Automatically Detecting Avalanche Events in Passive Seismic Data

Marc J. Rubin, Tracy Camp
Dept. of Electrical Engineering and Computer Science
Colorado School of Mines
Golden, CO USA
mrubin@mines.edu, tcamp@mines.edu

Alec van Herwijnen, Jürg Schweizer
WSL Institute for Snow and Avalanche Research SLF
Davos, Switzerland
vanherwijnen@slf.ch, schweizer@slf.ch

***Abstract*—During the 2010-2011 winter season, we deployed seven geophones on a mountain outside of Davos, Switzerland and collected over 100 days of seismic data containing 385 possible avalanche events (33 confirmed slab avalanches). In this article, we describe our efforts to develop a pattern recognition workflow to automatically detect snow avalanche events from passive seismic data. Our initial workflow consisted of frequency domain feature extraction, cluster-based stratified subsampling, and 100 runs of training and testing of 12 different classification algorithms. When tested on the entire season of data from a single sensor, all twelve machine learning algorithms resulted in mean classification accuracies above 84%, with seven classifiers reaching over 90%.**

We then experimented with a voting based paradigm that combined information from all seven sensors. This method increased overall accuracy and precision, but performed quite poorly in terms of classifier recall. We, therefore, decided to pursue other signal preprocessing methodologies.

We focused our efforts on improving the overall performance of single sensor avalanche detection, and employed spectral flux based event selection to identify events with significant instantaneous increases in spectral energy. With a threshold of 90% relative spectral flux increase, we correctly selected 32 of 33 slab avalanches and reduced our problem space by nearly 98%. When trained and tested on this reduced data set of only significant events, a decision stump classifier achieved 93% overall accuracy, 89.5% recall, and improved the precision of our initial workflow from 2.8% to 13.2%.

***Keywords*-avalanche, seismic, detection, automated, machine learning**

I. Introduction

This article describes how we used data mining and machine learning algorithms to automatically detect avalanche events from passive seismic data. Before describing our machine learning experimentation and results, we first list our motivations for this work and summarize the previous literature.

A. Motivation

Avalanches pose a significant natural hazard to humans. In terms of human impact, from 1990 to 2011, the United States averaged 25 avalanche fatalities each year, with over 500 total deaths [1]. In addition to the human cost, avalanches also affect U.S. highways. For example, in Colorado, 252 known avalanche paths impact roads, causing approximately 6,000 hours of avalanche mitigation and cleanup each year [2]. Furthermore, the Colorado Department of Transportation (CDOT) "regularly monitors and/or controls" 278 avalanche paths and triggered over 700 avalanches in 2009-2010 alone [2].

Knowing when and where avalanches occur is critical information to avalanche forecasters and mitigation crews [3], [4]. For example, a sudden increase in natural avalanche activity may lead highway mitigation crews to close sections of highway and artificially trigger avalanches using explosives. Though signs of recent natural avalanche activity is one of the best indicators of impending avalanche danger, current observation methods require human interaction, mainly in the form of visual surveillance. During periods of low visibility (e.g., storms, fog, darkness), humans simply cannot view avalanche paths until visibility returns, which may be several hours to many days later. To combat this discrepancy, researchers have and continue to use geophones to detect the seismic energy of avalanches rumbling down an avalanche path (e.g., [5], [6], [7], [8]).

B. Background

Though many researchers have used seismic sensors to detect avalanches, few have explored the use of machine learning and data mining algorithms to automatically and reliably classify when an avalanche event has occurred. Following is a summary of the previous literature involving automatic avalanche event detection from seismic data.

The first set of previous research focuses on the SARA (System for Avalanche Recognition Analysis) software suite, which uses an expert system approach based on fuzzy logic rules to detect avalanches from preprocessed seismic signals [9], [10], [11], [12]. As detailed in [10], the fuzzy logic rules were not created automatically by any type of rule extraction technique (e.g., RIPPER [13]). Instead, the authors used their "expert knowledge" from the analysis of 308 unambiguously identified seismic events to derive the properties, parameters, and boundaries for each fuzzy logic rule. The rules are mutually exclusive and use features from the time and time-frequency domains to identify the type of seismic events (e.g., helicopter, earthquake, thunder, avalanche, etc.). Though this detection scheme obtained 90%

978-0-7695-4913-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMLA.2012.12

accuracy during testing, the main drawback of SARA is that the fuzzy logic rules are based on manual analysis of previous data; in other words, the rules are not derived automatically or in a timely manner. The authors of [10] acknowledge this drawback, and even state that users must modify the rules to adapt to other sites; however, modifying rules for each site is impractical as it requires significant time and expert knowledge.

In an attempt to automate the training phase of avalanche event classification, [14] used a distanced weighted k-nearest neighbor approach (k = 3) on previously recorded seismic data. Over 10 years of seismic data was recorded from geophones installed in gullies along a dangerous stretch of highway on the coast of Iceland. These geophones recorded many seismic events, including rockfalls, landslides, traffic, and avalanches. The classification results of this automated method were unsatisfactory; specifically, only 65% (78 of 119) of all avalanches were correctly identified using their k-nearest neighbor algorithm approach. This work suggests that there is considerable room for improvement.

To the best of our knowledge, there is no other work that attempts to automatically train classification algorithms to successfully and reliably detect avalanche events from passive seismic data. Our contributions are (1) a thorough description of the preprocessing we do for feature extraction and (2) a successful pattern recognition workflow to automatically detect avalanches in passive seismic data with 93% overall accuracy and 13.2% precision.

II. Passive Seismic Dataset

A. Raw Data

During the winter of 2010-2011, we buried seven geophone sensors five to 10m apart in a snow slope in an active avalanche region near Davos, Switzerland (Figure 1). The geophones recorded over 100 days of seismic data with 24-bit precision at a sampling rate of 500 Hz. Thus, our classification experiments deal with a very large data set; i.e., each day contains roughly 43 million 24-bit samples of seismic data and the entire avalanche season is more than four billion 24-bit samples per sensor. More information regarding our deployment can be found in [5].

B. Identifying Avalanche Events

In addition to installing geophones, we also deployed a weather station, acoustic microphone, and automatic camera, which took two images of the adjacent slopes every five minutes (e.g., Figure 2). After manually analyzing all consecutive 30 minute spectrograms from a single sensor, camera images, and microphone recordings, we identified 385 potential avalanche events. It is important to note that, despite the microphone and camera systems, there is little ground truth validation of the data set. This was caused by a variety of reasons including poor visibility, lousy microphone data, and camera downtime. In particular, of the 385 events, only 33 were confirmed slab avalanches while the remaining 352 were assumed to be small sluff events. Briefly, a sluff is a small surface slide of newly fallen powder snow and is the least dangerous type of avalanche [3], [4]. Slab avalanches, on the other hand, involve large volumes of fast moving, densely packed snow and are known to kill people, destroy structures, and bury roads. It is important to note that the primary goal for this pattern recognition task is to reliably identify the large (confirmed) slab avalanches; correctly identifying the small sluff events is a bonus.

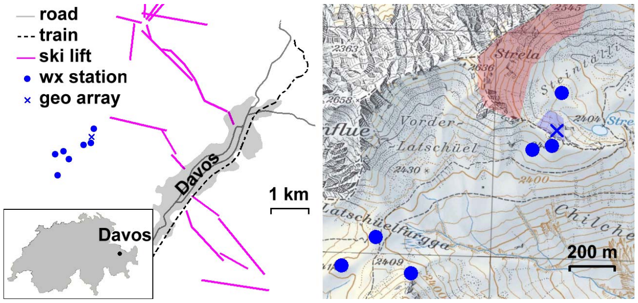


Figure 1: The geophone array (blue 'x') was installed in a snow slope near Davos, Switzerland. The red shaded portion represents the field of view for two automatic cameras.

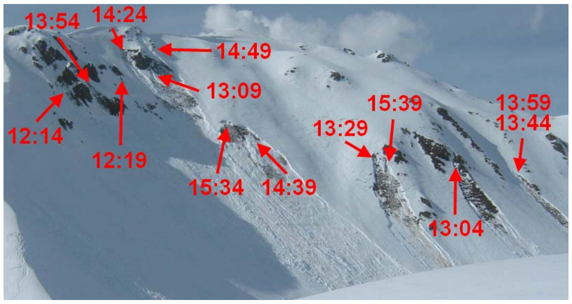


Figure 2: Camera images were sometimes used to help confirm the avalanches events (e.g., avalanches from 23 March 2011).

Each assumed avalanche event has an estimated start time and length in the passive seismic data. From this data we constructed the classes (i.e., avalanche vs. non-avalanche) for our classification experiments. The avalanches ranged from three seconds to nearly two minutes in length, with a mean length of roughly 25 seconds. We note that the avalanche events only occur approximately 0.2% of the time; in other words, avalanches do not occur approximately 99.8% of the time. We also note that the non-avalanche events are noisy; specifically there is background and spurious noise caused by wind, helicopters, ski lifts, snow cats, airplanes, and earthquakes (Figure 3).

Though the avalanche events are anomalies in terms of the number of occurrences, we found that statistical and clustering based anomaly detection did not work well with our data set. This result is most likely due to the fact

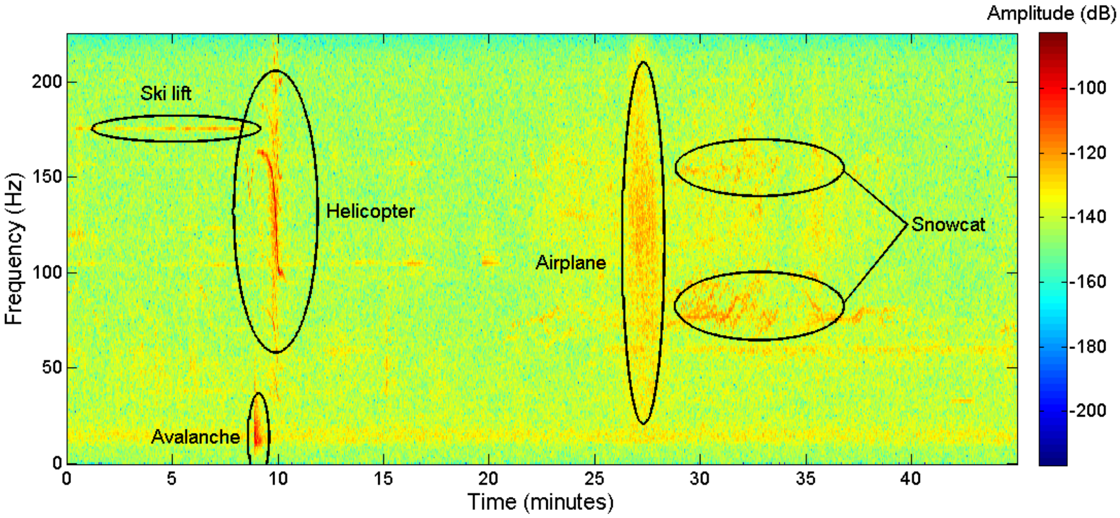


Figure 3: Seismic signals generated by various noises in the environment and by a confirmed avalanche.

that avalanches were not statistical outliers in the data. We, therefore, experimented with a one-class support vector machine (OCSVM) to detect avalanches, similar to how [15] used an OCSVM to detect anomalous network intrusions. Specifically, we trained and tested an OCSVM using two-fold cross-validation on data from the entire season that was processed using the feature extraction procedure discussed in Section III. Though the results of the OCSVM (Table I) were better than both the statistical and clustering based methods (which both performed no better than random chance), the classification results were much worse than our pattern recognition workflow discussed next.

One-Class Support Vector Machine				
Accuracy[1]	Recall[2]	Specificity[3]	Precision[4]	AUC [5]
0.661	0.649	0.661	0.004	0.345

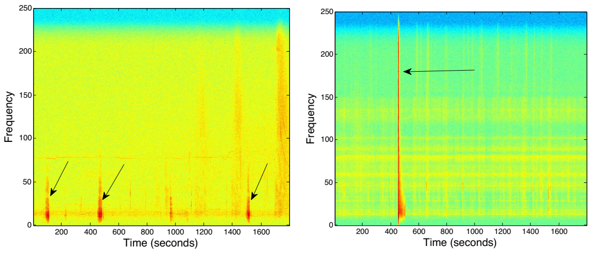
Table I: Results from two-fold cross-validation of a one-class support vector machine (OCSVM) trained and tested on all five-second windows within the entire seismic data set.

III. Feature Extraction

In data mining and machine learning research, the majority of time is spent determining the best features to extract to maximize class separability. This section specifies how we processed the raw seismic data into individual frames containing 10 features appropriate for classification.

A. Avalanches in the Frequency Domain

A windowed 2048-bin fast Fourier transform (FFT) was used to convert the data from the time to frequency domain. Based on our observations and previous research, avalanches tend to have high energy in the low frequency bins of the spectral domain (Figure 4a). Unfortunately, increased spectral power in low frequency bins did not always equate to an avalanche, as other non-events (e.g., earthquakes, wind) had significant low frequency amplitudes as well. Additionally, many avalanche events had significant high frequency content and wide spectral traces (Figure 4b). Needless to say, the ambiguity and variability of our dataset leads naturally to using automated machine learning algorithms to generate a pattern recognition model.



(a)

(b)

Figure 4: (a) Avalanches tend to have a strong presence in the low frequency bins of the spectral domain. (b) However, many avalanche events had significant energy in high frequency bins as well.

B. Signal Processing and Feature Generation

We extracted 10 features from the frequency domain using an open-source Matlab toolbox (i.e., MIRToolBox [16]) designed for audio signal processing. Each frame was labeled as either an avalanche or non-avalanche, depending on the location of the frame to each suspected avalanche event. Each frame was transformed from the time to frequency domain using a 2048-bin, non-overlapping, windowed FFT.

[1] The percentage of all frames correctly predicted.

[2] The percentage of avalanche frames correctly predicted.

[3] The percentage of non-avalanche frames correctly predicted.

[4] The percentage of frames labeled as avalanches that were actually avalanches.

[5] The **A**rea **U**nder the ROC (Receiver Operating Characteristic) **C**urve.

Similar to signal processing techniques commonly found in music signal processing, several spectral features were calculated for each frame (Table II), including the centroid, 85% rolloff, kurtosis, spread, skewness, regularity, and flatness [17].

Source	Features
Spectral Frame	centroid, 85% rolloff, kurtosis, spread, skewness, regularity, flatness
Top 1% Energy Bins	maximum, mean, standard deviation

Table II: The features we extracted all came from the frequency domain. After calculating the spectral features, the frequency spectrum was sorted based on amplitude and the top 1% most energetic frequency bins were used.

The seven spectral frame features attempt to create a quantifiable summary of the entire spectral frame. The spectral centroid represents the "center of gravity" of the spectrum, and is a metric often used in music signal processing to measure timbre [17]. 85% rolloff is the frequency at which 85% of all spectral energy is below. Spectral kurtosis is often described as the "peakiness" of the spectrum, with higher kurtosis meaning more jagged peaks. Spectral spread is simply the standard deviation of the frequency domain. Skewness measures the symmetry of the spectrum. Regularity measures the variability between peaks. Lastly, spectral flatness is a measure of how "flat" the data is, and is calculated as a ratio between the geometric and arithmetic mean.

In addition to the seven features calculated from the entire spectral frame, we sorted the spectral powers based on amplitude and selected the top 1% most energetic frequency bins for further processing. From these top 1% most powerful frequencies, we calculated the maximum, mean, and standard deviation (Figure 5).

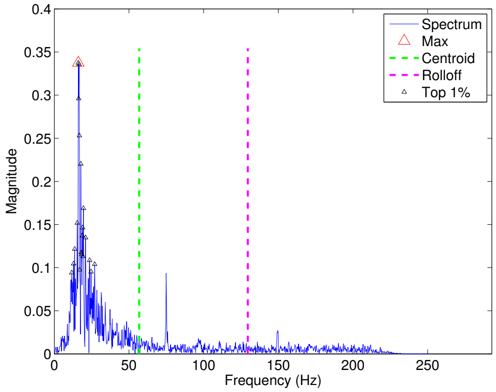


Figure 5: For each five second frame, we calculated several features from the frequency spectrum (2048-bin FFT). This event is a confirmed slab avalanche recorded on 16 January, 2011.

It is important to note that our experimentation with time domain features (e.g., ratio of short-term over long-term average amplitude) proved fruitless, and will not be discussed further. In addition, due to the limited spatial distribution of the seven geophone sensors (i.e., five to 10m apart) and relatively slow sampling rate (i.e., 500 Hz), we could not reliably calculate information about the seismic waveforms (e.g., arrival times, velocity, back azimuth, etc.).

IV. Avalanche Pattern Recognition

In this section we describe our workflow to detect avalanches from passive seismic data. This section is divided into four parts. First we describe cluster-based subsampling used to help offset the deleterious effects of class imbalance. Second, we provide details of our experimentation with 12 different machine learning algorithms on seismic data from the entire season. Third, we test a voting based paradigm that combined information from all seven sensors. Finally, we explore using spectral-flux based event selection to decrease the problem space and increase the overall performance of our classifier.

A. Cluster Based Subsampling

Before we could begin training and testing, we first addressed the issue of extreme class imbalance. In our entire data set, of the 1,264,648 total 5 second frames, only 2,293 frames contain possible avalanches. Thus, our class distributions were 99.82% non events and 0.18% avalanche events. Classifiers trained and tested according to these imbalanced class distributions could naively obtain over 99% overall accuracy by always selecting the non-event class [18], [19], [20]. In such an extreme case, the classifier will be overtrained to recognize the majority class and will be unable to detect the minority class.

To overcome the issue of class imbalance, we employed a clustering based subsampling methodology (e.g., Figure 6). More specifically, we used K-Means clustering to organize the non-avalanche event data into seven different groups. The number of groups was based on the number of known noise events, i.e., planes, helicopters, ski lifts, snow cats, wind, earthquakes, and miscellaneous. Before each iteration of the classifier training and testing, we used stratified cluster-based subsampling to help generate the training and testing datasets. Compared to random subsampling, this method insured that our subsampled data proportionally represented the majority class accurately [13], [20].

B. Full Season Avalanche Classification

We trained and tested 12 different classification algorithms on seismic data from the entire season. More specifically, we trained each classifier on a subset of the passive seismic data from a single sensor and tested on the entire season (excluding the data used for training). The training data consisted of 10% of the 2293 avalanche frames (randomly selected) and an equal number of non-avalanche frames selected via the stratified cluster based subsampling method

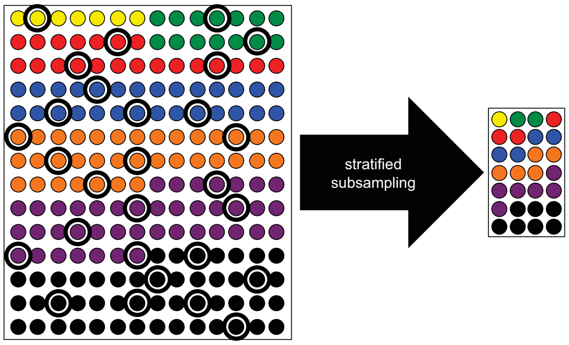


Figure 6: Conceptual diagram of cluster-based stratified subsampling. The colors represent possible groups formed during K-means clustering. Of the seven groups in this example, the data is subsampled using stratified (proportional) subsampling, which helps guarantee that the subsampled data accurately represents the majority class.

Algorithm	Accuracy	Recall	Specificity	Precision	AUC
ANN	0.879	0.846	0.879	0.012	0.863
Bayes	0.875	0.831	0.875	0.011	0.853
BayesNet	0.944	0.779	0.944	0.023	0.862
CART	0.914	0.809	0.914	0.017	0.861
Fuzzy	0.842	0.857	0.842	0.009	0.850
GaussProc	0.922	0.827	0.922	0.017	0.874
J48	0.883	0.842	0.883	0.013	0.862
KNN	0.870	0.791	0.870	0.010	0.830
RandForest	0.943	0.818	0.943	0.023	0.881
RIPPER	0.924	0.812	0.924	0.019	0.868
STUMP	0.951	0.770	0.951	0.028	0.861
SVM	0.913	0.833	0.914	0.016	0.873

Table III: Mean results from testing 100 iterations of 12 machine learning algorithms on a single sensor's data. Each algorithm was trained on 10% of the avalanches and an equal number of non-avalanche events selected using cluster-based subsampling. Testing occurred on all data not used for training.

described previously. This process was simulated 100 times for each algorithm. Training and testing was performed using a combination of open-source data mining tools (i.e., KNIME and WEKA [21], [22]).

For the classification experiments, we tested 12 different classifiers: i.e., artificial neural network (ANN), naive Bayes, Bayes network, CART tree, fuzzy logic rules, Gaussian processes (GaussProc), J48 Tree, k-nearest neighbors (KNN), random forest (RandForest), RIPPER, decision stump, and support vector machine (SVM). This variety was chosen because they represent a wide spectrum of classification algorithms, from probabilistic and statistical (i.e., Bayes, Gaussian processes) to highly non-linear (i.e., ANN, SVM). Furthermore, rule-based classifiers (i.e., fuzzy logic, RIPPER) were used to compare our results with the previous machine learning work done by [9], [10], [11], [12]. We chose to use a KNN (k=7) because it represents a simple distance based classifier and was the algorithm of choice in [14]. Lastly, several types of decision trees were tested (i.e., stump, CART, J48, random forest) because the model's decision boundaries can be interpreted by humans.

The results are quite promising (Table III): i.e., all algorithms reached 84% mean overall classification accuracy or above. Additionally, all classifiers reported mean recall above 77%, meaning that the classifiers could discern the avalanche events.

Though these results show that the dataset is indeed classifiable, the low precision leaves room for much improvement. To reiterate, a classifier with 1% precision will identify 99 false positive avalanches for every one true positive avalanche. It is important to note that neither [10] nor [14] report the precision of their classification models. In Section IV-D, we investigate ways to improve precision by using changes in spectral flux to select events before classification.

We also experimented with a voting based paradigm, where information from all seven sensors was shared to inform classification. In other words, independent classifiers were trained on each sensor, and voting was used during testing to inform the class label. We describe this paradigm next.

C. *Voting Based Avalanche Detection*

Since data was collected concurrently from seven adjacent geophone sensors, we tested a voting based classification paradigm to increase the precision of our model. That is, we trained random forest classifiers on each of the seven sensors (using the same methodology detailed previously) and, for each five-second frame, tallied the number of avalanche labels. Thus, each frame had zero to seven potential "avalanche" votes. For example, if sensor2 and sensor5 identified an avalanche in frame 17, there would be two votes for classifying an avalanche in frame 17.

We evaluated thresholds for avalanche detection from one to seven votes, and our results are presented in Table IV and Figure 7. With increasing votes, our combined classifier approach obtained 99% overall accuracy and 13.4% precision, which indicates a better ratio of false positives to true positives compared to single sensor classification (e.g., Table III). At the same time, however, increasing votes led to decreasing recall (true positive rate). Such poor classifier recall results (i.e., 48.7%) results led us to further experiment with signal preprocessing, as detailed in the next section.

Voting with Random Forest Classifiers					
Votes	Accuracy	Recall	Specificity	Precision	AUC
1	0.625	0.949	0.624	0.002	0.786
2	0.851	0.888	0.850	0.005	0.869
3	0.933	0.815	0.933	0.011	0.874
4	0.967	0.740	0.967	0.020	0.854
5	0.984	0.674	0.984	0.037	0.829
6	0.992	0.601	0.994	0.071	0.797
7	0.997	0.487	0.997	0.134	0.742

Table IV: Results from testing the voting based classification scheme over the entire season.

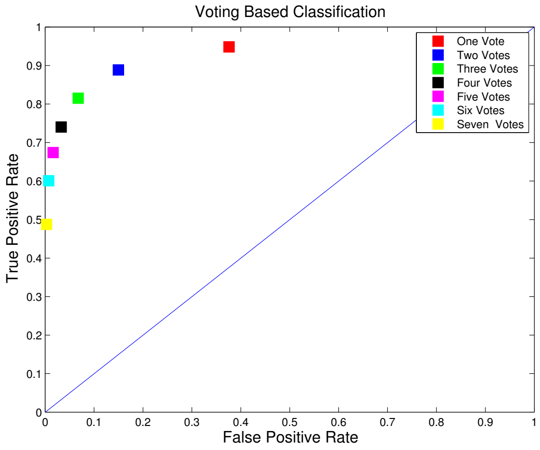


Figure 7: Results from testing our voting based classification scheme over the entire season. A perfect classifier is point (0,1). The blue line indicates random classification.

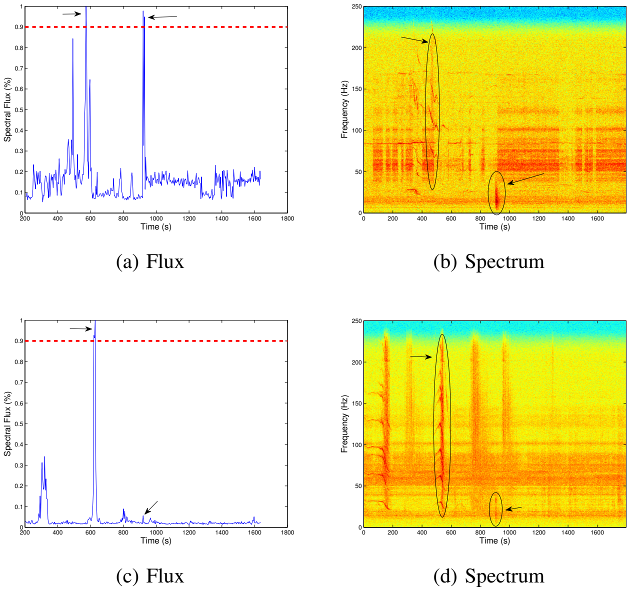
D. Automated Event Selection and Detection

To improve the overall performance of our classification model, we used a more intelligent way to select events before classification. Instead of naively training and testing on *all* data from the entire season, we used spectral flux based event selection to extract only the events with instantaneous increases in spectral energy. This technique is often used in music signal processing to extract beat patterns for automated transcription [17].

Spectral flux is simply the Euclidean distance between two successive spectral frames. In our workflow, we calculated a 2048-bin FFT for each five second frame and determined the Euclidean distance (spectral flux) between consecutive frames. An event was selected if there was an instantaneous increase of spectral flux above a predetermined threshold.

In our workflow, we used spectral flux to select events as follows. First we divided the data into consecutive five minute subsets. Second, for each five minute subset of data, we calculated the spectral flux between all consecutive five second spectral frames. Third, we normalized the spectral flux by dividing each value by the maximum flux for that five minute subset. Lastly, we selected the five second frames with normalized spectral flux values above a predetermined threshold (Figure 8). We experimented with thresholds from 10% to 90% instantaneous increase in spectral flux (Table V).

For an entire seasons worth of data, using spectral flux to select interesting events resulted in a 97.6% reduction in the problem space: from 1,264,648 to 30,822 five second frames. Our results show that with a spectral flux threshold of 90%, we selected 97% (32 of 33) slab avalanches and 70% (246 of 352) of the smaller sluffs. It is important to note that avalanche forecasters care mostly about automatically identifying the confirmed slab avalanches; in other words, it is ok for our automated workflow to miss some of the smaller sluffs in favor of improved results on detecting slabs.



(a) Flux

(b) Spectrum

(c) Flux

(d) Spectrum

Figure 8: Spectral flux based event selection was used to identify only significant events. The red dotted line represents the 90% threshold. (a), (b) An unknown event around time 600 and confirmed slab avalanche at time 900. (c),(d) An unknown event around time 600 and unconfirmed sluff avalanche at time 900 (from a different date).

Threshold	Slabs (%)	Sluffs (%)	Total Events
0.10	97	96	251,391
0.20	97	93	234,063
0.30	97	87	201,290
0.40	97	83	165,563
0.50	97	81	129,816
0.60	97	77	96,207
0.70	97	73	67,301
0.80	97	72	45,599
0.90	**97**	**70**	**30,822**

Table V: Results from experiments testing the spectral flux based event selection. With a strict threshold of 90% instantaneous increase in spectral flux, we select 97% (32 of 33) slab avalanches and 70% (246 of 352) of smaller sluffs.

After using spectral flux to select events of interest, we trained and tested all 12 classifiers on each of the nine subsets of data (one for each of the nine thresholds). The classification experiments used the same workflow as described previously. The results show that by increasingly limiting our problem space to only the most significant spectral events, i.e., higher thresholds, we increased the precision of all classification models (e.g., Table III vs. Table VI).

The best performing model was the decision stump classifier, reaching 93.0% accuracy, 89.5% recall, and 13.2% precision as threshold increased (Table VII, Figure 9). Though

Algorithm	Accuracy	Recall	Specificity	Precision	AUC
ANN	0.903	0.881	0.904	0.081	0.893
Bayes	0.913	0.903	0.913	0.104	0.908
BayesNet	0.904	0.899	0.904	0.079	0.902
CART	0.924	0.897	0.924	0.123	0.910
Fuzzy	0.897	0.905	0.897	0.082	0.901
GaussProc	0.924	0.912	0.924	0.099	0.918
J48	0.909	0.886	0.909	0.107	0.898
KNN	0.814	0.828	0.814	0.036	0.821
RandForest	0.921	0.895	0.921	0.102	0.908
RIPPER	0.916	0.888	0.916	0.117	0.902
STUMP	0.930	0.895	0.930	0.132	0.913
SVM	0.929	0.897	0.929	0.105	0.913

Table VI: Mean results from testing 100 iterations of 12 machine learning algorithms on a single sensor's data when using spectral flux based event selection with a 90% threshold. Each algorithm was trained on 10% of the avalanches and an equal number of non-avalanche events selected using cluster-based subsampling. Testing occurred on all data not used for training.

the accuracy decreased slightly when compared to previous tests (Table III), the increased recall and precision makes our model much more reasonable.

Decision Stump Classifier					
Threshold	Accuracy	Recall	Specificity	Precision	AUC
0.1	0.941	0.854	0.941	0.039	0.898
0.2	0.937	0.858	0.938	0.040	0.898
0.3	0.948	0.851	0.948	0.051	0.899
0.4	0.939	0.868	0.939	0.052	0.904
0.5	0.945	0.881	0.945	0.060	0.913
0.6	0.937	0.877	0.937	0.072	0.907
0.7	0.934	0.879	0.934	0.089	0.907
0.8	0.935	0.882	0.935	0.111	0.909
0.9	0.930	0.895	0.930	0.132	0.913

Table VII: Mean results for a decision stump classifier trained and tested on events selected using spectral flux based event selection with different thresholds.

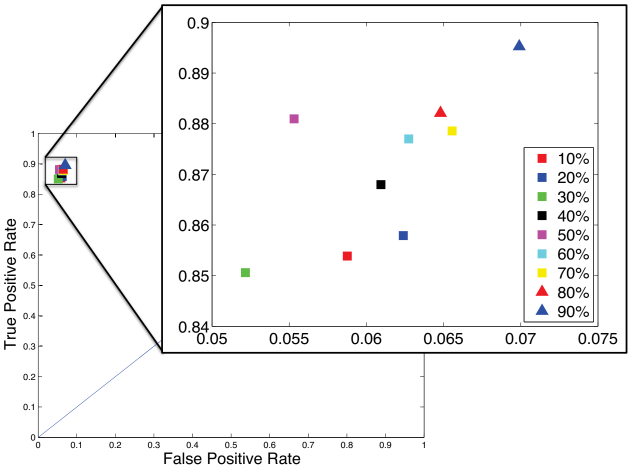


Figure 9: Results from training and testing a decision stump classifier using various thresholds for the spectral flux based event selection. The blue line represents a random classifier, and point (0,1) would be a perfect classifier.

We note that performing event selection on all seven sensors and employing voting based classification proved unsuccessful; in other words, there was little consistency of selected events between all sensors. That is, an event selected as an avalanche by sensor one would very rarely be selected from a different sensor. Thus, because few events were shared between all sensors, voting could not be conducted effectively.

E. Observation: Reducing the Majority Class

Our results indicate that using spectral flux to select only the large events helped improve the precision of each classifier. We hypothesize that this improvement in classifier precision is due to the significant reduction in the number of majority (non-avalanche) class objects (see Table V). To test our hypothesis, we trained and tested a classifier 100 times using different proportions of the majority class in the test data sets. In other words, we trained a decision stump classifier using the stratified cluster-based subsampling method described previously and tested it using data sets with subsequently fewer randomly selected majority class objects (e.g., 10%, 20%, ..., 90% of original), ultimately testing with balanced test sets containing the same number of avalanche and non-avalanche events (99.83% reduction). It is important to note that we did not employ spectral flux based event selection in this effort.

The results (Figure 10) show the effects of reducing the majority class in the test dataset, and reveal a strong relationship between reducing the size of the majority class and increasing classifier precision. Our results demonstrate the importance of balancing test datasets before attempting classification, in order to decrease the number of false positives. More specifically, we should utilize techniques to help reduce the majority class before classification.

V. Conclusions

In this article, we present a pattern recognition workflow to automatically detect avalanche events from passive seismic data. Previous attempts to detect avalanches from passive seismic data either required human experts to develop fuzzy logic rules [9], [10], [11], [12], or failed to reach favorable classification accuracies [14]. This work improves upon the current literature by reaching 93% classification accuracy (with over 13% precision) using automated machine learning algorithms to train classification models.

Though we obtained successful classification accuracies, the relatively low precision rates of our model indicates that there is still room for improvement. First, we intend to experiment with features beyond the frequency domain (e.g., time-frequency domain of Gabor atoms). Second, we will investigate adaptive windowing techniques such that the events selected could be short or longer than five seconds. Lastly, in future (planned) geophone deployments, the sensors will be spaced further apart (e.g., 30m), allowing

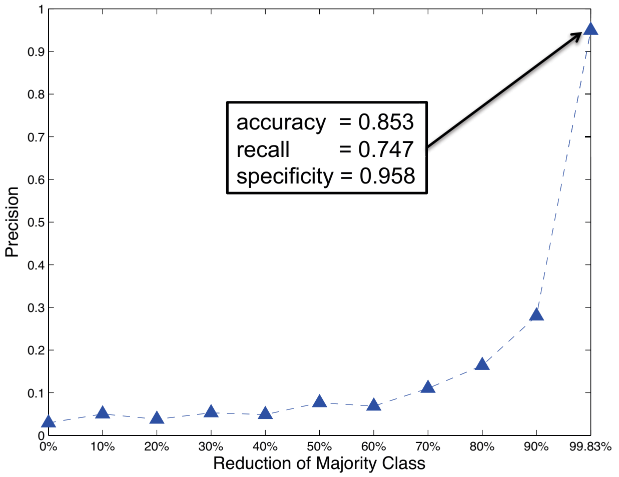


Figure 10: Reducing the size of the majority class during testing led to increased precision (and specificity) for our decision stump classifier trained using our pattern recognition workflow; however, we note accuracy and recall results drop dramatically. We did not employ spectral flux based event selection in this work.

us to estimate seismic wave arrival times and calculate the back azimuth.

Acknowledgment

This work is supported in part by NSF Grant DGE-0801692.

References

[1] Colorado Avalanche Information Center, "Accidents: Statistics," http://avalanche.state.co.us/acc/acc_us.php, 2010, Retrieved 07-20-2012.

[2] Colorado Department of Transportation, "Transportation facts 2011," http://www.coloradodot.info/library/FactBook/FactBook2011/view, 2011, Retrieved 07-20-2012.

[3] B. Tremper, *Staying Alive in Avalanche Terrain*, The Mountaineers Books, Seattle, WA, 2008.

[4] D. McClung and P. Schaerer, *The Avalanche Handbook*, The Mountaineers Books, Seattle, WA, 2006.

[5] A. Herwijnen and J. Schweizer, "Monitoring avalanche activity using a seismic sensor," *Cold Regions Science and Technology*, vol. 69, no. 2-3, pp. 165–176, 2011.

[6] B. Biescas, F. Dufour, G. Furdada, G. Khazaradze, and E. Suriñach, "Frequency content evolution of snow avalanche seismic signals," *Surveys in Geophysics*, vol. 24, pp. 447–464, 2003.

[7] K. Nishimura and K. Izumi, "Seismic signals induced by snow avalanche flow," *Natural Hazards*, vol. 15, pp. 89–100, 1997.

[8] E. Suriñach, G. Furdada, F. Sabot, B. Biescas, and J. Vilaplana, "On the characterization of seismic signals generated by snow avalanches for monitoring purposes," *Annals of Glaciology*, vol. 32, no. 7, pp. 268–274, 2001.

[9] B. Leprettre, J.-P. Navarre, and A. Taillefer, "First results from a pre-operational system for automatic detection and recognition of seismic signals associated with avalanches," *Journal of Glaciology*, vol. 42, no. 141, pp. 352–363, 1996.

[10] B. Leprettre, N. Martin, F. Glangeaud, and J.-P. Navarre, "Three-component signal recognition using time, time-frequency, and polarization information– application to seismic detection of avalanches," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 83 –102, 1998.

[11] B. Leprettre, J.-P. Navarre, J.M. Panel, F. Touvier, A. Taillefer, and J. Roulle, "Prototype for operational seismic detection of natural avalanches," *Annals of Glaciology*, vol. 26, pp. 313–318, 1998.

[12] J.-P. Navarre, Ekatherina Bourova, Jacques Roulle, and Deliot Yannick Delio, "The seismic detection of avalanches: an information tool for the avalanche forecaster," *Proceedings of the International Snow Science Workshop*, 2009.

[13] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley, Boston, MA, 2005.

[14] B. Bessason, G. Eiriksson, O. Thorarinsson, A. Thorarinsson, and S. Einarsson, "Automatic detection of avalanches and debris flows by seismic methods," *Journal of Glaciology*, vol. 53, no. 182, pp. 461–472, 2007.

[15] Y. Wang, J. Wong, and A. Miner, "Anomaly intrusion detection using one class SVM," *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop*, 2004.

[16] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," *Proceedings of the 10th Int. Conference on Digital Audio Effects*, 2007.

[17] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[18] A. Orriols and E. Bernadó-Mansilla, "The class imbalance problem in learning classifier systems," *Proceedings of the 2005 Workshop on Genetic and Evolutionary Computation*, pp. 74–78, 2005.

[19] V. Garcia, J. Sanchez, R. Mollineda, R. Alejo, and A. Sotoca, "The class imbalance problem in pattern classification and learning," *Congreso Español de Informática*, pp. 283–291, 2007.

[20] S. Yen and Y. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, pp. 5718–5727, 2009.

[21] M. Berthold, N. Cebron, F. Dill, T. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "KNIME: The Konstanz Information Miner," *Studies in Classification, Data Analysis, and Knowledge Organization*, 2007.

[22] P. Reutemann, B. Pfahringer, and E. Frank, "A toolbox for learning from relational data with propositional and multi-instance learners," *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, vol. 3339, pp. 1017–1023, 2004.