Single Detector Gravitational Wave Detection Using Multivariate Analysis

Emily Thompson

Clemson University, Clemson, SC 29632 and
Cardiff University, Cardiff, CF24 3AA, UK

Supervisor: Dr. Patrick Sutton

Cardiff Universtiy, Cardiff, CF24 3AA, UK

Dated: August 10, 2014

Abstract

We investigate the possibility of including multivariate analysis (MVA) in the standard X-Pipeline gravitational wave (GW) burst search analysis for a single detector. MVA has been very effective in data analysis involving high energy particle physics, and is very promising in showing similar results in the GW data analysis community as well. We use the software package Toolkit for Multivariate Analysis (TMVA) with a focus on the use of the Boosted Decision Tree (BDT) classifier for the discrimination between GW signals and background detector noise. We find that we are able to achieve an increase in sensitivity to both circular sine Gaussian and inspiral waveforms when an MVA step is integrated into the standard X-Pipeline GW burst search process while analyzing data from a single detector.

1 Introduction

Predicted in 1916 by Albert Einstein in his theory of general relativity, gravitational waves are ripples in the curvature of space-time that travel at the speed of light and transport energy as gravitational radiation. In order to be detectable on earth, they must propagate from very large masses being accelerated in the universe, since the amplitudes of gravitational waves are relatively very small. For example, if a neutron star is in the Virgo cluster (r$\sim$18 Mpc), has a mass of 1.4 $M_{\odot}$, and is formed in a highly-nonspherical gravitational collapse,

the amplitude of the gravitational waves propagated from it will have an upper limit of approximately 1.5 X 10^{-21} (1).

Since 2010-2011, the LIGO and Virgo detectors have been offline to undergo upgrades to create the advanced network of detectors. With GEO 600 as the only GW detector able to take data, we have been in a single detector era. Typical GW burst searches rely on coincidence and correlations between data from two or more detectors to reject non-Gaussian background events as noise. Therefore, it is vital that an alternate way to distinguish GW burst signals from background noise be developed.

Multivariate analysis (MVA) may be the answer. MVA was developed at CERN for high-energy particle physics, and has proven very effective in separating signal from background in large quantities of high-dimensional data. It implements supervised machine learning to look at all variables that make up events, and finds patterns and correlations that best discriminate between true signals and background noise fluctuations.

We evaluate how well MVA, when included in the standard X-Pipeline GW burst search, improves the discrimination between signal and background for data from a single detector. In the Background section, we will outline the importance of GRBs and their associated GWs, and detail the basics of how X-pipeline searches for GW bursts. In the Multivariate Analysis section, we will go into detail about how MVA works and how it can help in GW burst data analysis. The Investigation section will describe step by step our analysis, and the Testing and Results section will describe how each MVA parameter was investigated, as well as present our results. Finally, the Conclusion section will summarize our findings and discuss possible future work.

2 Background

2.1 Gamma Ray Bursts

Gamma ray bursts (GRBs) are intense flashes of gamma ray radiation that have a duration of anywhere from milliseconds to minutes. They are the most luminous electromagnetic events that have been detected in the universe so far. They are thought to originate from the merger of compact binary objects such as neutron stars and black holes, or the core collapse of massive stars in supernovas. These cataclysmic events are also thought to generate gravitational wave bursts that are strong enough to be detected on Earth. These GW bursts are important to detect because they will provide an exciting new probe of the astrophysical systems that create GRBs. The GW emission associated with GRBs typically depends on poorly understood physics, and therefore the detection of a GW burst depends on our ability to distinguish between a rare weak signal with an unknown waveform from highly non-stationary and non-Gaussian background noise.

2.2 X-Pipeline

X-Pipeline is a software package used for the coherent analysis of data from a network of interferometers with the purpose of detecting GW bursts associated with GRBs and other astrophysical triggers. With a network of detectors, X-Pipeline is able to time shift the data and make cuts to reject noise glitches using correlations between the coherent and incoherent detector energies (2). With only data from a single detector to analyze, however, X-Pipeline is unable to apply these veto cuts.

Time-frequency maps of the signal streams are constructed, and clusters of pixels that have large energy values are selected as candidate signal events. Figure 1 is an example of a simple time frequency map that shows the process of clustering. Nearest-neighbor pixels (those that share an edge) are grouped into a single cluster. Each cluster in Figure 1 is denoted by a different color and is considered a candidate detection event. X-Pipeline then assigns a statistical significance based on a user-specified event variable. The higher the significance, the more likely an event is a signal. For a single detector, the 5 event variables that X-Pipeline records for each event cluster are listed and described in Table 1.

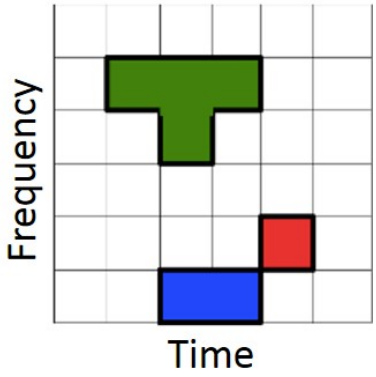


Figure 1: A time frequency map with pixels grouped into 3 clusters, denoted by different clusters. Each cluster is considered a candidate signal event.

Table 1: Single Detector Event Variables

Variable	Description
Energyitf1	Sum of the signal to noise ratios squared for each pixel in a cluster. The 'power' of the data.
Loghbayesian	Bayesian-inspired likelihood ratio for the hypothesis of polarized GWs versus Gaussian noise. Essentially the same as Energyitf1 but weighted so that when the noise is high, loghbaysian is downgraded and vice versa.
Number of pixels	The number of pixels in a given cluster.
Duration	Length of time that given cluster spans.
Bandwidth	Range of frequency of given cluster.

X-Pipeline can only assign the significance of each event based on one of these variables. For a single detector, this is usually loghbayesian. Therefore, events with a higher loghbayesian value will have a higher significance and according to X-Pipeline is more likely to be a signal. Events with a lower loghbayesian are more likely to be background noise.

3 Multivariate Analysis

X-Pipeline has been very effective in data analysis for GW Burst searches using a network of detectors. However, it is limited by its ability to discriminate between signal and background using only one event variable. In a single detector analysis, it is also limited by its inability to make cuts based on coherent veto tests. Therefore, we explore the use of MVA to improve X-Pipeline's effectiveness in data analysis with a single detector.

3.1 Toolkit for multivariate analysis

The ROOT (3) based software package TMVA (4) that was developed by the particle physics community is used in this investigation. TMVA first takes in a data stream consisting of known signal and background events. The data stream is then randomly split into two equal sets with equal amounts of signal and background events in each set. The method that is used to discriminate between signal and background is called the classifier. One of the sets is used to train the classifier, and the other is used to test it. First, the training set is passed through the classifier, and the results and details on how the classifier processed the data is stored in a"weight" file. Then, the testing set is passed through the classifier to test its performance using the weight file. Each event, after passing through the classifier, is given an MVA response value, which specifies how likely the classifier thinks the event is a signal or background. Events with a higher classifier response are more likely to be signal, while a lower classifier response is more likely to be background.

3.2 Boosted Decision Tree

In this investigation, we use the Boosted Decision Tree (BDT) classifier. TMVA has multiple different classifiers to choose from, but for our purposes, BDT is the simplest and exhibits best results in the shortest processing time. A decision tree is based on a binary tree-like structure and contains a series of branches (yes/no decision nodes) and leaves (end nodes). Figure 2 depicts a decision tree. The blue and yellow boxes (labeled 'Background' and 'Signal', respectively) represent the leaves, while the grey circles and boxes represent branches and thresholds, respectively.

The BDT classifier starts at the top of the tree (V_a in Figure 2), and for each event variable, performs a user-specified number of equally spaced cuts. Each cut is treated as a 'test threshold', and BDT notes how well signal and background is separated for each cut as if it was used as the threshold at that node. Then, BDT chooses the best cut for the best variable and makes this the first threshold to separate events in the tree. Each event, depending on the value of the chosen variable, either goes to V_b or V_c in the second layer of the tree. Then, the process repeats in each branch separately. The decision made at each branch node represents a cut in the phase space. The phase space is continuously split by this method until it is divided into many regions that end up as either signal or background, depending on the majority of training events present in that leaf node. The user must specify a stop criterion, which is either a limit on the number of layers the tree can contain, or a certain purity of signal or background that automatically creates a leaf node.

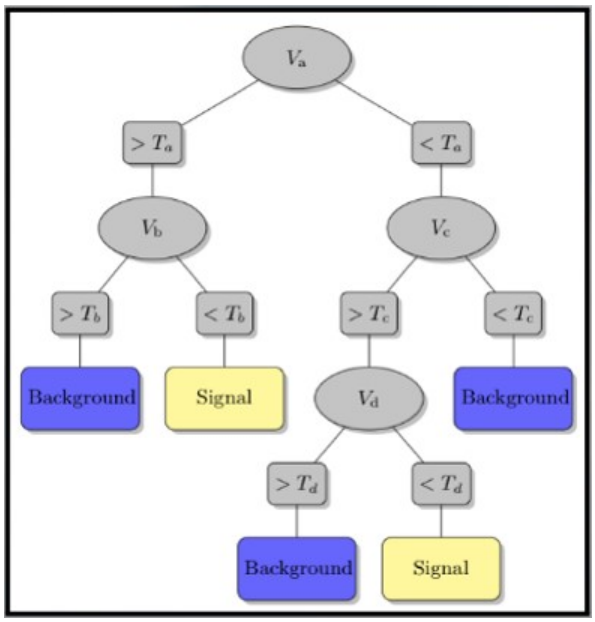


Figure 2: Shematic of a decision tree. V_a, V_b, V_c and V_d are event variables, and T_a, T_b, T_c, and T_d are values of those variables that are the thresholds to determine the sorting of the events. (5)

Decision trees are susceptible to statistical fluctuations within the training data set. To avoid this, multiple decision trees are made to create a "forest". Each tree in the forest is created using a random subset of the training set. The final classification of events is determined by a majority vote from the results of each tree. The procedure not only stabilizes the response of the individual trees but also enhances overall performance.

"Boosting" is another method used to statistically stabilize the classifier. During training, signal and background events that are misclassified in one tree are 'boosted' in the next tree. This means that they are given a higher weight in the next tree being constructed, so it is more likely to be sorted into the correct end node. Although a simple strategy, it has been shown to dramatically increase the performace of decision trees.

4 Investigation

In our investigation, we apply MVA to a GW burst search by integrating it into the standard X-Pipeline search process, depicted in Figure 3. Unfortunately, to this date no extraordinary Galactic event that is likely to have produced a GW burst has been observed. Therefore, we have chosen to test the MVA analysis on a high energy neutrino observed by IceCube (4) on August 8th, 2011. The event was nicknamed 'Bert' and it has the largest recorded neutrino energy recorded to date (6). Because we used Bert data, we assumed a high sky position accuracy in the location of where a GW burst would come from.

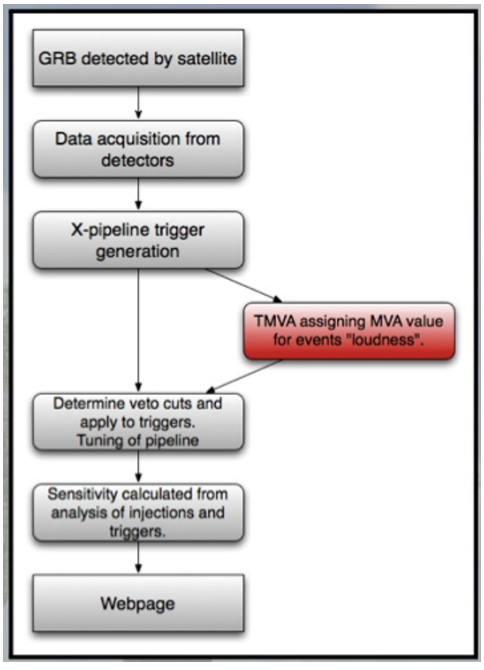
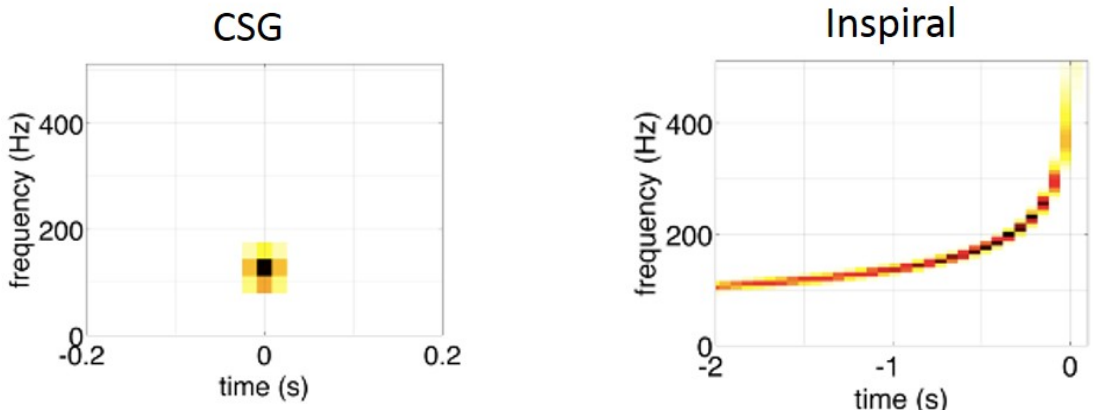


Figure 3: Shematic of MVA integrated into the standard X-Pipeline data analysis process. (7)

The data stream from GEO 600 during the time of Bert is acquired, and a set of simulated GW waveforms are generated to inject into the signal data set. Before these waveforms can be injected into the signal data set, all of their parameters must be slightly randomized. Otherwise, the MVA classifier will be severely over trained and will only consider waveforms with very specific parameters to be GWs.

We generated a random mix of circular sine Gaussians (CSGs) and binary neutron star inspirals to be injected into the training set. For the testing set, we injected CSGs and inspirals separately so we could investigate how will the classifier was sorting each waveform. Figure 4 shows time frequency maps of a CSG and an inspiral waveform. These two waveforms were chosen for training because as seen in Figure 4, they are very different from each other and will have a wide range of values for each event variable. This prevents MVA from training the classifier to only identify one kind of waveform as a GW burst. This is important because as mentioned before, the exact waveform of a GW burst is currently unknown and we must not overtrain the classifier to identify only certain types of waveforms. The set of waveforms is injected multiple times into the data stream at different amplitudes, or injection scales. As a result, we can easily evaluate the sensitivity of our analysis based on how many waveforms at each injection scale are recognized as signal.



(a) (b)

Figure 4: (a) Time-frequency map of a typical CSG waveform. (b) Time-freqency map of a typical inspiral waveform. Both of these waveforms are used to test and train the classifier. (5)

After testing and training the classifier, plots of the separation between signal and background are made in order to display how well the MVA did at discriminating between the two. Figure 5 is a standard output plot produced by TMVA. On the vertical axis is the significance X-Pipeline gave to each event. On the horizontal axis is the classifier response of MVA, rescaled to fall between 0 and 100. The blue points are background, and are overlaid on top of the red points, which are signal. This plot shows that with only the X-pipeline significance used as a measure to discriminate between signal and background, the cut would be very poor and either much signal would be considered background or vice versa. However,

with the addition of MVA into the X-pipeline burst search process, we are able to make a 2 dimensional cut (depicted by green line) that does a much better job at separating signal from background.

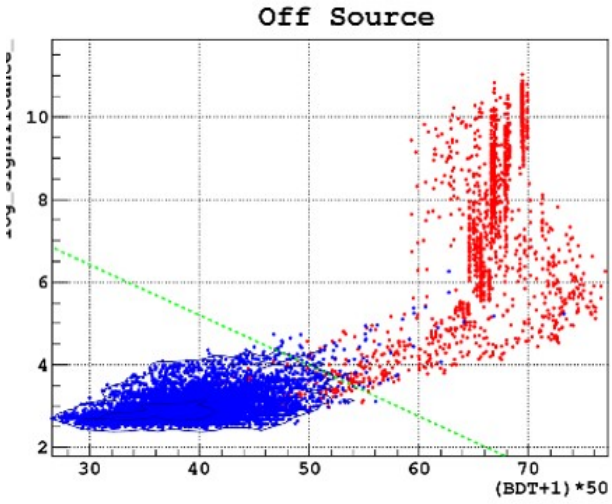


Figure 5: Standard output plot for TMVA. The Y-axis is the significance that X-Pipeline assigns to each event, while the X-axis is the classifier response for each events. Red events are signal and blue events are background. The green line shows the optimal cut between signal and background using both the X-Pipeline significance and classifier response.

Figure 6 is a standard histogram produced by TMVA. Not to be confused with Figure 5, the red here is background and the blue is signal. This histogram is to check for overtraining of the classifier. If the points do not land close to the top of the histogram bars, then it is an indication that the classifier has been overtrained. Figure 6 depicts this at around a BDT response of 0, where the dots curve slightly upward from the top of the histogram bars. This overtraining is thought to be from small amplitude injections in the data stream. Because they are so small, the classifier looks at the characteristics of the noise covering the injection instead of the actual injection waveform. Therefore, it is considering the event variables of the noise to be the signal's variables, and is incorrectly training the classifier. Recent significant improvements have been made to eliminate this overtraining by adding an additional cleaning step to the data stream before sending it through MVA.

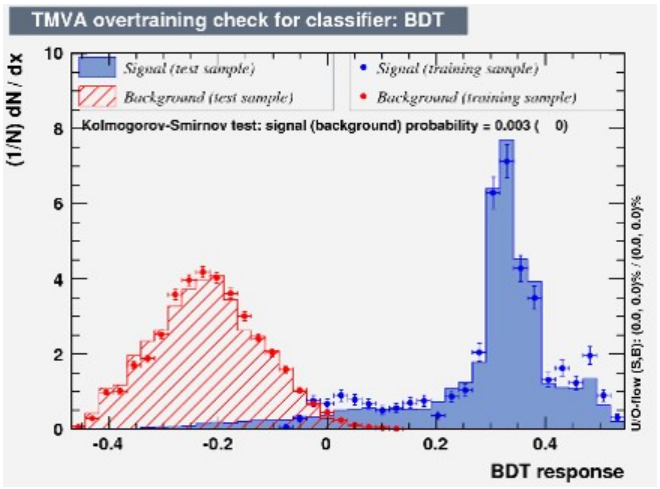


Figure 6: Standard output histogram for TMVA. If the points do not line up closely with the top of the histogram bars, the classifier is being overtrained. The slight overtraining seen around a BDT response of 0 is thought to be due to X-Pipeline recovering noise in place of the event due to small injection scales.

After passing through MVA, the classifier response replaces the X-Pipeline significance for each event and the results are passed back to X-Pipeline, where the sensitivity of the analysis is determined and displayed in a webpage. For each waveform in the testing set, a plot such as Figure 7 is created. Figure 7 shows the detection efficiency of the default TMVA analysis for the CSG waveform. Each point represents a set of waveforms at a certain injection scale. The horizontal axis is the root-sum-squared (h_{rss}) amplitude of the injections, which is given by Equation 1, where h_+ and h_x are the plus and cross polarizations of the gravitational wave, respectively (6). The units of h_{rss} are therefore $\mathrm{Hz}^{-\frac{1}{2}}$. The vertical axis is the fraction of the injections at the given amplitude that survive the analysis cuts and are recognized as signal. We want the steep increase in the curve to occur at the smallest h_{rss} amplitude possible in order to have the most sensitive analysis.

The red and yellow points on Figure 7 show the 90% and 50% upper limits (ULs), respectively. These are the amplitudes at which at least 90% or 50% of the injected waveforms are recognized as signal when passed through the classifier. In our investigation, these upper limits are used to quantify the sensitivity of a certain analysis. The smaller the upper limit, the more sensitive the analysis.

$$h_{rss} = \sqrt{\int_{-\infty}^{+\infty} [h_+^2(t) + h_x^2(t)]dt} \tag{1}$$

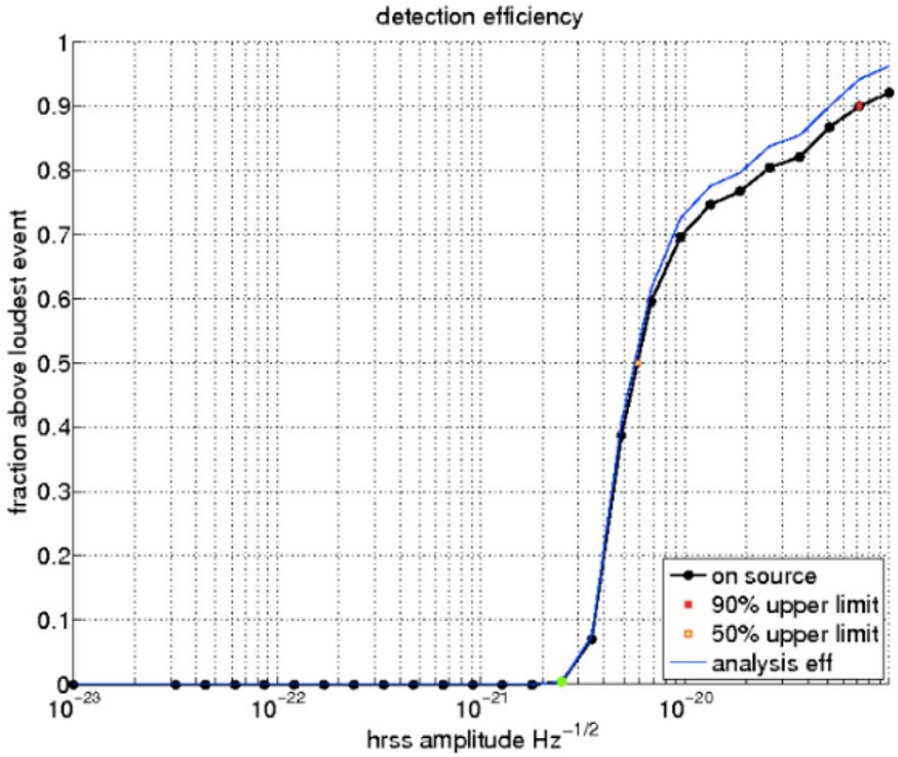


Figure 7: Example output plot of the detection efficiency for the CSG waveforms. Each point represents a set of injections, with each point having a continuously larger amplitude injection scale. The vertical axis represents the fraction of each set of injections that survive the analysis cuts and are recognized as signal.

5 Testing and Results

TMVA allows the user to specify multiple parameters that define how the classifier is trained. We chose to investigate 6 of these parameters that were available for the BDT classifier. These are listed and described in Table 2. The number after the Boosting method is the learning rate of the boosting method being used.

Table 2: MVA parameters investigated

MVA parameter	Description	Default
nCuts	Number of threshold values the BDT tests at each node	20
NTrees	Number of trees in the forest	400
MaxDepth	Maximum number of layers in a tree	3
MinNodeSize	Minimum percentage of training events requred in a leaf node	5%
Boosting Method	Method to determine re-weighting of misclassified events	AdaBoost .5
Weighting	Event variable that classifier should give most importance	No weight

We first ran a standard X-Pipeline GW burst search without the MVA step included. The upper limit values of this run are listed on the 2nd row of Tables 3-9, and are in units of $\mathrm{Hz}^{-\frac{1}{2}}$. Then, we looked at one MVA parameter at a time and found the value of this parameter that maximized the sensitivity of our analysis. While tweaking the chosen parameter, we kept all other parameters at their default value. To determine which value of each parameter maximized the sensitivity, we looked at the 50% and 90% upper limits for both the CSG and inspiral testing waveforms. After determining the best value for each of the 6 parameters in Table 2, we applied these values to each parameter and tried to find the best overall sensitivity that the X-Pipeline with MVA search process could obtain.

5.1 nCuts

nCuts is the number of cuts applied to each variable when BDT is searching for the best threshold value. The higher nCuts is, the better the chosen threshold should be. However, the trade-off is that with more cuts being made for each variable, there is more computing power and time that is needed. Therefore, we want to choose a value of nCuts that is high enough so it significantly increases the sensitivity of our analysis, while not excessively increasing the computing time needed. We tested nCuts from a value of 5 to 65 in increments of 5 for each waveform. We calculated the ratio of the upper limit obtained with each nCuts value to the original X-Pipeline upper limit and the results are listed in Table 3.

The optimal value of nCuts was 45 for the inspiral waveform ULs and the 50% UL for CSG. The best value for the 90% UL CSG was 25. However, both of these optimal values did not result in a dramatic increase in sensitivity over other values of nCuts, and may be just due to statistical fluctuations.

5.2 NTrees

NTrees is the number of trees created in the BDT forest. The more trees there are, the less susceptible to statistical fluctuations the BDT classifier is. However, more computing

power and time are needed as NTrees increases. Therefore, we want to choose a value of NTrees that statistically stabilizes the classifier to a certain degree while still not causing a substantial increase in computing time. We tested NTrees values from 200 to 1400 in increments of 200 trees. We once again calculated the ratio of the MVA upper limits to the original X-Pipeline upper limit, and the results are listed in Table 4.

We find a significant increase in sensitivity for all UL with NTrees larger than 200. However, further increase in NTrees above this value did not result in an increase in sensitivity. Therefore, the best value of NTrees is 400, because it gives the best sensitivity with the shortest computing time.

5.3 MaxDepth

MaxDepth is the maximum number of layers allowed in a tree. A small MaxDepth value will affect the analysis greatly, while larger values of MaxDepth will not because the trees will finish sorting before reaching the maximum depth. Therefore, we only tested MaxDepth values of 2 to 7 in increments of 1. The results are listed in Table 5.

We find a significant increase in sensitivity for all UL with a MaxDepth larger than 2. However, no further increase in sensitivity is observed for higher values of MaxDepth.

5.4 MinNodeSize

MinNodeSize is a stop criterion that limits how small a node can get. The smaller MinNodeSize, the more each tree will sort and discriminate between background and noise. However, having too small of a MinNodeSize will result in insignificant branches at the bottom of the trees, since each node will be sorting too few events. Therefore, we tested vales of MinNodeSize from 3-5% in increments of 1%, and then a value of 10% as well. The results are listed in Table 6.

A MinNodeSize of 10 resulted in a decrease in sensitivity for both inspiral waveform UL and the 50% CSG waveform. For the smaller MinNodeSize values, these same waveforms were essentially the same, but for the CSG 90% UL we saw an increase in sensitivity with a MinNodeSize of 3. Therefore, we consider a MinNodeSize of 3% to be the best value.

5.5 Boosting Method

Boosting is a way of enhancing the classifier's performance and increasing statistical stability. The two types of boosting that we investigated were Adaptive Boost (AdaBoost) and Gradient Boost (GradBoost). AdaBoost works by multiplying previously misclassified events by a common boost weight α, which is given below by equation 2.

$$\alpha = \frac{1 - err}{err} \tag{2}$$

In Equation 2, err is the misclassification rate in the previous tree. The learning rate of AdaBoost is controlled by a parameter, β, that is given as an exponent to the boost weight (α^{β}). According to TMVA, AdaBoost performs best with small individual trees and a slow learning rate (8). Therefore, we tested AdaBoost with a learning rate of 0.3, 0.5, and 1. The results are listed in Table 7.

We also investigated the GradBoost boosting method. GradBoost works similarly to AdaBoost, but uses a different loss-function, which determines the deviation between the model response and the true value obtained from the training sample. Just like AdaBoost, GradBoost works best on small individual trees, and its robustness can be enhanced by reducing its learning rate through the shrinkage parameter. The shrinkage parameter controls the weight of the individual trees. We tested GradBoost with shrinkage values of 0.5, 1.0, and 1.5. The results are listed in Table 7.

We found no significant increase in sensitivity for the 50% UL of either waveform for any of the AdaBoost or GradBoost runs. However, we saw an increase in sensitivity for the 90% UL of both waveforms using AdaBoost with a learning rate of 0.5. Therefore, we consider this the best boosting method and learning rate value.

5.6 Weighting

The last MVA parameter we investigated was weighting. Weighting values are user specified values that tell the classifier which event variable to give the most importance to when deciding where thresholds should be placed. We tried 4 different runs with energyitf1, log(energyitf1), significance, or log(significance) as the variable on which weighting was placed. The significance variable is the X-Pipeline significance, which in this case was the loghbayesian variable. The results are listed in Table 8.

All of the weights we tried proved to decrease sensitivity in one or more of the UL, and therefore we considered the best run to be without any weighting.

5.7 Mix

Taking all of the best MVA parameter values that we have discussed in the previous sections, we tried multiple runs using various different changes to parameters. We tweaked the values until we found the best result, which is listed in Table 9. These UL values were obtained with: nCuts=50, NTrees=600, MaxDepth=6, MinNodeSize=3, AdaBoost with a learning rate of 0.5, and no weighting. As shown in Table 9, we were able to increase the sensitivity in both the 50% and 90% UL for both waveforms.

Table 3: nCuts

nCuts	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
5	1.01	1.04	0.94	1.05
15	0.93	0.98	0.88	1.20
25	0.91	0.98	0.87	1.01
35	0.91	0.97	0.87	1.05
45	0.87	0.94	0.85	1.05
55	0.88	0.94	0.87	1.09
65	0.90	0.94	0.86	1.05

Table 4: NTrees

NTrees	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
200	0.93	0.94	0.89	1.15
400	0.87	0.90	0.87	1.09
600	0.86	0.90	0.87	1.09
800	0.88	0.92	0.88	1.09
1000	0.88	0.94	0.87	1.09
1200	0.87	0.96	0.87	1.09
1400	0.87	0.97	0.87	1.09

Table 5: MaxDepth

MaxDepth	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
2	0.96	0.98	0.92	1.18
3	0.87	0.90	0.87	1.09
4	0.89	0.92	0.88	1.09
5	0.86	0.96	0.86	1.16
6	0.84	0.90	0.86	1.09
7	0.86	0.94	0.86	1.09

Table 6: MinNodeSize

MinNodeSize (%)	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
3	0.86	0.92	0.85	1.02
4	0.85	0.89	0.86	1.09
5	0.87	0.90	0.87	1.09
10	0.97	1.03	0.89	1.09

Table 7: Boosting Method

Boosting Method	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
AdaBoost .3	0.93	0.95	0.86	1.14
AdaBoost .5	0.87	0.90	0.87	1.09
AdaBoost 1	0.88	0.97	0.89	1.18
GradBoost .5	0.85	0.95	0.86	1.14
GradBoost 1	0.89	0.97	0.89	1.20
GradBoost 1.5	0.89	0.99	0.88	1.22

Table 8: Weighting

Weighting	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
log(energyitf1)	0.88	0.90	0.87	1.15
energyitf1	0.92	0.95	0.87	1.14
log(significance)	0.92	0.94	0.87	1.15
significance	0.89	0.96	0.87	1.09
no weight	0.87	0.90	0.87	1.09

Table 9: Mix

Mix	50% UL (Insp)	90% UL (Insp)	50% UL (CSG)	90% UL (CSG)
X-Pipeline Original	2.82E-20	4.00E-20	6.70E-21	6.57E-20
Mix	0.81	0.90	0.82	0.97

6 Conclusion

In this investigation, we were successful in determining the optimal MVA parameter values that maximized the sensitivity of the single detector analysis. The best UL ratios we obtained are listed in Table 9, and the MVA parameter values that resulted in these are listed in section 5.7. We were able to increase the sensitivity of the X-Pipeline single detector GW Burst analysis on the 50% and 90% UL for both the inspiral and CSG waveforms. Therefore, MVA is a worthwhile method to explore further in implementing permanently in future GW burst searches, especially during this single detector era.

There is a tremendous amount of future work that can be done. Within the BDT classifier, there are more parameters that can be investigated such as pruning, which cuts the trees from the bottom up to remove insignificant branches, and bagging, which is a resampling technique that trains the classifier using resampled training events each time.

Our data from the Bert event assumed a very high sky position accuracy. It would be worthwhile to test the efficiency of using MVA with a range of different sky positions, as well as a lower sky position accuracy. In addition, testing our analysis with MVA using a hardware injection will ensure that it works as we expect it to in the case of real GW wave being detected. This is done by adding a signal into the control system of the interferometers, which makes the instrument behave as if a real GW signal was detected.

We trained the BDT classifier with a random mix of CSG and inspiral waveforms, and tested with the same two waveforms separately. However, we are unsure of what an actual GW burst signal will look like, and therefore it is important that our trained BDT classifier is able to pick out signals that are not exactly like a CSG or inspiral. This can be tested by injecting additional waveforms, such as chirplets and white noise bursts, into the training data set. If the trained BDT classifier is unable to pick up these waveforms in testing, then we should inject the training data set with different combinations of waveforms and determine which combination results in the classifier recognizing the most diverse range of signal waveforms.

Currently, X-Pipeline only records the 5 variables listed in Table 1 in a single detector analysis. While X-pipeline only uses one of these variables when assigning the significance for each event, the MVA step uses all of these variables to assign significance. Therefore, it would be worthwhile to have X-Pipeline record as many variables as possible for each cluster to be sent to MVA. One obvious variable that is missing is the central frequency of the clusters. In Figure 1, if the single-pixel red cluster was shifted to a higher central frequency (moved vertically up on the time frequency map), it would be considered the exact same to X-Pipeline. Adding a central frequency variable would allow X-Pipeline to take note of this change, and more importantly, for MVA to have another variable to use in the discrimination between signal and background. This simple change, along with the other various methods mentioned above, could significantly increase the sensitivity of our GW burst searches and elevate our chances of making a real GW burst detection in the near future.

References

[1] B. Sathyaprakash and B. F. Shutz, Living Rev. Relativity 12, (2009), 2. (2009).

[2] P. J. Sutton, New J.Phys.12:053034,2010 (2010).

[3] CERN, *ROOT User's Guide. An Object-Oriented Data Analysis Framework*, 2006.

[4] M. G. Aartsen, Physical Review Letters, 111(2):21103 (2013).

[5] T. S. Adams *et al.*, Physical Review D, 88(6) 62006. (2013).

[6] T. S. Adams, Detector characterisation and searches for gravitational waves using geo 600, Master's thesis, Cardiff University, 2014.

[7] D. Meacher, Distinguishing gravitational wave signals from detector noise using multivariate analysis, Master's thesis, Cardiff University, 2011.

[8] CERN, *Toolkit for Multivariate Data Analysis with ROOT*, 2013.