A BDT optimization study and assessment of deep learning in selecting VBF events in the $H \to ZZ^* \to 4l$ channel

[†]Alan Morningstar, [‡]Pierre Savard, and [‡]Richard Teuscher
University of Toronto Physics Department
Conseil Européenne pour la Recherche Nucléaire (CERN)
[†]*IPP CERN Summer Student*
[‡]*Supervisor*

(Dated: August 2015)

Since the discovery of the Higgs boson in 2012, the most recently confirmed and final piece of the Standard Model has been rigorously studied in various production modes and decay channels. However, Run 1 of the LHC did not yield sufficient statistics to resolve production modes in the $H \to ZZ^* \to 4l$ channel. Current and future runs of the LHC will offer sufficient data to do so, thus machine learning techniques used to separate the two most frequent Higgs boson production modes - gluon fusion and vector boson fusion - were studied and the findings are presented in this report. An optimization of boosted decision trees trained on $\sqrt{s} = 8$ TeV ATLAS Monte Carlo data is presented. The feasibility of improving classification efficiency by using deep neural networks is also studied and detailed below.

I. INTRODUCTION

The Standard Model and the LHC

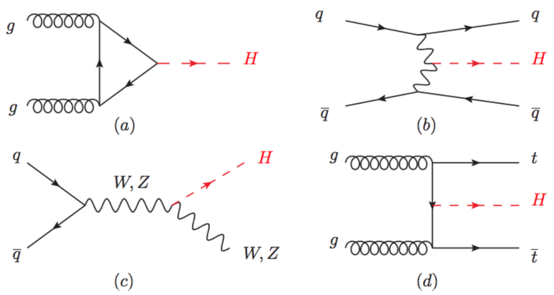
The Standard Model

The Standard Model (SM) is currently humankind's most complete and consistent theory of all known fundamental particles and their interaction via three of the four known forces of nature - electromagnetism, the weak nuclear force, and the strong nuclear force. It is a remarkably successful theory, having predicted the existence of particles such as the W, Z and Higgs boson before their discovery as well as being consistent with almost all experimental observations to date. However, there is a significant amount of experimental evidence demonstrating the incompleteness of the SM. Any truly complete description of nature should include gravity and a solution to the hierarchy problem - an explanation of why gravitation is so much weaker than the other three forces. Aside from that, the SM does not contain a dark matter candidate nor an explanation of dark energy leaving 84% of the observed matter in the universe and 95% of the energy content a mystery for now. Furthermore, the SM neutrinos are massless and thus do not undergo flavour oscillation which is inconsistent with observation. The question of how a matter antimatter asymmetry of one part per billion arose in the early universe is also unexplained by the current amount of CP violation in the SM.

In order to make progress on the problems mentioned above, it is imperative that the SM be rigorously tested in order to uncover any subtle inconsistencies with experimental observation at the LHC. Even small deviations from SM predictions could provide valuable hints as to what physics beyond the SM might be. Since the Higgs boson was discovered in 2012, at a mass of 125 GeV, physics in the Higgs sector has been the focus of many studies being conducted by ATLAS and other experiments at the LHC. At hadron colliders, namely the LHC, the Higgs boson can be produced in four main modes: gluon gluon fusion (ggF), vector boson fusion (VBF), associated production with gauge bosons (ZH & WH), and associated production with top quarks (tt̄H).

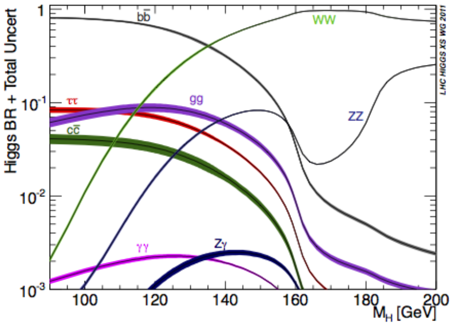
Table I: Higgs boson production modes at the LHC ($\sqrt{s} = 8$ TeV).

Diagram	Mode	σ(pb)
a)	ggF	$19.3^{+15\%}_{-15\%}$
b)	VBF	$1.58^{+3\%}_{-2\%}$
c)	ZH & WH	$1.11^{+5\%}_{+5\%}$
d)	tt̄H	$0.13^{+12\%}_{-18\%}$



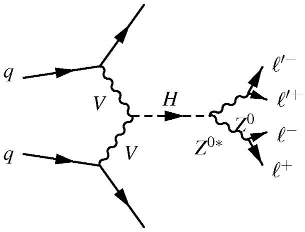
Once a Higgs boson is produced, it subsequently decays within approximately 10^{-22} s in its own rest frame. It can decay through various channels whose branching ratios are depicted as a function of Higgs mass in Figure 1.

Figure 1: Higgs boson branching ratios as a function of Higgs mass.



The decay channel of interest for this report is the Higgs decay into an on shell Z and an off shell Z which subsequently decay to lepton antilepton pairs, hence $H \to ZZ^* \to 4l$.

Figure 2: $VV \to H \to ZZ^* \to 4l$



This decay is referred to as the Golden Channel due to the high precision with which e and μ are measured and due to the fact that the final state is fully reconstructable with no missing transverse momentum. This means that even though the $H \to ZZ^* \to 4l$ decay has a small branching ratio (2.5%) when compared to other decay channels, its clean signature makes it valuable for analysis.

The LHC

The LHC accelerates protons to ultra relativistic energies in order to produce the required TeV scale collisions to probe the SM and beyond. It is a 27 km long pp collider designed to run at centre of mass energy $\sqrt{s} = 14$ TeV and luminosity $\mathcal{L} = 10^{34}$ cm^{-2}s^{-1}. There are 1232 high field 8.3 T liquid helium cooled superconducting dipole electromagnets which are required to direct the beams around the circular path as well as 392 quadrupoles to focus them. It has two ultrahigh vacuum beam pipes for opposite momentum proton beams which cross at four points along the ring. At these crossings, bunches of 10^{11} protons intersect every 25 ns which resulted in an average of approximately 20 collisions per bunch crossing during Run 1.

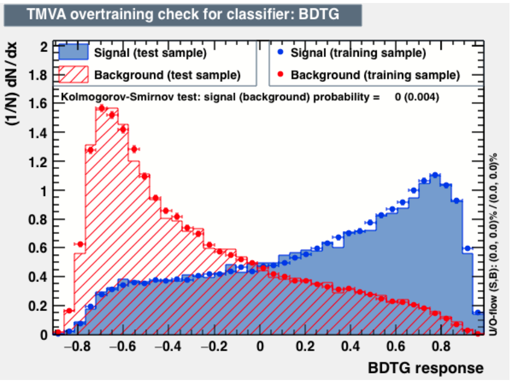
One of the intersection points is centred at the ATLAS detector, one of two general purpose detectors at the LHC and the largest ever built. The inner most part of ATLAS is a subsystem called the inner detector (ID). In order of increasing radius, the inner detector is comprised of the newly added insertable b-layer (IBL), pixel detector, semi-conductor tracker (SCT), and straw tube detectors (TRT). The IBL provides supporting high resolution vertexing, impact parameter measurement, and b-jet tagging alongside the original pixel layers. In the next layer, the SCT formed of silicon microstrip detectors contributes to momentum measurement as well as vertex reconstruction. Finally, the last part of the ID is the TRT which is used for additional particle tracking as well as electron identification. Following the inner detector, there is the electromagnetic calorimeter and hadronic calorimeter which are used to measure the energy of charged particles, photons, jets, and neutrons. However, some particles are deeply penetrating, such as muons, so there is a dedicated muon spectrometer surrounding ATLAS as the final layer of detectors.

Machine learning

Shallow Machine Learning

Machine learning is a collection of algorithms in which a classifier or statistical model is iteratively improved upon by successively processing a set of training data. The goal is for the model to learn patterns in the training data that are common to other independent data sets taken of the same underlying phenomena. Thus, the model would be a classification tool whose performance does not depend on the specific set of data given but is capable of pattern recognition to distinguish signal and background events. This results in the compression of multidimensional discriminatory information down to one variable which separates signal and background.

Figure 3: Classifier output example.

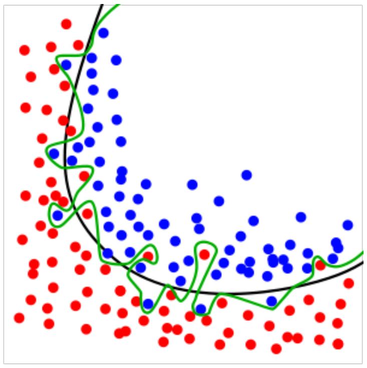


These models $\hat{y}(x_i)$ are based on receiving sets of input variables $x_i \in X$ i=1...n, where X is the n dimensional configuration space of inputs, and outputting for each event a value indicating the model's prediction of the event's class y which for the training set is known. For example, $y = 1$ and $y = 0$ or -1 are often used to indicate signal and background respectively. Many techniques of machine learning are based on stacking layers which isolate more and more complex regions of X in order to efficiently locate where the signal events $\mathcal{S} = \{x_i | y = 1\}$ lie. This leads to the existence of both shallow and deep models, where shallow and deep respectively refer to the level of abstraction that the classifier can attain. Namely, more layers of processing in the model correspond to a higher potential for learning abstract nonlinear features of the training data.

Machine learning is an attractive tool for use in particle physics because the problem defined above is found in data analysis done on particle collision data. The problem focused on in this report is that of isolating Higgs events where the production mode is VBF ($y = 1$) from those produced via other processes ($y = 0$ or -1), mostly ggF, and the $qq \rightarrow ZZ$ background given an input of kinematic characteristics ($x_i,\ i = 1..n$) of the particles involved. The study of machine learning in particle physics is motivated by the fact that it is more efficient to increase the statistical significance of physics results by using more sophisticated data analysis techniques than by producing more collisions.

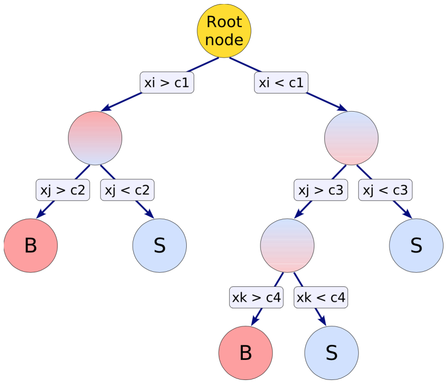
Before discussing specific techniques of machine learning, it is important to first define overfitting: the result of a classifier learning features specific to the training set used but not common to other data sets which represent the same phenomena. Overfitting is screened for by using separate training and testing data, where if the classification performance is much better for the training set than for the testing set it is clear that the model suffers from overfitting.

Figure 4: Overfitting.



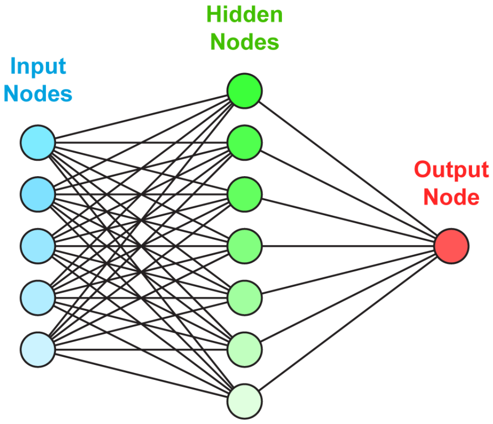
The simplest and most commonly used method for isolating regions of higher statistical significance when searching for a signal in data is a cut or multiple cuts $\{x_a > a_0,\ x_b > b_0 ...\}$. This method isolates a single hypercube in X and is a useful technique when high statistics are available. Going further, it is possible to use a decision tree with a specified depth to isolate multiple hypercubes in X. It is easy to see that these methods are not optimal, as there is no reason why signal regions would be rectangular and because blindly increasing the tree depth to extreme values in order to produce complex geometries leads to overfitting.

Figure 5: Decision tree.



The decision tree shown in Figure 5 is considered a shallow classifier due to its simple linear structure. However, the method of boosting uses the vote of many weak classifiers trained on re-weighted versions of the training set to form a strong classifier. Hence, boosted decision trees (BDTs) are a commonly used high performance classifier in particle physics because they are easy to validate yet highly effective. However, BDTs are still shallow and thus have limited ability to discover high level abstract features of the data. This means they perform significantly better when given training variables that are purposely crafted by physicists to discriminate between signal and background. Another classifier with similar performance characteristics is the artificial neural network (NN) seen below.

Figure 6: Artificial neural network diagram.



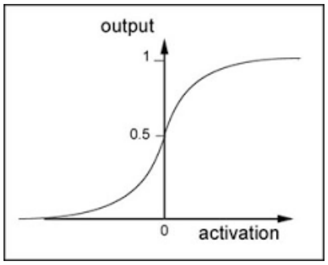
Artificial neural networks are biologically inspired classifiers which learn by adjusting the signal propagation strength (weight) associated with each connecting path (synapse) between nodes (neurons) and by adjusting biases at neurons in a way that minimizes the cost function $C(\hat{y}(x_i),\ y(x_i))$ of the output, where y is the desired output for a given set of inputs and $\hat{y}$ is the approximation to y computed by the model. Therefore, NNs are trained by a gradient descent algorithm in the space of weights which determine its behaviour. Single layer NNs are non-linear and in fact can be used to compute any function if allowed to be arbitrarily large [1]. In practice however, the required size of the NN needed to model complex data is unmanageably large.

The way that a neural network calculates an output is by propagating the standardized input values forward through the network. For a given neuron n_j^{k+1} , the j^{th} neuron in layer $k+1$ of the network, the output of the neuron is the activation function f applied to the biased weighted sum over all synapses connected from layer k of the output of neurons n_i^k.

$$a_j^{k+1} = f\left(b_j^{k+1} + \sum_i w_{ij}^k a_i^k\right)$$

The activation function is common to all neurons in a network and is meant to standardize the outputs of neurons, it is commonly a sigmoid or tanh function.

Figure 7: Activation function.



In the notation used above, the input values of Figure 7 would be $a_i^1,\ i = 1-5$ and the output value would be $\hat{y} = a_1^3$.

Deep Learning

In order to skip the step of manually constructing discriminatory input variables, one would desire a technique capable of automatically discovering useful non-linear functions of raw input data. For example, if a classifier was being made to recognize Higgs boson candidate events given the four momenta of all four final state leptons in $4l$ final state data, then one could give the $4l$ invariant mass m_{4l} as a training variable and obtain high efficiency selection by using a shallow classifier, in this case the cuts $m_{4l} > 120$ GeV and $m_{4l} < 130$ GeV would work well. However, if the only training variables were the raw components of the four momenta then a classifier would be needed that could discover the relationship $m_{4l} = \sqrt{(\sum_{i=1}^4 p_{l_i}) \cdot (\sum_{i=1}^4 p_{l_i})}$ itself. This task would necessarily require a deep classifier.

A classifier used for such abstract pattern recognition tasks is the deep neural network. Deep neural networks are multi-layered versions of the artificial neural network introduced above. They contain many hidden layers, often with hundreds or thousands of hidden neurons per layer in commercial applications or machine learning research. The multi-layer structure endows the deep network with the potential for many layers of non-linear input signal processing and can result in high level feature recognition of simple raw data. The process of training deep networks is more complicated than with their shallow counterparts. This is a result of their lack of transparency, the large number of parameters involved in defining a deep neural network, the dependence of performance on structural characteristics as well as hyper parameters like learning rate, and finally the vanishing gradient problem. The vanishing gradient problem is the tendency for the learning process to occur at different rates throughout the networks depth. Due to the rich nature of training deep neural networks, the use of deep neural networks has taken on its own name: deep learning.

In order to train a neural network, of any size, it is necessary to calculate the partial derivatives $\frac{\partial C}{\partial w_{ij}^k}$ as the optimization of a neural network is a question of finding the minimum cost in the space of possible configurations of w_{ij}^k. This is the rate of change of the cost function - or how wrong the model is for a given event or batch of events - with respect to the weight associated with the synapse connecting the i^{th} neuron in layer k to the j^{th} neuron in layer $k+1$. For large neural networks, it is not feasible to calculate these partial derivatives by numerically varying each weight and propagating the change forward through the network to calculate ΔC as this would have to be done for each parameter of the network. However, it is much less computationally intensive to use an algorithm called back propagation to calculate the required partial derivatives. This algorithm is based on using the chain rule of calculus to use the partial derivatives that have already been computed in order to minimize the amount of new computation that has to be done. By doing this, the algorithm propagates from output to input once, calculating all the partial derivatives along the way.

II. METHODS & RESULTS

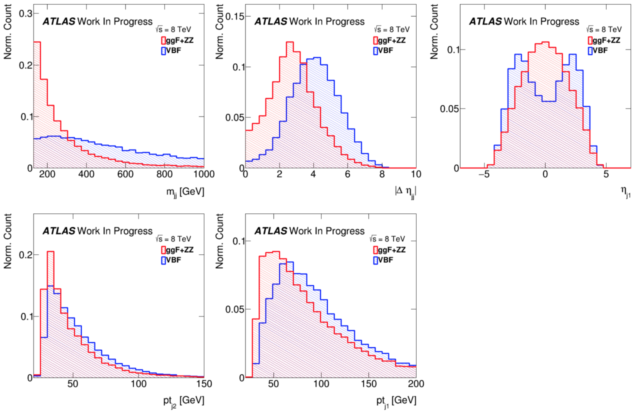
The goals of the study detailed within this report were as follows:

1. determine the effects of each training variable on BDT performance and optimize the BDT used to separate VBF signal from ggF and qqZZ background with respect to training variables
2. explore the feasibility of using deep learning to augment the performance of VBF selection

BDT Optimization

It is a kinematic characteristic of VBF events that there are 2 quark initiated jets which have a large separation in η ($\Delta\eta_{jj}$) and dijet invariant mass m_{jj}. Therefore, the problem of VBF selection lends itself well to using the higher level physically relevant variables m_{jj} and $\Delta\eta_{jj}$ for training a shallow classifier, namely a BDT. The previous analysis of this problem concluded that the core five variables $\{m_{jj},\ \Delta\eta_{jj},\ \eta_{j1},\ p^T_{j1},\ p^T_{j2}\}$, were an effective set for training BDTs [2].

Figure 8: Distributions of m_{jj}, $\Delta\eta_{jj}$, η_{j1}, p^T_{j1}, p^T_{j2} for VBF (blue) and ggF+ZZ (red).



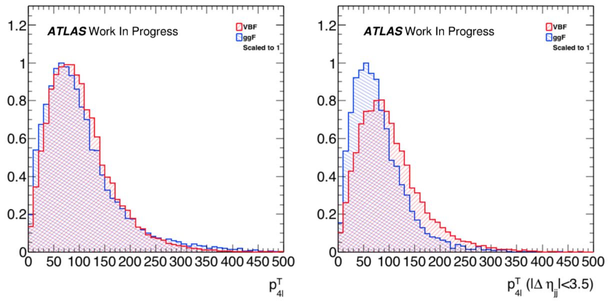
The performance of a classifier can be reliably determined by its receiver operating characteristic (ROC) curve. This curve plots the efficiency with which a cut on the output distribution of the classifier accepts signal and rejects background. An ideal classifier would have a point like ROC curve localized at (1,1) corresponding to perfect classification. Whereas a flat output distribution which contains no discriminatory power would yield the line BkgRej = 1 − SigAcc as an ROC curve. Therefore, a single scalar value representing the performance of a classifier can be chosen to be the integral of the ROC curve. The BDT trained on the core five variables shown above is the standard performance for this problem and corresponds to an ROC integral of 0.785. The previous analysis also concluded that it did not effect classification efficiency in any significant way to train on purely ggF background [2]. Another possible measure of performance is the statistical significance $\frac{S}{\sqrt{B}}$ of a cut at $BDT_{VBF} = 0$ when background and signal distributions are both normalized to unity. Below is a summarized table of these performance measures for BDTs trained on the variable set indicated by ten binary digits corresponding to the set $\{m_{jj},\ \Delta\eta_{jj},\ \eta_{j1},\ p^T_{j1},\ p^T_{j2},\ p^T_{4l},\ w_{j1},\ w_{j2},\ N^{Trk}_{j1},\ N^{Trk}_{j2}\}$ with a background indicated by either ggF, ZZ, or Both.

Table II: BDT training summary.

TrainingSet_Background	$\frac{S}{\sqrt{B}}$ @ $BDT = 0$ (N)	*ROC int*
1100000000_ggH	1.34	0.781
1111100000_ggH	1.35	0.785
1101110000_ggH	1.38	0.794
1111111111_ggH	1.48	0.820
1100000000_ZZ	1.68	0.845
1111100000_ZZ	1.72	0.854
1101110000_ZZ	1.72	0.858
1111111111_ZZ	1.83	0.875
1100000000_Both	1.41	0.799
1111100000_Both	1.42	0.802
1101110000_Both	1.43	0.806
1111111111_Both	1.51	0.828

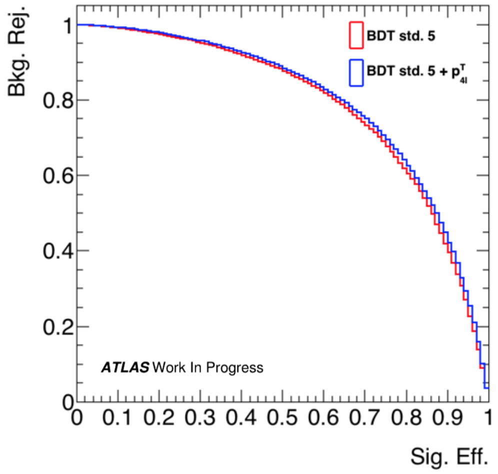
It is shown above that the first two variables, m_{jj} and $\Delta\eta_{jj}$, provide the majority of the separation power. Additionally, it was found that in fact η_{j1} contributed a negligible amount when $\Delta\eta_{jj}$ was included in the training set. For all training sets, from the above table it can be seen that it is easier to discriminate VBF from ZZ background than from ggF events. An initially surprising result is the fact that adding p^T_{4l} to the training set provides a significant boost of classification power, bringing the ROC integral from 0.785 to 0.794. This result is surprising because the signal and background distributions for p^T_{4l} on all of phase space have a small separation in comparison to the other training variables used. However it is worth considering the fact that while all of the discriminatory variables in the training set describe the jet kinematics, p^T_{4l} is a characteristic of the combined final state leptons and so while heavily correlated with the jet behaviour, it might have new information to contribute to the classifier. This can be seen by plotting p^T_{4l} over just half of phase space using a cut on $\Delta\eta_{jj}$.

Figure 9: p_{4l}^{T} distribution on all and half of phase space.



The effect on the ROC curve from adding p_{4l}^{T} to the set of training variables is shown below below.

Figure 10: ROC comparison for the addition of p_{4l}^{T} to the training set.



The BDT training summary in Table 2 contains training sets of all variables. That is, BDTs were trained using jet widths $w_{j[1,2]}$ and number of tracks $N_{j[1,2]}^{Trk}$. While these training variables yielded fantastic increases in performance of the BDT, they are unreliable due to differences between monte carlo and data. Jet widths and number of tracks are not modelled well enough to be training variables for a classifier. This is because the classifier must be trained on monte carlo data; yet it is used to classify real data, making it a difficult method to justify and validate if the training data was not accurate.

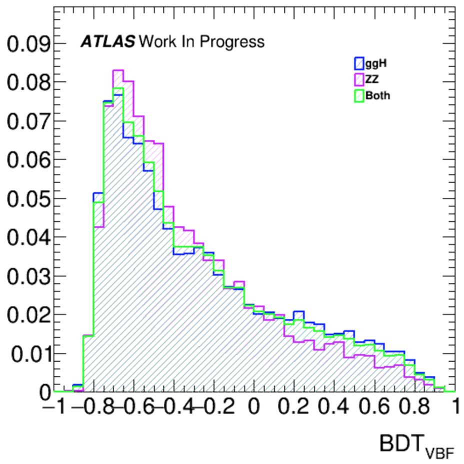
A series of applications of the trained classifiers to alternate backgrounds was done in order to validate the claim that BDTs trained on ggH background can classify ZZ events just as efficiently as BDTs trained on ggH + ZZ.

Table III: BDT applications.

TrainingSet_Background	Application	$\frac{S}{\sqrt{B}}$ @ $BDT = 0$ (N)
1100000000_ggH	ZZ	1.76
1111100000_ggH	ZZ	1.74
1101110000_ggH	ZZ	1.60
1101110000_ggH	Both	1.43
1100000000_ZZ	ggH	1.34
1111100000_ZZ	ggH	1.32
1101110000_ZZ	ggH	1.33
1101110000_ZZ	Both	1.36
1100000000_Both	ggH	1.35
1111100000_Both	ggH	1.35
1101110000_Both	ggH	1.39
1100000000_Both	ZZ	1.73
1111100000_Both	ZZ	1.75
1101110000_Both	ZZ	1.67

Both the training set 1101110000_ggH and 1101110000_Both applied to VBF vs. ggH classification seen in Table II and III give $\frac{S}{\sqrt{B}} = 1.43$. This was used as proof of the claim that the choice to train on ggH or ggF + ZZ background is irrelevant for the problem of VBF selection as stated in the previous note [2]. A visualization of this can be seen below where a BDT trained on ggH background is used to classify all backgrounds.

Figure 11: BDT trained on ggH only applied to all background types.



The conclusion of this BDT study is that the training set $\{m_{jj},\ \Delta\eta_{jj},\ p_{j1}^{T},\ p_{j2}^{T},\ p_{4l}^{T}\}$ performs better than the previous benchmark on this problem, and training on ggH background only is a valid option. As a final note, the BDT algorithms performance is not extremely sensitive to the hyper parameters used to structure the BDT. Therefore, these results are reproducible with a tree depth of >4 and a forest size of >600 as long as overtraining is carefully monitored.

Deep Learning

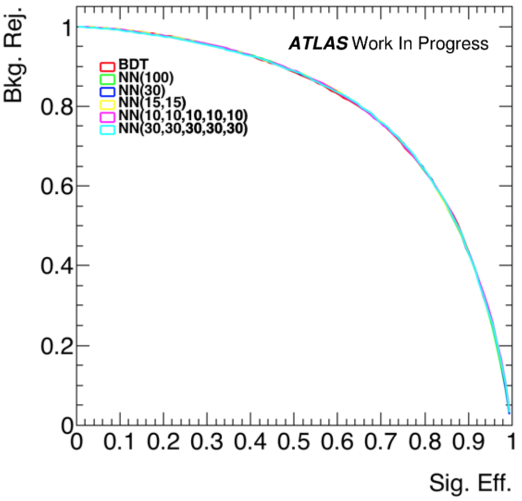
The use of deep learning for problems in particle physics has been shown to have promise recently [3]. Whether it is useful for the specific task of VBF selection in the $H \to ZZ^* \to 4l$ channel is what this work aims to study.

The prevalent package for multivariate analysis used by experimental physicists is the Toolkit for Multivariate Analysis (TMVA), which is an extension of ROOT, the data analysis framework used in high energy physics. This package contains an implementation of neural networks that allows deep networks to be built with any depth or width, however it does not have the ability to use some modern training algorithms like the so called dropout algorithm which stochastically ignores neurons while training in order to avoid overfitting. The TMVA implementation of neural networks as well as the pylearn2 implementation in python were both used in an attempt to improve upon the BDTs' performance for this problem. A range of neural network architectures was tested and it was found that for the problem of VBF selection, deep neural networks could match the performance of BDTs, but not significantly improve upon it. The comparison was done with TMVA using 8 TeV Monte Carlo data. Using both TMVA and pylearn2 frameworks for training neural nets, depths of 4-6 layers with widths in the range 5-100 were explored. The pylearn2 neural networks were trained using theano, a python library for gpu computing and a Tesla K40 graphics card however the speedup in training on a gpu was only seen to be a factor of about 2 improvement upon the use of a 6 core intel xeon cpu.

In general, deep neural network training is a problem of finding a tuned set of hyper parameters and the right architecture for a given problem. Since neural networks are computationally intensive to train, it is likely that the neural networks in this study were not optimal configurations for the problem of VBF selection as a dense grid search through the space of hyper parameters and architectures was not feasible. The following are highlights of the neural network study. The models mentioned below were all trained on the set of variables 110111 unless otherwise stated.

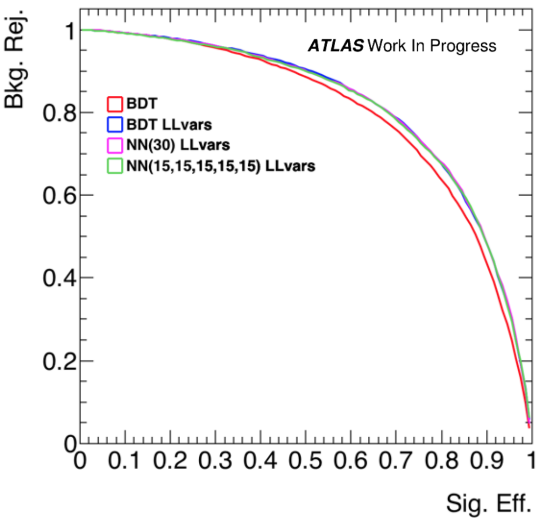
Firstly, shallow neural networks were tested in order to determine whether they could match the performance of BDTs. Indeed, 1 and 2 hidden layer neural networks were found to perform as well as BDTs, but not better. The single layer neural networks had 100 and 30 hidden neurons and were trained with a learning rate of 10^{-4}, and a batch size of 5 events. The 2 layer network had 15 hidden neurons in each layer and was trained with the same hyper parameters as the single layer network. The batch size is the number of events for which the cost function is computed before adjusting the weights of the network according to the average direction suggested by all of the batch events. Both were trained over 1000 epochs, meaning the networks processed the full training dataset 1000 times. The ROC integrals for the test data were 0.796 for the 2 layer network and 0.797 for the single layer networks. Importantly, the maximum performance of a deep neural network on this problem was also found to be equivalent to that of the BDTs and the shallow neural networks. A 5 hidden layer neural network with 10 hidden neurons in each layer trained with a learning rate of 10^{-4} and a batch size of 5 resulted in an ROC integral of .796 as well. The width of the 5 hidden layer network was increased to 30 neurons per layer and still the ROC integral was 0.797. These results were verified by repeating the training with randomized training and testing data to be constant to 2 parts in 1000, thus all methods result in the same performance within statistical fluctuation due to finite data. The ROC curves of all 4 of these neural networks and the BDTs method are plotted together below.

Figure 12: ROC curves of classifiers trained on 110111



Using the python libraries pylearn2 and theano, neural networks of depth 1 to 5 were trained with the standard training set 110111 mentioned above plus the low level variables: jet masses, $\phi_{j[1,2]}$, η_{4l}, $\eta_{j[1,2]}$ and there were no improvements in performance upon the TMVA results. However, when this was done using TMVA, a significant increase in ROC integral was seen. The discrepancy between using pylearn2 and TMVA is not understood, but it is certain that the TMVA networks are not overtrained as the ROC curves produces when using both training and testing samples overlap. This result was thought to be a result of deep learnings ability to extract useful information from low level inputs, however when shallow neural networks and BDTs were given the low level variables $m_{j[1,2]}$, $\phi_{j[1,2]}$, η_{4l}, $\eta_{j[1,2]}$ as well, there was an equal performance gain in all cases. The final ROC integrals were all 0.815.

Figure 13



III. DISCUSSION & CONCLUSIONS

The optimization of the boosted decision trees method used on the problem of VBF signal and ggF background separation in the $H \to ZZ^* \to 4l$ channel was studied, and it was found that the best training variable set consists of the dijet invariant mass (m_{jj}), the dijet separation in η ($\Delta\eta_{jj}$), both jet transverse momenta (p^T_{ji}), and the transverse momentum of the Higgs boson (p^T_{4l}). Furthermore, it was found that η_{j1} was not effective when added to the training set and it was verified that training on ggH background without the ZZ events was a valid method. Later on, it was shown that adding the low level variables $m_{j[1,2]}$, $\phi_{j[1,2]}$, η_{4l}, $\eta_{j[1,2]}$ yielded an increase in ROC integral across all 3 methods BDTs, shallow neural network, and deep neural network, from 0.796 to 0.815.

Shallow and deep neural networks were explored as a potential substitute for BDTs if they performed better, but it was found that both shallow and deep neural networks of varying size had the same effectiveness as BDTs when using the ROC curve as a metric for performance. It seems that when many classifiers of different type obtain the same result, then it is not the method that limits the performance, it is the information available itself. Therefore, deep learning is not a useful tool in augmenting classification performance in the problem of selecting VBF events in the $H \to ZZ^* \to 4l$ channel.

ACKNOWLEDGMENTS

I am grateful to both Pierre Savard and Richard Teuscher for attentive supervision, as well as S. Haider Abidi for consistently providing valuable advice. I would also like to thank Bernd Stelzer and Daniel Mori of Simon Fraser University for computing resources and guidance. Thanks to the University of Toronto and NSERC for supporting my research during the summer of 2015. Finally, I am immensely thankful that the IPP has provided me with the CERN Summer Student Fellowship allowing me to spend part of this summer at CERN.

[1] Michael A. Nielson, “Neural Networks and Deep Learning”, Determination Press, 2015

[2] ATLAS Collaboration, “Study of the 125 GeV SM-like Higgs boson properties using production mechanism specific signatures in the $H \to ZZ^{(*)} \to l^+l^-l^+l^-$ channel.”, 2014

[3] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for Exotic Particles in High-Energy Physics with Deep Learning”, 2014