PHYSICAL REVIEW D **95,** 104059 (2017)

Transient classification in LIGO data using difference boosting neural network

N. Mukund,[1,*] S. Abraham,[1,†] S. Kandhasamy,[2,‡] S. Mitra,[1,§] and N. S. Philip[3,‖]

[1]*Inter-University Centre for Astronomy and Astrophysics (IUCAA), Post Bag 4, Ganeshkhind, Pune 411 007, India*

[2]*LIGO Livingston Observatory, Livingston, Louisiana 70754, USA*

[3]*Department of Physics, St. Thomas College, Kozhencherry, Kerala 689641, India*

(Received 30 September 2016; published 31 May 2017)

Detection and classification of transients in data from gravitational wave detectors are crucial for efficient searches for true astrophysical events and identification of noise sources. We present a hybrid method for classification of short duration transients seen in gravitational wave data using both supervised and unsupervised machine learning techniques. To train the classifiers, we use the relative wavelet energy and the corresponding entropy obtained by applying one-dimensional wavelet decomposition on the data. The prediction accuracy of the trained classifier on nine simulated classes of gravitational wave transients and also LIGO's sixth science run hardware injections are reported. Targeted searches for a couple of known classes of nonastrophysical signals in the first observational run of Advanced LIGO data are also presented. The ability to accurately identify transient classes using minimal training samples makes the proposed method a useful tool for LIGO detector characterization as well as searches for short duration gravitational wave signals.

DOI: 10.1103/PhysRevD.95.104059

I. INTRODUCTION

Detection of short duration gravitational waves (GW) in LIGO data requires reliable identification and removal of noise transients produced by a variety of nonastrophysical sources [1,2]. Noise transients present in the data reduces the reliability of a GW detection by increasing its false alarm probability. Mitigation of noise transients is a major challenge in searches for GW, especially for short duration events where the signal can be easily mimicked by nonastrophysical transients of varied origin. These often have a waveform morphology close to that of the targeted signal, thus making the differentiation even more difficult [3].

With the advent of big data analysis, machine learning has emerged as a useful tool to handle huge volumes of data and to interpret meaningful results from them. In the past few decades, machine learning algorithms such as artificial neural network (ANN) [4,5], support vector machines [6,7], random forest [8], Gaussian mixture model [9] etc. found many applications in astronomy and occasionally, have been used for the study of noise artifacts in GW analysis. Since the visual inspection of individual events and their classification is time consuming and prone to errors, machine learning methods are more effective and reliable for the detection of hidden signatures of astrophysical GW in the data.

We present a hybrid classifier that combines features from supervised and unsupervised machine learning algorithms to do the transient classification. Our classifier performs an unsupervised hierarchical clustering on the incoming data to identify possible groups and a supervised Bayesian [10] classifier to do the final classification. The classifier code uses features extracted from wavelet analysis of the data in a fast and efficient manner using GPU and MPI parallelization techniques, whereby, making it a good candidate for real-time burst trigger classification and detector characterization. When used to predict the class labels for an input data, the classifier ranks the most likely classes each with an associated probability (confidence level) that may be used to set a threshold to discard unreliable predictions. This multiple class prediction is useful to identify borderline examples in the feature space. In our study, the classifier was first tested on simulated data consisting of astrophysical bursts along with commonly observed instrumental glitches and then on the LIGO sixth science run burst hardware injections. Targeted searches for specific glitch types seen in Advanced LIGO first observation data were also carried out, and the results are reported. Recent methods like deep learning [11,12] using convolutional neural networks require large number of training data and are computationally expensive. The fact that we are able to represent the transient classes with minimal features and fewer training data samples makes our method less susceptible to such issues and speeds up the training process, making it suitable for realtime applications.

*nikhil@iucaa.in
†sheelu@iucaa.in
‡skandhasamy@ligo-la.caltech.edu
§sanjit@iucaa.in
‖nspp@associates.iucaa.in

 © 2017 American Physical Society

TABLE I. Details of transients used. A = astrophysical, NA = nonastrophysical, O1: Advanced LIGO 1st science run, S6 = LIGO 6th science run, Sim = simulated.

Transient	Symbol	Type	Search type
Sine Gaussian [16,17]	SG	NA	S6, Sim
Ring down [18]	RD	A	S6, Sim
Gaussian [17,19]	GA	NA	S6, Sim
Supernova [20,21]	SN	A	S6, Sim
Cusp [22]	CSP	A	S6, Sim
White burst noise [19]	WNB	NA	S6, Sim
Black hole merger [23,24]	LBM	A	S6, Sim
Chirping sine Gaussian [25]	CSG	NA	Sim
Blip [3]	Blip	NA	Sim
Scattering [3,26,27]	SCT	NA	O1
Type A (low frequency)	A	NA	O1
Type B	B	NA	O1
Type C (blip, Fig. 6) [28]	C	NA	O1
Type D	D	NA	O1
Type E (koi fish) [28]	E	NA	O1
Type F (needle)	F	NA	O1
Type G	G	NA	O1
Lightning [3]	LGN	NA	Targeted

II. TRANSIENT EVENTS IN GW DATA

Table I lists the transients used in our analysis. Standard searches for compact binary coalescences use matched filtering as the base algorithm [13], while the burst searches primarily look for excess power in the data along with the time coincidence to trigger a detection [14,15]. Both these searches are followed by extensive sanity checks, where the auxiliary channels insensitive to astrophysical signals are inspected to rule out possible terrestrial coupling [3]. Auxiliary channels are often in the thousands, and their coupling with the GW strain sensitive channel is seen to fluctuate in time due to the dynamic nature of the instrument. This often makes the auxiliary channel veto procedure a daunting task. Incorporating a machine learning based veto procedure to identify well-known classes of nonastrophysical transients can help discern the trigger right at the strain channel and thus reduce false alarms.

III. FEATURE EXTRACTION

Raw time series is preprocessed by applying a whitening transformation, which enhances the short duration features seen in the data. Transient signals occurring in power systems and neuromagnetic brain responses have structural and temporal similarities with the glitch signals found in LIGO data streams. Wavelet based feature extraction for classifying these transients are detailed in [29,30]. A wavelet is a function having a smooth oscillatory pattern which vanishes near the ends [29]. With its desirable qualities like good localization in time and frequency domain, it seems to be a natural choice for extracting information from transient signals. A discrete wavelet transform results in sparser signal representation consisting of a reduced feature set, but still preserves information necessary to differentiate among the classes.

The mother wavelet is defined as

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right), \tag{1}$$

where $a, b \in R$ and $a \neq 0$ is scaled and time shifted to form the wavelet family.

The orthonormal basis of Hilbert space $L^2(R)$ consisting of finite-energy signals is obtained by discretizing scale and translation parameters $a_j = 2^{-j}$ and $b_{j,k} = 2^{-j}k$ giving the family wavelet as

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad \text{with} \quad j, k \in Z. \tag{2}$$

The all resolution level wavelet decomposition of the signal has the form,

$$S(t) = \sum_{j=-N}^{-1}\sum_{k} C_j(k)\psi j, \quad k(t) = \sum_{j=-N}^{-1} r_j(t), \tag{3}$$

where $N = \log 2$ (signal length).

The energy E_j at each resolution level is computed as

$$E_j = \|r_j\|^2 = \sum_k |C_j(k)|^2. \tag{4}$$

The relative wavelet energy at each resolution level

$$p_j = \frac{E_j}{E_{\text{tot}}}, \quad \text{where } E_{\text{tot}} = \sum_{j<0} E_j. \tag{5}$$

Wavelet entropy S_{WE} which encodes the degree of disorder in a signal can be written as

$$S_{WE} = -\sum_{j<0} p_j \ln[p_j]. \tag{6}$$

Here we carry out similar N-level one-dimensional wavelet decomposition [31] using an appropriately chosen mother wavelet. Feature extraction for simulated and LIGO O1 data is done using a Daubechies 2 (`db2`) wavelet, while for other search cases, a discrete Meyer (`dmey`) wavelet is used. We use $N = 12$ and 14, respectively, for data sampled at 4 and 16 KHz. Energy in the detail levels and wavelet entropy are then computed and are normalized to unity. In addition, kurtosis of the whitened signal is also used as a distinguishing feature. These features along with the class labels form the input for our Bayesian classifier. Figure 1 shows typical transients and their detail coefficient wavelet energy.

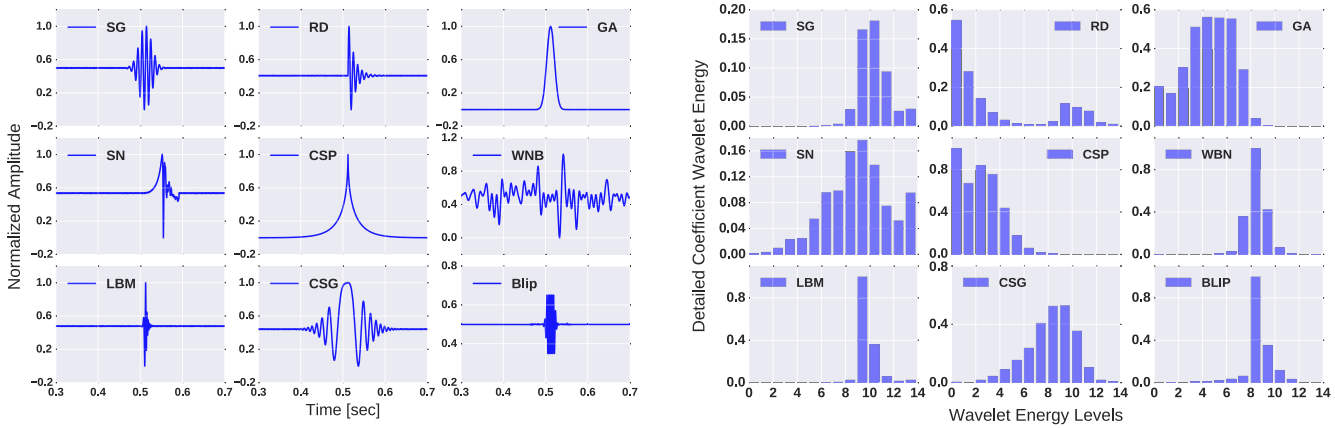

FIG. 1. Left panel depicts typical transient events (SNR set to 50 for better visualization). Wavelet energy median distribution for simulated data (SNR varied from 8 to 100) shown in the right panel.

IV. CLASSIFIER

Machine learning involves techniques which allow systems to automatically learn and improve prediction accuracies by exploring their past experiences on data. It mimics human decision-making ability by discovering the relationships between the variables of a system from a given set of examples that have both the variables and the observed outcomes. Here we use a hybrid classifier, a supervised Bayesian [10] one called difference boosting neural network (DBNN) [33,34] to classify the burst signals.

The DBNN can impose conditional independence on data without a loss of accuracy in the posterior computations. It does this by associating a threshold window with each of the attribute values of the examples [35]. The network is designed to work with discrete value input features, while GW data features are continuous. A simple method to deal with continuous feature value is to recast it into a suitable number of bins. There is no fixed criteria for the number of bins each feature may take. It might be argued that smaller the bin size, conditions can be imposed with better accuracy. However, in most practical situations, the optimal bin size is close to the square root of N, where N represents the number of discrete values present in the data for that variable. Once the bins are defined, for each feature bin and the given classes, the allowed ranges for all the remaining features are registered.

The DBNN, being a supervised neural network, requires a training data to configure the network before it can be used for classification of unseen data. The learning takes place by highlighting the difference between the features in two or more classes [35] by using Bayesian probability as its central rule for decision making. The confidence in a prediction [36] is the value of the posterior Bayesian probability for a given set of input features.

The working of DBNN can be divided into three units: Bayesian probability estimator, gradient descent boosting algorithm, and a discriminant function estimator [35]. The network starts with a flat prior for all the classes $P(C_k) = 1/N$, preventing the training from being biased to any specific prior distribution. The first unit in DBNN (executed by option 0 in the implementation) computes Bayesian probability and the threshold function for each of the training sample by constructing a grid for each class with columns representing the attributes and rows their respective values. The bin location for each attribute value is decided such that the full range of values can be uniformly covered by the set number of bins for that attribute across the classes. Initially, the content in attribute bins are all set to one. The training examples are taken one by one, and the bin corresponding to each attribute value for it's class is incremented by one. This sampled data is used to compute the likelihood for an attribute value to favor a class, $P(U_m|C_k)$, as the ratio of occurrences (counts) in it's bin for the class C_K to the total counts in all k classes for the same bin number that U_m holds for that attribute. The classifier also makes notes for each attribute value and it's class, the allowed maximum and minimum values taken by the remaining attributes in the entire training sample. This information is used to negate the possibility that the value of one feature may favor multiple classes, unless all other features also have values in the same range across the classes.

Though we started with a flat prior, to compute the Bayesian probability, we need to estimate the actual prior. In the Bayesian framework, prior has no special meaning. It is a weighted bias (belief) about the probable outcome of an experiment based on experiences in the past. In the second unit (executed by option 1 in the implementation), the DBNN estimates prior based on it's experience with the given training data. The DBNN does not make any change in the prior for correctly classified examples. In the case of failed examples, it attaches an additional weight to the attributes so that, it may also get correctly classified. To avoid random fluctuations due to the introduction

of arbitrary priers, this is done by modifying the flat prior incrementally by $\Delta W_m = \alpha(1 - \frac{P_k}{P_{k^*}})$ through a set of repeated rounds on the training data until the example gets correctly classified. That is, until $P(U|C_k) = \Pi_m P(U_m|C)$ goes to a maximum for the true class represented by the data. Here P_k and P_{k^*}, respectively, represent the calculated Bayesian probability for the true class and the wrongly estimated class, and α is a fraction called the learning rate [36]. Since the ratio of the probabilities are taken, this is much like the way humans arrive at their decisions based on their cumulative experiences in the past. This process is called training, and after training, the estimated likelihoods and prior are saved for future use. The assumption during the training process is that a representative training data is available that has suitable examples to represent all the variants in the target space.

The third unit (executed by options 2 and 3 in the implementation) computes the discriminant function. According to Bayesian theorem, the updated belief or the posterior is the product of the prior, and the evidence normalized over all possibilities. This can be written as

$$P(C_k|U) = \frac{\prod_m (P(U_m|C_k)W_m}{\sum_k \prod_m (P(U_m|C_k)W_m}, \tag{7}$$

where W_m represent the prior weight vector.

DBNN has been successfully applied to many astronomical problems such as star-galaxy classification [35], classification of point sources such as quasars, stars, and unresolved galaxies [37], transient classification [38] to indicate a few.

As for the case of all supervised networks, the accuracy of the predictions depend on the initial class selection and quality of the training data sets. When encountering real instrument data where it is difficult to know beforehand the actual groups present, running an unsupervised classifier prior to the Wavelet-DBNN classifier was seen to vastly improve the results. This step becomes more relevant for targeted searches looking for a particular transient class where unsupervised learning can yield insights into contamination from other glitch classes. Prior information about other glitches with very similar morphology can be made use of by the network to learn to differentiate between them whereby improving the accuracy.

We run an unsupervised classifier using Hierarchical clustering on the data to get an idea about the possible transient groups currently present in the data and their respective distribution (see Fig. 2). Classifier trained this way is observed to outperform the other scenarios where class selection is done either by visual inspection or by using predefined classes.

We employ a bottom up agglomerate clustering where the pairwise distance is calculated using the Mahalanobis distance measure [39]. The criterion for estimating the

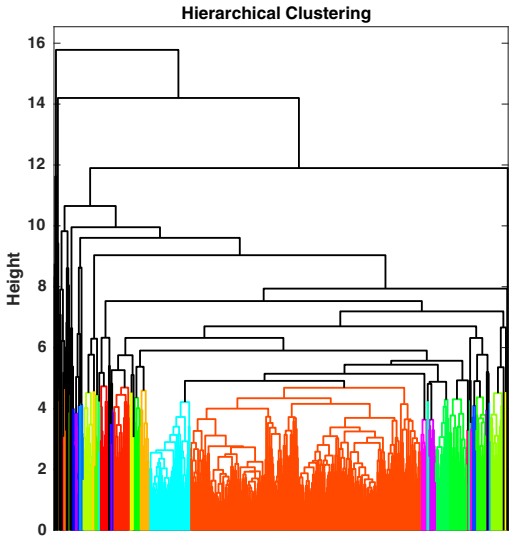

FIG. 2. Dendrogram showing hierarchical clustering of 1000 transient triggers identified in O1 Data from Hanford observatory by the `Omicron` algorithm [32]. The transient morphology changes progressively from left to right.

linkage between the clusters is based on the average distance between pairs of signals among the clusters, weighted by the numbers of elements in each cluster. Cluster linkage at each level of dendrogram is calculated recursively whose value for a given pair of clusters is given by

$$d(r,s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}} \|\tilde{x}_r - \tilde{x}_s\|_2. \tag{8}$$

The optimal distance measure used for linkage and the original mother wavelet used for decomposition are both selected based on the value of the cophenetic correlation coefficient, c [40], with a value close to unity being ideal.

$$c = \frac{\sum_{i<j}(Y_{ij} - y)(Z_{ij} - z)}{\sqrt{\sum_{i<j}(Y_{ij} - y)^2 \sum_{i<j}(Z_{ij} - z)^2}}. \tag{9}$$

Y_{ij} is the pairwise distance between parametrized waveforms while Z_{ij} is their linkage distance. y, z, respectively, represent the average value of the corresponding distance measures.

Optimal leaf ordering of the resulting dendrogram is achieved by maximizing the sum of similarities between adjacent leaves [41]. This step is carried out to identify the relationship between the various clusters and to locate possible subgroups. For example, in Fig. 2, transients at both ends are least related to each other. The schematic of the hybrid classifier useful for real time transient classification is shown in Fig. 3.

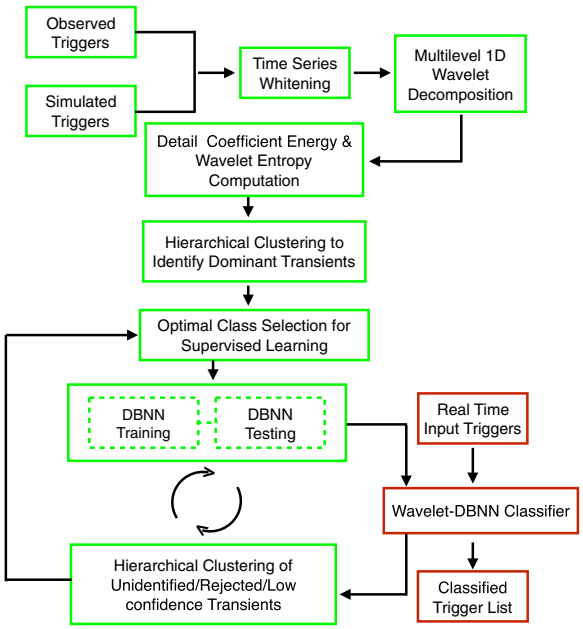

FIG. 3. Block diagram of the proposed hybrid classifier.

V. SIMULATED DATA

A simulated data set consists of 49845 transients from ten classes (refer to Table I) whose SNR is varied uniformly between 8 and 100 by means of Gaussian white noise addition. Signals in each class are generated for different values of parameters sampled from a wide range. Details of bursts used in simulation are given below (t_o is set to 0.5 Sec).

(i) Gaussian (GN)

These broadband nonastrophysical signals are modeled as simple Gaussians with a duration parameter τ taking values 0.0005, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.02, and 0.05 [17].

$$s(t) = \exp\left(-\frac{(t-t_o)^2}{\tau^2}\right)$$

(ii) Sine-Gaussian (SG)

SG models a nonastrophysical glitch which produces significant triggers in matched filtering analysis for coalescing compact binaries [16]. τ is set to $2/f_o$ with a central frequency (f_o) logarithmically spanning from 100 Hz to 2000 Hz.

$$s(t) = \exp\left(-\frac{(t-t_o)^2}{\tau^2}\right)\sin(2\pi f_o(t-t_o))$$

(iii) Ringdown (RG)

RG signals have a longer duration but a shorter bandwidth and are modeled as damped sinusoids. They are produced from quasinormal modes of a final black hole formed from coalescing compact binaries [18]. Here we set $\tau = 4/f_o$ with f_o similar to that of sine-Gaussian data set.

$$s(t) = \begin{cases} s(t) = \exp(-\frac{(t-t_o)}{\tau})\cos(2\pi f_o(t-t_o)) & \text{if } t \geq t_o \\ 0 & \text{if } t \leq t_o \end{cases}$$

(iv) Chirping sine Gaussian (CSG)

CSG is similar to SGs but with an additional chirping parameter [25]. This signal closely models the whistle glitches frequently seen in LIGO detector data. The equation below gives the waveform model where each of the parameter is varied as follows: $f_o : \{5, 100\}$, $\alpha : \{10, 100\}$ and $\tau : \{0.001, 0.025\}$

$$s(t) = \frac{\exp\left(\frac{-(1-1i\alpha)(t-t_o)^2}{(4\tau^2)} + 2\pi i(t-t_o)f_o\right)}{(2\pi\tau^2)^{\frac{1}{4}}}$$

(v) Supernova (SN)

Zwerger-Mueller waveforms [20], one of the Supernova waveforms, are produced by an axisymmetric core collapse of supernovae. These are obtained by hydrodynamical simulations of the stellar core collapse by varying the initial conditions like adiabatic index, spin, and differential rotation profile. We incorporate 78 models (with varying SNR) consisting of a simple analytic equation of state. We also make use of Ott-Burrows supernova waveforms [21] in our analysis.

(vi) Cusp (CSP)

Symmetry breaking phase transitions in the early Universe could generate cosmic strings [22] with a cusplike signal, $h(f) = A(f)f^{-4/3}$. Such waveforms are simulated with an exponential roll off after a cutoff frequency f_o which is varied from 50 Hz to 2000 Hz.

(vii) White noise bursts (WBN)

WBN have in general very complex time-frequency morphology. Their spectra is white in the specified band and zero outside [19]. Here we construct a set of burst signals which have central frequency spanning from 50–300 Hz, bandwidth 50 to 150 Hz and duration 0.1 to 0.4 seconds.

(viii) Black hole merger (LBM)

These waveforms capture the coalescence radiation emitted from a merger of binary black hole systems using Lazarus approach [23]. Analytic approximation [24] is used to construct time domain templates to replicate the merger scenarios. We considered black hole binaries with a chirp mass in range $\{20, 50\}$ and cosine of inclination angle varied between zero and one.

(ix) Blip (Blip)

Blips are observed frequently in both LIGO detectors but their origin is not well understood [28].

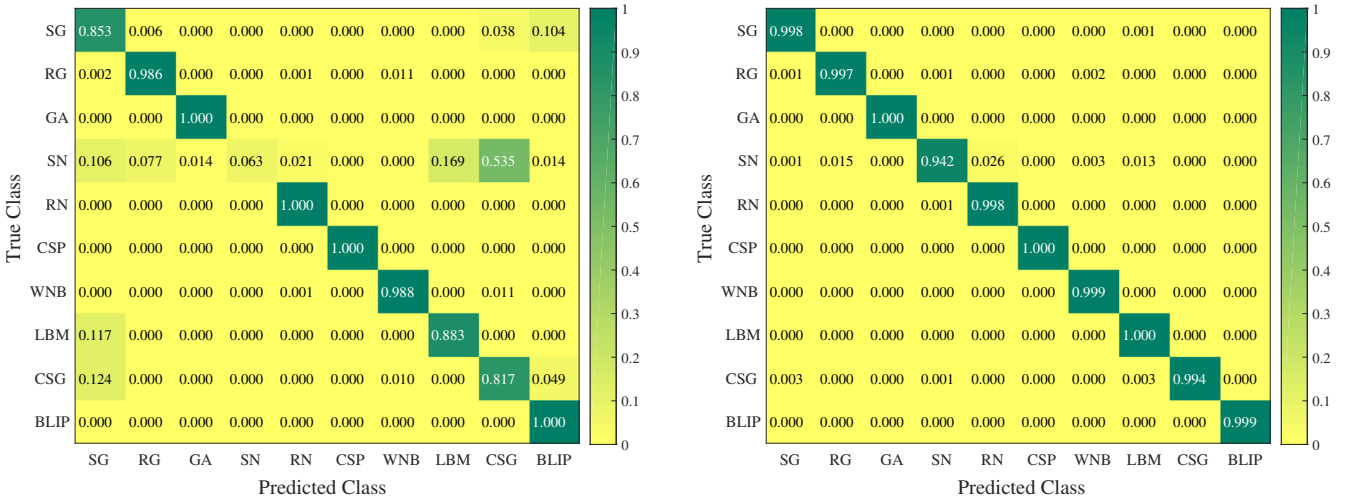

FIG. 4. Confusion matrix for simulated data: Results from traditional SVM (left) and DBNN (right) classifiers. The closer a diagonal element is to unity, the better is the classification for the corresponding type. Accuracy of our method is thus evident.

Hardware injections carried out at the observatories sometimes hit the saturation limit of the actuator resulting in signals which look similar to blips. Hence we simulate them by clipping sine-Gaussians at a few percent level around the mean amplitude.

While sampling the parameters, care was taken to ensure that the signals within a class are significantly different. Table II shows the performance of a Wavelet-DBNN classifier. Total number of samples, size of training set, true positives (TP), false positives (FP), precision, sensitivity, and specificity are reported (see [42] for terms definition). The resulting confusion matrix is shown on the right panel of Fig. 4.

For comparison with a standard classifier, we use publicly available support vector machine (SVM) implementation `LIBSVM` [7,43] on the same wavelet decomposed parameter sets. Figure 4 clearly shows how our Wavelet-DBNN classifier outperforms the traditional classifier. Stark difference is observed for Supernova signals where the SVM shows a very high misclassification, most likely due to the limited number of data samples and the inherent diversity in their morphology.

TABLE II. Simulated transient signals.

Name	Total	Train.	TP	FP	Preci.	Sensi.	Speci.
SG	5000	552	4991	22	0.99	0.99	1.00
RG	5000	311	4984	16	0.99	0.99	1.00
GA	5000	155	5000	0	1.00	1.00	1.00
SN	745	313	702	14	0.98	0.94	1.00
RN	5000	114	4992	19	0.99	0.99	1.00
CSP	5000	14	5000	0	1.00	1.00	1.00
WNB	10000	421	9994	15	0.99	0.99	1.00
LBM	5000	586	4999	31	0.99	1.00	0.99
CSG	5000	378	4969	0	1.00	0.99	1.00
BLIP	4100	153	4097	0	1.00	0.99	1.00

VI. S6 HARDWARE INJECTIONS

To check the performance of our classifier on the real data, we use the classifier on the LIGO strain data obtained from the sixth science run [44]. We apply our classifier to six different classes of hardware injected short duration transient signals as given in Table III. The strain data is whitened to better identify the transients and then down sampled to 4096 Hz. 1634 transients with SNR greater than 10 are used in the analysis. Table III gives the results after classification.

VII. TARGETED SEARCH: LIGO STRAIN CHANNEL

Detector characterization studies revealed several kinds of nonastrophysical transients in the Advanced LIGO's strain data during its first observation run [3]. Identification of these transients and establishment of their nonastrophysical origin were crucial for the detection of GW signal [1–3]. For those known classes, if we could automate their detection using machine learning methods, it will reduce the noise background in the astrophysical GW searches. We used LIGO Hanford Observatory (LHO) strain data (September 18 to January 12) consisting of 28354 transient triggers [32] with SNR ranging from 8 to 100 and

TABLE III. S6 hardware injections.

Name	Total	Train.	TP	FP	Preci.	Sensi.	Speci
SG	1476	69	1476	10	0.99	1.00	0.94
RG	36	25	33	0	1.00	0.92	1.00
GA	46	38	44	8	0.85	0.96	1.00
SN	41	34	33	4	1.00	0.86	1.00
CSP	28	27	24	0	1.00	0.86	1.00
WNB	29	27	24	0	1.00	0.83	1.00

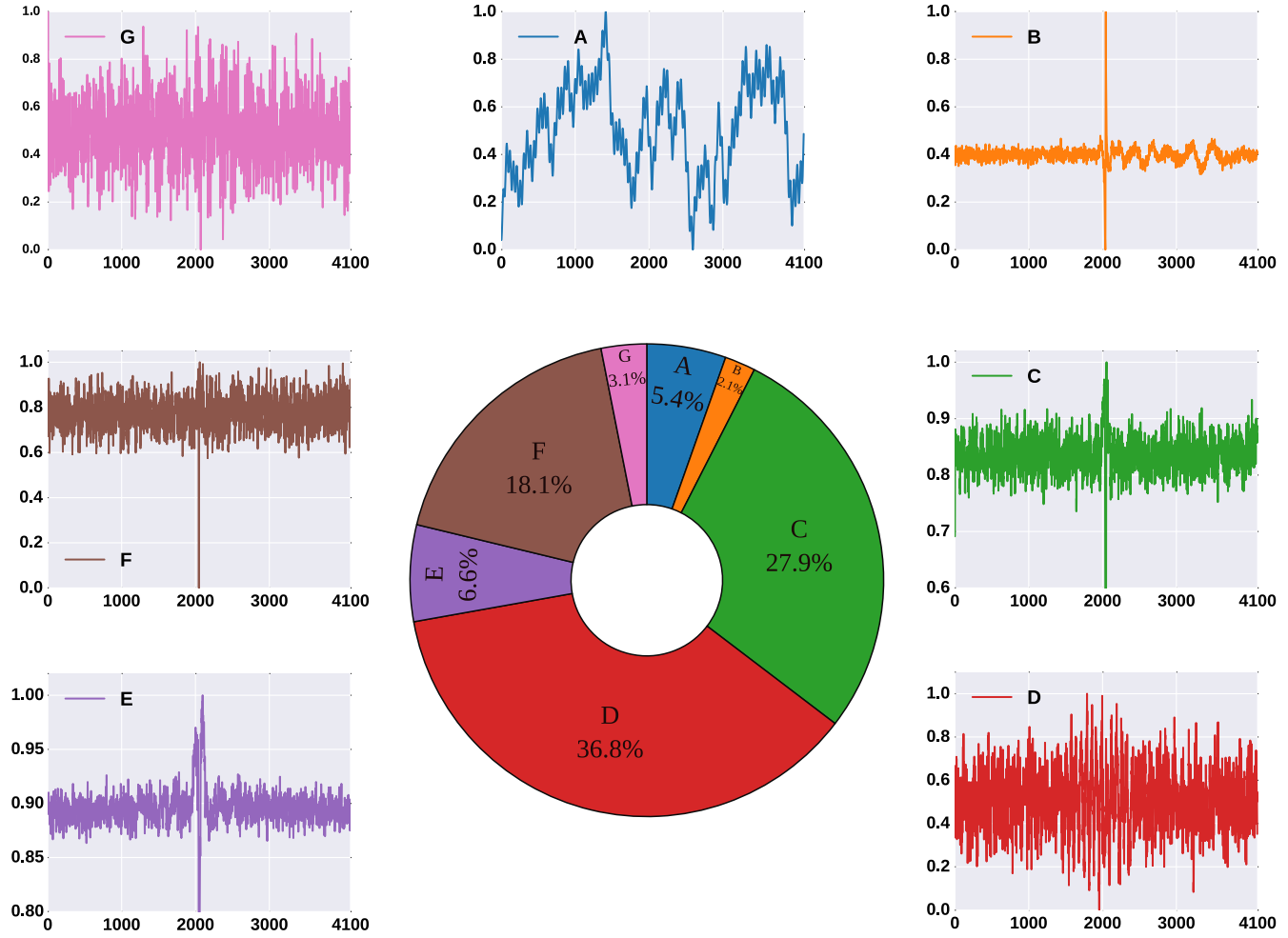

FIG. 5. Donut chart shows the distribution of O1 transients classified by DBNN-Wavelet classifier into the major classes identified by hierarchical clustering. Representative transients from each class (sampled at 4096 Hz) are also shown.

having a maximum frequency of 2096 Hz. Our classifier is then used to search for events which look similar to the major transient classes (see Table I) evident from the initial hierarchical clustering. Several of these could potentially limit generic burst searches, in particular, cosmic cusps and supernova events. One second whitened data around the trigger is used for feature extraction. Training set consisted of minimal samples ranging from five to ten per class. Figure 5 shows the distribution of classified transients with a similar morphology as the training set.

We also did the analysis with LIGO Livingston data for triggers in SNR range 10–100 and obtained comparable results. Classifier found a strong presence of glitches caused due to previously proposed scattering mechanism [26,27]. We observe coincidence (within one second) between some of the classified scattering glitches and the triggers seen in LIGO's auxiliary angular length sensing channels. These auxiliary channels carry information about the motion of signal recycling cavity optics. The coupling was seen to occur predominantly from the pitch and yaw degrees of freedom with a respective contribution of 17.3% and 12.5% with 50% of the glitches coincidentally seen in both the channels. Scattering happens when an off axis beam gets reflected back from the beam tube and recombines with the main beam. These morphology based identification coupled with coincident analysis would help one to narrow down to the region mostly likely to cause the transients and also help in applying appropriate data quality vetoes.

VIII. TARGETED SEARCH: LIGO AUXILIARY CHANNELS

Severe weather conditions can affect both the detectors and, if not properly vetoed, can be misinterpreted as a true signal. Variation in the ambient magnetic field during lightning and a thunderstorm around LIGO can affect sensors and actuators present in multistage suspension systems that isolate and control the LIGO test mass. They are seen in magnetometers with a very distinct time-frequency morphology (Fig. 6, right panel). These also induce currents in the beam tube and are detected by on site clamp meters. Here we apply our classifier to separate

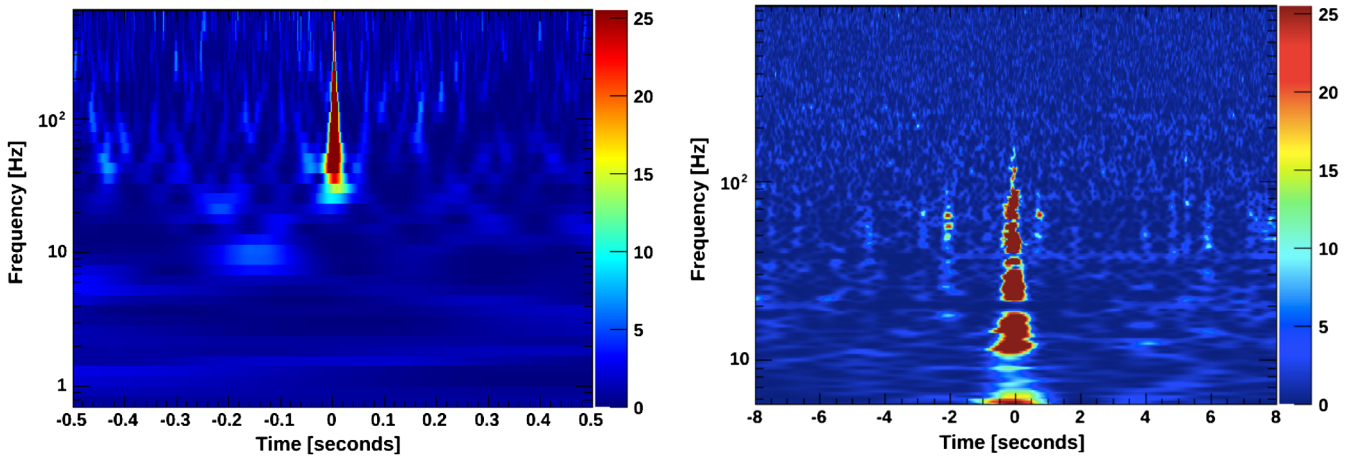

FIG. 6. OmegaScan: Type C glitch in LHO strain channel (left) and lightning glitch in LLO magnetometer (right). Plots generated through LIGOdv-web [45].

out lightning events from the other transients seen in the magnetometer data. We use LIGO Livingston Observatory (LLO) Y-arm magnetometer Omicron triggers generated from 16:00:00 to 23:00:00 UTC of December 16, 2015. Hierarchical clustering on the first 30 minutes of data generates the training set, which is fed to the supervised classifier that performs the final targeted search. Triggers with SNR 15 to 1000 and frequency 1 to 1024 Hz are used for the analysis. 42 out of 689 such triggers are identified to be caused by lightning. Similar search carried out in LLO X-arm magnetometer data for the same period identifies 45 lightning triggers. Our results are consistent with the local weather data, which reported lightning activity during the same period. Number of misclassifications in these cases turned out to be only one and six, respectively.

IX. CONCLUSIONS

We have convincingly demonstrated the resourcefulness of machine learning in detector characterization and burst signal analysis in LIGO like complex instruments. We showed that an effective feature extraction technique, in conjunction with an efficient classifier, can be used to classify a variety of transients in practical situation involving real data. We used relative wavelet energy, wavelet entropy, and kurtosis as a possible parameter set for classifier input. This, coupled with a difference boosting neural network, was very accurate in discerning between classes with slightly different morphology and possibly different physical origin. The usefulness of the method was shown in our analysis where we could do an accurate targeted search for a specific glitch using minimal training sets. The parameter set used here can be expanded to include other features which can aid the classification even when the corresponding values are unavailable for other classes. The special construction of the classifier makes sure that it does not suffer from the curse of dimensionality unlike most neural network classifiers. Hence, the feature set can be expanded in future without causing much computational overhead. Combining class information along with multichannel coincidence analysis will help to narrow down to the cause for a particular kind of transient present in the data. If there is a good enough reason to believe that the trigger is nonastrophysical then glitch based vetoes can be applied to those times. This would lower background triggers in search pipelines thus enhancing confidence in the true detections. We plan to develop such a data quality vector which can be used to directly veto low latency triggers produced by search pipelines looking for astrophysical signals.

ACKNOWLEDGMENTS

We would like to thank the detector characterization working group of the LIGO Scientific Collaboration for useful comments and suggestions. NM acknowledges the Council for Scientific and Industrial Research (CSIR), India for providing financial support as a Senior Research Fellow. SM acknowledges the support of the Science and Engineering Research Board (SERB), India through the fast track Grant No. SR/FTP/PS-030/2012. Authors express thanks to Siddhartha Chatterjee, Arun Aniyan, Shantanu Desai, and Bhooshan Gadre for their valuable comments and suggestions. 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 No. PHY-0757058. This paper has been assigned LIGO Document No. LIGO-P1600094.

[1] B. Abbott *et al.*, Phys. Rev. Lett. **116**, 061102 (2016).
[2] B. Abbott *et al.*, Phys. Rev. Lett. **116**, 241103 (2016).
[3] B. Abbott *et al.*, Classical Quantum Gravity **33**, 134001 (2016).
[4] A. Adams and A. Woolley, Vistas Astron. **38**, 273 (1994).
[5] R. Biswas *et al.*, Phys. Rev. D **88**, 062003 (2013).
[6] C. Cortes and V. Vapnik, Mach. Learn. **20**, 273 (1995).
[7] C.-C. Chang and C.-J. Lin, ACM Trans. Intell. Syst. Technol. **2**, 1 (2011).
[8] L. Breiman, Mach. Learn. **45**, 5 (2001).
[9] J. Powell, D. Trifirò, E. Cuoco, I. S. Heng, and M. Cavaglià, Classical Quantum Gravity **32**, 215012 (2015).
[10] T. Bayes, Phil. Trans. R. Soc. London **53**, 269 (1763).
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Proceedings of the 25th International Conference on Neural Information Processing Systems* (Curran Associates Inc., New York, USA, 2012), p. 1097.
[12] M. Zevin *et al.*, Classical Quantum Gravity **34**, 064003 (2017).
[13] B. S. Sathyaprakash and S. V. Dhurandhar, Phys. Rev. D **44**, 3819 (1991).
[14] S. Klimenko and G. Mitselmakher, Classical Quantum Gravity **21**, S1819 (2004).
[15] S. Klimenko, I. Yakushin, A. Mercer, and G. Mitselmakher, Classical Quantum Gravity **25**, 114029 (2008).
[16] T. D. Canton, S. Bhagwat, S. V. Dhurandhar, and A. Lundgren, Classical Quantum Gravity **31**, 015016 (2014).
[17] B. Abbott, R. Abbott, R. Adhikari, A. Ageev, B. Allen, R. Amin, S. Anderson, W. Anderson, M. Araya, H. Armandula *et al.*, Phys. Rev. D **69**, 102001 (2004).
[18] C. V. Vishveshwara, Nature (London) **227**, 936 (1970).
[19] J. Abadie *et al.*, Phys. Rev. D **85**, 122007 (2012).
[20] T. Zwerger and E. Mueller, Astron. Astrophys. **320**, 209 (1997).
[21] A. Burrows, E. Livne, L. Dessart, C. Ott, and J. Murphy, Astrophys. J. **640**, 878 (2006).
[22] T. Damour and A. Vilenkin, Phys. Rev. Lett. **85**, 3761 (2000).
[23] J. Baker, M. Campanelli, and C. O. Lousto, Phys. Rev. D **65**, 044001 (2002).
[24] J. Baker, M. Campanelli, C. O. Lousto, and R. Takahashi, Phys. Rev. D **65**, 124012 (2002).
[25] S. Bose, S. Dhurandhar, A. Gupta, and A. Lundgren, Phys. Rev. D **94**, 122004 (2016).
[26] T. Accadia *et al.*, Classical Quantum Gravity **27**, 194011 (2010).
[27] D. J. Ottaway, P. Fritschel, and S. J. Waldman, Opt. Express **20**, 8329 (2012).
[28] https://cqgplus.com/2016/06/06/how-do-we-know-ligo-detected-gravitational-waves/.
[29] M. Bhagat, C. Bhushan, G. Saha, S. Shimjo, K. Watanabe, and J. Bhattacharya, PLoS One **4**, e7173 (2009).
[30] T. Jayasree, D. Devaraj, and R. Sukanesh, Int. J. Comp. Electrical Engineering **1**, 590 (2009).
[31] MATLAB and Wavelet Toolbox Release 2013a, *Version 8.1.0.604* (The MathWorks Inc., Natick, Massachusetts, 2013).
[32] F. Robinet, Omicron: An Algorithm to Detect and Characterize Transient Events in Gravitational-Wave Detectors, Technical Report, 2015.
[33] N. S. Philip and K. B. Joseph, Intell. Data Anal. **4**, 463 (2000).
[34] http://www.iucaa.ernet.in/~nspp/dbnn.html.
[35] N. S. Philip, Y. Wadadekar, A. Kembhavi, and K. B. Joseph, Astron. Astrophys. **385**, 1119 (2002).
[36] N. S. Philip, Paladyn **1**, 160 (2010).
[37] S. Abraham, N. S. Philip, A. Kembhavi, Y. G. Wadadekar, and R. Sinha, Mon. Not. R. Astron. Soc. **419**, 80 (2012).
[38] N. S. Philip *et al.*, in *Astronomical Society of India Conference Series*, edited by P. Prugniel and H. P. Singh, Vol. 6 (Bulletin of the Astronomical Society of India, India, 2012), p. 151.
[39] P. C. Mahalanobis, Proc. Natl. Inst. Sci. India. **2**, 49 (1936).
[40] P. Sneath and R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, A Series of Books in Biology (W. H. Freeman, San Francisco, USA, 1973).
[41] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, Bioinformatics **17**, S22 (2001).
[42] D. Powers, School of Informatics and Engineering Technical Reports, 2007.
[43] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring, GNU Octave Version 4.0.0 Manual: A High-Level Interactive Language for Numerical Computations (2015).
[44] J. Aasi *et al.*, Classical Quantum Gravity **32**, 115012 (2015).
[45] J. Areeda and J. Smith, *LigoDV-Web*, https://ldvw.ligo.caltech.edu/ldvw/view.