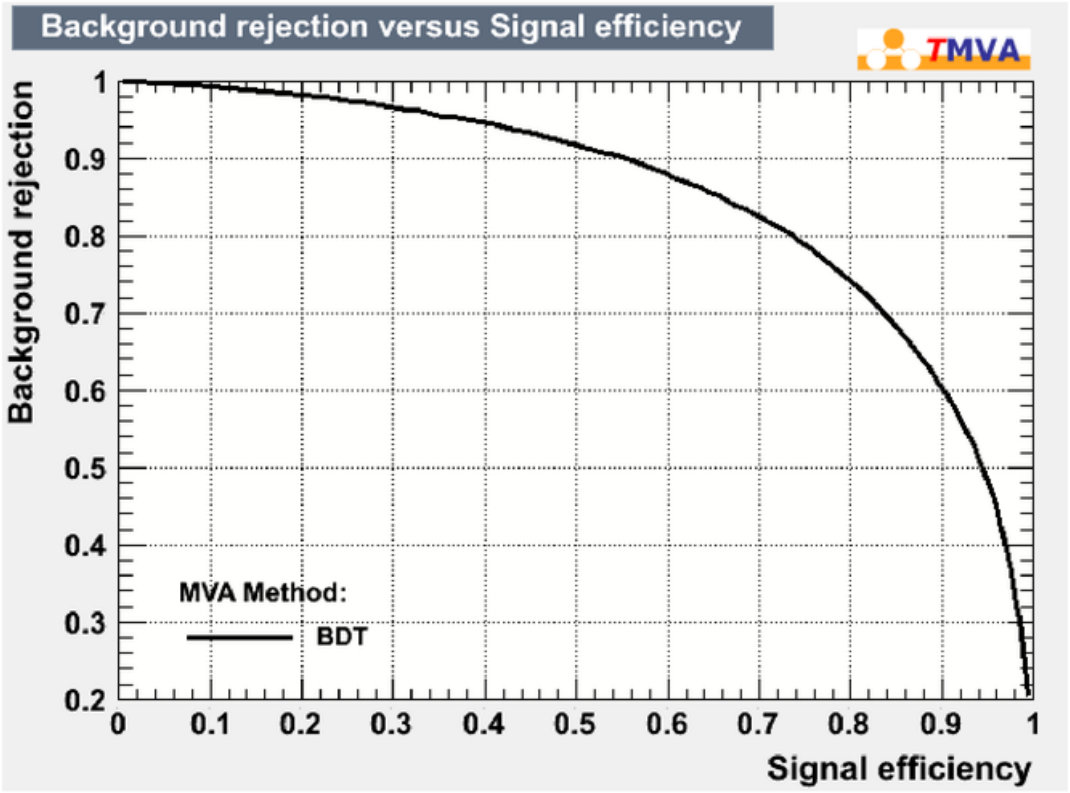


Figure 51: ROC curve for the final classifier, using 14 variables and a BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3. The area under the curve is 0.851.

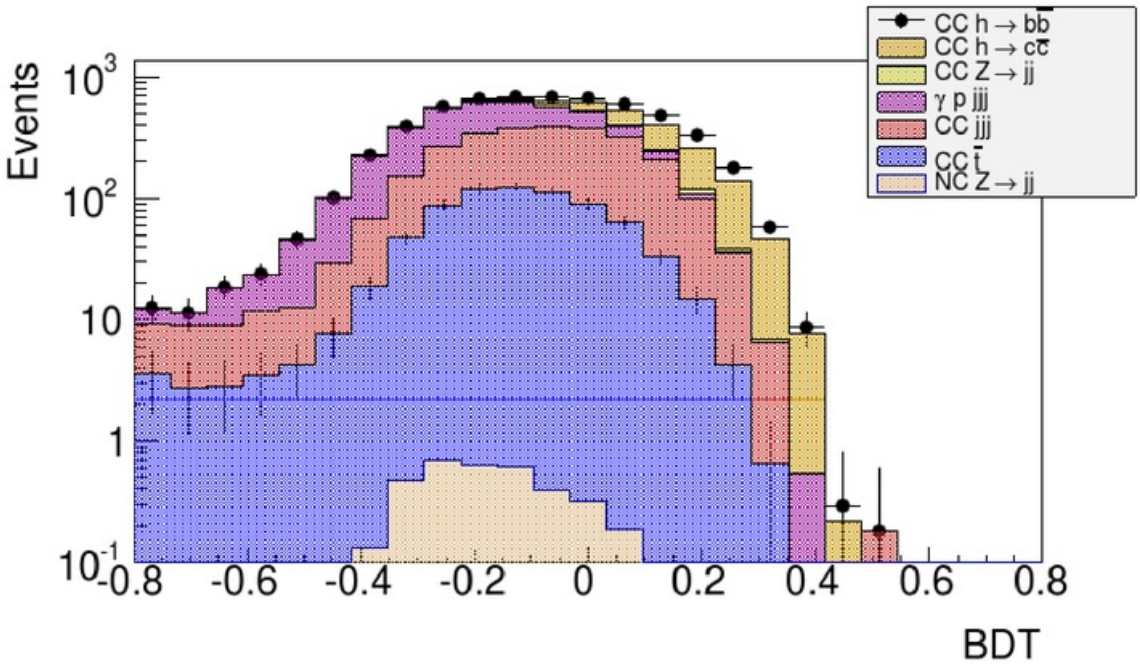


Figure 52: BDT output distribution for the final classifier, using 14 variables and a BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3.

BDT Cut	Number of Signal Events	Number of Background Events	Coupling Error (%)	Significance
-0.40	739	4794	5.0	10.4
-0.35	739	4605	4.9	10.6
-0.30	738	4331	4.8	10.9
-0.25	735	3948	4.7	11.4
-0.20	729	3480	4.5	12.0
-0.15	717	2993	4.2	12.6
-0.10	697	2488	4.0	13.4
-0.05	666	2024	3.9	14.7
0.00	602	1396	3.7	15.1
0.05	532	977	3.7	15.7
0.10	440	623	3.7	16.0
0.15	331	356	4.0	15.5
0.20	219	173	4.5	14.3
0.25	116	71	5.9	11.5
0.30	47	20	8.7	8.3
0.35	12	4	16.5	4.6
0.40	1	0	80.0	0.9

Figure 53: Table of results showing measured number of Higgs-to-charm events and background events for the final classifier, using 14 variables and a BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3.

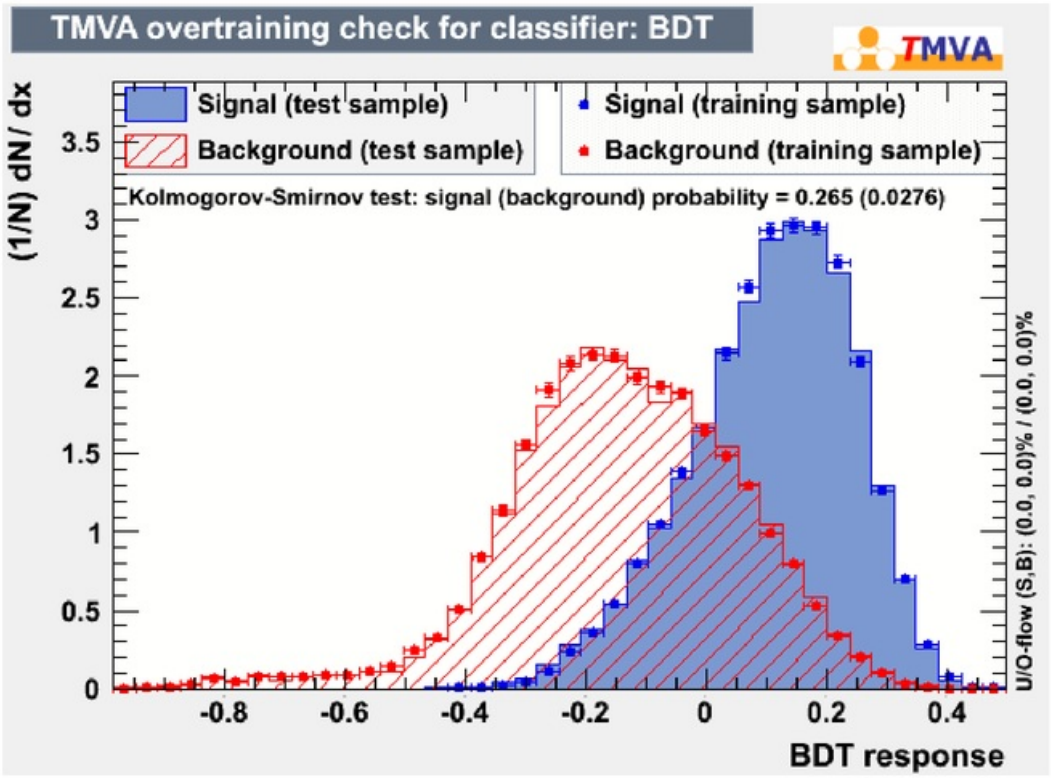


Figure 54: Superposition of classifier output distributions for test and training data for the final classifier, using 14 variables and a BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3. The output distribution for the test sample is shown in the shaded areas, and the output distribution for the training sample is shown by the points.

Signal Efficiency (test)			Signal Efficiency (training)		
B = 0.01	B = 0.1	B = 0.3	B = 0.01	B = 0.1	B = 0.3
0.126	0.552	0.837	0.139	0.563	0.844

Figure 55: Table showing test and training signal efficiencies at the 3 benchmark background efficiencies of B = 0.01, B = 0.1 and B = 0.3 for the final classifier, using 14 variables and a BDT configuration of N = 800, D = 5, and $AdaBoostBeta$ = 0.3.

The data in figure 55 indicates that the classifier has not been overtrained, as the small deviations between the test and training signal efficiencies can be accounted for by statistical fluctuations, therefore the issue of BDT overtraining in this project has been resolved. The similarity of the shapes of the test and training distributions in figure 54 is further evidence for this. Using the data collected when investigating both optimisation and overtraining allowed for the BDT method to be refined for the given classification problem and produce better results without overtraining bringing the validity of the results into question.

The BDT output distribution in figure 52 shows a greater separation between the signal and background events classified during the application stage, with more background events collected at lower output values, allowing for cuts with higher signal purity to be made at greater output values. This is further reinforced by the table of data shown in figure 53, where large signal-to-background ratios for cuts in the range of 0 – 0.15 allow for a greater signal significance. The greatest significance and lowest coupling error are seen at a BDT cut of 0.1, where the expected number of Higgs-to-charm coupling events is 440, with a coupling error of 3.7 % and a signal significance of 16.0. This is more precise than the original predicted value with a higher significance, demonstrating that the changes made to the analysis framework allow for better results to be obtained.

In addition, the framework had been made more robust by reducing the number of variables from 26 to 14, decreasing the risk of errors which are not accounted for in the case that the kinematic variables had not been accurately simulated. This meant that the primary objectives of this project, which were reducing the number of variables and improving the classifier performance, had both been achieved. The changes made had resulted in slower data processing by the analysis framework, however the speed of the analysis was very much a secondary consideration in this project. It was inevitable that enhancing the classifier performance would result in some loss in classification speed. It is likely that further improvements could be made to this classifier to optimise it for the task of identifying Higgs-to-charm decays and increase the speed of this process. However, within the scope of this project, the idea of optimisation was to attain better performance by altering the BDT booking options, and the final results show that this had been realised.

8 Conclusion

The multivariate analysis framework was improved for the purpose of identifying Higgs-to-charm coupling events by reducing the number of variables used by the framework and altering the BDT configuration options to produce better classification results. The number of variables used was reduced from 26 to 14, allowing for a more robust framework which was less susceptible to inaccuracy resulting from the model data being simulated. Optimising the chosen method of BDT analysis enhanced the classification power of the MVA framework and gave a final result of 440 for the expected number of Higgs-to-charm decays, with a coupling error of 3.7 % and a signal significance of 16.0, an improvement on the initial value in terms of both precision and signal significance. These results assume all backgrounds to 2 % and a luminosity of $1000 fb^{-1}$.

To improve this analysis framework further, more BDT configuration options could be explored as a means of refining the method for this specific task, and new combinations of variables could be considered. The number of variables may be reduced further by combining variables or replacing several variables with a single variable more useful in classification. In addition, it would be beneficial to explore ways to increase the BDT processing speed without reducing the classifier performance, as a decrease in speed appeared to be an inevitable consequence of improving the analysis results in this project.

9 Bibliography

[1] C. Delaunay, T. Golling, G. Perez and Y. Soreq, "Charming the Higgs," 2013.

[2] G. T. Garvey, D. A. Harris, H. A. Tanaka, R. Tayloe and G. P. Zeller, "Recent advances and open questions in neutrino-induced quasi-elastic scattering and single photon production," Elsevier, 2015.

[3] B. Povh, K. Rith, C. Scholz, F. Zetsche and W. Rodejohann, Particles and Nuclei: An Introduction to the Physical Concepts, Springer, 2015.

[4] U.-k. Yang, "Particle Physics Group," 2011. [Online]. Available: http://www.hep.manchester.ac.uk/u/ukyang/fpp2/dis/dis_lec1.pdf. [Accessed April 2017].

[5] W. Verkerke, "Measurement of charm production deep inelastic scattering," University of Amsterdam, 1998.

[6] K. Chakravarthula, "Study of jet tranverse momentum and jet rapidity dependence of dijet azimuthal decorrelations with the DØ detector," Louisiana Tech University, 2012.

[7] P. Hansson, "The Parton Model," KTH Royal Institute of Technology, 2004.

[8] A. collaboration, "The electric and magnetic form factors of the proton," 2014.

[9] M. Breidenbach, J. I. Friedman and H. W. Kendall, "Observed Behavior of Highly Inelastic Electron-Proton Scattering," SLAC National Accelerator Laboratory, 1969.

[10] M. Klein and H. Schopper, "Electrons at the LHC: a new beginning," *CERN Courier,* 2014.

[11] O. Brüning and M. Klein, "The Large Hadron Electron Collider," *Modern Physics Letters A,* vol. 28, no. 16, pp. 247- 253, 2013.

[12] L. S. Group, "A Large Hadron Electron Collider at CERN," CERN, Geneva, 2012.

[13] E. Cruz-Alaniz, "LHeC Accelerator Development," in *International Workshop on Deep-Inelastic Scattering and Related Subjects*, Warsaw, 2014.

[14] J. Alwall, M. Herquett, F. Maltoni, O. Mattelaer and T. Stelzer, "MadGraph 5 : Going Beyond," *Journal of High Energy Physics,* vol. 2, no. 6, 2011.

[15] J. Alwall, P. Artoisenet, S. de Visscher, C. Duhr, R. Frederix, M. Herquet and O. Mattelaer, "New Developments in MadGraph/MadEvent," SLAC National Accelerator Laboratory, 2008.

[16] F. Maltoni and T. Stelzer, "MadEvent: automatic event generation with MadGraph," Institute of Physics Publishing , 2003.

[17] T. Sjöstrand, S. Mrenna and P. Skands, "PYTHIA 6.4 Manual," Fermilab Technical Publications, 2006.

[18] Université catholique de Louvain, "Delphes," [Online]. Available: https://cp3.irmp.ucl.ac.be/projects/delphes. [Accessed May 2017].

[19] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne and H. Voss, "http:// tmva.sourceforge.net," 4th October 2013. [Online]. [Accessed 2007].

[20] P. Cunningham, M. Cord and S. J. Delany, "Supervised Learning," in *Machine Learning Techniques for Multimedia*, Springer Berlin Heidelberg, 2008, pp. 21-49.

[21] U.S. National Library of Medicine, "Open-i," [Online]. Available: https://openi.nlm.nih.gov/imgs/512/261/3861891/PMC3861891_CG-14-397_F10.png. [Accessed March 2017].

[22] T. D. Straszheim, "Introduction to overtraining," 2009. [Online]. Available: http://software.icecube.wisc.edu/documentation/projects/pybdt/man_overtraining.html. [Accessed March 2017].

[23] D. Hampson, "Precision Higgs coupling measurements at a high luminosity LHeC," 2016.

[24] G. Cowan, K. Cranmer, E. Gross and O. Vitells, "Asymptotic formulae for likelihood-based tests of new physics," 2013.

[25] J. R. Quinlan, "Simplifying Decision Trees," *International Journal of Human-Computer Studies,* vol. 55, pp. 497-510, 1999.

[26] Analytics Vidhya, "Analytics Vidhya," June 2015. [Online]. Available: https://www.analyticsvidhya.com/wp-content/uploads/2015/06/Picture7.jpg. [Accessed February 2017].

[27] B. P. Roea, H.-J. Yanga and J. Zhub, "Boosted Decision Trees, A Powerful Event Classifier," University of Michigan, 2005.

[28] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics,* vol. 33, no. 3, pp. 1065-1076, 1962.

[29] D. Dannheim, T. Carli, A. Voigt, K.-J. Grahn and P. Speckmayer, "PDE-Foam - a probability-density estimation method using self-adapting phase-space binning," CERN–PH–EP, 2008.

[30] D. Siganos and C. Stergiou, "Neural Networks," Imperial College of Science Technology and Medicine, London, 1996.

[31] D. D. Team, "Deep Learning for Java," Skymind, 2017. [Online]. Available: https://deeplearning4j.org/img/perceptron_node.png. [Accessed January 2017].

[32] R. Taleb, A. Meroufel and P. Wira, "Neural Network Control of Asymmetrical Multilevel Converters," *Leonardo Journal of Sciences,* no. 14, pp. 53-70, 2009.

[33] V. J. Martin, "School of Physics and Astronomy," [Online]. Available: http://www2.ph.ed.ac.uk/~vjm/Lectures/SHParticlePhysics2012_files/PPNotes3.pdf. [Accessed April 2017].

[34] M. Dam, "Physics of Higgs Boson(s) at the LHC," in *Aspen Center for Physics*, 2003.

10 Appendix A: Application Results Using 14 Variables With Default BDT Method

The default BDT method was applied to an unlabelled dataset after the number of variables had been reduced from 26 to 15, with newly defined variables included in the framework. The BDT output distribution and results obtained by cutting in different places along the x-axis are shown in figures 56 and 57, respectively:

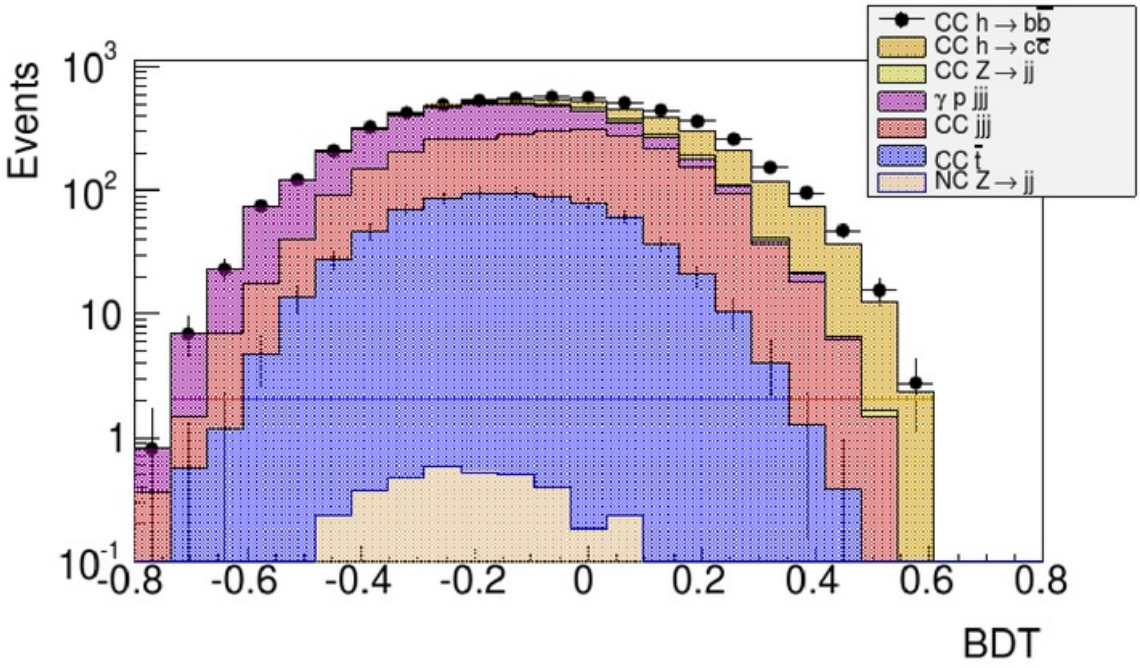


Figure 56: BDT output distribution for the default BDT method using 14 variables.

BDT Cut	Number of Signal Events	Number of Background Events	Coupling Error (%)	Significance
-0.40	738	4561	4.9	10.6
-0.35	736	4303	4.8	10.9
-0.30	732	4002	4.7	11.2
-0.25	725	3643	4.6	11.6
-0.20	715	3267	4.4	12.1
-0.15	702	2886	4.3	12.6
-0.10	683	2492	4.1	13.1
-0.05	656	2093	4.0	13.7
0.00	602	1573	3.9	14.3
0.05	548	1223	3.8	14.7
0.10	485	897	3.8	15.0
0.15	410	629	3.9	14.9
0.20	328	417	4.2	14.5
0.25	248	255	4.5	13.7
0.30	173	142	5.1	12.5
0.35	114	81	6.1	10.7
0.40	55	31	8.5	8.0

Figure 57: Table of results showing measured number of Higgs-to-charm events and background events for the default BDT method using 14 variables

Using a BDT cut of 0.1 would give the measurement with both the highest precision and significance. The expected number of Higgs-to-charm coupling events measured using this cut is 485, with a coupling error of 3.8 % and a signal significance of 15.0. This result was obtained prior to BDT optimisation, but was already an improvement on the initial value of 475 events with a coupling error of 3.9 % and a signal significance of 14.7. In addition, it can be demonstrated by plotting the coupling error against the BDT cut that this combination of variables stabilised the classifier performance by reducing the variation in the coupling error, and allowed for a lower error when cutting in the range -0.40 to 0.10. This is shown by the plot in figure 58.

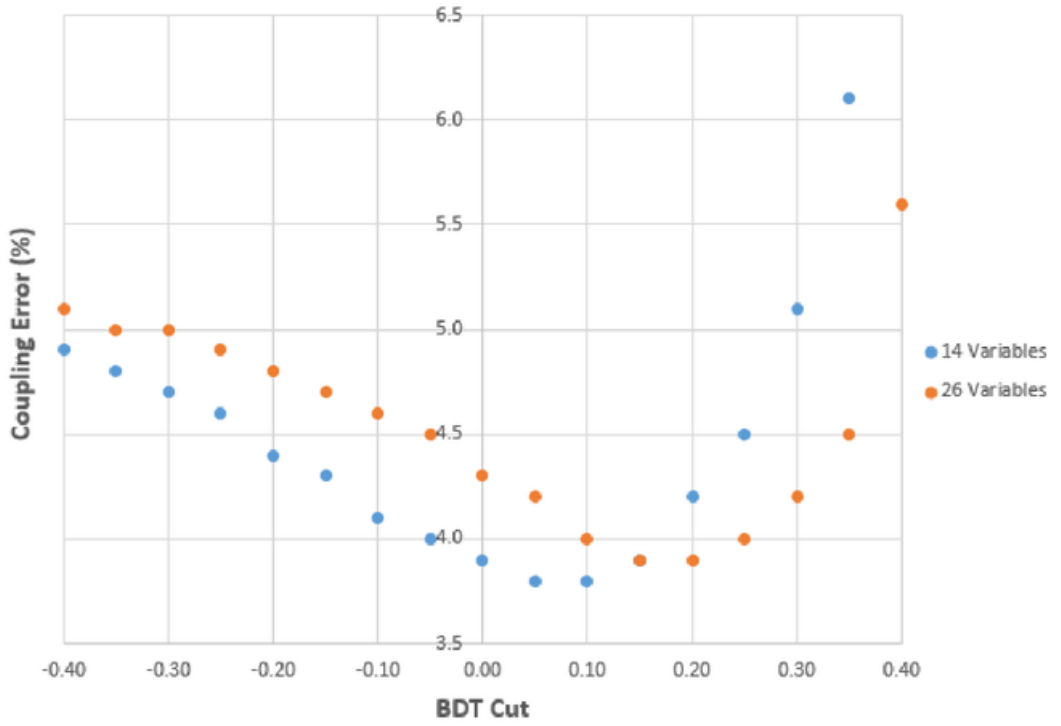


Figure 58: Graph showing variation in coupling error when cutting along the x-axis of the BDT output distribution for the original 26 variables and the final combination of 14 variables.

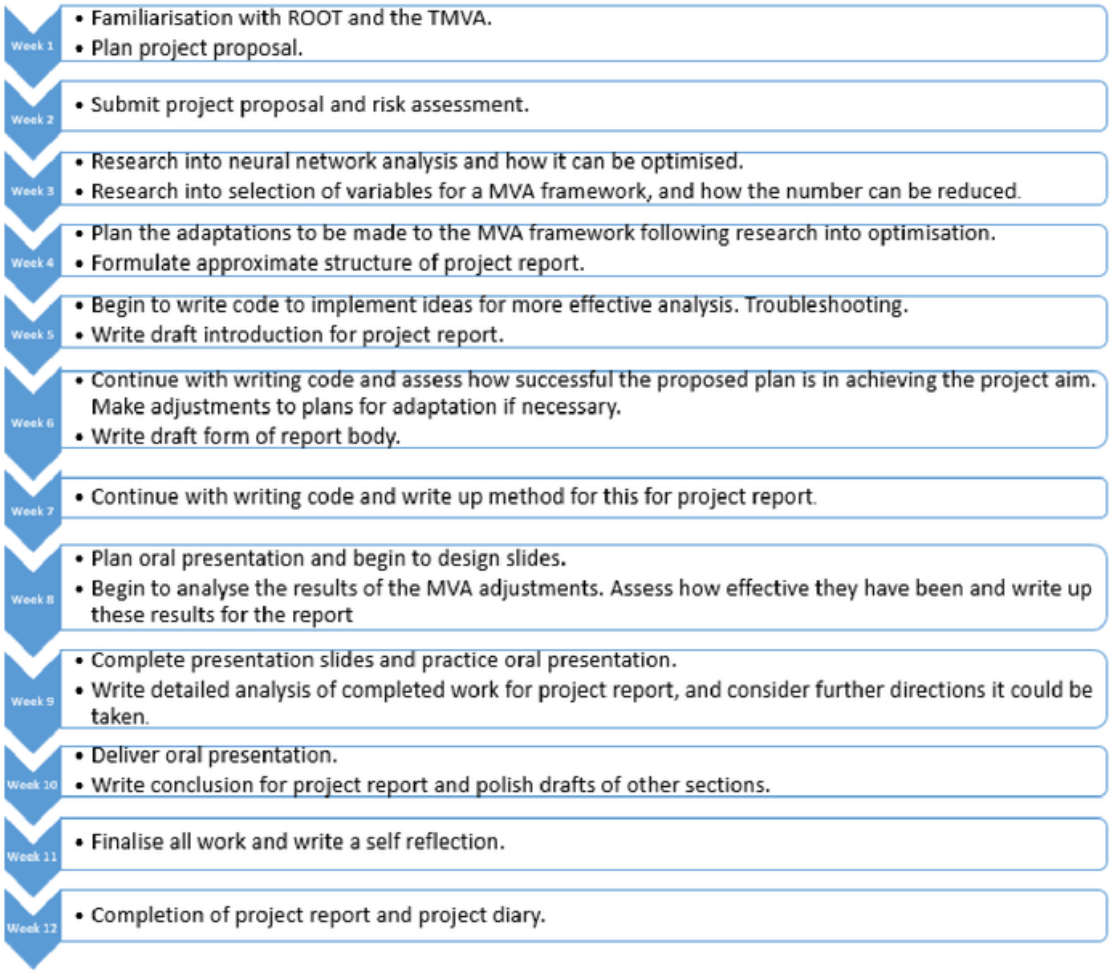
11 Appendix B: Project Proposal

The project proposal shown below was written in the second week of the project and included a working plan with timescales in which each aim was anticipated to be achieved.

PHYS379 Project Proposal: Finding Higgs to charm decays in electron-proton collisions

The main objective of this project is to learn how to use multi-variate analysis techniques in ways which optimise the analysis framework to produce better results for a physics experiment. Namely, the multi-variate analysis techniques will be used to identify Higgs to charm decays in data simulating electron-proton collisions. The broader aim will be to improve the general use of multi-variate analysis techniques in identifying a charm signal from background data, with a particular emphasis on how neural network analysis is used within the framework, and how this technique may be optimised for this purpose. The ideas which are most likely to be considered to optimise the use of the TMVA are finding ways in which neural network analysis could be performed more quickly, and reducing the number of variables used in analysis.

The proposed timeline for this project is shown below:



Overall, the proposed timescales and objectives were met and the overall aims of the project were accomplished. There were some slight deviations from this plan, most notably that neural networks were disregarded in favour of BDT analysis, as it became clear throughout the research into MVA methods that using BDT analysis was most pragmatic for the purposes of this project. The aims of reducing the number of variables and improving the use of the chosen multivariate analysis technique were both achieved within the given time constraints.

Finding Higgs to charm decays in electron-proton collisions

GRADEMARK REPORT

FINAL GRADE

/60

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68