PHYSICAL REVIEW D **88,** 062003 (2013)

Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data

Rahul Biswas,[1] Lindy Blackburn,[2] Junwei Cao,[3] Reed Essick,[4] Kari Alison Hodge,[5] Erotokritos Katsavounidis,[4] Kyungmin Kim,[6,7] Young-Min Kim,[8,7] Eric-Olivier Le Bigot,[3] Chang-Hwan Lee,[8] John J. Oh,[7] Sang Hoon Oh,[7] Edwin J. Son,[7] Ye Tao,[9] Ruslan Vaulin,[4,*] and Xiaoge Wang[9]

[1]*University of Texas-Brownsville, Brownsville, Texas 78520, USA*
[2]*NASA Goddard Space Flight Center, Greenbelt, Maryland 20771, USA*
[3]*Research Institute of Information Technology, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China*
[4]*LIGO-Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[5]*LIGO-California Institute of Technology, Pasadena, California 91125, USA*
[6]*Hanyang University, Seoul 133-791, Korea*
[7]*National Institute for Mathematical Sciences, Daejeon 305-811, Korea*
[8]*Pusan National University, Busan 609-735, Korea*
[9]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China*

(Received 29 April 2013; published 23 September 2013)

The sensitivity of searches for astrophysical transients in data from the Laser Interferometer Gravitational-wave Observatory (LIGO) is generally limited by the presence of transient, non-Gaussian noise artifacts, which occur at a high enough rate such that accidental coincidence across multiple detectors is non-negligible. These ''glitches'' can easily be mistaken for transient gravitational-wave signals, and their robust identification and removal will help any search for astrophysical gravitational waves. We apply machine-learning algorithms (MLAs) to the problem, using data from auxiliary channels within the LIGO detectors that monitor degrees of freedom unaffected by astrophysical signals. Noise sources may produce artifacts in these auxiliary channels as well as the gravitational-wave channel. The number of auxiliary-channel parameters describing these disturbances may also be extremely large; high dimensionality is an area where MLAs are particularly well suited. We demonstrate the feasibility and applicability of three different MLAs: artificial neural networks, support vector machines, and random forests. These classifiers identify and remove a substantial fraction of the glitches present in two different data sets: four weeks of LIGO's fourth science run and one week of LIGO's sixth science run. We observe that all three algorithms agree on which events are glitches to within 10% for the sixth-science-run data, and support this by showing that the different optimization criteria used by each classifier generate the same decision surface, based on a likelihood-ratio statistic. Furthermore, we find that all classifiers obtain similar performance to the benchmark algorithm, the ordered veto list, which is optimized to detect pairwise correlations between transients in LIGO auxiliary channels and glitches in the gravitational-wave data. This suggests that most of the useful information currently extracted from the auxiliary channels is already described by this model. Future performance gains are thus likely to involve additional sources of information, rather than improvements in the classification algorithms themselves. We discuss several plausible sources of such new information as well as the ways of propagating it through the classifiers into gravitational-wave searches.

DOI: 10.1103/PhysRevD.88.062003 PACS numbers: 04.80.Nn, 07.05.Mh, 07.05.Kf

I. INTRODUCTION

The Laser Interferometer Gravitational-wave Observatory (LIGO) is a two-site network of ground-based detectors designed for the direct detection and measurement of gravitational-wave signals from astrophysical sources [1,2]. The LIGO detectors, in their initial configuration [1], have operated since 2001 and conducted several scientific runs, collecting data with incrementally increased sensitivity in each run [1,3,4]. Although no gravitational waves were detected, these runs tested and refined key technologies, as well as provided a large amount of data characterizing the detectors. The next generation of detectors, referred to as the advanced LIGO detectors, are currently under construction and are expected to be operational by 2015 [1,2]. Major upgrades to lasers, optics, and seismic isolation/sensing will provide roughly a factor of ten improvement to sensitivity, which corresponds to a factor of 1000 in the observable volume of space and the number of detectable sources. Based on our current knowledge of potential astrophysical sources, the advanced LIGO detectors are expected to make routine gravitational-wave detections (see, for example, Ref. [5]) and will open the era of gravitational-wave astronomy.

The LIGO detector noise may be characterized by an approximately stationary component of colored Gaussian

*vaulin@ligo.mit.edu

noise, with the addition of short-duration non-Gaussian noise artifacts called "glitches." (Other noise sources, such as nonstationary lines and broadband nonstationarity, do not always fit neatly into this framework.) The stationary noise in the instrument is dominated by low-frequency seismic noise coupling to mirror motion, thermal noise in the mirrors and suspensions, 60 Hz power lines and harmonics, and shot noise. Sources of transient noise can include temporary seismic, acoustic, or magnetic disturbances, power transients, scattered light, dust crossing the beam, instabilities in the interferometer, channel saturations, and other complicated and often nonlinear effects. To monitor these disturbances and keep the instrument in a stable operating condition through active feedback, each detector records hundreds of auxiliary channels along with the gravitational-wave channel. These auxiliary channels keep track of important non–gravitational wave degrees of freedom in the interferometer, as well as information about the local environment. They are critical to understanding the state of the instrument at any particular time.

One of LIGO's main scientific goals is the detection of transient gravitational-wave signals, which can come from the coalescence of a compact binary or core-collapse supernova, among other astrophysical sources [6]. The presence of glitches is problematic for searches targeting these signals because glitches can be easily confused with transient gravitational-wave signals. They dominate background at moderate and high signal-to-noise ratios, where Gaussian noise contribution is completely negligible, making the detection of most realistic transient gravitational-wave signals with a single LIGO detector alone virtually impossible. The primary method to distinguish a real gravitational-wave transient from an instrumental artifact is to check that a signal appears in two or more spatially separated detectors. While this coincidence requirement is extremely effective, a high rate of glitches means that the accidental coincidence of noise transients across multiple detectors still dominates the search background, resulting in weaker upper limits and making the confident detection of real signals challenging [7]. Even in the case of the search for gravitational waves from binary neutron stars—in which the waveforms are well modeled and powerful signal consistency tests are employed to reject glitches—the volume search sensitivity is 30% less than what it could be in the presence of only Gaussian noise [8]. The problem is most severe in the searches for transients with poorly modeled or little identifying waveform structure, such as generic gravitational-wave bursts or intermediate-mass binary black-hole coalescence, which spend only a short amount of time (a few cycles) in the LIGO sensitive band.

For instance, using the KleineWelle analysis algorithm [9] as a proxy for a generic search for gravitational-wave bursts, one finds that the rate of moderately significant Gaussian noise fluctuations (with $\rho_{\mathrm{Gauss}} \geq 15$) is of order 10^{-3} Hz in a single detector. The rate of such fluctuations occurring at two detectors within 10 ms (the light travel time between two LIGO detectors) of each other and characterized by a similar central frequency is 10^{-3} Hz $\times$ 10^{-3} Hz $\times$ 10^{-2} s $\times$ $10^{-3} \approx 10^{-11}$ Hz, making them approximately one-in-3000-years events.[1] Taking this as a reasonable threshold for detection, one finds [e.g. from Fig. 5(b) showing the distribution of glitches] that in the presence of glitches bursts with a significance of $\rho_{\mathrm{glitch}} \geq 1000$ occur at similar rate, 10^{-3} Hz. Given that the ratio of burst significances is equal to the ratio of their energies, it is inversely proportional to the square of the distances to the astrophysical sources ($\rho_{\mathrm{Gauss}}/\rho_{\mathrm{glitch}} \approx E_{\mathrm{Gauss}}/E_{\mathrm{glitch}} = D^2_{\mathrm{glitch}}/D^2_{\mathrm{Gauss}}$). The sensitivity of the burst search is degraded by glitches by a factor of $D_{\mathrm{Gauss}}/D_{\mathrm{glitch}} \approx (1000/15)^{1/2} \approx 8$ relative to the same search in Gaussian noise. The number of astrophysical sources grows as distance cubed, so this reduction in sensitivity is very detrimental to searches for generic bursts of gravitational radiation. While these order-of-magnitude estimates correspond to gravitational-wave signals of unknown form, they give a sense of the severity of the problem glitches impose.

While the precise noise characteristics in the advanced detectors will be different from those of the initial LIGO, glitch sources for future data will exist and the detection problem for short-duration signals will persist. Thus, it is critical to develop data analysis methods for the robust identification of glitches in LIGO data. Many algorithms have been developed to look for glitches. These algorithms typically involve generating a statistic that measures the pairwise correlation between glitches from a single auxiliary channel and glitches from the gravitational-wave channel. In particular, algorithms have used the *use percentage* (ratio of gravitational-wave channel glitches removed to auxiliary glitches used) [10,11], the *veto efficiency* (fraction of gravitational-wave channel glitches removed), the *fractional deadtime* (the fraction of analysis time removed by vetoes), *veto significance* (the probability of observing at least as many coincident gravitational-wave channel and auxiliary glitches assuming two uncorrelated Poisson processes) [12], and the ratio of *veto efficiency to fractional deadtime* [13]. Other algorithms include those in Refs. [7,14–17]. Machine-learning algorithms (MLA)s are distinct from these other algorithms in that they can consider information from many different auxiliary channels simultaneously, rather than assuming a pairwise correlation or some other restriction on the glitch-coupling mechanism.

We use the ordered veto list (OVL) algorithm as a benchmark for our investigations [13]. OVL has been

[1]We require that central frequencies of the bursts in two detectors coincide to within 1 Hz. Given the detector's approximate bandwidth of 1 kHz, this leads to a probability of chance coincidence of 10^{-3}.

used in recent LIGO science runs as one of the primary glitch-detection algorithms. In particular, an earlier version of OVL described in Ref. [18] was used during LIGO's fifth science run [19]. OVL attempts to measure the degree of likelihood that a gravitational-wave candidate can be associated with a transient instrumental disturbance found in one of the many auxiliary channels using the ratio of efficiency to fractional deadtime.

Glitches are induced by the detector's environment, noise in the detector subsystems, or a combination thereof. These sources may appear in the auxiliary channels as well. In order to avoid potential bias we use a subset of auxiliary channels shown to be insensitive to test signals introduced in the nominal gravitational-wave channel. This subset is generated through hardware injections at the detectors [12]. The hardware injections involve driving the test masses through magnetic couplings[2] with an expected gravitational-wave signal and searching for evidence of that signal in auxiliary channels. If the signal does not systematically appear in an auxiliary channel, that channel is deemed ''safe'' and we include it in our analysis. By analyzing information from these auxiliary channels, one may be able to distinguish glitches from genuine gravitational-wave signals and ideally establish their cause. The main difficulty in such an analysis is processing the information from hundreds of channels which may manifest nontrivial correlations between themselves when they respond to an instrumental disturbance. Given the high dimensionality and the absence of reliable models for noise and couplings between auxiliary channels, traditional computational methods are not well suited to this problem. On the other hand, MLAs have been used to solve problems like this since the 1970s in other fields, such as computer science, biology, and finance.

This paper presents the use of MLAs for the purpose of glitch identification in gravitational-wave detectors. The main goal of the paper is to establish the feasibility of applying MLAs in the context of the LIGO detectors. We consider three well-known algorithms: the artificial neural network (ANN), the support vector machine (SVM), and the random forest (RF). We explore their properties and test their performance by analyzing data from past scientific runs. Based on these tests, we discuss the prospects for using MLAs for glitch identification in the advanced LIGO detectors.

This paper is organized as follows. In Sec. II, we describe the process for reducing raw time-series data and preparing feature vectors for the MLA classifiers. This is followed by a general formulation of the glitch detection problem in Sec. III. Then, in Sec. IV, we briefly describe the classifiers' algorithms. Training and testing of the classifiers is discussed in Sec. V. Finally, we evaluate and compare the classifiers' performances using the standard receiver operating characteristic (ROC) curves in Sec. VI and investigate various ways of combining classifiers in Sec. VII. In Appendix B, we explore several optimization criteria used by the classifiers and verify their theoretical consistency.

[2]In advanced LIGO detectors, test masses will be driven via electrostatic actuation.

II. DATA PREPARATION

We use data taken by the 4 km arm detector at Hanford, WA (H1) during LIGO's fourth science run (S4: 24 February–24 March 2005), and data taken by the 4 km arm detector at Livingston, LA (L1) during one week (28 May–4 June 2010) of LIGO's sixth science run (S6: 7 July 2009–20 October 2010). Hereafter we refer to these data sets as the S4 and the S6 data.

In the time between the fourth and the sixth science runs, the detectors underwent major commissioning and improvements to their sensitivity. Thus, while the H1 and L1 detectors are identical by design, the data taken by H1 during S4 and the data taken by L1 during S6 are quite different. These data sets represent evolutionary changes in both the detector-noise power spectral density and the non-Gaussian transient artifacts. Differences in the detectors' environments due to their distant geographical locations add another degree of freedom. Processing data from detectors separated in time and location allows us to determine how adaptable and robust these analysis algorithms are. This is important when extrapolating their performance to advanced detectors.

Classification, or the separation of input data into various categories, is one of the MLAs' main uses; thus, they are often referred to as classifiers. We have two categories of data: glitches (Class 1) and ''clean'' data (Class 0). If one was to perform a search for gravitational-wave transient signals, the first category, glitches, would generally be identified as candidate transient events and considered false alarms. The second category, ''clean'' data, contain only Gaussian detector noise in the gravitational-wave channel. A true gravitational-wave signal, when it arrives at the detector, is superposed on the Gaussian detector noise. If the signal's amplitude is high enough, it also would be identified by the search algorithm as a candidate transient event. Since it is a genuine gravitational-wave transient, it would constitute an actual detection, as opposed to glitches which act as noise. Hereafter we refer to such candidate gravitational-wave transients, either genuine gravitational-wave transients or glitches, as transient events or simply as events.

We characterize a transient event in either class by information from the detector's auxiliary channels. Importantly, we record the same information for both classes of events. Each channel records a time series measuring some non-gravitational wave degree of freedom, either in the detector or its environment. We first reduce the time series to a set of non-Gaussian transients using the

KleineWelle analysis algorithm [9], which finds clusters of excess signal energy in the dyadic wavelet domain. The detected transients are ranked by their statistical significance, ρ, defined as the negative logarithm of the probability that a random cluster of wavelet coefficients subject to Gaussian noise would contain the same or greater signal energy,

$$\rho = -\ln P(E_{\text{random}} \geq E_{\text{observed}}). \quad (1)$$

The MLA classifiers use the properties of auxiliary-channel transients coincident in time with the gravitational-wave channel event to classify the gravitational-wave event. Given an event in the gravitational-wave channel at time t, we build a feature vector x out of the nearby auxiliary-channel transients. Each channel contributes five features:

(i) ρ: The significance of the single loudest transient in that auxiliary channel within ± 100 ms of t.
(ii) Δt: The difference between t and the central time corresponding to the auxiliary-channel transient.
(iii) d: The duration of the auxiliary-channel transient.
(iv) f: The central frequency of the auxiliary-channel transient.
(v) n: The number of wavelet coefficients clustered to form the auxiliary-channel transient (a measure of time-frequency volume).

The KleineWelle triggers are recorded for transients with significance $\rho \geq 15$; below this threshold there is substantial contribution from random (uncorrelated) Gaussian noise which is uninformative. If no such auxiliary transient is found within 100 ms of t, the five fields for that channel are set to default values of zero. The 100-ms window covers most transient coupling timescales identified by previous studies [12]. However, there is no guarantee that this window is an optimal choice, or that it should be the same for all auxiliary channels. In total, we analyze 250 (162) auxiliary channels from S6 (S4) data, resulting in 1250 (810) dimensions for the auxiliary feature vector, x. In addition, we record certain bookkeeping information about the original gravitational-wave channel event, the state of nearby non-Gaussian transients in the gravitational-wave channel, and other information about data quality. These values are stripped before classifier training and evaluation so that we train the classifiers on only information contained in the auxiliary features.

The set of ''glitch'' (Class 1) samples, $\{x\}_1$, is generated by running KleineWelle over the gravitational-wave channel from one of the LIGO detectors. This set of non-Gaussian transients from the gravitational-wave channel can, in principle, contain true gravitational waves. However, prior to the coincidence requirement, they are overwhelmingly dominated by noise artifacts. Even for the most sensitive data set (S6), the expected rate of detectable gravitational-wave transients from known astrophysical sources is extremely low ($\sim 10^{-9}$ Hz [5]) with respect to the rate of single-detector noise transients (~ 0.1 Hz). Should there be a significant contribution of real gravitational waves in our single-detector glitch sample, the effect would be a reduction of training quality as the gravitational waves provide no useful correlations with auxiliary channels across the disjoint training and evaluation data sets. For the advanced LIGO detectors, it may be appropriate to remove coincident gravitational-wave candidates from the glitch training samples to avoid contamination from detectable gravitational-wave events. In both classifiers' training and performance evaluation, we treat all KleineWelle transients from the gravitational-wave channel as artifacts. In total, we identify 2832 (16 204) noise transients above a nominal significance threshold of $\rho \geq 35$ from the Livingston L1 (Hanford H1) detector during one week of the S6 (four weeks of the S4) science run. At this threshold, effectively all detected transients are non-Gaussian outliers (Fig. 5), and contribute to the bulk of non-Gaussian background in searches for gravitational-wave transients. The central time from each event is used to trigger feature-vector generation, so that $\{x\}_1$ is a set of 2832 (16 204) sample vectors, each described by 1250 (810) features derived from coincident auxiliary-channel information. The samples are most representative of the background in gravitational-wave burst searches which generally target short, unmodeled signals.

''Clean'' (Class 0) samples, $\{x\}_0$, are formed by first generating 10^5 randomly distributed times to estimate the auxiliary states at times when no glitch is present. To further aid in distinguishing times when there is no disturbance, we exclude Class 0 samples which fall within ± 100 ms of a Class 1 sample. As with Class 1, the full set of Class 0 samples $\{x\}_0$ is built from auxiliary-channel information nearby each randomly selected time.

III. GENERAL FORMULATION OF THE DETECTION PROBLEM

The data analysis problem which we address here can be formulated as the robust identification of transient artifacts (glitches) in the gravitational-wave channel based on the information contained solely in the safe auxiliary detector channels. Clearly, the solution to this problem is directly related to the solution to the ultimate problem of the robust detection and classification of gravitational-wave transients in LIGO data. The identification of glitches will reduce the non-Gaussian background and improve the sensitivity of gravitational-wave searches. We leave the question of how the results of our current analysis of the auxiliary channels can be incorporated into the search for transient gravitational waves to future work.

For a given transient event in the gravitational-wave channel, we construct a feature vector of auxiliary information, x, following the procedure outlined in Sec. II. Our detection problem reduces to binary prediction on whether this transient is a glitch (Class 1) or a clean sample (Class 0) based on x and only x. In feature space, $x \in V_d$, this binary

decision can be mapped into identifying domains for Class 1 events, V_1, and Class 0 events, V_0. We call the surface separating the two regions the decision surface. Unless the two classes are perfectly separable, which is typically not the case, there is a nonzero probability for an event of one class to occur in a domain identified with a different class. In this case, one would like to find an optimal decision surface separating two classes in such a way that we maximize the probability of finding events of Class 1 in V_1 at a fixed probability of miscategorizing events from Class 0 in V_1. This essentially minimizes the probability of incorrectly classifying events. P_1 represents the probability of glitch detection, which we also call glitch *detection efficiency*, and P_0 is called the *false-alarm probability*. This optimization principle is often referred to as the Neyman-Pearson criteria [20].

The probability of detection and the probability of false alarm can be expressed in terms of conditional probability density functions for the feature vector, x,

$$P_1 = \int_{V_d} \Theta(f(x) - F^*) p(x|1) p(1) \mathrm{d}x, \tag{2a}$$

$$P_0 = \int_{V_d} \Theta(f(x) - F^*) p(x|0) p(0) \mathrm{d}x. \tag{2b}$$

Here $p(x|1)$ and $p(x|0)$ define probability density functions for the feature vector in the presence and absence of a glitch in the gravitational-wave data, respectively. $p(1)$ and $p(0)$ are prior probabilities for having a glitch or clean data, related to one another via $p(1) + p(0) = 1$. The Heaviside step function $\Theta(f(x) - F^*)$ defines the region V_1 which signifies a glitch in the gravitational-wave data, and $f(x) = F^*$ defines the decision surface. F^* is a threshold parameter, which corresponds to a specific value of the probability of false alarm through Eq. (2b).

The optimal decision surface is found by maximizing the functional

$$S[f(x)] = P_1[f(x)] - l_0(P_0[f(x)] - P_0^*), \tag{3}$$

where P_0^* is a tolerable value for the probability of false alarm and l_0 is a Lagrange multiplier. Setting the variation of this functional with respect to $f(x)$ to zero leads to a condition for the points on the decision surface,

$$\frac{p(x|1)p(1)}{p(x|0)p(0)} = \text{Constant}. \tag{4}$$

The ratio of conditional probability density functions,

$$\Lambda(x) \equiv \frac{p(x|1)}{p(x|0)}, \tag{5}$$

is called the likelihood ratio (sometimes also referred to as the Bayes factor). The Constant in the optimality condition (4) does not carry any special meaning, and the condition can be satisfied if the decision surface is defined as the surface of constant likelihood ratio [21],

$$f(x) = \Lambda(x) = F^*, \tag{6}$$

with F^* set by the probability of false alarm, P_0^*, through Eq. (2b). Actually, the decision surface can be defined by any monotonic function of the likelihood ratio with a trivial redefinition of F^*. There is a unique decision surface for each value of $P_0^* \in [0, 1]$, and we can label decision surfaces by their corresponding values of P_0^*. See Appendix A and Fig. 9 for an illustration of the concept of the likelihood-ratio decision surfaces in a toy example.

The optimization of Eq. (3) maximizes the detection probability, $P_1 \rightarrow P_1^{\mathrm{OPT}}$, for every value of the probability of false alarm, $P_0 = P_0^*$. The curve $P_1^{\mathrm{OPT}}(P_0)$ is called the ROC curve. It is a standard measure of any detection algorithm's performance. We can think of optimizing Eq. (3) as maximizing the area under the ROC curve. For further details on the use of the likelihood ratio in the gravitational-wave searches, see Refs. [21,22].

Finding the optimal decision surfaces by direct estimation of the conditional probability density functions, $p(x|1)$ and $p(x|0)$, is an extremely difficult task if the feature vector contains more than a few dimensions. For high-dimensional problems, when no parametric model for these probability distributions is known and with a limited number of experimental samples that can be used to estimate these probability density functions, one has to resort to some other method. MLAs are well suited for these detection problems.

In this paper, we consider three popular MLAs: ANN, SVM and RF. They differ significantly in their underlying algorithms and their approaches to classification. This allows us to investigate the applicability of different types of MLAs to glitch identification in the gravitational-wave data. However, all MLAs require training samples of events from both Class 1 and Class 0. The MLA classifiers use the training sets to find an optimal classification scheme or decision surface. In the limit of infinitely many samples and unlimited computational resources, different classifiers should recover the same theoretical result: the decision surface defined by the constant likelihood ratio (6). To this end, it is critical that classifiers are trained and optimized using criteria consistent with this result. In Appendix B, we explore several standard optimization criteria and derive the decision surfaces they generate in this theoretical limit. We find that all of these criteria lead to a decision surface with a constant likelihood ratio. In particular, this is true for the fraction of correctly classified events and the Gini index criteria that are used by ANN/SVM and RF, respectively.

While all classifiers we investigate here should find the same optimal solution with sufficient data, in practice, the algorithms are limited by the finite number of samples in the training sets and by computational cost. The classifiers have to handle a large number of dimensions efficiently, many of which might be redundant or irrelevant. By no means is it clear that the MLA classifiers will perform well

under such conditions. It is our goal to demonstrate that they do.

We evaluate their performance by computing ROC curves.[3] These curves, which map the classifiers' overall efficiencies, are objective and can be directly compared. In addition to comparing the MLA classifiers to one another, we benchmark them using ROC curves from the OVL algorithm [13]. This method constructs segments of data to be vetoed using a hard time window and a threshold on the significance of transients in the auxiliary channels. The veto segments are constructed separately for different auxiliary channels and are applied in the order of decreasing correlation with the gravitational-wave events. By construction, only pairwise correlations between a single auxiliary channel and the gravitational-wave channel are considered by the OVL algorithm. These results have a straightforward interpretation and provide a good sanity check.

In order to make the classifier comparison as fair as possible, we train and evaluate their performances using exactly the same data. Furthermore, we use a round-robin procedure for the training-evaluation cycle, which allows us to use all available glitch and clean samples. Samples are randomized and separated into ten equal subsets. To classify events in the kth subset, we use classifiers trained on all but the kth subset. In this way, we ensure that training and evaluation are done with disjoint sets so that any overtraining that might occur does not bias our results.

An MLA classifier's output is called a rank, $r_{\mathrm{MLA}} \in [0, 1]$, and a separate rank is assigned to each glitch and clean sample. Higher ranks generally denote a higher confidence that the event is a glitch. A threshold on this rank maps to the probability of false alarm, P_0, by computing the fraction of clean samples with greater or equal rank. Similarly, the probability of detection or efficiency, P_1, is estimated by computing the fraction of glitches with greater or equal rank. Essentially, we parametrically define the ROC curve, $P_1^{\mathrm{OPT}}(P_0)$, with a threshold on the classifier's rank. Synchronous training and evaluation of the classifiers allow us to perform a fair comparison and to investigate various ways of combining the outputs of different classifiers. We discuss our findings in detail in Secs. VI and VII.

[3]More traditional veto approaches to data quality in gravitational-wave searches use another measure of veto quality. For a given veto configuration consisting of a list of disjoint segments of data, the fractional "deadtime" is computed from the sum of the durations of all data segments to be vetoed. While not precisely the same, this quantity is related to the probability of false alarm, P_0, which accounts only for the fraction of clean data removed from the search. For a typical rate of glitches of $\sim$0.1 Hz, the two measures are almost identical in the most relevant region of $P_0 \leq 10^{-2}$. Thus, in that interval the ROC curves of this paper can be directly compared to the often-used figure of merit, efficiency vs fractional deadtime. See for example Ref. [12].

IV. OVERVIEW OF THE MACHINE-LEARNING ALGORITHMS

In this section, we give a short overview of the basic properties of the classifiers and the tuning procedures used to determine the best-performing configurations for each classifier. Throughout this section, we will use the notation x_i where $i = 1, 2, \ldots N$ to denote the set of N sample feature vectors. Similarly, y_i will denote the actual class associated with the the ith sample feature vector, either Class 0 or Class 1. Predictions about a feature vector's class will be denoted by $y(x_i)$.

A. Artificial neural network

ANNs employ a machine-learning technique based on simulating the data processing in human brains and mimicking biological neural networks [23,24]. As is well known, the human brain is composed of a tremendous number of interconnected neurons, with each cell performing a simple task (responding to an input stimulus). However, when a large number of neurons form a complicated network structure, they can perform complex tasks such as speech recognition and decision making.

A single neuron is composed of dendrites, a cell body, and an axon. When dendrites receive an external stimulus from other neurons, the cell body computes the signal. When the total strength of the stimulus is greater than the synapse threshold, the neuron is fired and sends an electrochemical signal to other neurons through the axon. This process can be implemented with a simple mathematical model including nodes, a network topology and learning rules adopted to a specific data-processing task. Nodes are characterized by their number of inputs and outputs (essentially how many other nodes they talk to), and by the connecting weights associated with each input and output. The network topology is defined by the connections between the neurons (nodes). The learning rules prescribe how the connecting weights are initialized and evolve.

There are a large number of ANN models with different topologies. For our purpose, we choose to implement the multilayered perceptron (MLP) model, which is one of the most widely used models. The MLP model has input and output layers as well as a few hidden layers in between. The input vector for the input layer is the auxiliary feature vector, x, while the input for hidden layers and the output layer is a combination of the output from nodes in the previous layer. We will call these intermediate vectors z to distinguish them from the full feature vector. Each layer has several neurons which are connected to the neurons in the adjacent layers with individual connecting weights. The initial structure—the number of layers, neurons, and the initial value of connecting weights—is chosen by hand and/or through an optimization scheme such as a genetic algorithm (GA).

When a neuron's input channels receive an external signal exceeding the threshold value set by an activation

function, the neuron is fired. This process can be expressed mathematically as

$$y(z) = f(w \cdot z + b), \tag{7}$$

where $y(z)$ is the output, z is an input vector, w are connecting weights, f is an activation function, and b is a bias. One may choose the activation function to be either the identity function, the ramp function, the step function, or a sigmoid function. We use the sigmoid function defined by

$$f(w \cdot z + b) = (1 + \mathrm{e}^{-2s(w \cdot z + b)})^{-1}. \tag{8}$$

We set the activation steepness $s = 0.5$ in hidden layers and $s = 0.9$ at the output layer. There is a single neuron at the output layer, and the value of that neuron's activation function is used as the ANN's rank, r_{ANN}.

The topological parameters determine the number of connection weights, which must be sufficiently large that the ANN has enough degrees of freedom to classify a given datum. The network's flexibility depends on the number of connection weights and should be matched with the size of the training sets and the input data's dimensionality. In our work, the numbers of layers and neurons are chosen so that the total number of connection weights is on the order of 10^4 when using the entire data set. In order to avoid overtraining, we decrease the number of layers and neurons in the runs in which either the dimensionality or the number of training samples are reduced.

The learning scheme finds the optimal connection weights, w, in each layer. In this paper we use the improved version of the resilient back-propagation algorithm [25], which minimizes the error between the output value, $y(x_i)$, and the known value, y_i. In this algorithm, the increase (decrease) factor and the minimum (maximum) step size determine the change in the connection weights, Δw, at each iteration during the training. In our work, the default values in the Fast Artificial Neural Network library [26] are used for all parameters except the increase factor, which is set to $\eta^+ = 1.001$. The same learning rules were used in all runs. We should note that ANNs can be optimized in an alternative way, via a GA or other similar methods. When using a GA, a combined optimization algorithm for topology, feature and weight selection can be applied to improve the performance of an ANN [27–31]. We explore these options in a separate publication [32].

In addition to choosing the ANN configuration parameters, we found that the ANN requires data preprocessing. The input variables with high absolute values have a greater effect on the output values, and thus we normalize all components of the feature vector, x, to the range [0,1]. To better resolve small Δt values, Δt is transformed via a logarithmic function before normalization,

$$\Delta t' = -\mathrm{sign}(\Delta t) \log |\Delta t|. \tag{9}$$

This transformation improves the ANN's ability to identify glitches, which tend to have smaller values of Δt. One can find a more detailed description of the procedure for tuning the ANN configuration parameters in Ref. [32].

B. Support vector machines

The SVM is an MLA for binary classification on a vector space [33,34]. It finds the optimal hyperplane that separates the two classes of training samples. This hyperplane is then used as the decision surface in feature space, and classifies events depending on which side of the hyperplane they fall.

As before, let $\{(x_i, y_i)|i = 1, 2, \ldots N\}$ be the training data set, where x_i is the feature vector of auxiliary transient information near time t_i, and $y_i \in \{1, -1\}$ is a label that marks the sample as either Class 1 or Class 0, respectively. Then assume that the training set is separable by a hyperplane $w \cdot x - b = 0$, where w is the normal vector to the hyperplane and b is the bias. Then the training samples with $y_i = 1$ satisfy the condition $w \cdot x_i - b \geq 1$, and the training samples with $y_i = -1$ satisfy the condition $w \cdot x_i - b \leq -1$. SVM uses a quadratic programming method to find the w and b that maximize the margin between the hyperplanes $w \cdot x - b = 1$ and $w \cdot x - b = -1$.

If the training samples are not separable in the original feature space, V_{d}, SVM uses a mapping, $\phi(x)$, into a higher-dimensional vector space, V_ϕ, in which two classes of events can be separated. The decision hyperplane in V_ϕ corresponds to a nonlinear surface in the original space, V_{d}. Thus, mapping the problem into a higher-dimensional space allows SVM to consider nonlinear decision surfaces. The dimensionality of V_ϕ grows exponentially with the degree of the nonlinearity of the decision surfaces in V_{d}. As a result, SVM cannot consider arbitrary decision surfaces and usually has to deal with nonseparable populations. If the training samples are not separable after mapping, a penalty parameter, C, is introduced to weight the training error. Finding the optimal hyperplane is reduced to the following quadratic programming problem:

$$\min_{w,b,\xi} \left(\frac{1}{2} w \cdot w + C \sum_{i=1}^{N} \xi_i \right), \tag{10a}$$

$$\text{subject to } yi \cdot (w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \tag{10b}$$

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, N. \tag{10c}$$

When the solution is found, SVM classifies a sample x by the decision function,

$$y(x_i) = \mathrm{sign}(w \cdot \phi(x_i) + b). \tag{11}$$

In solving the quadratic programming problem, the function ϕ is not explicitly needed. It is sufficient to specify $\phi(x_i) \cdot \phi(x_j)$. The function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is called the kernel function. The form of the kernel function implicitly defines the family of surfaces in V_{d} over which SVM is optimizing. In this study we use the radial basis function as the kernel function. It is defined as

$$K(x_i, x_j) = \exp\{-\gamma \|x_i - x_j\|^2\}, \quad (12)$$

where γ is a tunable parameter.

The SVM algorithm was implemented by using the open source package libsvm [35]. As part of this package, the kernel function parameter, γ, and the penalty parameter, C, are tuned in order to achieve the best performance for a specific application. The best parameters (C and γ) are selected through an exhaustive search. For each pair of parameters ($\log C$, $\log \gamma$) on a grid, we calculate a figure of merit. The parameters with the best figure of merit are then used for classifying events. The default figure of merit in libsvm is the accuracy (fraction of correctly classified events). However, we replace it with a figure of merit better adapted to glitch detection. Instead of using the accuracy, our code calculates the area under the estimated ROC curve $[P_1^{\mathrm{EST}}(P_0)]$ in the interval of the probability of false alarm $P_0 \in [0.001, 0.05]$ on a log-linear scale ($[0.001, 0.05]$ is a range of acceptable probability of false alarm for practical glitch detection),

$$\text{figure of merit} = \int_{P_0=0.001}^{P_0=0.05} \mathrm{d}(\ln P_0) P_1^{\mathrm{EST}}(P_0). \quad (13)$$

Performing an exhaustive search for the best SVM parameters is computationally expensive. We can speed up this tuning process by exploiting the fact that the tuning time grows nonlinearly with the training sample size. By using smaller training sets, we can reduce the total time spent determining the optimal parameters. We randomly selected p subsets of vectors from the training set, with each subset 10 times smaller than the full training set. The best pair of the SVM parameters for each of the p subsets was then calculated, with each subset optimization running on a single CPU core. This gives p sets of best parameters, calculated in parallel. The parameters C and γ that are selected the most often were then chosen as the final best SVM parameters. This modified parameter optimization algorithm was applied to various training sets (described in Sec. V). We found that the optimal SVM parameters do not depend on the training set. We therefore use the same parameters for all calculations reported in this paper ($C = 8$ and $\gamma = 0.0078125$).

In its standard configuration, SVM classifies samples by a discrete label, $y \in \{1, -1\}$. However, the libsvm package can provide a probability-based version of Eq. (11) that yields continuous values, $r_{\mathrm{SVM}} \in [0, 1]$ [36]. We use these continuous values as the output of the SVM classifier.

C. Random forest technology

RF technology [37,38] improves upon the classical decision tree [39,40] approach to classification. The classifying decision tree performs a series of binary splits on any/all of the dimensions of the feature vector, x, that describes an event. The goal is to distribute events into groups consisting of only a single class. In a machine-learning context, the decision tree is formed by training it on a set of events of known class. During the training, a series of splits are made, where each split chooses the dimension and its threshold that optimizes a certain criterion, such as the fraction of correctly classified training events or the Gini index, defined by Eq. (B7). Splitting stops once no split can further improve the optimization criterion or once the limit on the minimum number of events allowed on a branch (the furthest reaches of a decision tree) is reached; at this point the branch becomes a leaf. Once a tree is formed, an event of unknown class is fed into the tree, and depending on its feature vector, x, it will be labeled as either Class 0 or Class 1. However, a single decision tree can be a victim to both false minima and overtraining. To guard against this, we create a forest of decision trees and average over their answers; this results in a continuous ranking, $r_{\mathrm{RF}} \in [0, 1]$, rather than a binary classification, as events can be placed on a continuum between Class 0 and Class 1.

Each decision tree in the forest is trained on a bootstrap replica of the original training set. If the original training set has N events, each bootstrap replica will also have N events, which are chosen randomly with replacement, meaning any given event may be picked more than once. Therefore, each tree gets a different set of training events. To further avoid false minima, random forest technology chooses a different random subset of the features to be available for splitting at each node. This ensures that a peculiarity in a particular dimension does not dominate the decision-making process.

We use the STATPATTERNRECOGNITION software package's [41] implementation of RF. The key input parameters are the number of trees in the forest, the number of features randomly selected for splitting at each tree node (branching point), the minimum number of samples on the terminal tree nodes (leaves) and the optimization criterion. To determine the best set of the RF parameters, we perform a search over a coarse grid in the parameter space, maximizing efficiency or the detection probability, P_1, at the probability of false alarm, $P_0 = 0.01$. We find that beyond a certain point, the RF efficiency grows very slowly with the number of trees and the number of features selected for splitting at the cost of a significant increase in running time during the training process. Taking this into account, we arrive at the following configuration, which we use in all runs: 100 trees in the forest, 64 different randomly chosen features at the branching points, a minimum of eight samples on a leaf, and the Gini index as the optimization criterion.

D. Ordered veto list algorithm

The OVL algorithm operates by looking for coincidences between the transients in gravitational-wave and auxiliary channels. Specifically, the transients identified in the auxiliary channel are used to construct a list of time segments. All transients in the gravitational-wave channel

occurring within these segments are removed from the list of transient gravitational-wave candidates. In effect, the data in these time segments are vetoed prior to any search for gravitational-waves.

The algorithm assumes that transients in certain auxiliary channels are more correlated with the glitches in the gravitational-wave channel and looks for a hierarchy of correlations between auxiliary and gravitational-wave glitches. Specifically, a series of veto configurations is created, corresponding to different auxiliary channels, the time windows around transients and the threshold on their significance. The ordered list corresponds to a list of these configurations, and veto configurations are applied to the data in order of decreasing correlation. For this study, the maximum time window is set to ± 100 ms to match the one we use to create auxiliary feature vectors for the MLA classifiers (Sec. II). Similarly, the lowest threshold on significance, ρ, is set to the auxiliary-channel nominal threshold of 15. For each channel, the number of possible veto configurations is equal to the number of unique combinations that can be constructed from the list of the time windows [± 25 ms, ± 50 ms, ± 100 ms] and the significance thresholds [15, 25, 30, 50, 100, 200, 400, 800, 1600].

Importantly, a segment removed by a veto configuration is not seen by later configurations. This prohibits duplicate vetoes and results in a measurement of the additional information contained in subsequent veto configurations. The performance of each configuration is evaluated and they are reranked accordingly. The OVL algorithm defines the veto-configuration rank, r_{OVL}, as the ratio of the fraction of gravitational-wave glitches removed to the fraction of analysis time removed. Repeated application of the algorithm produces an ordered list with the better-performing configurations appearing higher on the list.

Only some of the veto configurations make it to the final list. Those which perform poorly ($r_{\mathrm{OVL}} \leq 3$) are discarded. This is done in order to get rid of irrelevant or redundant channels and to speed up the algorithm's convergence. Typically, the OVL algorithm converges within less than ten iterations to a final ordered list. We find that only 47 out of 162 auxiliary channels in S4 data and 35 out of 250 auxiliary channels in S6 data appear on the final list. Below, we refer to this subset of channels as the ''OVL auxiliary channels.'' For a more detailed description of the OVL algorithm, see Ref. [13].

The procedure for optimizing the ordered list of veto configurations can be considered a training phase. An ordered list of veto configurations optimized for a given segment of data can be applied to another segment of data. Veto segments are generated based on the transients in the auxiliary channels and the list of configurations. The performance of the algorithm is evaluated by counting fractions of removed glitches and clean samples, and computing the ROC curve. As with MLA classifiers, we use the round-robin procedure for OVL's training-evaluation cycle.

V. TESTING THE ALGORITHMS' ROBUSTNESS

One of the main goals of this study is to establish if MLA methods can successfully identify transient instrumental and environmental artifacts in LIGO gravitational-wave data. The potential difficulty arises from high dimensionality and the fact that information from a large number of dimensions might be either redundant or irrelevant. Furthermore, the origin of a large fraction of glitches is unknown in the sense that their cause has not been pinpointed to a single instrumental or environmental source. In the absence of such deterministic knowledge, one has to monitor a large number of auxiliary channels and look for statistically significant correlations between transients in these channels and transients in the gravitational-wave channel. These correlations, in principle, may involve more than one auxiliary channel and may depend on the transients' parameters in an extremely complicated way. Additionally, new kinds of artifacts may arise if one of the detector subsystems begins to malfunction. Likewise, some auxiliary channels' coupling strengths to the gravitational-wave channel may be functions of the detector's state (e.g. optical cavity configuration and mirror alignment). Depending on the detector's state, the same disturbance witnessed by an auxiliary channel may or may not cause a glitch in the gravitational-wave channel. This information cannot be captured by the KleineWelle-derived parameters of the transients in the auxiliary channels alone and requires extending the current method. We leave this problem to future work.

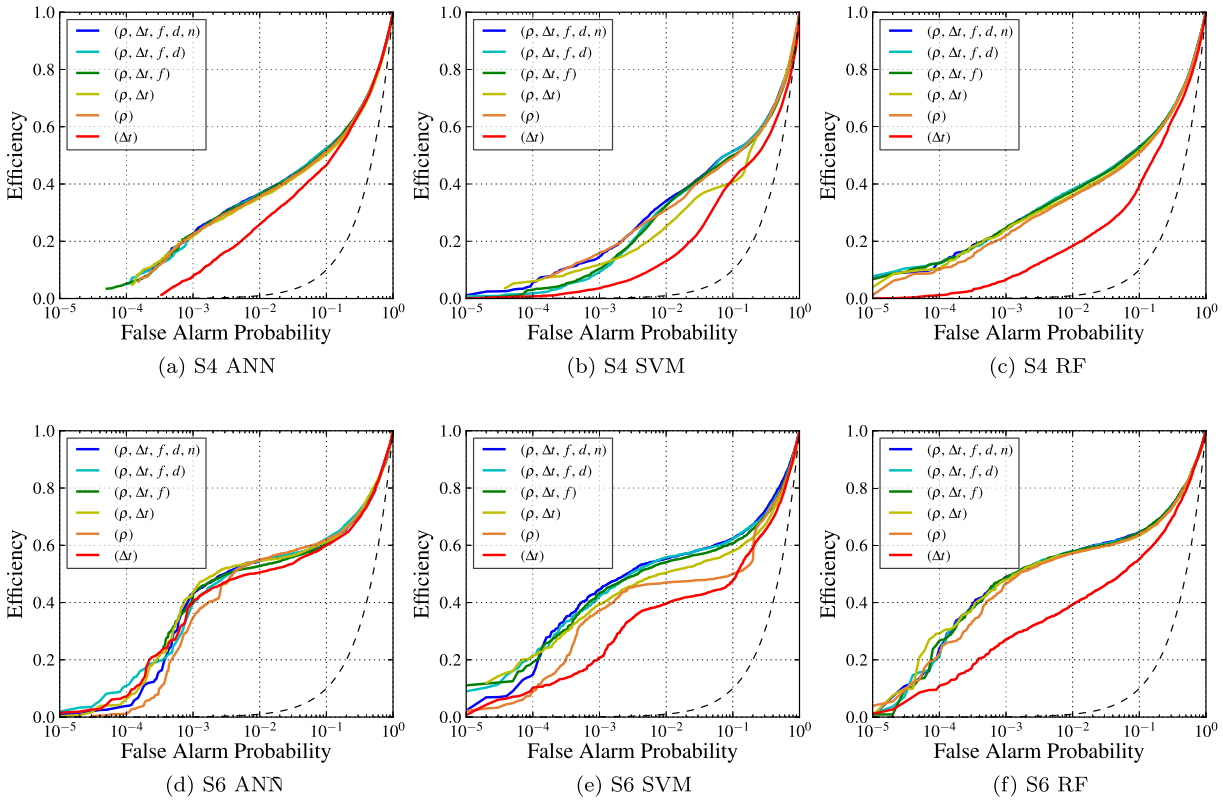
Because of the uncertainty in the types and locations of correlations, we include as many auxiliary channels and their transients' parameters as possible. However, this forces us to handle a large number of features, many of which might be either redundant or irrelevant. The MLA classifiers may be confused by the presence of these superfluous features and their performance may suffer. One can improve performance by reducing the number of features and keeping only those that are statistically significant. However, this requires preprocessing the input data and tuning, which may be extremely labor intensive. On the other hand, if the MLA classifier can ignore irrelevant dimensions automatically without a significant decrease in performance, it can be used as a robust analysis tool for real-time glitch identification and detector characterization. By efficiently processing information from all auxiliary channels, a classifier will be able to identify new artifacts and help to diagnose problems with the detector.

In order to determine our classifiers' robustness, we perform a series of runs in which we vary the dimensionality of the input data and evaluate the classifiers' performance. First, we investigate how their efficiency depends on which transient parameters are used. We expect that not all of the five parameters (ρ, Δt, f, d, n) are equally informative. Naively, ρ and Δt, reflecting the disturbance's amplitude in the auxiliary channel and its degree of coincidence with the

transient in the gravitational-wave channel, respectively, should be the most informative. Potentially, the frequency, f, duration, d, and the number of wavelet coefficients, n, may carry useful information if only certain auxiliary transients produce glitches. However, it is possible that these parameters are only correlated with the corresponding parameters of the gravitational-wave transient, which we do not incorporate in this analysis. Such correlations, even if not broadened by frequency-dependent transfer functions, would require analysis specialized to specific gravitational-wave signals and goes beyond the scope of this paper. We perform a generic analysis, not relying on the specific characteristics of the gravitational-wave transients.

Anticipating that some of the parameters could be irrelevant, we prepare several data sets by removing features from the list: $(\rho, \Delta t, f, d, n)$. We prepare these data sets for both S4 and S6 data and run each of the classifiers through the training-evaluation round-robin cycles described in Sec. III. We evaluate their performance by computing the ROC curves, shown in Fig. 1.

We note the following relative trends in the ROC curves for all classifiers. The omission of the transient's duration, d, and the number of wavelets, n, has virtually no effect on efficiency. The ROC curves are the same to within our error, which is less than $\pm 1\%$ for our efficiency measurement, based on the total number of glitch samples and the normal approximation for the binomial confidence interval, $\sqrt{P_1(1-P_1)/N}$. The omission of the frequency, f, slightly reduces the efficiency of SVM [Figs. 1(b) and 1(e)], but has no effect on either ANN or RF. A comparison between the ROC curves for the $(\rho, \Delta t)$, (ρ) and (Δt) data sets shows that while a transient's significance is the most informative parameter, including the time difference generally results in better overall performance. Of the three MLA classifiers, SVM seems to be the most sensitive to whether the time difference is used in addition to significance. RF, as it appears, relies primarily on significance, which is reflected in the poor performance of the (Δt)-only ROC curves in Figs. 1(c) and 1(f). The trend for ANN is not as clear. In S4 data, including timing does not change the ROC curve



(a) S4 ANN (b) S4 SVM (c) S4 RF

(d) S6 ANN (e) S6 SVM (f) S6 RF

FIG. 1 (color online). Varying sample features. We expect some of the five features recorded for each auxiliary channel to be more useful than others. To quantitatively demonstrate this, we train and evaluate our classifiers using subsets of our sample data, with each subset restricting the number of auxiliary features. We observe the general trend that the significance, ρ, and time difference, Δt, are the two most important features. Between these two, ρ appears to be marginally more important than Δt. On the other hand, the central frequency, f, the duration, d, and the number of wavelet coefficients in an event, n, all appear to have very little affect on the classifiers' performance. Importantly, our classifiers are not impaired by the presence of these superfluous features and appear to robustly reject irrelevant data without significant efficiency loss. The black dashed line represents a classifier based on random choice. Panels (a), (b), and (c) show the ROC curves with S4 data for ANN, SVM, and RF, respectively. Panels (d), (e), and (f) show the ROC curves with S6 data for ANN, SVM, and RF, respectively.

[Fig. 1(a)] while in S6 data it improves it [Fig. 1(d)]. Overall, we conclude that based on these tests, most if not all of the information about detected glitches is contained in the $(\rho, \Delta t)$ pair. At the same time, keeping irrelevant features does not seem to have a negative effect on our classifiers' performance.

The OVL algorithm, which we use as a benchmark, ranks and orders the auxiliary channels based on the strength of correlations between transient disturbances in the auxiliary channels and glitches in the gravitational-wave channel. The final list of OVL channels includes only a small subset of the available auxiliary channels: 47 (of 162) in S4 data and 35 (of 250) in S6 data. The rest of the channels do not show statistically significant correlations. It is possible that these channels contain no useful information for glitch identification, or that one has to include correlations involving multiple channels and/or other features to extract the useful information. In the former case, throwing out irrelevant channels will significantly decrease our problem's dimensionality and may improve the classifiers' efficiency. In the latter case, classifiers might be capable of using higher-order correlations to identify classes of glitches missed by OVL.

We prepare two sets of data to investigate these possibilities. In the first data set, we use only the OVL auxiliary channels and exclude information from all other channels. In the second data set, we further reduce the number of dimensions by using only ρ and Δt. We apply classifiers to both data sets, evaluate their performance and compare it to the run over the full data set (all channels and all features). Figure 2 shows the ROC curves computed for these test runs.

In both S4 and S6 data, the three curves for RF [Figs. 2(c) and 2(f)] lay on top of each other, demonstrating that this classifier's performance is not affected by the data reduction. ANN shows slight improvement in its performance for the maximally reduced data set in the S6 data [Fig. 2(d)], and no discernible change in the S4 data [Fig. 2(a)]. SVM exhibits the most variation of the three classifiers. While dropping the auxiliary channels not included in the OVL list has a very small effect on SVM's

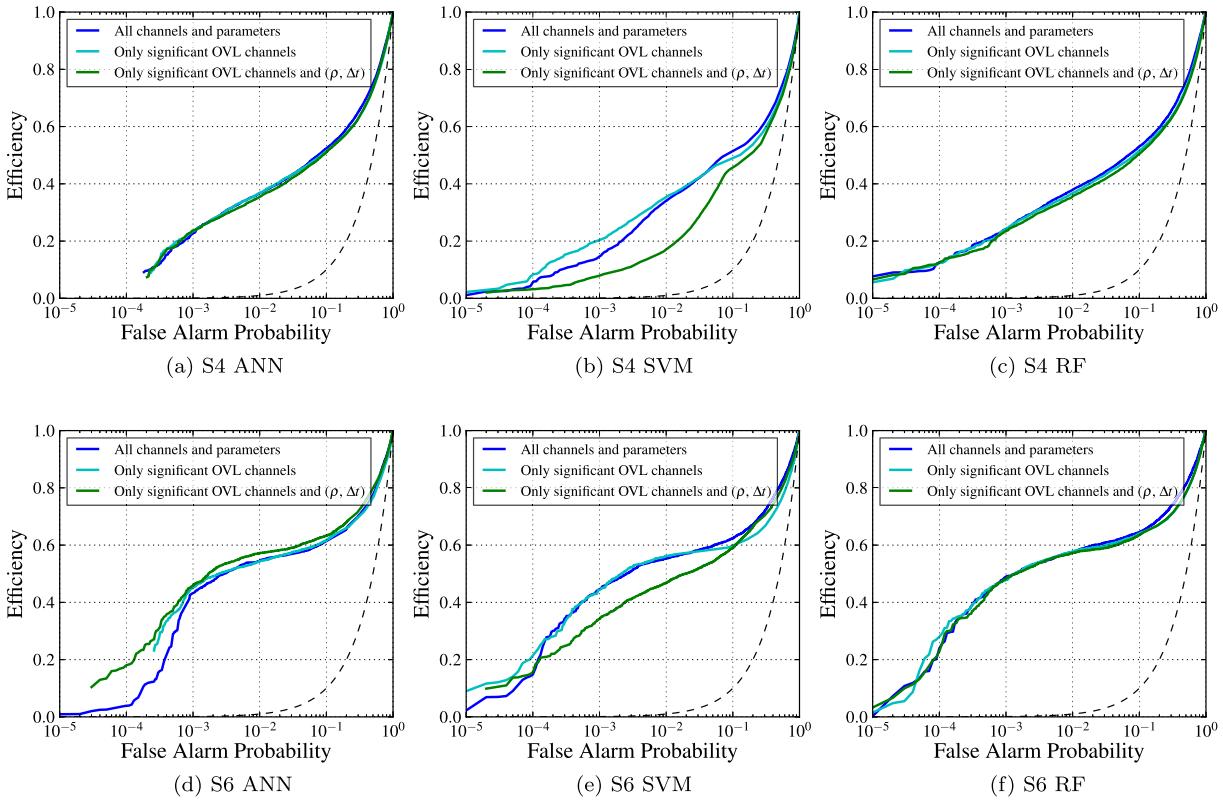


FIG. 2 (color online). Reducing the number of channels. One way to reduce the dimensionality of our feature space is to reduce the number of auxiliary channels used to create the feature vector. We use a subset of auxiliary channels identified by OVL as strongly correlated with glitches in the gravitational-wave channel (light blue). We notice that for the most part, there is not much efficiency loss when restricting the feature space in this way. This also means that very little information is extracted from the other auxiliary channels. The classifiers can reject extraneous channels and features without significant loss or gain of efficiency. We also restrict the feature vector to only include the significance, ρ, and the time difference, Δt, for the OVL auxiliary channels (green). Again, there is not much efficiency loss, suggesting that these are the important features and that the classifiers can robustly reject unimportant features automatically. The black dashed line represents a classifier based on random choice. Panels (a), (b), and (c) show the ROC curves with S4 data for ANN, SVM, and RF, respectively. Panels (d), (e), and (f) show the ROC curves with S6 data for ANN, SVM, and RF, respectively.

ROC curve, further data reduction leads to an efficiency loss [Figs. 2(b) and 2(e)]. Viewed together, the plots in Fig. 2 imply that, on the one hand, non-OVL channels can be safely dropped from the analysis, but on the other hand the presence of these uninformative channels does not reduce our classifiers' efficiency. This is reassuring. As previously mentioned, one would like to use these methods for real-time classification and detector diagnosis, in which case monitoring as many channels as possible allows us to identify new kinds of glitches and potential detector malfunctions. For example, an auxiliary channel that previously showed no sign of a problem may begin to witness glitches. If excluded from the analysis based on its previous irrelevance, the classifiers would not be able to identify glitches witnessed by this channel or warn of a problem.

Another way in which input data may influence a classifier's performance is by limiting the number of samples in the training set. Theoretically, the larger the training sets, the more accurate a classifier's prediction. However, larger training sets come with a much higher computational cost and longer training times. In our case, the size of the glitch training set is limited by the glitch rate in the gravitational-wave channel and the duration of the detector's run. We remind the reader that we use four weeks from the S4 run from the H1 detector and one week from the S6 run from the L1 detector to collect glitch samples. One would like to use shorter segments to better capture the nonstationarity of the detector's behavior. However, having too few glitch samples would not provide a classifier with enough information. Ultimately, the size of the glitch training set will have to be tuned based on the detector's behavior. We have much more control over the size of the clean training set, which is based on completely random times when the detector was operating in the science mode. In our simulations, we start with 10^5 clean samples, but it might be possible to reduce this number without loss of efficiency, thereby speeding up classifier training.

We test how the classifiers' performance is affected by the size of the clean training set in a series of runs in which we gradually reduce the number of clean samples available. Runs with 100%, 75%, 50%, and 25% of the total number of clean samples available for training are supplemented by a run in which the number of clean training samples is equal to the number of glitch training samples (16% in S4 data and 2.5% in S6 data). In addition, we performed one run in which we reduced the number of glitch training samples by half, but kept 100% of the clean training samples. While not completely exhaustive, we believe these runs provide us with enough information to describe the classifiers' behavior. In all of these runs, we use all available samples for evaluation, employing the round-robin procedure. Figure 3 demonstrates changes in the ROC curves due to the variation of training sets.

RF performance [Figs. 3(c) and 3(f)] is not affected by the reduction of the clean training set in the explored range, with the only exception being the run over S6 data where size of the clean training set is to 2.5% of the original. In this case, the ROC curve shows an efficiency loss on the order of 5% at a false-alarm probability of $P_0 = 10^{-3}$. Also, cutting the glitch training set by half does not affect RF efficiency in either S4 or S6 data.

SVM's performance follows very similar trends, shown in Figs. 3(b) and 3(e), demonstrating robust performance against the reduction of the clean training set and suffering appreciable loss of efficiency only in the case of the smallest set of clean training samples. Unlike RF, SVM seems to be more sensitive to variations in the size of the glitch training set. The ROC curve for the 50% glitch set in S6 data drops 5%–10% in the false-alarm probability region of $P_0 = 10^{-3}$ [Fig. 3(e)]. However, this does not happen in the S4 run [Fig. 3(e)]. This can be explained by the fact that the S4 glitch data set has five times more samples than the S6 set. Even after cutting it in half, the S4 set provides better sampling than the full S6 set.

ANN is affected most severely by training-set reduction [Figs. 3(a) and 3(d)]. First, its overall performance visibly degrades with the size of the clean training set, especially in the S6 runs [Fig. 3(d)]. However, we note that the ROC curve primarily drops near a false-alarm probability of $P_0 = 10^{-3}$, and it remains the same near $P_0 = 10^{-2}$ (for all but the 2.5% set). The higher P_0 value is more important in practice because the probability of false alarm of 10^{-2} is still tolerable and, at the same time, the efficiency is significantly higher than at $P_0 = 10^{-3}$. This means that we are likely to operate a real-time monitor near $P_0 = 10^{-2}$ rather than near 10^{-3}. Reducing the training sample introduces an artifact on ANN's ROC curves not seen for either RF or SVM. Here, the false-alarm probability's range decreases with the size of the clean training set. This is due to the fact that with the ANN configuration parameters used in this analysis, ANN's rank becomes more degenerate when less clean samples are available for training, meaning that multiple clean samples in the evaluation set are assigned exactly the same rank. This is in general undesirable, because a continuous, nondegenerate rank carries more information and can be more efficiently incorporated into gravitational-wave searches. The degeneracy issue of ANN and its possible solutions are treated in detail in Ref. [32].

We would like to highlight the fact that in our test runs, we use data from two different detectors and during different science runs, and that we test three very different classifiers. The common trends we observe are not the result of peculiarities in a specific data set or an algorithm. It is reasonable to expect that they reflect generic properties of the detectors' auxiliary data as well as the MLA classifiers. Extrapolating this to future applications in advanced detectors, we find it reassuring that the classifiers, when suitably configured, are able to monitor large numbers of auxiliary channels while ignoring irrelevant channels and features. Furthermore, their performance is robust against variations

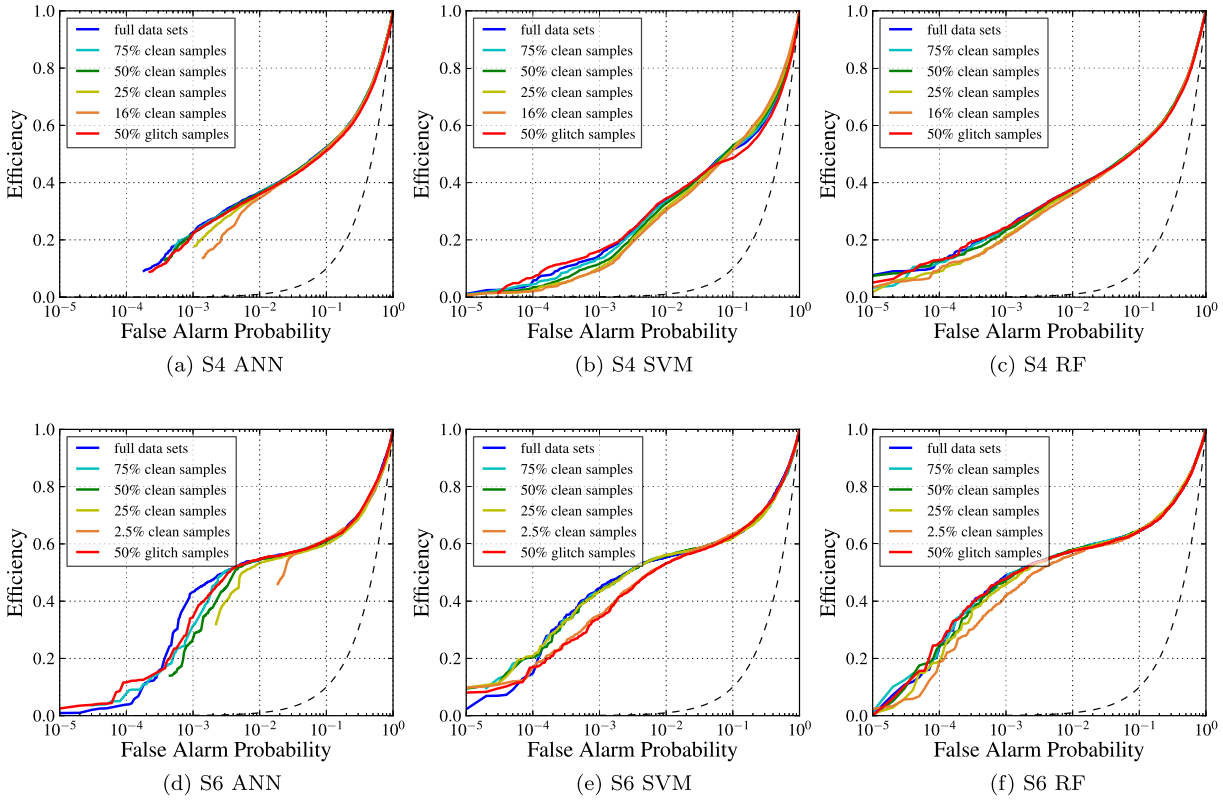


FIG. 3 (color online). Varying the size of training data sets. In our sample data, the number of glitches is limited by the actual glitch rate in the LIGO detectors and the length of the analysis time we use. However, we can construct as many clean samples as necessary because we sample the auxiliary channels at random times. In general, the classifiers' performance will increase with larger training data sets, but at additional computational cost. We investigate the effect of varying the size of training sets on the classifiers' performance, and observe only small changes even when we significantly reduce the number of clean samples. We also reduce the number of glitch samples, observing that the classifiers are more sensitive to the number of glitches provided for training. This is likely due to the smaller number of total glitch samples, and reducing the number of glitches may induce a severe undersampling of feature space. The black dashed line represents a classifier based on random choice. Panels (a), (b), and (c) show the ROC curves with S4 data for ANN, SVM, and RF, respectively. Panels (d), (e), and (f) show the ROC curves with S6 data for ANN, SVM, and RF, respectively.

in the training set size. In the next sections we compare different classifiers in their bulk performance as well as in sample-by-sample predictions using the full data sets.

VI. EVALUATING AND COMPARING CLASSIFIERS' PERFORMANCE

The most relevant measure of any glitch-detection algorithm's performance is its detection efficiency, the fraction of identified glitches, P_1, at some probability of false alarm, P_0. The ROC curve is the key figure of merit and can be used to assess the algorithm's efficiency, and objectively compare it to other methods throughout the entire range of the probability of false alarm. This is useful because the upper limit for acceptable values of the probability of false alarm depends on the specific application. In the problem of glitch detection in gravitational-wave data, we set this value to be $P_0 = 10^{-2}$, which corresponds to 1% of true gravitational-wave transients falsely labeled as glitches. Another way to interpret this is that 1% of the clean science data are removed from searches for gravitational waves.

Our test runs, described in the previous section, demonstrate the robustness of the MLA classifiers against the presence of irrelevant features in the input data. We are interested in measuring the classifiers' efficiency in the common case where no prior information about the relevance of the auxiliary channels is assumed. For this purpose, we use the full S4 and S6 data sets, including all channels with a wide selection of parameters. Using exactly the same training/evaluation sets for all our classifiers allows us to assign four ranks, (r_{ANN}, r_{SVM}, r_{RF}, r_{OVL}), to every sample and compute the probability of false alarm, $P_0(r_i)$ and efficiency, $P_1(r_i)$ for each of the classifiers. While the ranks cannot be compared directly, these probabilities can. Any differences in the classifiers' predictions, in this case, are from the details and limitations of the methods themselves, and are not from the training data.

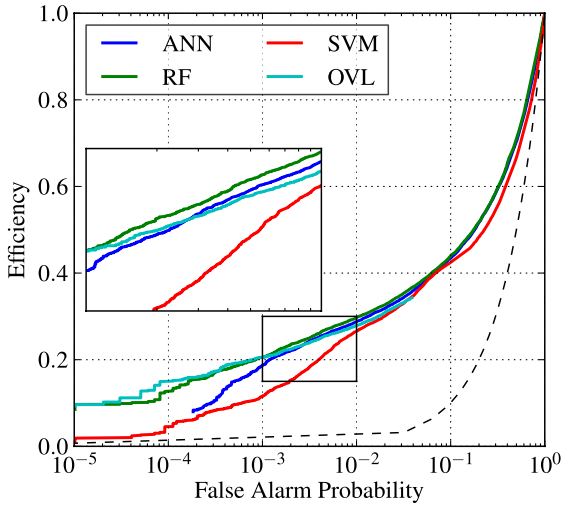
Glitch samples that are separated in time by less than a second are likely to be caused by the same auxiliary

disturbance. Even if they are not, gravitational-wave transient candidates detected in a search are typically "clustered" with a time window ranging from a few hundred milliseconds to a few seconds, depending on the length of the targeted gravitational-wave signal. Clustering implies that among all candidates within the time window, only the one with the highest statistical significance will be retained. In order to avoid double counting of possibly correlated glitches and to replicate conditions similar to a real-life gravitational-wave search, we apply a clustering procedure to the glitch samples, using a one-second time window. In this time window, we keep the sample with the highest significance, ρ, of the transient in the gravitational-wave channel.

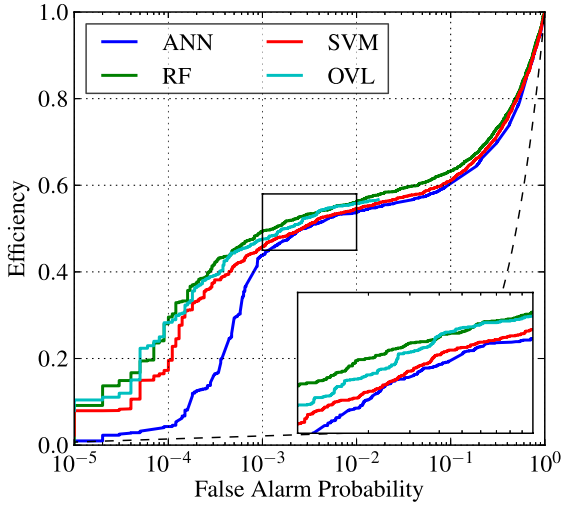
The ROC curves are computed after clustering. Figure 4 shows the ROC curves for ANN, SVM, RF and OVL for both S4 and S6 data.

All our classifiers show comparable efficiencies in the most relevant range of the probability of false alarm for practical applications (10^{-3}–10^{-2}). Of the three MLA classifiers, RF achieves the best efficiency in this range, with ANN and SVM getting very close near $P_0 = 10^{-2}$. Relative to other classifiers, SVM performs worse in the case of S4 data, and ANN's efficiency drops fast at $P \leq 10^{-3}$. The most striking feature on these plots is how closely the RF and the OVL curves follow each other in both S4 and S6 data [Figs. 4(a) and 4(b), respectively]. In absolute terms, the classifiers achieve significantly higher efficiency for S6 than for S4 data, 56% versus 30% at $P_0 = 10^{-2}$. We also note that the clustering procedure has more effect on the ROC curves in S4 than in S6 data. In the former case, the efficiency drops by 5%–10% [compare to the curves in Figs. 3(a) to 3(c)], whereas in the latter it stays practically unchanged [compare to Figs. 3(d) to 3(f)]. The reason for this is not clear. In the context of detector evolution, the S6 data are much more relevant for advanced detectors. At the same time, we should caution that we use just one week of data from the S6 science run and a larger scale testing is required for evaluating the effect of the detector's nonstationarity.

The ROC curves characterize the bulk performance of the classifiers, but they do not provide information about what kind of glitches are identified. To gain further insight into the distribution of glitches before and after classification, we plot cumulative histograms of the significance, ρ, in the gravitational-wave channel for glitches that remain after removing those detected by each of the classifiers at $P_0 \leq 10^{-2}$. We also plot a histogram of all glitches before any glitch removal. These histograms are shown in Fig. 5. They show the effect of each classifier on the distribution of glitches in the gravitational-wave channel. In both the S4 and S6 data sets, the tail of the glitch distribution, which contains samples with the highest significance, is reduced. At the same time, as is clear from the plots, many glitches in the mid range of significances are also removed,



(a) S4 ROC curves

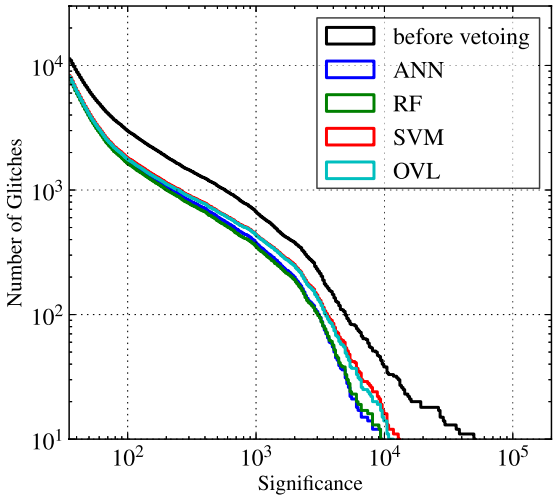


(b) S6 ROC curves

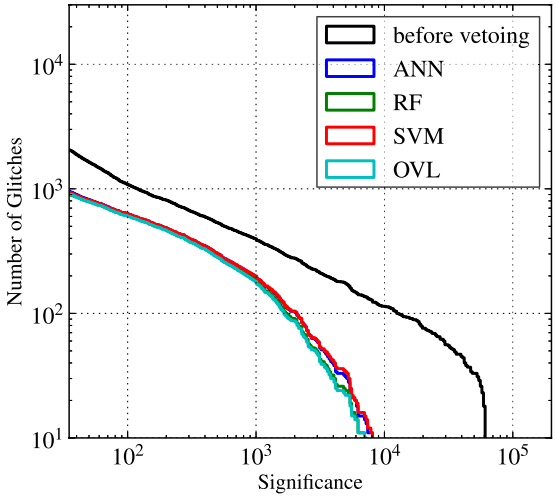
FIG. 4 (color online). Comparing algorithmic performance. We directly compare the performance of RF (green), ANN (blue), SVM (red), and OVL (light blue) using the full data sets. Glitches are clustered in time. We see that all the classifiers perform similarly, particularly in S6. There is a general trend of higher performance in S6 than in S4, which we attribute to differences in the types of glitches present in the two data sets. We should also note that all the MLA classifiers achieve performance similar to our benchmark, OVL, but RF appears to perform marginally better for a large range of the false-alarm probability. The dashed line corresponds to a classifier based on random choice. Panel (a) shows the ROC curves for S4 data. Panel (b) shows the ROC curves for S6 data. Insets on both panels show the ROC curves in the region of a false-alarm probability $P_0 \in [10^{-3}, 10^{-2}]$.

contributing to an overall lowering of the background for transient gravitational-wave searches. The fact that our classifiers remove low-significance glitches while some of the very high-significance glitches are left behind indicates that there is no strong correlation between the

amplitude of the glitches in the gravitational-wave channel and their detectability using auxiliary-channel information. This in turn implies that we either do not provide all necessary information for the identification of these high-significance glitches in the input feature vector or the classifiers somehow do not take advantage of this information. Given the close agreement between various classifiers that we observe in the ROC curves (Fig. 4) and

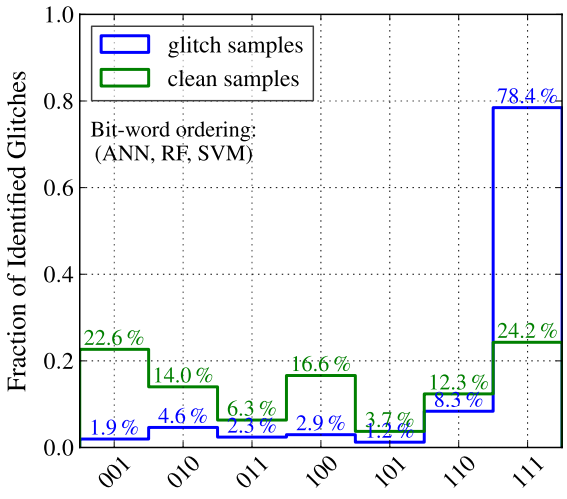


(a) S4 glitches

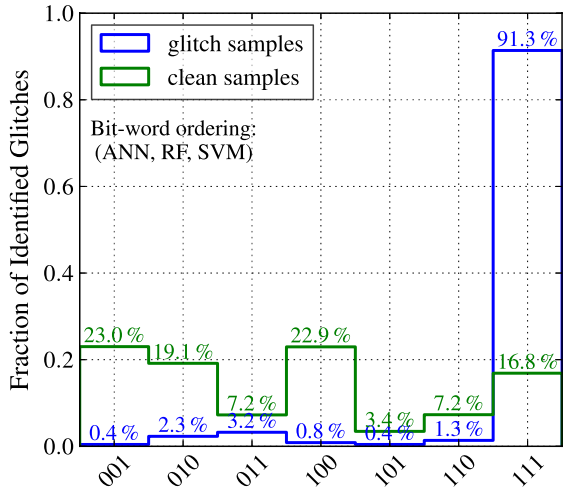


(b) S6 glitches

FIG. 5 (color online). Comparing the distribution of glitches before and after applying classifiers at 1% probability of false alarm. This cumulative histogram shows the number of glitches that remain with at least as high a significance in the gravitational-wave channel. We see that all our classifiers remove similar fractions of glitches at 1% probability of false alarm. This corresponds to their similar performances in Fig. 4, with efficiencies near 30% and 55% for S4 and S6 data, respectively. We also see that the classifiers tend to truncate the high-significance tails of the non-Gaussian transient distributions, particularly in S6. For reference, in Gaussian noise the odds of observing in a week of data a transient above the nominal significance threshold used here ($\rho \geq 35$) are extremely small ($\approx 10^{-6}$). Thus virtually all of the transients on the plot are non-Gaussian artifacts. Panel (a) shows the distributions of glitches in S4 data. Panel (b) shows the distributions of glitches in S6 data.



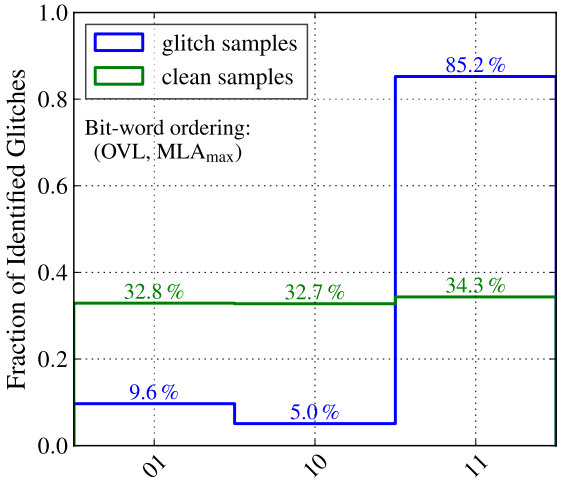
(a) S4 bit-word histogram for MLAs



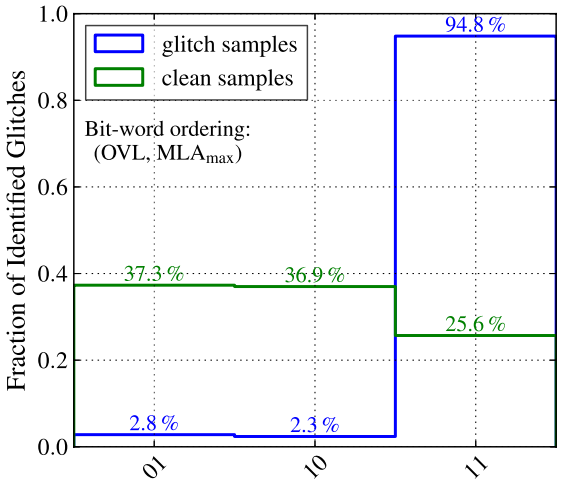
(b) S6 bit-word histogram for MLAs

FIG. 6 (color online). Redundancy between MLA classifiers. These histograms show the fractions of detected glitches identified in common by a given set of classifiers at 1% probability of false alarm (blue). The abscissa is labeled with bit-words, which are indicators of which classifier(s) found that subset of glitches (e.g. 011 corresponds to glitches that were not found by ANN, but were found by RF and SVM). The quoted percentages represent the fractions of detected glitches so that 100% represents those glitches which were successfully identified by at least one of the classifiers at 1% false-alarm probability. The three classifiers show a large overlap for glitch identification (bit-word = 111), meaning the classifiers are largely able to identify the same glitch events. Also shown is the fraction of clean samples (green) misidentified as glitches, which shows a comparatively flat distribution across classifier combinations. Panel (a) shows the bit-word histograms for S4 data. Panel (b) shows the bit-word histograms for S6 data.

the histograms of glitch distributions (Fig. 5), the former alternative seems to be more plausible. Alternatively, our choices of the thresholds and the coincidence windows that went into the construction of the feature vectors might not be optimal. Also, heretofore unincluded features characterizing the state of the detector, which may amplify transient disturbances in the auxiliary channels and induce glitches in the gravitational-wave channel, might be crucial for identifying glitches missed in the current analysis. We leave the investigation of these possibilities to future work.



(a) S4 bit-word histogram for MLA_{max} and OVL



(b) S6 bit-word histogram for MLA_{max} and OVL

FIG. 7 (color online). Redundancy between MLA_{max} and OVL. This figure is similar to Fig. 6, except that these histograms only compare the results of combining the MLA classifiers into a single unified classifier (MLA_{max}) and OVL. Even though OVL only considers pairwise correlations between auxiliary channels and the gravitational-wave channel, we see that it identifies the same glitches as MLA_{max}. This suggests that the glitches identified by the MLA classifiers are effectively characterized by pairwise correlations between a single auxiliary channel and the gravitational-wave channel, and that considering multichannel correlations does not add much. We also see that these classifiers are highly correlated in their selection of glitches (blue), but less correlated across the set of misidentified clean samples (green). Panel (a) shows the bit-word histograms for S4 data. Panel (b) shows the bit-word histograms for S6 data.

Although the ROC curves (Fig. 4) and the histograms (Fig. 5) provide strong evidence that all classifiers detect the same glitches, they do not give a clear quantitative picture of the overlap between these methods. To see this more clearly, we define subsets of glitches based on which combination of classifiers detected them with a probability of false alarm less than 10^{-2}. We determine overlaps between the MLA classifiers by constructing a bit-word diagram (Fig. 6). It clearly demonstrates a high degree of redundancy between the classifiers. The fraction of glitches detected by all three MLA classifiers is 91.3% for S6 data and 78.4% for S4 data. For comparison, we show in the same figure the bit-word diagram representation for clean samples that are falsely identified as glitches with a probability of false alarm less than 10^{-2}. The classifiers' predictions for clean samples are distributed almost uniformly. This suggests that our classifiers select clean samples nearly independently, or at least with a much lower level of correlation than for glitches.

Next, we compare the MLA classifiers to OVL. In order to reduce the number of possible pairings, we combine the MLA classifiers following the maximum-likelihood-ratio algorithm described in more detail in Sec. VII. In short, this algorithm picks the most statistically significant prediction out of the three MLA classifiers for each event. We denote the combined classifier as MLA_{max}. As in the previous case, we construct the bit-word diagram for both glitch and clean samples detected with the probability of false alarm less than 10^{-2} (Fig. 7). The redundancy is even stronger; the fraction of glitches detected by MLA_{max} and OVL is 94.8% for S6 data and 85.2% for S4 data. The full bit-word histograms show the same behavior and we omit them here.

VII. METHODS FOR COMBINING CLASSIFIERS

On a fundamental level, the MLA classifiers search for a one-parameter family of decision surfaces in the feature space, $x \in V_d$, by optimizing a detection criterion. The parameter labeling the decision surfaces can be mapped into a continuous rank, $r_{MLA}(x) \in [0, 1]$. This rank reflects the odds for a sample, x, to correspond to a glitch in the gravitational-wave channel. As we discuss in Sec. III and Appendix B, theoretically, if the classifiers use consistent optimization criteria, they should arrive at the same optimal decision surfaces and make completely redundant predictions. In other words, their ranks would be functionally dependent. In practice, however, different classifiers often lead to different results, primarily due to the limitations in the number of samples in the training sets and/or computing resources. For instance, different classifiers may be more or less sensitive to different types of glitches. In this case, one

should be able to detect a larger set of glitches by combining their output. Furthermore, the classifiers may be strongly correlated in the ranks they assign to glitch samples, but only weakly correlated when classifying clean samples. Again, by combining the output of different classifiers, we may be able to extract information about these correlations and improve the total efficiency of our analysis.

This last case appears to be applicable to our data set. From Sec. VI, we see that at a probability of false alarm of 1%, all classifiers remove nearly identical sets of glitches (to within 10% for the S6 data). However, the classifiers agree to a significantly lesser extent on the clean samples they remove (Fig. 6). This suggests that the correlations between the classifiers' predictions are different for glitches and clean samples, and combining the classifiers' output could possibly lead to an improved analysis.

The general problem of combining the results from multiple, partially redundant analysis methods has been addressed in the context of gravitational-wave searches in Ref. [22]. Treating the output of the classifiers—namely their ranks—as new data samples, one arrives at the optimal combined ranking given by the joint likelihood ratio,

$$\Lambda_{\text{joint}}(\vec{r}) = \frac{p(\vec{r}|1)}{p(\vec{r}|0)}, \tag{14}$$

where $\vec{r} \equiv (r_{\text{ANN}}, r_{\text{SVM}}, r_{\text{RF}})$ is the vector of the MLA ranks assigned to a sample, x, and $p(\vec{r}|1)$ and $p(\vec{r}|0)$ are the probability density functions for the rank vector in the case of glitch and clean samples, respectively. We should point out that we can modify this ranking by multiplying by the ratio of prior probabilities $[p(1)/p(0)]$ to match the rankings for individual classifiers without affecting the ordering assigned to samples. Typically, these conditional probability distributions are not known and computing the joint likelihood ratio from first principles is not possible. One has to develop a suitable approximation. We try several different approximations when combining algorithms.

Our first approximation, and perhaps the simplest, estimates the likelihood ratio for each classifier separately and assigns the maximum to the sample. This method should be valid in the two limits: extremely strong correlations and extremely weak correlations between the classifiers. It was first suggested and applied in the context of combining results of multiple gravitational-wave searches in Ref. [22]. We estimate the individual likelihood ratios in two ways: 1) as the ratio of CDF and 2) as the ratio of kernel density estimates for the PDF. Though a proper estimate should involve the PDFs, the advantage of using CDFs is that we already calculate them when evaluating the efficiency and probability of false alarm for each classifier. They should approximate the ratio of PDFs reasonably well in the tail of the distributions, when the probability of false alarm is low. This assumes that PDFs are either slowly varying or simple (e.g. power law or exponential) decaying functions of the rank. However, at large values of the probability of false alarm or in the case when the probability distributions exhibit complicated functional dependence on the rank, our approximation may break down and we will have to resort to the more fundamental ratio of the PDFs. Explicitly, we estimate the joint likelihood ratio using

$$L_1(\vec{r}) \equiv \max_{r_j} \left\{ \frac{\int_{r_j}^1 p(r'_j|1)\mathrm{d}r'_j}{\int_{r_j}^1 p(r'_j|0)\mathrm{d}r'_j} \right\} = \max_{r_j} \frac{P_1(r_j)}{P_0(r_j)}. \tag{15}$$

We refer to this method as MLA_{max} when introducing it in the context of Fig. 7.

We also construct smooth one-dimensional PDFs for clean and glitch samples from their ranks using Gaussian kernel density estimation [42]. These estimates were built using a constant bandwidth equal to 0.05 in the rank space, which ranges from 0 to 1. Based on this, we define the approximate combined rankings,

$$L_2(\vec{r}) \equiv \max_{r_j} \left\{ \frac{p(r_j|1)}{p(r_j|0)} \right\}. \tag{16}$$

It is by no means true that we can always approximate the multidimensional likelihood ratio (14) with the maximum over a set of one-dimensional likelihood ratios. If we can better model the multidimensional probability distributions, we should be able to extract more information. To this end, we also implement a slightly more complicated combining algorithm. We observe that the algorithms are highly correlated on which glitches they remove, and less correlated on the clean samples (see Fig. 6). We therefore approximate $p(\vec{r}|1) \approx \max_{r_j}\{p(r_j|1)\}$ and $p(\vec{r}|0) \approx \prod_j p(r_j|0)$, which assumes that the algorithms are completely uncorrelated for the clean samples. Λ_{joint} is then approximated by

$$L_3(\vec{r}) \equiv \frac{\max_{r_j}\{p(r_j|1)\}}{\prod_i p(r_i|0)}. \tag{17}$$

Again, we compute the individual PDFs using Gaussian kernel density estimation.

More subtle—but still useful—correlations between the ranks assigned by different classifiers cannot be accounted for by these simple analytical approximations. Estimating the multidimensional probability distributions is a difficult task and undersampling quickly becomes the dominant source of error when expanding to higher than two dimensions. Rather than developing a complicated analytic model, we can use one of the MLA classifiers to compute the combined rank. We use RF to attempt to combine the ranks from each classifier and construct an estimate of the full (three-dimensional) joint likelihood ratio.

We compare the methods for combining the classifiers by computing ROC curves, which are shown in Fig. 8. We reproduce only the S6 curves because the S4 data shows the same trends.

All combined methods result in very similar ROC curves and, when compared to the OVL curve, they do not seem to improve the overall performance by more than a few

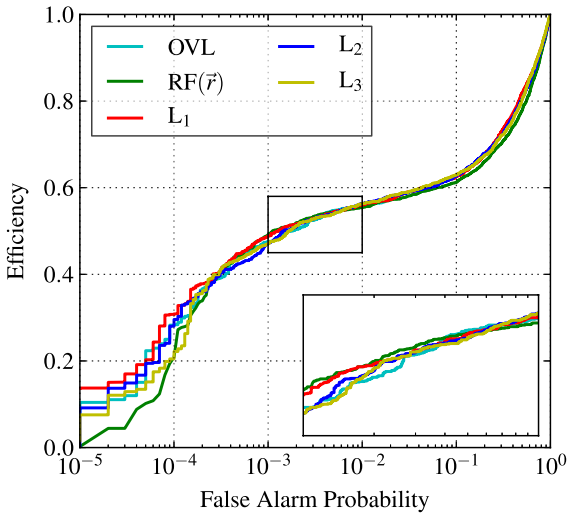


FIG. 8 (color online). Comparison of different combining algorithms using S6 data. This figure compares the performance of our various schemes for combining the output of the three MLA classifiers. We note that all four algorithms, L_1 (15), L_2 (16), L_3 (17), and using RF to classify events based on the MLA output vector $\vec{r}$, agree to a remarkable degree. The fact that our simple analytic algorithms perform just as well as the RF suggests that there are not many subtle correlations between the classifiers' output. The MLA combining algorithms do not perform much better than OVL. Comparing these curves with Fig. 4 shows that the combined performance does not exceed the individual classifier's performances. This suggest that the individual MLA classifiers each extract almost all of the useful information from our feature vectors, and that they identify the same types of glitches. These conclusions are further supported by Fig. 6.

percent. These combined results lead us to conclude that the individual classifiers have already reached nearly optimal performance for the given input data, and that their combination, while increasing their robustness, cannot improve the overall efficiency. Basically, all the useful information has been extracted already.

Although it is not immediately apparent, these combining schemes do add robustness to our identification of glitches. The combining algorithms are able to ignore underperforming classifiers and reject noisy input fairly well, and we see that they tend to select the best performance from the individual classifiers. By comparing Fig. 8 with Fig. 4, we see that the combining algorithms follow the best ROC curve from Fig. 4, even when individual classifiers are not performing equitably. This is most evident at extremely low probabilities of false alarm. This robustness is important because it can protect a combined glitch-identification algorithm from bugs in a single classifier. In this way, the combining algorithm essentially acts as an automatic cross reference between individual MLA classifiers.

VIII. CONCLUSION

In this study, we applied various machine-learning algorithms to the problem of identifying transient noise artifacts (glitches) in gravitational-wave data from LIGO detectors. Our main goal was to establish the feasibility of using MLAs for the robust detection of instrumental and environmental glitches based on information from auxiliary detector channels. We considered several MLA classifiers: the artificial neural networks, the support vector machine, and the random forest. We formulated the general detection problem in the context of glitch identification using auxiliary-channel information and showed that, theoretically, all classifiers lead to the same optimal solution. In a real-life application, our classifiers have to monitor a large number of channels. Even after data reduction, the dimensionality of our feature space could be as high as 1250, making classification a truly challenging task. We tested classifiers using data sets from the S4 and S6 LIGO scientific runs. We used standard ROC curves as the main figure of merit to evaluate and compare the classifiers' performances.

Our tests showed that the classifiers can handle extraneous features efficiently without affecting their performance. Likewise, we found that the classifiers are generally robust against changes in the size of the training set. The most important result of our investigation is the confirmation that the MLA classifiers can be used to monitor a large number of auxiliary channels, many of which might be irrelevant or redundant, without a loss of efficiency. These classifiers can be used to develop a real-time monitoring and detector characterization tool.

After establishing the robustness of the classifiers against changes in the input data and the presence of nuisance parameters, we evaluated the algorithms' performance in terms of the ROC curve and carried out a detailed comparison between the classifiers. This included their impact on the overall distribution of glitches in the gravitational-wave channel and the redundancy of their predictions. We found that at a false-alarm probability of 1%, all classifiers demonstrate comparable performance and achieve 30% and 56% efficiency at identifying single-detector glitches above our nominal threshold when tested on the S4 and S6 data, respectively. While not superb, this is a step toward the ultimate goal of producing a cleaned data set, which is indistinguishable from Gaussian noise.

In all tests we benchmarked the MLA classifiers against the OVL classifier, which was optimized to detect pairwise correlations between transients in single auxiliary channels and transients in the gravitational-wave channel. Somewhat unexpectedly, the MLA classifiers demonstrate a very high level of redundancy with the OVL classifier, achieving similar efficiency as measured by the ROC curves. The thorough event-by-event comparison shows 85% and 95% redundancy in glitch detection between the MLA and the OVL classifiers for S4 and S6 data, respectively. Moreover, only a small subset of all channels, 47 (of 162) in S4 data and 35 (of 250) in S6 data, contributes to the total efficiency.

This indicates that the input data are dominated by simple pairwise correlations, and that the higher-order correlations are either subdominant or altogether not present in the data. This interesting insight into the structure of the data could not have been gained without the application of MLAs. In future work we will investigate how the current performance can be improved through boosting MLAs using OVL predictions and including other types of information in the input data.

One advantage of the MLA classifiers is that they provide a continues rank, $r_{\mathrm{MLA}} \in [0, 1]$, rather than a binary flag. The OVL classifier's output can also be converted into a rank which, although by construction is discrete, is not a binary flag. This rank can be incorporated directly into the searches for gravitational waves as a parameter characterizing a candidate event along with the rest of the data from the gravitational-wave channel, as opposed to a standard approach of vetoing entire segments of flagged data based on a hard threshold on data quality. This approach offers more efficient use of auxiliary information and opens new possibilities for tuning and applying the MLA and OVL classifiers in the specific gravitational-wave searches. We will pursue this direction in our future work.

Another advantage of the MLA classifiers is that they can incorporate various potentially diverse types of information and establish correlations between multiple parameters. In the future, in addition to the fast, transient noise data used in this study, we plan to include more slowly varying baseline information about the detector subsystems. For example, the state of alignment in the interferometer may be important for predicting the amount of noise that couples into the gravitational-wave channel from elsewhere in the instrument. In the current study, we recorded only significant, short-duration disturbances in the alignment channels detected by the KleineWelle analysis algorithm. In our future work, we will also include the instantaneous level of misalignment in the optical cavity at the moment of a glitch, which will amount to a different representation of the data from the alignment channels. Machine learning should be able to automatically identify such nonlinear correlations, even if they are not known previously. We would also like to include information from other transient detection algorithms [9,43] that may be better equipped to find and parametrize different classes of signals.

As a final test of our study, we explored several ways of combining the output of several classifiers (including OVL) in order to increase the robustness of their predictions and possibly improve combined efficiency. Following general principles for combing multiple analysis methods, we suggested several approximations for the optimal combined ranking given by the joint likelihood ratio. We tested our approximations and found that they perform similarly to and do not improve upon the efficiencies of individual classifiers.

Based on these results, we conclude that the three MLA classifiers used in this study are all able to achieve robust and competitive classification performance for our set of data. The RF classifier was the most robust against the form (range, shape, scaling, number) of input data, while ANN and SVM benefit from reshaping certain input parameters along physical arguments. Since all classifiers achieve similar limiting performance and identify most of the same events, we conclude that they are near-optimal in their use of the existing data. Future improvement in classification efficiency is therefore likely to come from including additional sources of useful information, rather than refinements to the algorithms themselves. In the future we plan to add a different and/or better parametrization of auxiliary-channel transients, and other baseline information from the auxiliary channels which may represent nonlinear couplings to the gravitational-wave channel.

ACKNOWLEDGMENTS

L. B. was supported by an appointment to the NASA Postdoctoral Program at Goddard Space Flight Center, administered by Oak Ridge Associated Universities through a contract with NASA. K. K., Y. M. K., C. H. L., J. J. O., S. H. O., and E. J. S. were supported in part by the Global Science experimental Data hub Center (GSDC) at KISTI. K. K., Y. M. K., and C. H. L. were supported in part by National Research Foundation Grant funded by the Korean Government (NRF-2011-220-C00029). C. H. L. was supported in part by the BAERI Nuclear R&D program (M20808740002). J. C., E. O. L., and X. W. were supported in part by the Ministry of Science and Technology of China under the National 973 Basic Research Program (Grants No. 2011CB302505 and No. 2011CB302805). T. Y. was supported in part by the National High-Tech Research and Development Plan of China (Grant No. 2010AA012302). R. E., K. H., E. K., and R. V. were supported by LIGO laboratory. LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation and operates under cooperative agreement PHY-0757058.

APPENDIX A: A TOY EXAMPLE OF LIKELIHOOD-RATIO DECISION SURFACES

To illustrate the optimization of decision surfaces, discussed in Sec. III, we consider a toy example in which feature-space, V_{d}, is two dimensional. See Fig. 9.

Class 0 samples (black circles) follow a symmetric Gaussian probability distribution centered at the origin, $p(x|0) = (1/2\pi)\exp\{-(x_1^2 + x_2^2)/2\}$. Class 1 samples (blue diamonds) follow the same distribution but with its center shifted to (3.0, 3.0), $p(x|1) = (1/2\pi)\exp\{-[(x_1 - 3)^2 + (x_2 - 3)^2]/2\}$. In this case, the optimal decision surfaces—defined as the surfaces of constant likelihood ratio

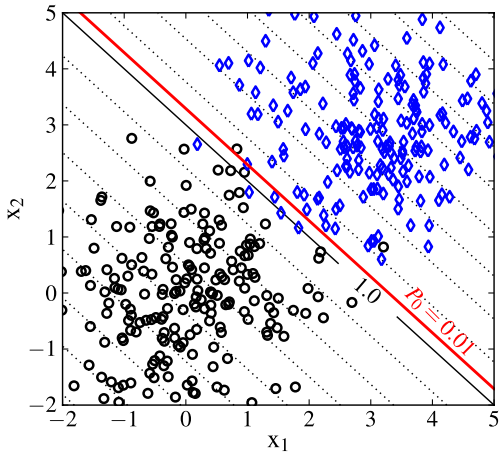


FIG. 9 (color online). A toy example of the likelihood-ratio decision surfaces.

(5), $\Lambda(x) = p(x|1)/p(x|0)$—are linear surfaces with normal vector $w = (1, 1)$. They are shown as dotted lines in the plot. The solid black line identifies the surface for which $\Lambda(x) = 1$. The likelihood ratio grows in the direction of the normal vector w. Each dotted line represents a tenfold increase (decrease) in $\Lambda(x)$. We assume that *a priori* probabilities for a given sample to be of Class 0 or Class 1 are equal. Using expressions for $p(x|0)$ and $p(x|1)$, it is easy to compute the detection and the false alarm probabilities, Eq. (2), for any constant likelihood-ratio surface. For example, the surface designated by the solid red line in the plot corresponds to the false-alarm probability, $P_0 = 10^{-2}$, meaning that 1% of all Class 0 events are expected to fall above this line. If the likelihood ratio itself is used as a decision function in Eq. (2), $f(x) = \Lambda(x)$, then the threshold value corresponding to $P_0 = 10^{-2}$ is $F^* = 2.39$. When applying this decision threshold to classify events, samples with $\Lambda(x) \geq 2.39$ will be classified as Class 1 and all others as Class 0. This implies that on average 1% of the samples that are Class 0 will be misclassified as Class 1 and approximately 97% ($P_1 = 0.97$) of Class 1 events will be correctly classified. This is the theoretical upper limit on the performance of any classification algorithm for this example. The form and orientation of the decision surfaces are defined by the probability distributions for both Class 0 and Class 1 events. For example, in the absence of any information about Class 1 events, it would be natural to assume a uniform distribution for them, $p(x|1) = \text{const}$. This would lead to the decision surfaces described by the concentric circles centered at the origin.

APPENDIX B: DERIVATION OF THE DECISION SURFACES FOR SOME MLA OPTIMIZATION SCHEMES

In Sec. III, we stress that for the detection (or the two-class classification) problem, the most natural optimizing criterion is the Neyman-Pearson criterion, which requires a maximum probability of detection at a fixed probability of false alarm. Optimizing this criterion, which in functional form is given by Eq. (3), leads to the one-parameter family of decision surfaces defined as surfaces of constant likelihood ratio (6). Each decision surface is labeled by a corresponding value of the likelihood-ratio, $\Lambda(x)$, providing a natural ranking for unclassified events. The higher the likelihood ratio, the more likely it is that that event belongs to Class 1. The likelihood ratio can be mapped to the particular value of the false-alarm probability, $P_0(\Lambda)$, which assigns it a statistical significance. In practice, it is often more convenient to define ranking in terms of some monotonic function of the likelihood ratio, $r(\Lambda)$ [e.g. $r(\Lambda) = \ln \Lambda$]. Classifying and ranking samples based on the likelihood ratio is guaranteed to maximize the ROC curve $P_1(P_0)$.

In their standard configurations, most MLA classifiers apply other kinds of optimization criteria (e.g. the fraction of correctly classified events or the Gini index). Many of these criteria treat the two classes of the events symmetrically, which often is more appropriate than the asymmetric Neyman-Pearson criterion. In this appendix, we would like to explore some of the most popular criteria used by the MLA classifiers in their relation to the Neyman-Pearson criterion. Specifically, we are interested in establishing consistency between the various criteria used in this study, in that they lead to the same optimal decision surfaces and compatible rankings.

1. The fraction of correctly classified events

First, we consider probably the most popular criterion: the fraction of correctly classified events. This criterion is used by both ANN and SVM in our analysis. Following the approach of Sec. III, we define it as a functional of the decision function, $f(x)$, on the feature space, V_d,

$$C = \int_{V_d} \Theta(f(x) - F^*) p(1|x) p(x) \mathrm{d}x + \int_{V_d} \Theta(F^* - f(x)) p(0|x) p(x) \mathrm{d}x, \quad \text{(B1)}$$

where $p(1|x)$ and $p(0|x)$ are the probabilities for a sample with feature vector x to be classified as Class 1 and Class 0, respectively, and $p(x)$ is the probability distribution of obtaining a feature vector, x, regardless of its class. To elucidate this expression, we note that $p(c|x)p(x)\mathrm{d}x$ corresponds to the fraction of total events that fall in the hypervolume $\mathrm{d}x$ and are of class c. Without the Θ functions, these integrals would evaluate to $C = p(1) + p(0) = 1$, and we see that the Θ functions select only those events that are correctly classified. With this interpretation, $f(x)$ is the decision function. For a given threshold F^*, it defines two regions of samples as Class 1 and Class 0 through $\Theta(f(x) - F^*)$ and its complement $\Theta(F^* - f(x))$, respectively. Thus, the first term in Eq. (B1) accounts for

correctly classified events of Class 1 and the second term does the same for Class 0 events.

Using Bayes' theorem, one can express $p(1|x)p(x)$ and $p(0|x)p(x)$ in the first and second term of Eq. (B1) as

$$p(1|x)p(x) = p(x|1)p(1), \tag{B2a}$$
$$p(0|x)p(x) = p(x|0)p(0). \tag{B2b}$$

Here $p(x|1)$ and $p(x|0)$, defined in Sec. III, are the probability density functions for feature vectors in the presence and absence of a glitch in gravitational-wave data, respectively. $p(1)$ and $p(0)$ are the prior probabilities for having a glitch or a clean datum, related to each other via $p(1) + p(0) = 1$ as always. The fraction of correctly classified events is then given by

$$C = \int_{V_d} \Theta(f(x) - F^*)p(x|1)p(1)\mathrm{d}x + \int_{V_d} \Theta(F^* - f(x))p(x|0)p(0)\mathrm{d}x. \tag{B3}$$

The requirement that the variation of C with respect to $f(x)$ must vanish,

$$\delta C = \int_{V_d} \delta(f(x) - F^*)\delta f(x)[p(x|1)p(1) - p(x|0)p(0)]\mathrm{d}x = 0, \tag{B4}$$

leads to the following condition for the points, x^*, satisfying $f(x^*) - F^* = 0$:

$$p(x^*|1)p(1) - p(x^*|0)p(0) = 0. \tag{B5}$$

This equation defines the decision surface consisting of points for which

$$\Lambda(x^*)\frac{p(1)}{p(0)} \equiv \frac{p(x^*|1)p(1)}{p(x^*|0)p(0)} = 1. \tag{B6}$$

Thus, optimizing the fraction of correctly classified events leads to a specific decision surface, for which the likelihood ratio $\Lambda(x^*) = p(0)/p(1)$. By construction, the optimization criterion (B1) treats events of both classes symmetrically, maximizing the number of correctly classified events. As a result, it selects a specific decision surface for which evidence that the sample belongs to either one class or the other is equal.

2. The Gini index

Next, we consider the Gini index criterion, which is used in RF. For two-class problems, the Gini index of a region in the feature space, V, is defined as

$$G(V) = 1 - p^2 - q^2 = 2pq, \tag{B7}$$

where p and q are the fraction of Class 1 and Class 0 samples in the region, with $p + q = 1$. The Gini index for multiple regions is given by the average

$$G = \sum_i G(V_i)p(V_i), \tag{B8}$$

where $p(V_i)$ is the probability for a sample to be in the region V_i.

For the two-class problem, there are two distinct regions: the region where samples are classified as Class 1, V_1, and the region where samples are classified as Class 0, V_0. We make the following definitions:

$$P = \int_{V_d} \Theta(f)p(1|x)p(x)\mathrm{d}x = \int_{V_d} \Theta(f)p(x|1)p(1)\mathrm{d}x, \tag{B9a}$$
$$Q = \int_{V_d} \Theta(f)p(0|x)p(x)\mathrm{d}x = \int_{V_d} \Theta(f)p(x|0)p(0)\mathrm{d}x, \tag{B9b}$$

where P (Q) is the probability that a glitch (clean sample) will fall into V_1 and $\Theta(f)$ is shorthand for $\Theta(f(x) - F^*)$. We recognize that the expected fractions of events in V_1 can be described as $p = P/p(V_1)$ and $q = Q/p(V_1)$, where $p(V_1) = \int_{V_d} \Theta(f)p(x)x$ is the probability for any event (either glitch or clean sample) to fall in V_1. Furthermore, we can immediately write the corresponding relations for V_0 in terms of P, Q, and $p(V_1) = 1 - p(V_0)$. We then obtain

$$\frac{G}{2} = \frac{PQ}{p(V_1)} + \frac{(p(1) - P)(p(0) - Q)}{1 - p(V_1)}, \tag{B10}$$

where $p(1)$ and $p(0)$ are the prior probabilities for an event to be Class 1 and Class 0, respectively. We also note that $p(V_1)$, P, and Q are functionally related through

$$p(V_1) \equiv \int_{V_d} \Theta(f)p(x)\mathrm{d}x = \int_{V_d} \Theta(f)[p(1|x)p(x) + p(0|x)p(x)]\mathrm{d}x = P + Q. \tag{B11a}$$

We now optimize the Gini index, Eq. (B10), to determine the optimal decision surface. For simplicity, let us first optimize G while holding $p(V_1)$ constant. We are interested in the optimal decision surface's shape, and this optimization will determine it while keeping the ratios of the number of samples in V_1 and V_0 constant. We obtain

$$\frac{1}{2}\frac{\delta G}{\delta f(x)} = \frac{p(x^*)Q + q(x^*)P}{p(V_1)} + \frac{p(x^*)Q + q(x^*)P}{1 - p(V_1)} - \frac{p(1)q(x^*) + p(0)p(x^*)}{1 - p(V_1)}, \tag{B12}$$

where

$$p(x^*) \equiv \frac{\delta P}{\delta f(x^*)} = \delta_{x,x^*} \int_{V_d} \delta(f(x) - F^*) p(x|1) p(1) \mathrm{d}x, \tag{B13a}$$

$$q(x^*) \equiv \frac{\delta Q}{\delta f(x^*)} = \delta_{x,x^*} \int_{V_d} \delta(f(x) - F^*) p(x|0) p(0) \mathrm{d}x, \tag{B13b}$$

where $\delta_{x,x^*} = \delta f(x)/\delta f(x^*) = \{1 \text{ if } x^* = x, 0 \text{ otherwise}\}$. We see that $p(x^*)$ and $q(x^*)$ are probability density functions defined on the decision surface ($x^* \in \{f(x^*) = F^*\}$). We can write these variations in terms of conditional probabilities on the decision surface,

$$p(x^*) = p(x^*|1)p(1), \tag{B14a}$$

$$q(x^*) = p(x^*|0)p(0). \tag{B14b}$$

Requiring the variation of the Gini index, Eq. (B12), to vanish leads to the following relation:

$$\frac{p(x^*)}{q(x^*)} = -\frac{p(1)p(V_1) - P}{p(0)p(V_1) - Q}. \tag{B15}$$

Identifying the left-hand side of this equation with the likelihood ratio for a point on the decision surface, $\Lambda(x^*)(p(1)/p(0)) = p(x^*|1)p(1)/p(x^*|0)p(0)$, and using Eq. (B11) we find that

$$\Lambda(x^*)\frac{p(1)}{p(0)} = 1, \tag{B16}$$

which will hold for all points on the decision surface. Remarkably, this condition is independent of P, Q and $p(V_1)$. As in the case of the fraction of correctly classified events (B6), it implies that the optimal decision surface is the surface on which the likelihood ratio is equal to a ratio of the priors.

If we consider the more general maximization problem, in which we allow $p(V_1)$ to vary, we must maximize

$$\frac{G}{2} = \frac{PQ}{p(V_1)} + \frac{(p(1) - P)(p(0) - Q)}{1 - p(V_1)} + \lambda(p(V_1) - P - Q), \tag{B17}$$

where we use a Lagrange multiplier (λ) to enforce the condition $p(V_1) - P - Q = 0$ and consider variations of $p(V_1)$ to be independent of variations in $f(x)$.

The variation with respect to $p(V_1)$ defines the Lagrange multiplier,

$$\lambda = \frac{PQ(1 - 2p(V_1)) - p(V_1)^2[p(1)p(0) - p(1)Q - p(0)P]}{p(V_1)^2(1 - p(V_1))^2}. \tag{B18}$$

Variation with respect of $f(x)$ is given by Eq. (B12) minus $\lambda(p(x^*) + q(x^*))$. Setting it to zero for all independent variations of $f(x^*)$ leads to a more general condition on the likelihood ratio,

$$\Lambda(x^*) = -\frac{\lambda p(V_1)(1 - p(V_1)) + p(1)p(V_1) - P}{\lambda p(V_1)(1 - p(V_1)) + p(0)p(V_1) - Q}, \tag{B19}$$

which holds separately for all points x^* on the decision surface.

First of all, note that this condition still requires a constant likelihood ratio on the decision surface. For each point on the surface, $\Lambda(x^*)$ is determined by P and Q, which are constants for a given surface. We recover Eqs. (B15) and (B16) when $\lambda = 0$. In all other cases, the likelihood ratio on the decision surface is given by a quite complicated expression, obtained by plugging Eqs. (B18) and (B11) into Eq. (B18). In practice, the likelihood ratio on the decision surface is set by the desired value for the probability of false alarm, P_0, but the ratio will be constant over the entire surface. Optimizing of the Gini index, then, will be equivalent to optimizing the ROC curve in the region of interest, e.g. in our study near $P_0 = 10^{-2}$.

3. Asymmetric criteria

Both criteria considered so far are symmetric in their treatment of events in Class 1 and Class 0. While the asymmetry can be imposed by tuning the ratio of prior probabilities $[p(1)/p(0)]$, in some cases it might be desirable to use an explicitly asymmetric criteria. The RF implementation in the STATPATTERNRECOGNITION package contains two different asymmetric criteria, which we explore in our study: signal purity (P) and signal significance (S) [41]. By construction, they identify one class of events as signal and the other as background and place more emphasis on correctly classifying signal rather than background.

The signal purity and signal significance are defined as

$$P = \frac{\omega_1}{\omega_1 + \omega_0}, \tag{B20}$$

$$S = \frac{\omega_1}{\sqrt{\omega_1^2 + \omega_0^2}}, \tag{B21}$$

where ω_1 and ω_0 are the fraction of Class 1 (signal) and Class 0 (background) events in the signal region. The aim is to identify the signal region with the highest signal purity or signal significance. In the process of the decision-tree construction, the classifier identifies the terminal nodes with the highest values of P or S and orders nodes using these criteria as a rank. For a given terminal node of a tree, one can express ω_1 and ω_0 in terms of conditional probabilities,

$$\omega_1 = p(x|1), \tag{B22}$$

$$\omega_0 = p(x|0). \tag{B23}$$

In terms of these, one can rewrite the signal purity and the signal significance as

$$P = \frac{p(x|1)}{p(x|1) + p(x|0)} = \frac{\Lambda(x)}{\Lambda(x) + 1}, \tag{B24}$$

$$S = \frac{p(x|1)}{\sqrt{p(x|1)^2 + p(x|0)^2}} = \frac{\Lambda(x)}{\sqrt{\Lambda^2(x) + 1}}. \tag{B25}$$

Both quantities are monotonic functions of the likelihood ratio $\Lambda(x) = p(x|1)/p(x|0)$. Ordering nodes in the tree would be equivalent to ranking events by the likelihood ratio, which in turn is equivalent to using decision surfaces of the constant likelihood ratio for classification.

In practice, different classifiers have various limitations which result in suboptimal performance. Depending on the application, one algorithm or criteria may be more optimal than another, but we establish here that on a theoretical level they all recover the same optimal solution. In the two-class classification problem, the decision surfaces are surfaces of constant likelihood ratio, which also defines the optimal ranking for samples.

[1] B. Abbott *et al.* (LIGO Scientific Collaboration), Rep. Prog. Phys. **72**, 076901 (2009).
[2] G. M. Harry and the LIGO Scientific Collaboration, Classical Quantum Gravity **27**, 084006 (2010).
[3] J. Abadie *et al.* (LIGO Scientific Collaboration and the Virgo Collaboration), arXiv:1003.2481.
[4] J. Abadie *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), arXiv:1203.2674.
[5] J. Abadie *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), Classical Quantum Gravity **27**, 173001 (2010).
[6] C. Cutler and K. S. Thorne, in *Proceedings of GR16*, edited by N. T. Bishop and S. D. Maharaj (WorldScientific, Singapore, 2002).
[7] J. Aasi *et al.*, Classical Quantum Gravity **29**, 155002 (2012).
[8] S. Babak *et al.*, Phys. Rev. D **87**, 024033 (2013).
[9] S. Chatterji, L. Blackburn, G. Martin, and E. Katsavounidis, Classical Quantum Gravity **21**, S1809 (2004).
[10] A. D. Credico and the LIGO Scientific Collaboration, Classical Quantum Gravity **22**, S1051 (2005).
[11] T. Isogai (The Ligo Scientific Collaboration and The Virgo Collaboration), J. Phys. Conf. Ser. **243**, 012005 (2010).
[12] J. R. Smith, T. Abbott, E. Hirose, N. Leroy, D. MacLeod, J. McIver, P. Saulson, and P. Shawhan, Classical Quantum Gravity **28**, 235005 (2011).
[13] R. Essick, L. Blackburn, and E. Katsavounidis, Classical Quantum Gravity **30**, 155010 (2013).
[14] P. Ajith, M. Hewitson, J. R. Smith, H. Grote, S. Hild, and K. A. Strain, Phys. Rev. D **76**, 042004 (2007).
[15] L. Blackburn *et al.*, Classical Quantum Gravity **25**, 184004 (2008).
[16] N. Christensen (for the LIGO Scientific Collaboration and the Virgo Collaboration), Classical Quantum Gravity **27**, 194010 (2010).
[17] J. Slutsky *et al.*, Classical Quantum Gravity **27**, 165023 (2010).
[18] L. Blackburn, Ph.D. thesis, MIT, 2010.
[19] B. P. Abbott *et al.* (LIGO Scientific), Phys. Rev. D **80**, 102001 (2009).
[20] J. Neyman and E. S. Pearson, Phil. Trans. R. Soc. A **231**, 289 (1933).
[21] R. Biswas *et al.*, Phys. Rev. D **85**, 122008 (2012).
[22] R. Biswas *et al.*, Phys. Rev. D **85**, 122009 (2012).
[23] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data mining, Inference, and Prediction* (Springer, Berlin, 2009), 2nd ed.
[24] R. Hecht-Nielsen, in *Proceedings of International Joint Conference on Neural Networks* (IEEE, New York, Washington, 1989), Vol. 1, p. 593.
[25] C. Igel and M. Hüsken, in *Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000)*, edited by H.-H. Bothe and R. Rojas (ICSC Academic Press, New York, 2000), p. 115.
[26] F. A. N. N. library, http://leenissen.dk/fann/wp/.
[27] J. R. Koza and J. P. Rice, in *Proceedings of International Joint Conference on Neural Networks, Seattle, 1991* (IEEE Computer Society Press, Los Alamitos, 1991), p. 397.
[28] S. Bornholdt and D. Graudenz, Neural Netw. **5**, 327 (1992).
[29] E. J. Chang and R. P. Lippmann, Adv. Neural Inf. Process. Syst. **3**, 797 (1991).
[30] R. J. Collins and D. R. Jefferson, in *Parallel Problem Solving from Nature* (Springer-Verlag, Berlin, 1990), p. 259.

[31] F. Gruau, in *Proceedings of the IEEE Workshop on Combinations of Genetic Algorithms and Neural Networks* (IEEE, New York, 1992).
[32] Y.-M. Kim, S. H. Oh, E. J. Son, K. Kim, C.-H. Lee, and J. J. Oh (unpublished).
[33] C. Cortes and V. Vapnik, Mach. Learn. **20**, 273 (1995).
[34] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge University Press, Cambridge, England, 2000), 1st ed.
[35] C.-C. Chang and C.-J. Lin, ACM Transactions on Intelligent Systems and Technology **2**, 27 (2011), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[36] T.-F. Wu, C.-J. Lin, and R. C. Weng, J. Mach. Learn. Res. **5**, 975 (2004).
[37] L. Breiman, Mach. Learn. **24**, 123 (1996).
[38] L. Breiman, Mach. Learn. **45**, 5 (2001).
[39] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees* (Wadsworth and Brooks, Monterey, CA, 1984).
[40] J. R. Quinlan, Mach. Learn. **1**, 81 (1986).
[41] I. Narsky, arXiv:physics/0507143.
[42] P. Wand and C. Jones, *Kernel Smoothing* (Chapman and Hall, London, 1995).
[43] W. G. Anderson, P. R. Brady, J. D. E. Creighton, and E. E. Flanagan, Phys. Rev. D **63**, 042003 (2001).