IDAHO TRANSPORTATION DEPARTMENT

RESEARCH REPORT

RP 219

Real Time Avalanche Detection for High Risk Areas

By

Scott Havens,

Dr. Hans-Peter Marshall,

Gabriel Trisca,

Dr. Jeffery B. Johnson

Boise State University

Prepared for

Idaho Transportation Department

Research Program

Contracting Services

Division of Engineering Services

http://itd.idaho.gov/highways/research/

December 2014

Standard Disclaimer

This document is disseminated under the sponsorship of the Idaho Transportation Department and the United States Department of Transportation in the interest of information exchange. The State of Idaho and the United States Government assume no liability of its contents or use thereof.

The contents of this report reflect the view of the authors, who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official policies of the Idaho Transportation Department or the United States Department of Transportation.

The State of Idaho and the United States Government do not endorse products or manufacturers. Trademarks or manufacturers' names appear herein only because they are considered essential to the object of this document.

This report does not constitute a standard, specification or regulation

.

1. Report No. FHWA-ID-14-219	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Real Time Avalanche Detection for High Risk Areas		5. Report Date December 2014 6. Performing Organization Code
7. Authors Scott Havens, Dr. Hans-Peter Marshall, Gabriel Trisca, Dr. Jeffery B. Johnson		8. Performing Organization Report No.
9. Performing Organization Name and Address Boise State University 1910 University Ave Boise ID 83706		10. Work Unit No. (TRAIS) 11. Contract or Grant No. RP219
12. Sponsoring Agency Name and Address Idaho Transportation Department Division of Engineering Services, Contracting Services, Research Program PO Box 7129 Boise, ID 83707-7129		13. Type of Report and Period Covered Final Report 09/01/2012 - 12/1/2014 14. Sponsoring Agency Code
15. Supplementary Notes Project performed in cooperation with the Idaho Transportation Department and FHWA.		

16. Abstract

Avalanches routinely occur on State Highway 21 (SH21) between Lowman and Stanley, Idaho each winter. The avalanches pose a threat to the safety of maintenance workers and the traveling public. A real-time avalanche detection system will allow the Idaho Transportation Department (ITD) avalanche forecasters to remotely monitor the major avalanche paths on this corridor. Information from the real-time system will aid avalanche forecasters in determining the current avalanche hazard, provide information about the timing and location of avalanche activity, initiate response/clearing of roadway, and provide information for when the road may be safe to open.

The overall project objective was to develop a real-time avalanche detection system that could be easily deployed and expanded to other avalanche areas within Idaho. In addition, the system developed through the research needed to be capable of providing information to avalanche forecasters in a format they can interpret. In this project, we demonstrate that:

1. Avalanches can be detected using our low-cost sensor arrays that measure infrasound (sound below the threshold of human hearing, less than 20 Hz).
2. These arrays can be reliably deployed for continuous remote monitoring throughout the winter.
3. The multi-sensor high resolution data can be processed on-site and high level information can be relayed back to ITD avalanche forecasters.

17. Key Words Avalanche, operational, avalanche detection		18. Distribution Statement Copies available online at http://itd.idaho.gov/highways/research/	
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 64	22. Price None

FHWA Form F 1700.7

[redacted]

APPROXIMATE CONVERSIONS TO SI UNITS					APPROXIMATE CONVERSIONS FROM SI UNITS				
Symbol	When You Know	Multiply By	To Find	Symbol	Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH					LENGTH		
in	inches	25.4		mm	mm	millimeters	0.039	inches	in
ft	feet	0.3048		m	m	meters	3.28	feet	ft
yd	yards	0.914		m	m	meters	1.09	yards	yd
mi	Miles (statute)	1.61		km	km	kilometers	0.621	Miles (statute)	mi
		AREA					AREA		
in^2	square inches	645.2	millimeters squared	cm^2	mm^2	millimeters squared	0.0016	square inches	in^2
ft^2	square feet	0.0929	meters squared	m^2	m^2	meters squared	10.764	square feet	ft^2
yd^2	square yards	0.836	meters squared	m^2	km^2	kilometers squared	0.39	square miles	mi^2
mi^2	square miles	2.59	kilometers squared	km^2	ha	hectares (10,000 m^2)	2.471	acres	ac
ac	acres	0.4046	hectares	ha					
		MASS (weight)					MASS (weight)		
oz	Ounces (avdp)	28.35	grams	g	g	grams	0.0353	Ounces (avdp)	oz
lb	Pounds (avdp)	0.454	kilograms	kg	kg	kilograms	2.205	Pounds (avdp)	lb
T	Short tons (2000 lb)	0.907	megagrams	mg	mg	megagrams (1000 kg)	1.103	short tons	T
		VOLUME					VOLUME		
fl oz	fluid ounces (US)	29.57	milliliters	mL	mL	milliliters	0.034	fluid ounces (US)	fl oz
gal	Gallons (liq)	3.785	liters	liters	liters	liters	0.264	Gallons (liq)	gal
ft^3	cubic feet	0.0283	meters cubed	m^3	m^3	meters cubed	35.315	cubic feet	ft^3
yd^3	cubic yards	0.765	meters cubed	m^3	m^3	meters cubed	1.308	cubic yards	yd^3
Note: Volumes greater than 1000 L shall be shown in m^3									
		TEMPERATURE (exact)					TEMPERATURE (exact)		
^oF	Fahrenheit temperature	5/9 (^{o}F-32)	Celsius temperature	^oC	^oC	Celsius temperature	9/5 ^{o}C+32	Fahrenheit temperature	^oF
		ILLUMINATION					ILLUMINATION		
fc	Foot-candles	10.76	lux	lx	lx	lux	0.0929	foot-candles	fc
fl	foot-lamberts	3.426	candela/m^2	cd/cm^2	cd/cm^2	candela/m^2	0.2919	foot-lamberts	fl
		FORCE and PRESSURE or STRESS					FORCE and PRESSURE or STRESS		
lbf	pound-force	4.45	newtons	N	N	newtons	0.225	pound-force	lbf
psi	pound-force per square inch	6.89	kilopascals	kPa	kPa	kilopascals	0.145	pound-force per square inch	psi

Acknowledgements

The authors would like to acknowledge the ITD avalanche forecasters Bill Nicholson and Chantel Astroga for their help with avalanche observations and data collection that was invaluable for this project over the past four winter seasons. We would like to thank IRIS PASSCAL and Ethan Greene from the Colorado Avalanche Information Center (CAIC) for data loggers. Jeffery B. Johnson helped provide the necessary background and discussion for infrasound processing as well as building new sensors for this project. The authors would like to thank all those that have helped with field work, installing and uninstalling multiple arrays over the years: Andrew Hedrick, Patrick Kormos, Tim Ronan, and Jesse Dean.

The authors thank the Technical Advisory Committee, specifically Dan Babcock, Bob Koeberlein, Tom Points, Mike Ponozzo, Inez Hopkins of ITD and Brent Inghram and Lance Johnson of the Federal Highway Administration - Idaho for their assistance.

Table of Contents

List of Tables

List of Figures

Executive Summary

Research Objective and Background

Avalanches routinely occur on State Highway 21 (SH21) between Lowman and Stanley, Idaho each winter, which poses a threat to the safety of maintenance workers and the traveling public (Figure 1). A real-time avalanche detection system will allow the Idaho Transportation Department (ITD) avalanche forecasters to remotely monitor the major avalanche paths on this corridor. Information from the real-time system will aid avalanche forecasters in determining the current avalanche hazard, provide information about the timing and location of avalanche activity, and provide information for initial response and cleanup to help determine when the road may be safe to open.

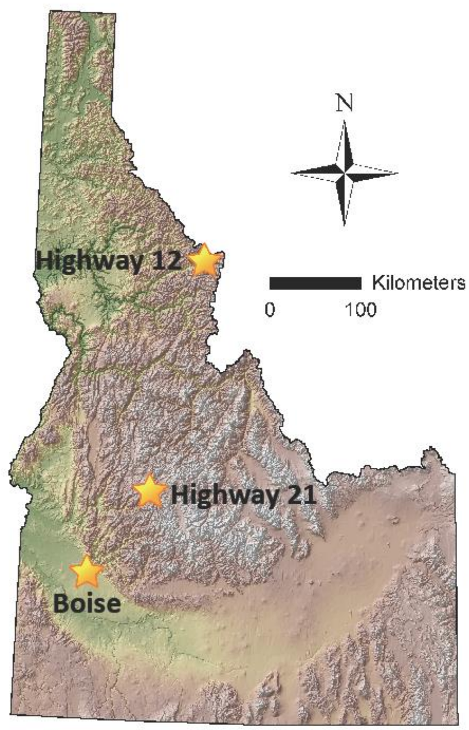

Figure 1. Avalanche Prone Highways in Idaho

The overall project objective was to develop a real time avalanche detection system that could be easily deployed and expanded to other avalanche areas within Idaho. The system needed to be capable of providing information to avalanche forecasters in a format they can interpret. In this project, we demonstrate that

1. Avalanches can be detected using our low-cost sensor arrays that measure infrasound (sound below the threshold of human hearing, less than 20 Hz).

2. These arrays can be reliably deployed for continuous remote monitoring throughout the winter.

3. The multi-sensor high resolution data can processed on-site and high level information can be relayed back to ITD avalanche forecasters.

Avalanche Detection and Classification

Three steps are needed for avalanche detection:

1. Data Processing.
2. Event Detection.
3. Event Classification.

This project employs arrays which are a set of multiple sensors arranged in a pattern, with spacing between each sensor of 100 to 150 feet. The use of arrays allows array processing techniques to calculate a coherence metric (called the Fisher statistic) from the raw infrasound signals. Once the data has been processed, event detection determines if a coherent signal of interest was recorded by the array. A new method of event detection was developed for this project using non-parametric methods. This method accounts for changes in the background signal of the Fisher statistic due to correlated noise (i.e. microbaroms or digitization noise) across the array.

Artificial neural networks (ANN) were used for event classification; they are based on how the biological brain works and perform much better than standard classification approaches. A trained observer manually picked events from a 2-day avalanche cycle during the winter of 2012/2013. The events consisted of avalanches, vehicles, planes, signals reflected from the atmosphere originated from Mountain Home Air Force Base (MHAFB), and many unknown signals. These events were used to train the ANN to classify each group of events separately, as accurate classification of non-avalanche related coherent signals is required to obtain accurate avalanche classifications.

Avalanche Detection System

All data processing occurs on-site with a small low power computer to reduce the amount of data transmission for remote deployments. The computer connects to the data logger and continuously downloads data to the computer. The infrasound data is processed to determine times of coherent signals. Event detection occurs after the data is processed and automatically determines times of possible events. Once an event is detected, the event is sent to the ANN for classification. If the ANN determines the event is an avalanche, a small text message is sent over Iridium satellite communications to a Boise State University (BSU) server and results are visible in near real-time on a webpage.

BSU has deployed infrasound arrays for the past four winters that operate continually and with minimal service throughout the winter season. The full detection system with on-site computer and telemetry was deployed for one week in August 2014 to ensure that it would function properly detecting vehicle traffic. Due to the complex terrain of SH21 and the location of the targeted avalanche paths, the system was deployed to detect events between mile markers 99.8 and 102.0.

Results showed that the infrasound data was processed quickly on-site, detected events, correctly classified vehicles, and sent a message for each positive vehicle detection. The message was displayed within a few minutes on a dedicated webpage. This successful test shows the potential for an operational real time avalanche detection system that is fully automated and sends only the relevant data to avalanche forecasters.

Conclusions

This report shows that the avalanche detection system developed by BSU will be a vital new tool for ITD avalanche forecasters to use operationally. We developed a robust infrasound array avalanche detection system, optimized sensor geometry, and successfully recorded infrasound nearly continuously over four winters. A prototype system including onboard processing, near real-time event detection, along with remote communication via satellite modem was developed and extensively tested. The prototype detection system has been deployed successfully short term using new processing techniques and advanced computer science classification methods. The successful remote deployment shows that the system could be expanded to other avalanche prone areas in Idaho, such as US12.

Implementation

The scope of this project provides proof-of-concept for using infrasound arrays, on-site beam-forming processing, neural network classification, and communication methods for a real-time avalanche detection system. Implementation of this work for practical forecasting operations, should ITD choose to deploy the system, will require additional funding from ITD to purchase and install the detection system. BSU recommends a part-time research technician be used for the winter of 2014/2015 to deal with any problems that may occur. In addition, the technician would be able to provide extra support, troubleshooting, maintenance, quality control, and retraining of the ANN with new data. The total cost of the avalanche detection system with part time research technician is estimated at $66,865.00.

Chapter 1
Introduction

Purpose of the Project

A real-time avalanche detection system would allow the Idaho Transportation Department (ITD) avalanche forecasters to remotely monitor the major avalanche paths on selected corridors. Such a system would:

- Alert ITD forecasters when smaller avalanches are occurring that do not block the road, which indicates regional avalanche activity and instability trends.
- Monitor avalanches at night when traffic volume is low and emergency response is limited.
- Provide an emergency alert in the unlikely event that the road is open when avalanches occur, as the highway could be closed immediately following any remotely detected avalanche events.
- Improve information about timing of avalanche events when the road is already closed, aiding in future forecasts and in turn providing avalanche information for when the road is safe to open.
- Provide an expandable avalanche detection system design that could be deployed in other areas around Idaho, especially the State Highway 12 (SH12) area.

The objective of this research project was to develop and test the performance of an avalanche detection system on selected sections of SH21 between Lowman and Stanley, Idaho. Researchers were charged with establishing and maintaining infrasound instrument arrays for a minimum of two years and developing user-friendly software for processing and interpretation of avalanche signals by ITD forecasters. At the end of Year 2, Boise State University (BSU) was to prepare a report evaluating system performance with recommendations for ongoing operation of the system.

Description of Research Problem

Avalanches routinely occur on SH21 between Lowman and Stanley, Idaho each winter, which poses a threat to the safety of maintenance workers and the traveling public. Before a formal avalanche forecasting and hazard mitigation program was developed, SH21 was closed for the majority of the winter (Figure 2). In the past, the number of avalanches on SH21 was small compared with recent activity due to the difficultly in distinguishing multiple avalanche events from a single debris pile on SH21 or from the debris pile being completely covered with new snowfall. After the avalanche forecasting and hazard mitigation program was developed in 2007, the number of days SH21 was closed was drastically reduced. The number of avalanches reported reaching SH21 has increased due to more accurate records, the avalanche forecasters observing avalanche activity within days after the avalanche occurring, and from avalanche mitigation which increases the number of smaller avalanches while decreasing the number of larger avalanches. The goal of the real time avalanche detection system is to provide another tool for ITD avalanche forecasters to utilize when making decisions about avalanche hazard along SH21.

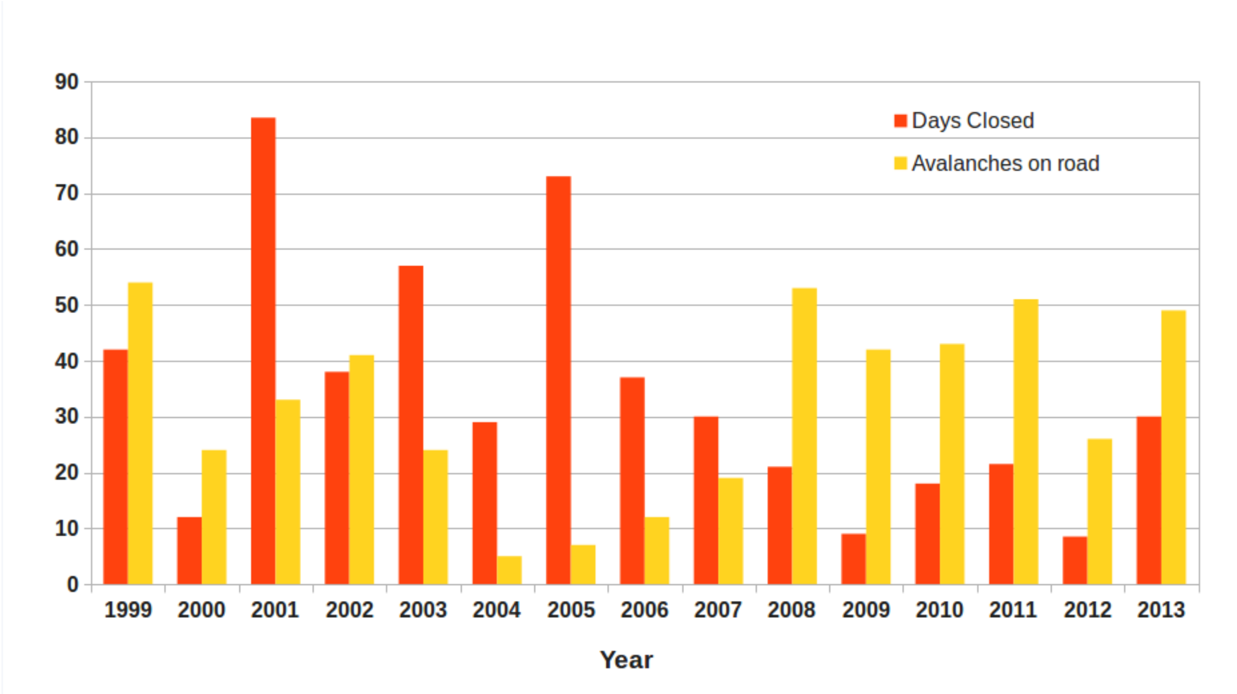

Figure 2. Days Closed Versus Number of Avalanches that Reached SH21

Project Task and Objectives

The project had the following project tasks and objectives:

1. Record avalanche events with existing infrasound arrays.

2. Experiment with different array configurations to determine the most efficient deployment for avalanche detection.

3. Identify avalanche events in infrasound data and how avalanche generated signals differ from other sound sources within the canyon. This will require extensive signal processing that will lead to automatic detection and classification of avalanche events.

4. Develop interactive software for ITD forecasters to monitor avalanche activity in real time.

5. Training for ITD forecasters on software operation.

Project Scope

This project was intended to develop and test for a real-time avalanche detection system. In this project, we demonstrate that avalanches can be detected using the low-cost BSU infrasound array system design with remote onboard processing, and the information can be relayed back to ITD avalanche forecasters via satellite modem.

Chapter 2
Literature Review

Avalanche Flow and Infrasound Generation

There are two types of avalanches, dry and wet. A dry avalanche occurs when the snowpack is below freezing temperature and there is minimal (< 3 percent) free water within the snow. Dry avalanches typically move faster than wet avalanches, at speeds of 30 - 50 m/s and generate a powder cloud. Wet snow avalanches typically occur due to a decrease in strength of the snowpack due to warming and free water within the snow. Since there typically is an excess of liquid water, a wet snow avalanche moves slower (5 - 30 m/s) and does not develop a powder cloud. Wet snow avalanche infrasound generation is not well understood or characterized and typically signal amplitudes are much lower. Therefore, this review focuses on the infrasound generation from dry snow avalanches, which is currently better understood.

A dry avalanche typically is made up of three layers (Figure 3). The dense flow portion of the avalanche flows over the snowpack, entraining snow. The movement over the snowpack generates seismic signals that can be detected and are a function of the avalanche path characteristics. The shear stress caused by interaction with the air against the dense flow leads to a saltation layer that moves faster than the dense flow. If avalanche speeds are great enough, turbulent eddies form to create a powder cloud or suspension layer that covers the avalanche core.[(1)]

Infrasound (1-20 Hz) is generated from the turbulent flow from the suspension layer of the avalanche, which is moving at a higher speed than the dense flow and saltation layers.[(2)] The suspension layer typically forms in the upper part of the avalanche path and loses energy as the avalanche slows down.[(3,2)] Kogelnig, et al. concluded that the size of the suspension layer observed on pressure pylons and the avalanche velocity were proportional to the infrasound amplitude.[(2)]

Even though the entire avalanche core is covered by the powder cloud, the theory is that only a small portion of the avalanche creates the measurable infrasound signals. Infrasound signals are generated when the avalanche displaces a large amount of the atmosphere. The most violent part of a dry avalanche is the front and where you can typically see large, forceful vertical eruptions of the powder cloud. These eruptions should coincide with large spikes in the infrasound amplitude. Between eruptions, the turbulent eddies at the avalanche front generate less intense infrasound signals. However, previous research has not verified these theories with quantitative data.

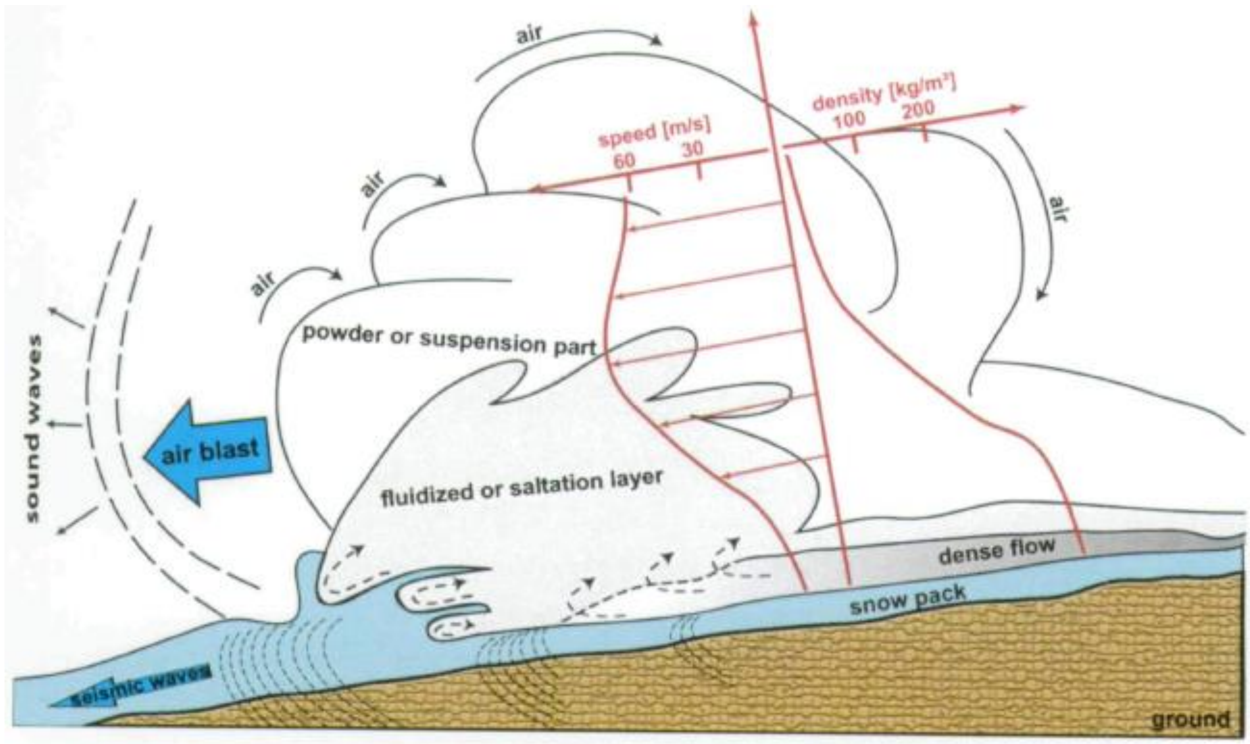

Figure 3. Dry Avalanche Flow

Dry avalanche flow, showing the three layers and possible sources of both infrasound and seismic signals.[(2)]

Current Avalanche Infrasound Research

The first avalanche infrasound research was performed at the National Oceanic and Atmospheric Administration (NOAA) in Boulder, CO. The authors found that avalanches generate acoustic signals in the 1-5 Hz region.[(4,5,6)] They could detect avalanche signals on their infrasound sensors deployed for atmospheric studies.

In Europe, Chritin, Rossi and Bolognesi developed an avalanche detection system using infrasound sensors.[(6)] Their system was called ARFANG and was used to detect avalanches using an array of four sensors to update a database of avalanche occurrences. With the accurate timing of the avalanche known from the infrasound, a nearest-neighbor avalanche forecasting model could be updated to more accurately evaluate the current avalanche hazard. The ARFANG system is still operational, but has not been updated since it was first implemented.

In the United States, using infrasound for avalanche detection was first developed at Teton Pass, WY and Jackson Hole Mountain Resort, WY using a single infrasound sensor. Several winters of data were collected to create a catalogue of avalanche signals.[(8)] The avalanche detection algorithm for the one sensor was based off the catalogue of events, statistics, digital filtering, and weighted threshold decision-making to determine if a signal was a potential avalanche. The results were mixed as wind noise can interfere with signal detection and classification.[(9)] The next step was to use multiple sensors to detect coherent signals.[3,10] An array of sensors was placed in a line 150 meters wide. With this array, the authors could beam from where the signal was generated. Since it's a commercial detection system, the actual processing steps were neither well known nor published, but most likely used correlation to determine the travel time differences between sensors.

Currently, the system developed by Scott, et al. is applied operationally at Teton Pass, WY and Alta, UT.[10] Multiple arrays of six sensors in a circle are used to identify coherent avalanche signals. The data is processed near real-time using Matlab functions and displayed with a Graphical User Interface (GUI). This commercial system has become the standard system in the United States but lacks many array processing techniques that could more accurately detect and locate avalanches, and is very high cost (close to $1M for recent Alta, UT system).

Recently, work by in the Aosta Valley, Italy has brought modern array processing techniques to avalanche detection.[11] Ulivieri deployed a four sensor array and detected all events using multi-channel correlation method.[12] The authors have had great results, but with only one array in a busy valley, it was difficult to determine what was an avalanche signal.

Array Processing

Arrays are a grouping of multiple instruments within a small area that decrease the signal-to-noise (SNR) ratio from summation of the signal using all sensors. An advantage of arrays is the ability to determine the source direction for sensor location information.[13] For example, the use of arrays in the seismological community has helped to refine velocity models of the Earth's interior and improve the detection capability of underground nuclear explosions.[14,15] Arrays have also been used with infrasound to detect avalanches, studying eruptions from volcanoes, and localization of thunder.[10,11,16,17]

Many array processing techniques have been developed by the seismic community to interpret array seismic data and can be adapted for use with infrasound. Rost and Thomas provide an excellent overview of many of the methods listed below.[13]

Array processing techniques assume a plane wave crossing the array where the wave front can be assumed to be infinite parallel planes (Figure 4b). This is a good assumption for a large source-receiver distance. The two concepts for the foundation of array processing are the slowness vector and beam forming.

Slowness Vector

The slowness vector describes the direction that a sound wave propagated from. A wave and the slowness vector can be described by two parameters:

1. The incidence angle i.
2. The back azimuth θ (Figure 5).

The incidence angle is related to the inverse of the apparent velocity (v_{app}) of the wave across the array.

The slowness *s* is defined as:

$$s = \frac{1}{v_{app}} = \frac{\sin i}{v_0}$$

Figure 4. Slowness Vector Equation

where v_0 is the speed of sound through air.

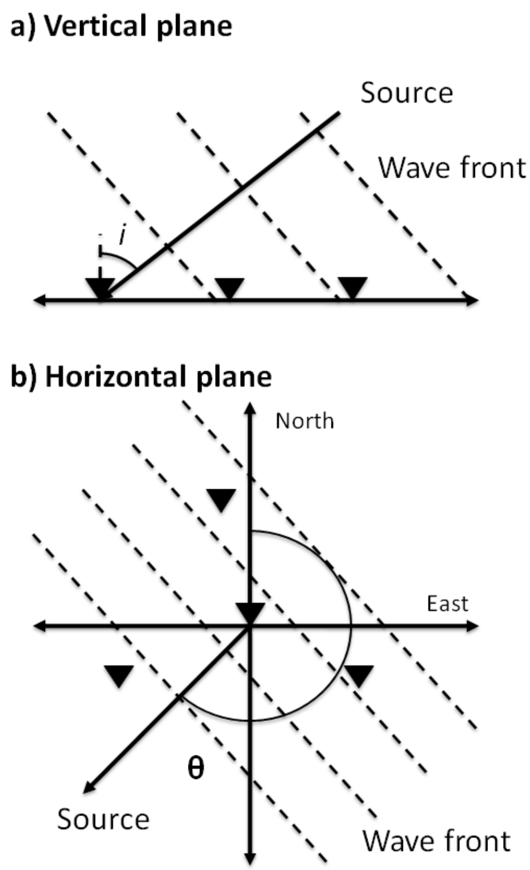

Figure 5. Incidence Angle and Back Azimuth

a. The incidence angle i is the angle from vertical at which the wave front reaches the array.
b. The bearing from North to the source is the back azimuth θ.

The back azimuth is the direction in degrees from North that the wave front passes the array in the horizontal plane. Both incidence angle and back azimuth are combined to determine the slowness vector *s* which points in the direction of wave propagation (Figure 7), and has units that are the inverse of velocity [s/m]. The slowness vector **s** is related to the incidence angle and back azimuth by:

$$\begin{aligned}\boldsymbol{s} &= (s_x, s_y, s_z)\\ &= \left(\frac{\sin\theta}{v_{app}}, \frac{\cos\theta}{v_{app}}, \frac{1}{v_{app}\tan i}\right)\\ &= s_{hor}\left(\sin\theta\,, \cos\theta\,, \frac{1}{\tan i}\right)\\ &= \frac{1}{v_0}(\sin i \sin\theta\,, \sin i\cos\theta\,, \cos i)\end{aligned}$$

Figure 6. Slowness Vector in 3 Dimensions

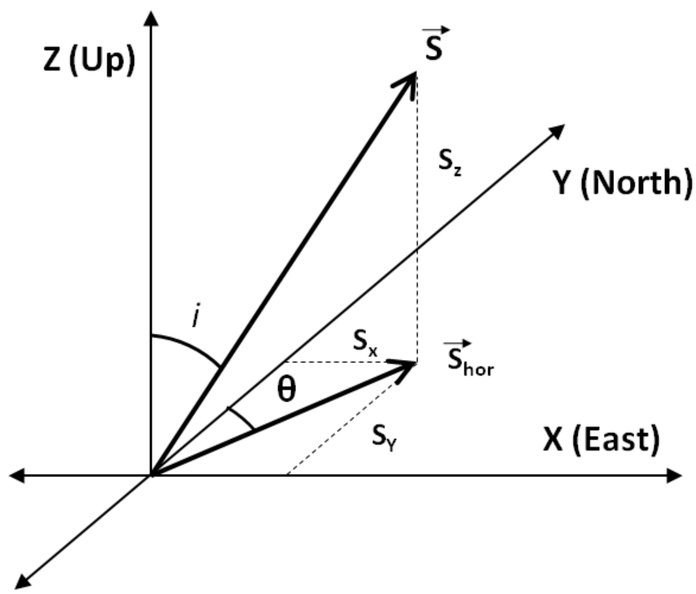

Figure 7. Slowness Vector

Slowness vector s broken down to its components s_x, s_y, and s_z. $\boldsymbol{s_{hor}}$ is the slowness vector in the x-y plane.

Beam Forming

Beam forming allows for separation of coherent signals and noise by utilizing all sensors in the array. When a wave crosses the array, the sensors record the signal with a time shift based on the wave's slowness vector. By applying the correct slowness vector to the recorded data, and shifting each sensor in time, the coherent signal can be aligned on all sensors, and the SNR can be increased.

Each sensor in the array has a direction vector $\mathbf{r_j}$ from the array center and represents the sensor's position in the array. The absolute value $r = |\boldsymbol{r}_j|$ describes the sensors absolute distance from the array center. The travel time of the wave between sensors is dependent on the slowness vector and sensor's direction vector. Given a recorded signal *f(t)* with noise *n(t)*, the signal recorded at each station *j* is:

$$x_j(t) = f\left(t - r_j \cdot s_{hor}\right) + n_j(t)$$

Figure 8. Equation Representing Recorded Signal

To calculate the signal with the time shift removed given the sensor's location r_j and the horizontal slowness vector $\boldsymbol{s_{hor}}$:

$$\overline{x_j}(t) = x_j\left(t + r_j \cdot s_{hor}\right) = f(t) + n_j\left(t - r_j \cdot s_{hor}\right)$$

Figure 9. Equation Applying Time Shift to Signal

Summing the signals over all *M* sensors produces the beam trace *b(t)* and increases the signal to noise by approximately $\sqrt{M}$:

$$b(t) = \frac{1}{M}\sum_{j=1}^{M} \overline{x_j}(t) = f(t) + \frac{1}{M}\sum_{j=1}^{M} n_j(t + \boldsymbol{r_j} \cdot s_{hor})$$

Figure 10. Equation to Calculate Beam

Chapter 3
Avalanche Detection with Infrasound

Study Site

ITD forecasts for SH21 located 2.5 hours northeast of Boise, Idaho in an intermountain climate. The area typically sees moderate snowfall (300 inches average), often extremely cold temperatures between storms, and rain on snow events throughout the winter. ITD has a limited explosive avalanche mitigation program due to the complex terrain of the start zones and highway location. Avalanche activity is mainly direct action avalanches due to storm snow or rain on snow, with at least one major wet slide cycle during the spring. Both lanes of SH21 are frequently covered during avalanche cycles and the road is often closed for several days at a time. The average return frequency for avalanches reaching the highway in relation to the infrasound arrays are shown in Figure 11.

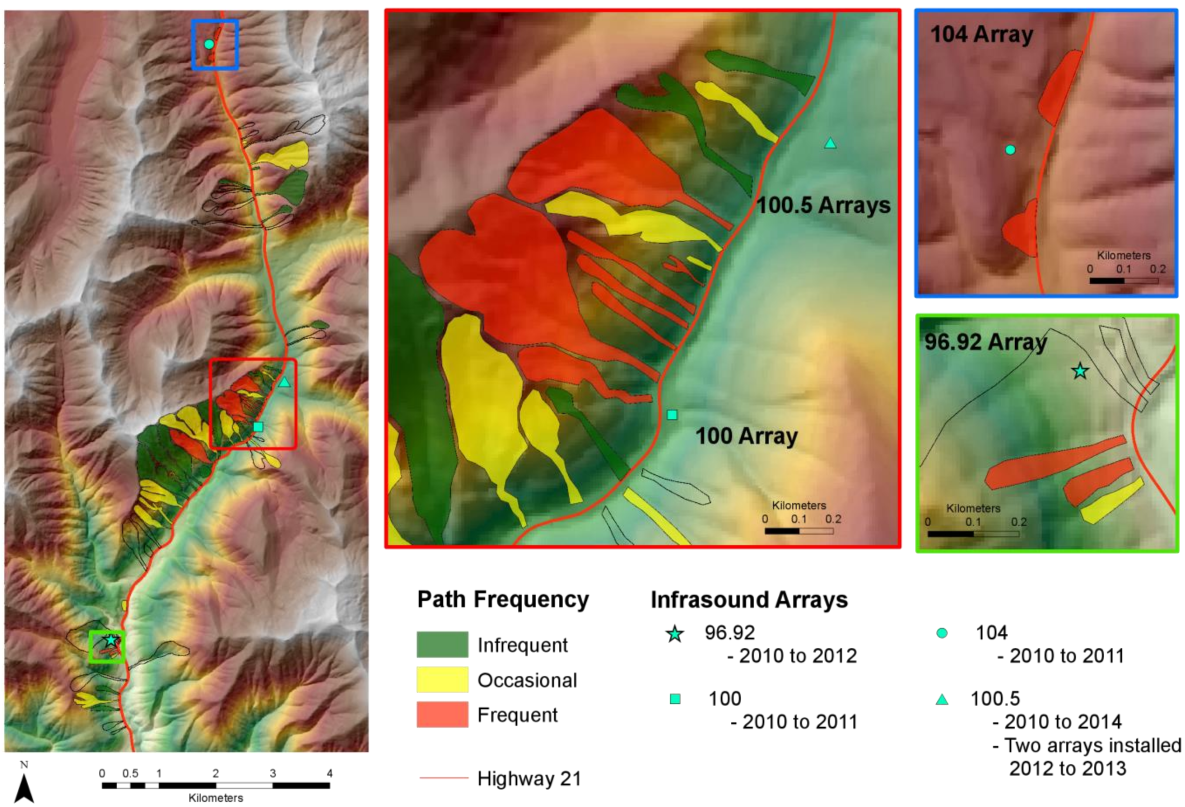

Figure 11. Infrasound Array Installation and Avalanche Frequency

Leveraging a previous project with ITD to forecast direct action avalanches using snow stability modeling infrasound arrays were deployed to determine accurate avalanche timing for model improvement.[18,19] Infrasound arrays have been deployed along SH21 since the winter of 2010/2011 with the number of arrays ranging from a maximum of 4 to 1 targeted array. The following table summarizes the array types installed:

Table 1. Summary of Infrasound Array Installations

Winter	Number of Arrays	Number of Sensors	Configuration
2010/2011	4	3	Arranged in a triangle with 30 meter spacing
2011/2012	2	3	Arranged in a triangle with 30 meter spacing
2012/2013	2	7	Spoke with 15 and 30 meter spacing, 1 geophone
2013/2014	1	5	Square with 1 in middle, 1 geophone, real time prototype

Background

Processing the raw infrasound data and outputting an event classification requires multiple steps. The following outlines the steps required to produce an event classification:

1. Take the raw infrasound data and use array processing techniques to calculate a coherency metric.

2. Use non-parametric event detection to determine if a signal is present from the previously calculated coherency metric.

3. If an event is detected, pass the relevant data to the classification scheme to determine the type of event.

Array Processing

Fisher Statistic

The Fisher statistic was first suggested by as a method to detect events across a seismic array.(20) The theory was further developed by others.(21,22) Blandford states the assumptions for the Fisher statistic as:

1. Signals must be perfectly correlated across the array.
2. Noise is normally distributed, stationary, and uncorrelated across the array.(21)

These assumptions are usually violated for small aperture infrasound and seismic arrays where the signals may not be perfectly correlated and the noise may also be correlated.

Following Blandford and Arrowsmith, et al. the Fisher statistic in the time domain is defined as the power of the beam over the residual power.(21,23)

$$F = \left(\frac{M-1}{M}\right) \times \frac{\sum_{n=1}^{N}\left[\sum_{j=1}^{M} x_j(n+l_j)\right]^2}{\sum_{n=1}^{N}\left[\sum_{j=1}^{M}\left\{x_j(n+l_j) - \left(\frac{1}{M}\cdot\sum_{m=1}^{M} x_m(n+l_m)\right)\right\}^2\right]}$$

Figure 12. Equation for the Fisher Statistic

Where M is the number of array elements, N is the size of the processing window in samples, l_j is the time lag in samples applied to the signal for a given element, and x_j is the signal measured at each element. When a signal is present, the numerator will increase from the power of the beam and the denominator will be reduced to the residual noise.

Frequency Wave-Number Analysis

The frequency wave-number (fk) analysis allows for the simultaneous calculation of the incidence angle and back azimuth.(13) The fk analysis determines the time shifts required for varying slowness vectors and calculates the Fisher statistic. When the Fisher statistic is maximized, the slowness vector provides a direct estimate of the back azimuth and incidence angle. If the source location is unknown, a grid search is performed over a range of back azimuths and incidence angles.

The fk analysis is performed for a short moving window with a constant step size. Within each window, the Fisher statistic is calculated using a grid search to time shift the infrasound signal depending on the slowness vector at each grid node. To provide the best results, the window size must be large enough to capture the dominant frequency moving across the array. For the small aperture arrays on SH21, a 6 second window with 3 second step size has proved to be sufficient to determine the location of moving sources.

Non-Parametric Event Detection

Non-Parametric Methods

The Fisher statistic follows the Central F-distribution when the data meets the assumption criteria of a perfectly correlated signal and uncorrelated noise.(21) However, with small aperture arrays, the assumptions are violated due to correlated noise recorded across the array. There are two methods to handle this situation:

1. Assume that the data can fit the F-distribution to use parametric methods.
2. Estimate the distribution directly from the data with non-parametric methods.

The Central F-distribution has two parameters that determine the shape, based on the signal bandwidth, time length of processing window, and the number of elements in the array. Using a moving window of Fisher statistic values, Arrowsmith, et al. fit the window of values to the F-distribution by scaling the values so that the peaks in the distribution lined up.(23) This method will work well with larger aperture arrays (~1 - 2 km in the study) but will have great difficulty if the Fisher statistic values do not follow the F-distribution.

Non-parametric methods allow for the Fisher statistic values within the window to be used directly to estimate the distribution. No assumptions are made about the underlying distribution making the method more robust, especially when values do not necessarily follow the Central F-distribution. The non-parametric probability density function (PDF) of Fisher statistic values is estimated using kernel density estimation.(24) The probability of an observed value x_j at time t is

$$P(x_t) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x_t - x_i)$$

Figure 13. Non-Parametric Kernel Estimate

where K_σ is a Gaussian kernel function with bandwidth σ, N is the number of background samples, and x_i is the background Fisher Statistic values. Other kernel functions can be used, but the results are typically not very sensitive to the kernel and primarily depend on the choice of kernel bandwidth.

Background Model

Signal noise will create a Fisher statistic that falls into a certain type of distribution. The distribution can be modeled by creating a non-parametric PDF for a moving window of Fisher statistic values (similar to Arrowsmith.(23) This becomes the background model for noise with which to compare new values (Figure 14). The Fisher statistic value from noise will fall somewhere within the distribution and have a higher probability. An event will fall to right of the distribution and have a low probability that the value has been observed before. A given value is considered an event if $P(x_t) < threshold$.

A moving background model will adapt to changes in the Fisher statistic from to correlated noise. As the background window moves forward in time, the model is updated by removing the oldest sample and adding the newest sample to the model. This type of update adds all samples to the model and will include samples that are not part of the background model. However, this will lead to false negatives as an event may not be detected if a significant amount of events are already within the model. The effect can be reduced by increasing the window size but this will also increase the number of false positives as adaptation to changes will be slower.

False detections occur when the noise becomes correlated and produces a high Fisher statistic value with a probability just below the threshold. To account for these false positives, we calculate the probability of the current observation from multiple frequency bands (similar to Brachet and Coyne).(25) This helps to determine if the signal is confined to one frequency band or is connected over multiple frequency bands. The detection is performed for nine 2 Hz frequency bands between 2 and 20 Hz. If the intersection (product) of the probabilities for all the frequency bands is below a threshold (similar to Elgammal, the current time is considered an event.(26) This process helps reduce the number of false positives due to correlated noise in a specific frequency band.

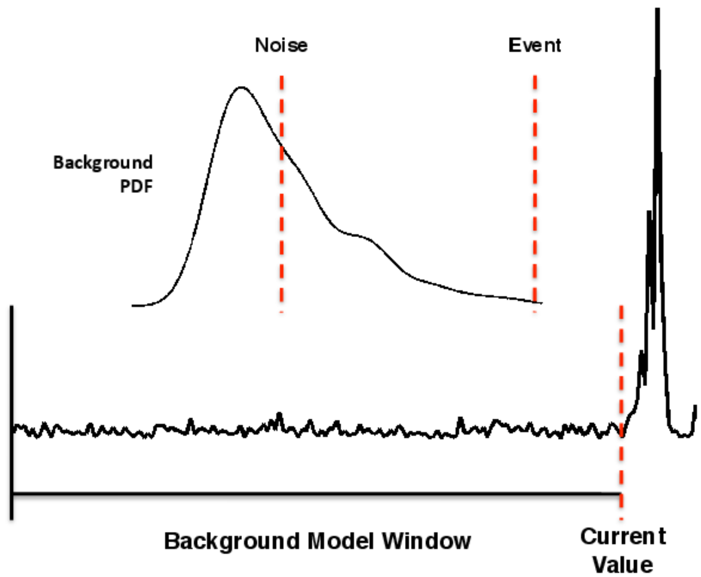

Figure 14. Fisher Statistic Background Model

A window of Fisher statistic values creates the background model PDF. A new value of the Fisher statistic is then compared to the PDF with noise expected to fall within the distribution and an event to fall to the right of the distribution. Red dashed lines show how the current value, either noise or an event, will fall on the PDF.

Event Classification

Artificial neural networks (ANNs) are based on biological brains, and as such, are conformed by units (neurons) that are connected to each other. These connections can vary in *strength* and the weight given to each connection is the basic factor that modifies the output of the network. In order to use artificial neural networks for classification, these connections need to be modified to achieve a desired output through training.[27]

Recurrent neural networks (RNNs) are a type of neural network that can maintain an internal state and allow for complex temporal decision-making and classification of continuous data.[28] These networks maintain "memory," as their output depends not only on current conditions but on those over some previous window in time.

Just as with biological neural networks, ANNs are trained by providing an input and an "ideal output." The input - also called a sample, can be an image, a sound snippet or any other data that can be represented as numbers along with a label (ideal output) that describes what the network should output when faced with the current input. The output of the network is usually a number which is greater for a positive output and small for a negative output. In other words, if the network is being trained to detect avalanches, the ideal output for an avalanche sample could be 1, and the ideal output for a sample that belongs to any other event would be 0. The output of a network trained in this manner would be a number close to 1 for all avalanche samples, and close to zero for any other samples.

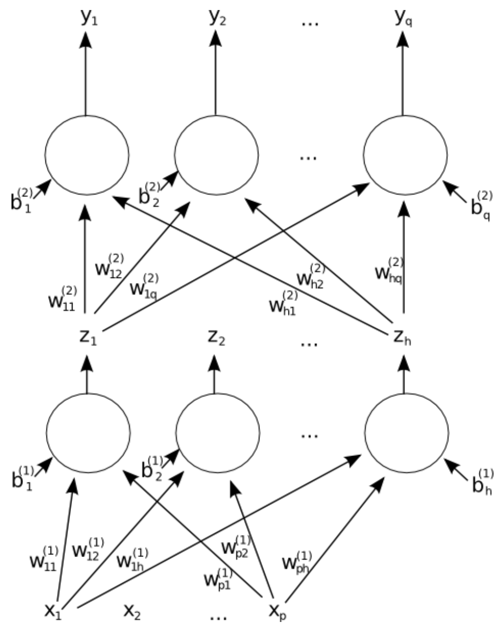

Figure 15. Basic Structure of an Artificial Neural Network

Training a network is a computationally expensive process.(27) As long as there are enough samples (labeled inputs), the network will continue to get more accurate in its classification through training. In the opposite scenario - where there are only a few samples to train the network, there is a high probability that the training will result in a network that is accurate in classifying a narrow subset of all the possible samples. However, the network will become very inaccurate when dealing with samples that belong to the same classification but are substantially different (the network *over-learns* to classify). To prevent this, classification approaches in general are typically trained to a subset of the available data, and then the classification model is tested on the remainder of the data.

Methods

Sensitivity Analysis of Input Parameters

Two parameters control the event detection results: the window size and the significance level (alpha). To determine how these parameters affect event detection, an avalanche signal and a 2-day avalanche cycle was tested for varying window sizes and significance levels. To quantify the error, the l^2-norm (i.e. the vector magnitude) was calculated for the difference between the true event indices and the detected events indices:

$$|\boldsymbol{r_j}| = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

Figure 16. l^2-norm Distance Metric

where x_k is the true event indices, y_k is the detected event indices, and $|\boldsymbol{r_j}|$ is the l^2-norm. The l^2-norm value will include information about false negative detections and any error between the actual event and the automatic detection. A low l^2-norm indicates the automatic detection matched the true event indices.

Avalanche Signal

A small wet avalanche was recorded during the 2-day avalanche cycle (Figure 17). The avalanche provides a typical signal for wet avalanches that we expect to encounter throughout deployment. The signal has a small pressure amplitude and was hard to visually distinguish above the noise. However, the Fisher statistic value was well above 55 in the 4 - 12 Hz bandwidth, indicating a highly correlated signal. The signal also has some high frequency instrument noise just prior to the avalanche, with higher Fisher statistic values and a large amplitude, uncorrelated signal caused by wind. This proved to be a great test as the detection algorithm should not detect this noise.

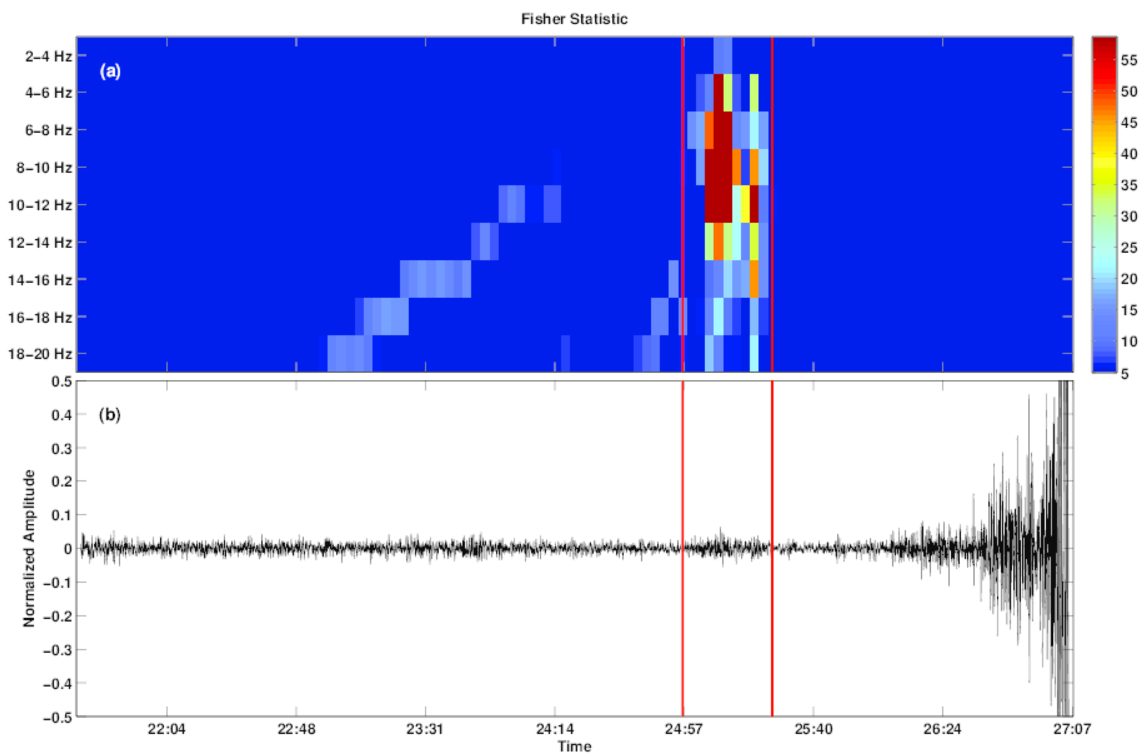

Figure 17. Small Wet Avalanche Signal

Small wet avalanche that occurred during the 2 day avalanche cycle.
a. The Fisher Statistic was calculated over 9, 2 Hz frequency bands with most correlated energy in the 4-12 Hz bandwidth.
b. Amplitude signal of the avalanche. The start and end times of the avalanche are shown in red.

Note: that the avalanche has a small amplitude signal followed by uncorrelated wind signal.
This signal is an avalanche as the back azimuth and v_{app} correspond to a known avalanche path.

Avalanche Cycle

126 events were identified through manual observation for a 60-hour period between January 25, 2013 12:00 GMT and January 28, 2013 00:00 GMT. Six different types of events were classified (Table 2) based on the timing, back azimuth, and apparent velocity of the signal. Forty vehicle signals were identified before-and-after the road was closed at approximately January 25, 2013 19:40 GMT. Four possible avalanche signals were identified during the 3 hours after January 26, 2013 17:32 GMT, while the road was closed. These small wet avalanches are fairly common with two to three avalanche cycles per year. Other events include signals from MHAFB, planes flying overhead, and the possible cleanup effort. Sixty five unknown signals were found which may contain already identified events but could not be classified without reasonable uncertainty.

Table 2. Identified Events

Event Type	Number of Events
Mt. Home Air Force Base	*4*
Plane	*6*
Possible Avalanche	*4*
Rotary (Possible)	*7*
Unknown	*65*
Vehicle	*40*
Total	***126***

The Fisher statistic values for each event are shown in Figure 14 as compared to the background Fisher statistic values determined when no events were detected (in black). Most of the events had a higher median Fisher statistic value than the background.

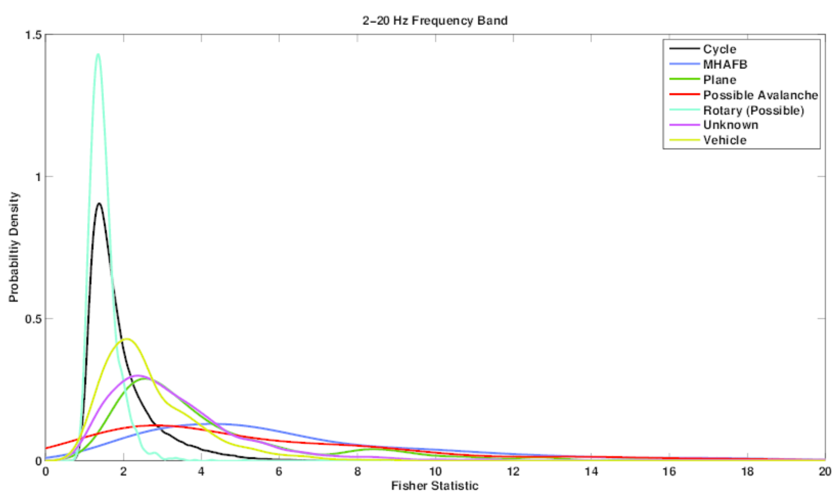

Figure 18. Cycle Background (Black) Compared to the Six Events

Event Classification

Once an event was detected, the necessary event properties were calculated for event classification. The neural network uses the Fisher statistic, power between 1 - 5Hz, power between 5 - 10Hz, power between 10 - 15Hz, and the back azimuth for classification. The properties were inputted to the neural network, which returns a time series of the network output. If the maximum output was above a given threshold, the event was classified for the given network.

Training a neural network requires a large dataset of previous events. However, we do not have a large number of events picked from previous seasons. To test the detection and classification, we have created two neural networks, one for vehicle classification, which has the largest number of events, and another network for avalanches.

The network has been trained with the same statistics from archived avalanches and outputs a number based on the inputs and how it is being classified. If the output is close to one, the event will be treated as an avalanche, otherwise it will be labeled “Not an Avalanche.”

Results

In this section, we will first discuss the sensitivity analysis of the event detection for the single avalanche and the 2-day avalanche cycle. Secondly, the results from the neural network training and classification will be examined for an avalanche and vehicle neural network.

Event Detection

Avalanche Signal

A range of values for the window size and alpha value were chosen for detection of a single avalanche event. The l^2-norm of the residual is shown in Figure 19. Small windows sizes under 600 seconds tended to have a larger error for all alpha values. Large alpha values over 10^{-9} had a large error for all window sizes. For both large window sizes and small alpha values, the error was attributed to not detecting the start and end times perfectly, but detecting the middle of the avalanche. Almost perfect detection (dark blue in Figure 19) occurred for alpha values in the 10^{-12} to 10^{-10} range with window sizes generally larger than 1,000 seconds.

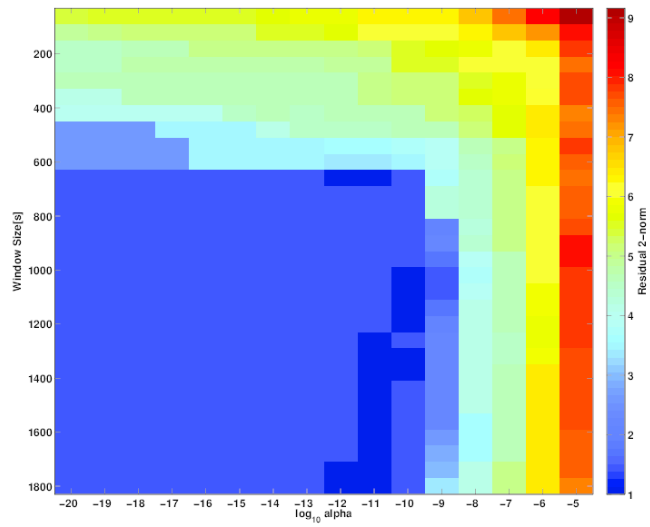

Figure 19. Sensitivity of the Avalanche Signal

Sensitivity of the window size and significance level on the l^2-norm of the residual. High significance level and small window sizes produce the most error in the detection, with almost perfect detection in dark blue. The most accurate detection occurs with an alpha value around 10^{-10} and a window size greater than 600 seconds.

Using the results obtained from Figure 19, a window size of 900 seconds and an alpha value of 10^{-10} were chosen for this avalanche. The detection results in Figure 20b show low probabilities that the Fisher statistic has been previously observed when the avalanche occurs. The instrument noise has low probabilities but are confined to non-overlapping frequency bands which do not trigger an event detection. The start and end times of the automatic detection match well with the manually picked start and end times for the avalanche (Figure 20c).

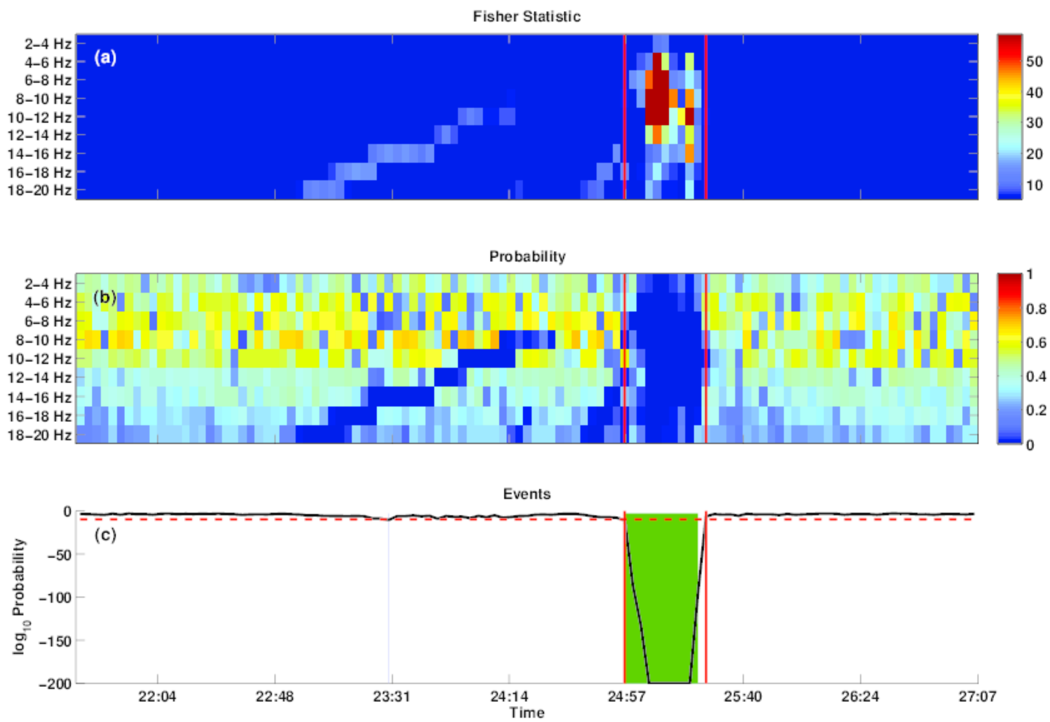

Figure 20. Detection of Small Wet Avalanche

Detection of the small wet avalanche. a. Fisher statistic values. b. Probability of the current observation, low values indicate a new observation and high values indicate a value that has been observed before. c. Product of the probabilities in black with the automatic detection in green. The red lines indicate the actual start and end time of the avalanche.

Avalanche Cycle

Each event was treated as a single detection to determine how well the various window sizes and alpha values could detect the start and end times as well as reducing false positives around the event. The mean and standard deviation of the l^2-norm is shown in Figure 21. Small windows sizes under 600 seconds tended to have slightly higher l^2-norm values. Large alpha values over 10^{-9} tended to have a large error for all window sizes. However, when the alpha value was decreased, the standard deviation of the l^2-norm increased indicating less consistent detections. Therefore, the best values will be around an alpha of 10^{-10} and a window size of around 900 seconds. These values provide a good tradeoff between consistent and accurate detections while reducing the number of false positives.

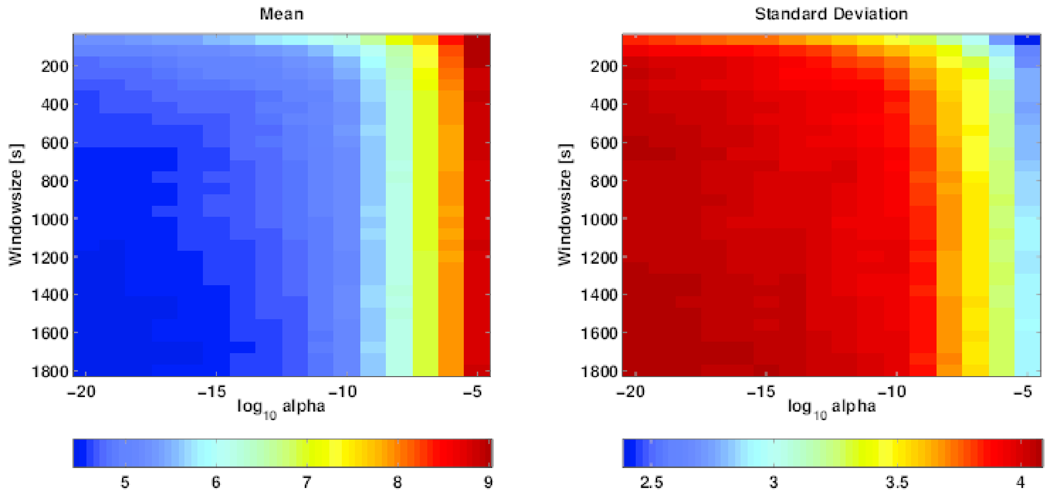

Figure 21. The Mean and Standard Deviation of the l^2-Norm for All the Events

Each event type has a different window size and alpha value combination that optimizes the detection (Figure 22). All events have larger l^2-norm when the alpha value is above 10^{-9}. Window sizes around 900 seconds and an alpha value of 10^{-10} have lower l^2-norm values for all events.

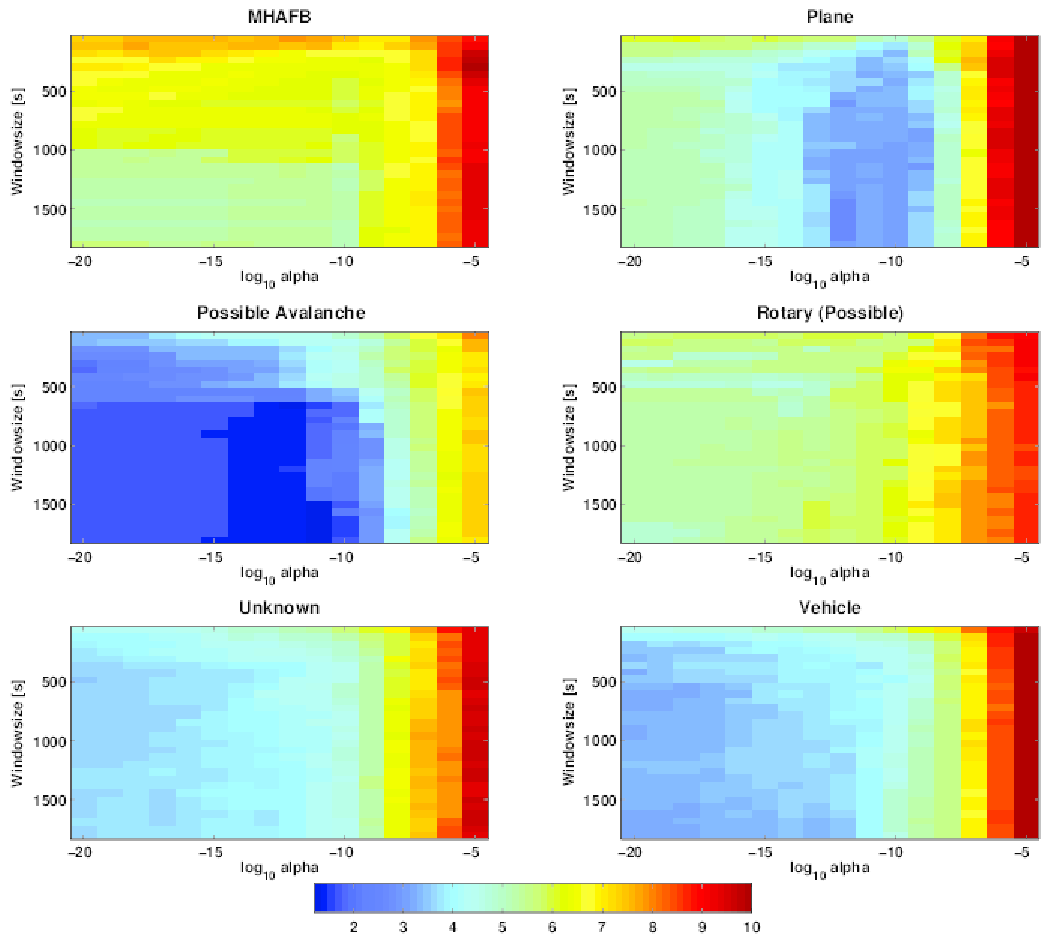

Figure 22. Median Values of the l^2-Norm for All Six Event Types

Event Classification

Two neural networks were created for classification, one for vehicles and the second for avalanches. The avalanche neural network (Figure 23 top) outputs a high value that is above the threshold, which indicates that the event was classified as an avalanche. The avalanche neural network outputs close to zero for an airplane signal, which is below the threshold and is classified as "not an avalanche." The same was true for the vehicle neural network (Figure 23 bottom) that correctly classifies a vehicle and does not classify the avalanche as a vehicle

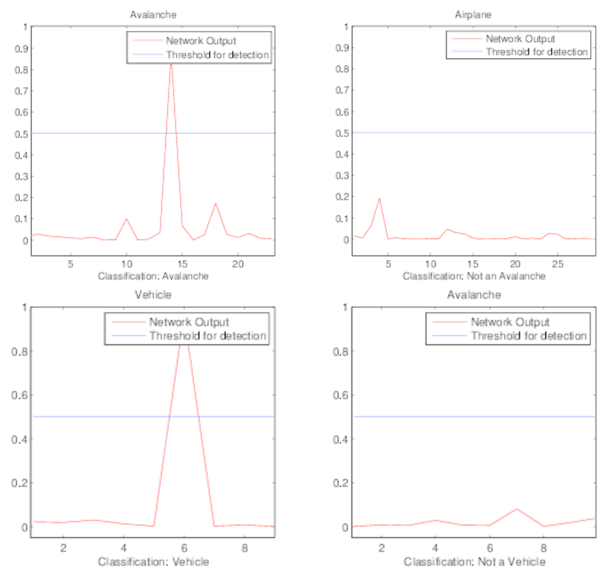

Figure 23. Network Output for Individual Events

Over a longer series of events, the avalanche classification network can reliably classify avalanches that have occurred (Figure 24). The network outputs a value close to zero for events that are not avalanches and outputs a value close to one for an avalanche. However, without more avalanche events, we cannot be certain if the network has been over-trained to the small number of avalanche events.

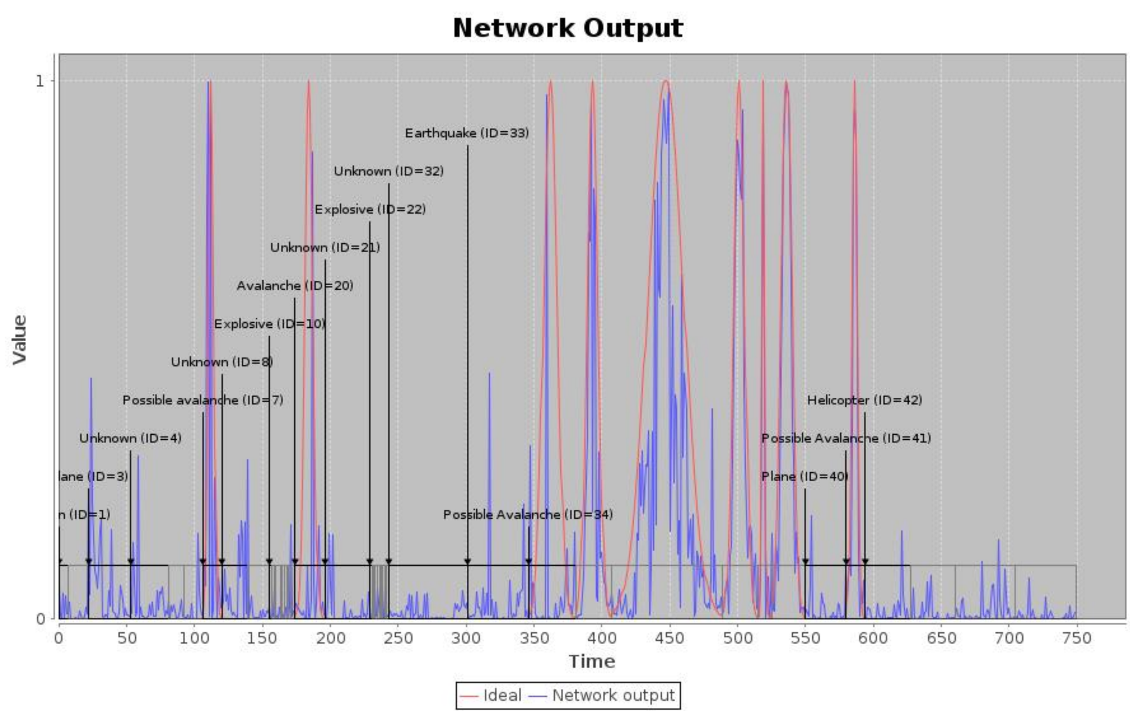

Figure 24. Avalanche Classification Network Output

Discussion and Conclusion

This chapter has outlined a new method for signal detection using non-parametric methods and array processing techniques. Non-parametric methods are advantageous over fitting the data to a certain distribution as no assumptions need to be made about the underlying distribution.[23] A window of Fisher statistic data was used as a background model with which to compare new values. A larger window will create a more stable background model that will be less adaptive to changes in the noise, but will be better at filtering out false positives.

Sensitivity analysis performed on a 2-day avalanche cycle and a single avalanche showed that the automatic event detection was highly dependent on the alpha value. A large alpha increased the number of false positives and decreasing alpha lowered the number of false positives but still had trouble correctly identifying the full event. A trade-off between more accurate detection and increased false positives occurred with a window size of 900 seconds and an alpha value of 10^{-10}.

The neural networks could correctly classify the6 event types observed in the 2-day avalanche cycle. However, the limited number of training events for avalanches, airplanes, MHAFB, and the rotary make the neural network output sensitive to new events due to over-training on the samples. This may lead to false negatives where an event was incorrectly misclassified due to an underrepresentation in the training sample. Therefore, a larger dataset for avalanches will be required for robust classification. The larger dataset can be obtained in two ways. First, the winter season of 2013/2014 contains multiple avalanche cycles and the data will be processed in the future to look for more avalanche signals. Secondly, once the system is deployed, the avalanche forecasters will provide feedback by either confirming or rejecting the automatic classifications, which will help to create a continually growing catalogue of events. Then new neural networks can be created from the growing catalogue. This technique will make the detection system more robust through time as the neural networks learn to recognize new patterns.

It is better to automatically detect more events using a slightly higher alpha rather than missing a small but significant event. All detected events are sent to the neural networks for classification which will filter out significant events from non-significant events. However, the neural network is just the first step and a human analyst will still be required to confirm the network's classification. The human analyst needs to be trained in viewing infrasound data and can be either the avalanche forecasters or a part-time research technician from BSU. As more events are confirmed, the network will need less human interaction but the time frame is unknown.

The ability to train separate artificial neural networks to classify different types of events allows the system to be highly accurate in its results. In other words, instead of having an expert that can always correctly classify any kind of event, we use multiple expert classification systems that are only accurate in classifying a single kind of event and then conducting a vote among them to determine the most likely classification.

The value of this approach is that, while the network training is more time consuming than having a single network, we can detect events in two scenarios: by direct vote or by exclusion. We can confidently say that vehicles will always produce a similar signal in our sensor array, while other natural phenomena, like avalanches, are bound to change. With this information, we can assume that an event is an avalanche not only if the *"avalanche classification network"* labels the event as an avalanche, but also if all the other networks (a vehicle, explosion, airplane network etc.) classify the event as not any of those labels. If the event is not any of the well-known types of events, there is a higher probability that it is an avalanche.

Chapter 4
Avalanche Detection System

Real-Time Application Overview

To decrease the amount of data that must be transmitted, a small low-power computer is used to process the infrasound data on-site. The infrasound signals from the sensors are recorded on the data logger which is connected to the on-site computer. The on-site computer continuously collects data from the data logger and performs the necessary steps for event detection. The events are stored in a database on the computer, which communicates over telemetry to an event database on the server. The avalanche forecasters will access the event database through a web application. The avalanche forecasters will confirm events in the event database which will be used to periodically retrain the neural network. A flow chart of the entire real time application is shown in Figure 25.

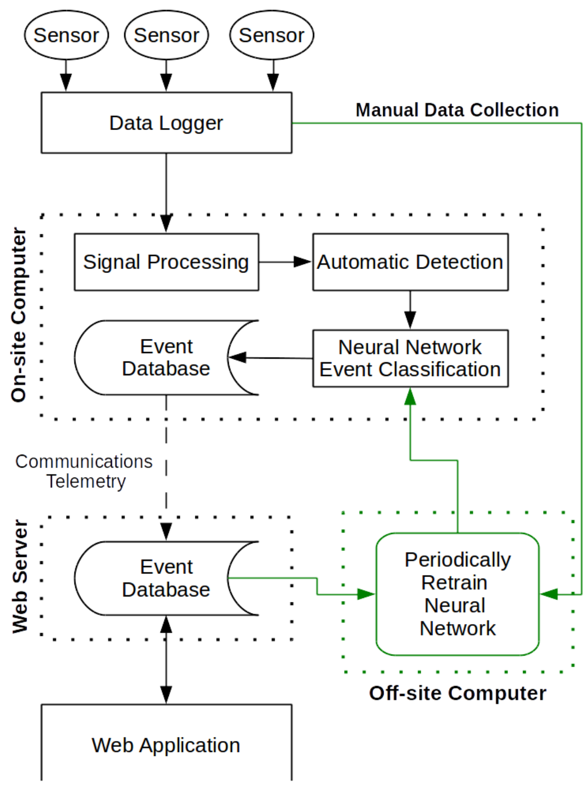

Figure 25. Real Time Flow Chart

Hardware Development

Hardware

The infrasound hardware consists of four main components: seismic data logger, small low power computer on-site, infrasound sensors, and telemetry. The hardware is powered using a bank of deep cycle batteries charged by a solar panel.

Data Logger

The data acquisition is performed using the Quanterra Q330S data logger (Figure 26) from Kinemetrics (www.kmi.com). The Q330S is a 6 channel, high resolution, ultra-low power seismic data logger that has proven performance in the polar regions and excels at telemetering data. Various sample rates exist for the Q330S and we have chosen to use 100Hz sampling rate to ensure that all waveforms in the infrasound bandwidth (1-20Hz) are captured. Timing is obtained from a GPS receiver attached to the Q330S. Table 3 shows selected specifications that are used for this project, more can be found at www.kmi.com.

Figure 26. Quanterra Q330S

Table 3. Q330S Specifications

Channels	*6*
Gain	*30*
Sample Rate	*100*
Resolutions	*24-bit A/D*
Timing	*Precision TCXO, locked to GPS*
Telemetry	*UDP or TCP IP over Ethernet connection.*
Temperature	*Fully specified -20 to +50C* *Operative -40 to +70C*
Memory	*32Mb RAM standard*
Network	*Dual Ethernet*
Serial Ports	*1 console ports*
Media	*Dual USB up to 32G total*
Power	*<0.8W avg. 12VDC, additional 1.2W for continuous Baler operation.*
Size	*17 x 4 x 6 in., 9 lbs.*

On-site Computer

On-site computing is performed by a low-power computer from fitPC (Figure 27). For this project, we have selected the fit-PC2i which is a small energy-efficient computer that has 2GB of RAM, a 1.6GHz processor and runs at 10 percent the power of a regular PC (Table 4). A 120GB solid state hard drive ensures fast reading and writing of data with better performance at lower temperatures. The fitPC runs the XUbuntu Linux distribution. The fitPC automatically restarts when power is applied in the event that power is lost.

Figure 27. Fit-PC2i from fitPC

The data processing is very fast and does not require the computer to be on full-time, but enters sleep mode between processing windows to further conserve power. Typically the computer may be on for 10 to 20 minutes per hour which is equivalent to a 2 - 4 Watt power consumption.

Table 4. fit-PC2i Specifications

Weight	*13 oz*
Size	*4 x 4.5 x 1.05 in.*
Power	*Low load 6W* *Full load 8W* *<1W standby*
Network	*2 x ethernet ports, 802.11 wifi*
Ports	*miniSD, HDMI, audio, 2 ethernet ports, 4 USB 2.0*
Memory	*2GB*
Processor	*Intel Atom Z530 1.6 GHz*

Infrasound Sensors

During the first two years of this project (2010-2012), we used a piezoresistivity pressure transducer (PPT) manufactured with micro-electromechanical systems (MEMS) technology developed at the New Mexico Institute of Mining and Technology (Infra-NMT). [29] Infra-NMT sensors use the change in resistance due to the displacement of a diaphragm to measure the infrasound signals. The sensors have a flat frequency response in the infrasound bandwidth and a sensitivity of ~45 $\mu V\ Pa^{-1}$, making it an ideal sensor for research purposes. However, this sensor was developed for volcanic studies where signals are generally above 1 Pa. The signals generated from avalanches are typically much less than 1 Pa and are difficult to detect with the Infra-NMT sensors.

Therefore, we developed a new infrasound sensor that uses an electrical condenser microphone (ECM) typically found in phones and computers. ECM's use one plate of a capacitor as a diaphragm that changes capacitance when deflected, which can be measured. Since ECM's were developed for phones which operate in the threshold of human hearing (20 to 20,000 Hz), the sensors must be adapted for use in the infrasound bandwidth and can have different frequency response for each sensor. However, these sensors have proven useful for infrasound studies of volcanoes and previous avalanche detection studies. [10,30,31] The extreme low cost of these sensors and higher signal-to-noise ratio (SNR) than the Infra-NMT, make them ideal for detection of small amplitude signals.

Telemetry

Telemetry is achieved with an Iridium satellite modem for proof of concept telemetry. The modem sends small text files (<1 KB) created by the fitPC based on detected events. The satellite modem is for testing purposes only as sending large amount of data is not economically feasible.

Real Time Installation

Figure 28 shows the Real Time System's main components deployed in the field. Power from the solar panel is connected to a battery bank and is controlled with the charge controller. Power from the battery bank is distributed through the power box where the Q330S, fitPC, and modem are connected. The infrasound sensors come into the box and are connected to the sensor input cables of the Q330S. The console port serves two uses, first to connect with the fitPC through the ethernet cable and second, to apply power. When an event is detected, the fitPC creates a telemetry file which is sent to the Iridium modem for transmission to the off-site server.

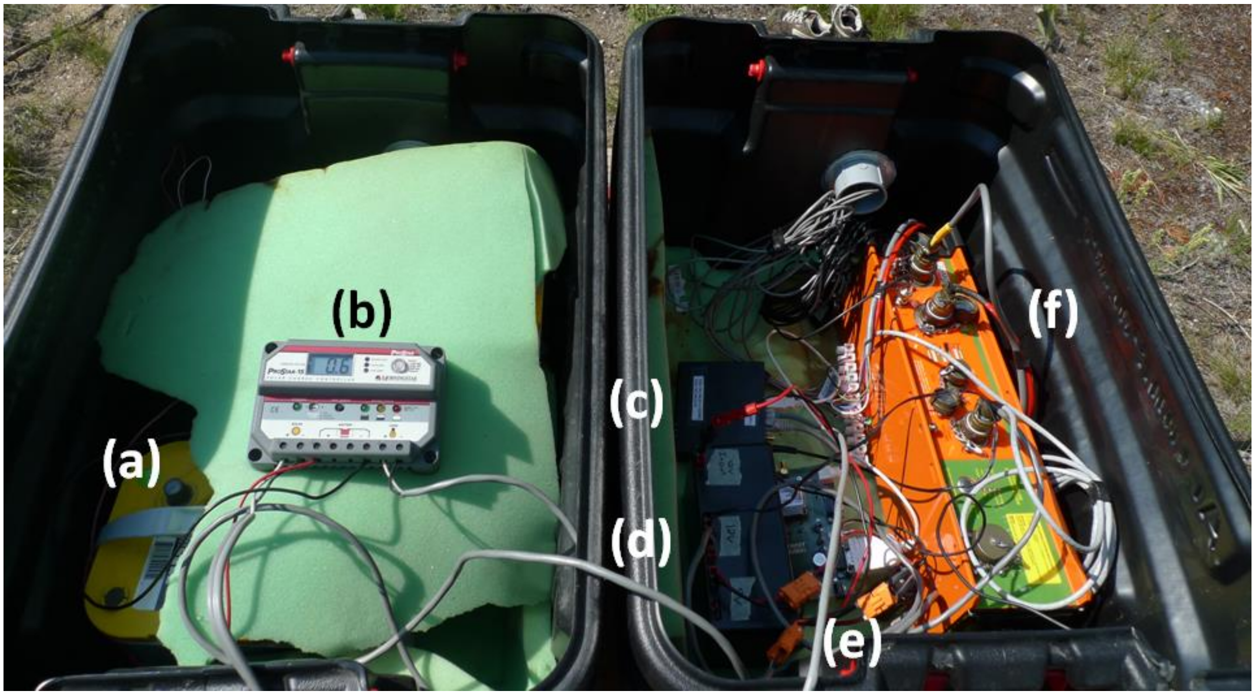

Figure 28. Real Time System

*Real time system components deployed in the field. a. Deep cycle batteries, ~150Ah.
b. Charge controller to manage charging of batteries from solar panels.
c. FitPC. d. Power box with two 12V and two 5V outputs.
e. Iridium satellite modem. f. Q330S.*

Service Life

We expect the Q330S to have significant service life due to its rugged design and proven use. After each season, laboratory tests would be needed to ensure that the data logger is still functioning properly and after 5 years, it should be sent to the manufacturer for service. With proper maintenance and service, we might expect the Q330S to last up to 10 years.

The fitPC comes with a 3 year manufacturer warranty. Similar to the data logger, it should be inspected and tested after each season to ensure that all systems are functioning properly. Due to being out in the elements, we would want to thoroughly inspect the computer should be thoroughly inspected after 3 years to ensure that the hard drive and computer hardware are functioning properly. With proper maintenance and service we would expect that the computer last approximately 5 years. At that point, the computer will most likely be obsolete and an upgrade to a newer and faster computer would greatly enhance the detection.

The ECM infrasound sensors are expected to have a service life of about 2 years. We have deployed ECM's for two winters with no major problems and have only needed to replace 2 sensors. Since the sensors are very inexpensive, if there are any sign that the signal is not perfect, the sensor can be replaced.

Software Development

On-Site Control

A Graphical User Interface (GUI) controls the two major processes on the fitPC, communication with the Q330S and data processing (Figure 29). When the computer is initially powered up either through startup, rebooting, or connecting power, the RunNetmon GUI initializes. When RunNetmon starts, it checks whether a connection with the Q330S has been established. If the connection is not present, RunNetmon will start communication and if communication has already been established, RunNetmon will not try to restart the connection. After the connection with the Q330S is detected or started, RunNetmon will start the data processing scripts. The processing flow is shown in Figure 30 and will be covered in detail below.

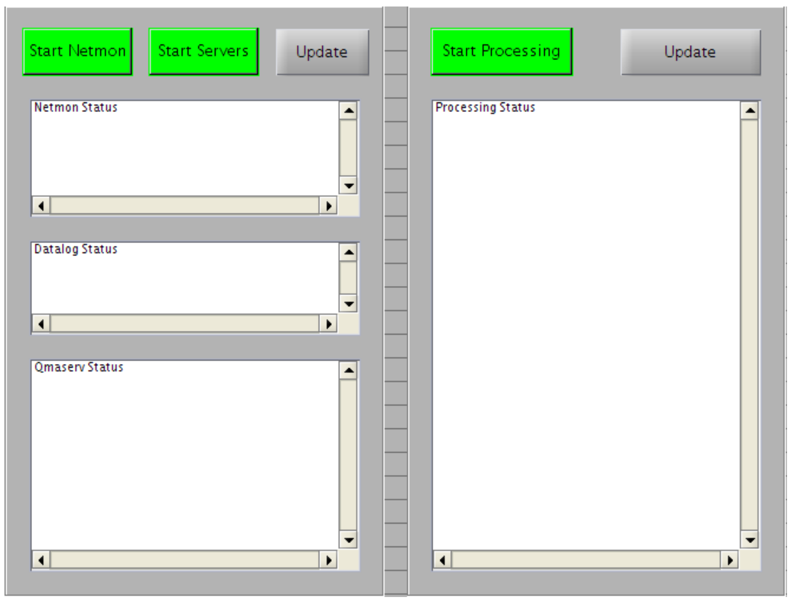

Figure 29. RunNetmon GUI

GUI on-site that controls the Q330S communication and processing. The left panel buttons control starting and stopping of netmon, qmaserv, and datalog (through netmon). The left panel update button will update the status for netmon, datalog, and qmaserv from the log files. The right panel controls the processing flow and the update button will update the status of the processing from the log file.

The RunNetmon GUI allows a user to visit the site and easily control the major aspects of the avalanche detection system. When connected to the fitPC, the user will see the RunNetmon GUI (Figure 29). The left panel controls netmon, which controls the communication server (qmaserv) and data logging program (datalog). The left button controls the start/stop of netmon which will also start/stop qmaserv and datalog. The middle button controls the start/stop of communication and data logging but leaves netmon running. The right button updates the status obtained from the netmon, qmaserv, and datalog log files.

The right panel controls the data processing. The left button controls the start/stop of the processing and is useful when visiting the site. The processing algorithm will put the fitPC to sleep and this button allows the user to stop processing for servicing. The right button updates the processing status from the log file.

RunNetmon is a simple GUI with limited functionality making it a robust method of controlling the communications and processing. If for some reason RunNetmon should fail, a script will restart RunNetmon automatically which ensures that the communications and data processing are always running.

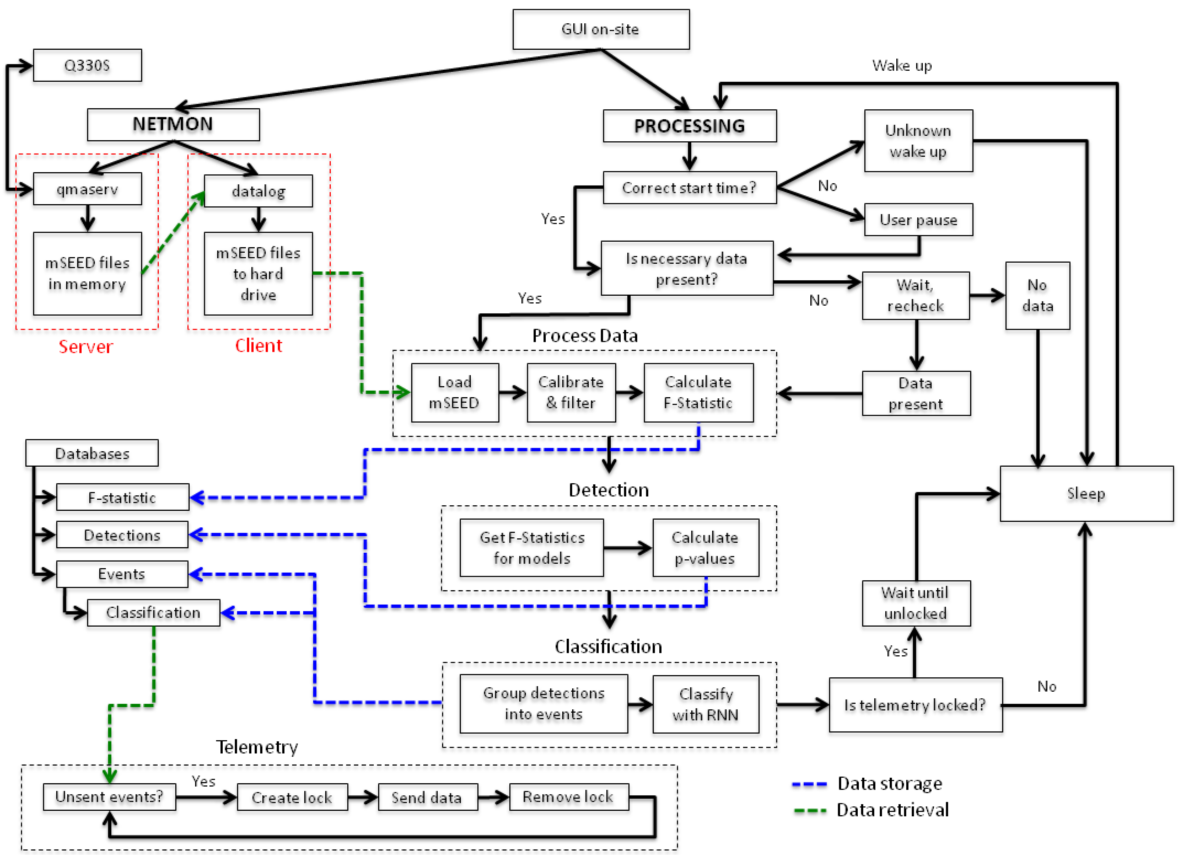

Figure 30. On-Site Data Retrieval and Processing Flow Chart

Q330S and fitPC Communication

Communication between the Q330S and fitPC is through Comserv software bundle, maintained by ISTI (www.isti.com), UC Berkley, and Quanterra. The library was developed to communicate with Quanterra digitizers in a Windows or Linux environment. Mountain Air (qmaserv) was developed to utilize the advanced IP communication ability of the Q330's. The software bundle consists of the communication server (qmaserv), a set of clients that communicate with qmaserv, and netmon which controls the startup and shutdown of multiple servers and clients.

qmaserv

The communications server qmaserv controls communication between the fitPC and Q330S through an ethernet cable. (Figure 30) qmaserv generates miniSEED (Standard for the Exchange of Earthquake Data) packets to provide data and related information from the Q330S to the clients. The data packets are stored in memory for the client programs to read. Each station requires a dedicated qmaserv server and multiple servers can run on a single computer.

datalog

datalog is a client that receives miniSEED data packets from a single qmaserv server and records the data to disk (Figure 30). datalog has been configured to only write the miniSEED packets from the five infrasound and one geophone channels. This ensures that only the necessary data is written to disk and reduces unwanted communication between datalog and qmaserv for fast real time application.

netmon

netmon is a management program used to control the startup, shutdown, and restart of the communication servers and clients for multiple stations (Figure 30) netmon can be used to manually check the status, perform startup, or shut down operations for a station. For continuous operation, netmon is run in background mode and automatically monitors the status and restart of servers and clients if one were to stop.

Processing Flow

The following section outlines the data processing occurring in near real-time on the fitPC. The processing algorithms have been developed by BSU for the purpose of avalanche detection using infrasound.

Database Creation

A SQLite database (www.sqlite.org) is an SQL database engine that excels at accessing a local database on the hard drive and is not designed to be used over an internet connection. A database can contain multiple tables which have columns (fields) representing a variable and rows (records) that represent the data to be stored. Data is inserted into or accessed from a table using SQL commands.

The avalanche detection system uses one database that contains three tables: Fisher statistic data (Table 6), detection data (Table 7), and events (Table 8). The database stores and provides data during processing depending on the current processing step (Figure 30). Using a database like SQLite makes the processing more efficient by only retrieving data that is necessary for the current step instead of loading a large data file that contains more data than needed.

Signal Processing

Table 5. Configuration File for Processing Parameters

Variable	Value	Description
WindowLength	600.0 seconds	Processing window length
MaxWindowLength	3.0 hours	Number of hours back to look for miniSEED files
TimeOffset	10.0 minutes	Offset back from current time to end processing window
master_node	1	Channel to base time shifts off for Fisher statistic
c	320.0 m/s	Speed of sound
num_nodes	31	Number of grid nodes to divide slowness space in one dimension
nfft	2048	Number of points to perform the Fast Fourier Transform for Power Bands
WindowSize	6 seconds	Length of window to calculate Fisher statistic
WindowOverlap	3 seconds	Length of overlap between windows
Channels	1 2 3 4 5	Channels to use
sampleRate	100 Hz	Sample rate of Q330S
Freq1	2 - 4 Hz	Frequency band for processing
Freq2	4 - 6 Hz	Frequency band for processing
Freq3	6 - 8 Hz	Frequency band for processing
Freq4	8 - 10 Hz	Frequency band for processing
Freq5	10 - 12 Hz	Frequency band for processing
Freq6	12 - 14 Hz	Frequency band for processing
Freq7	14 - 16 Hz	Frequency band for processing
Freq8	16 - 18 Hz	Frequency band for processing
Freq9	18 - 20 Hz	Frequency band for processing
FreqB	2 - 20 Hz	Broadband for Neural Network processing
Power1	1.0 - 5.0 Hz	Frequency range to calculate power
Power2	5.0 - 10.0 Hz	Frequency range to calculate power
Power3	10.0 - 15.0 Hz	Frequency range to calculate power
Power4	15.0 - 20.0 Hz	Frequency range to calculate power
Power5	20.0 - 50.0 Hz	Frequency range to calculate power
EventBuffer	5	Number of windows to buffer the front and back for event database
WindowBack	900 seconds	Non-parametric window length
alpha	10^{-10}	Significance level for event detection
NumberofTries	8	Number of tries to check if data has been loaded
EventThresh	0.5	Neural network output threshold for event classification

The main function of the on-site processing is to reduce the amount of data to transfer. Since this requires a small computer to be continually acquiring data from the data logger and computation expense of data processing, power consumption becomes a primary concern. To enhance the power efficiency of the computer, we put the computer to sleep after processing and telemetry are completed. At a pre-determined time, the computer will wake, begin processing, and go back to sleep. The processing flow follows the flow chart in Figure 30 with processing parameters defined in a configuration file (Table 5).

The processing window length (in seconds) is defined in the processing configuration file. The processing window defines the start and end time for data processing and controls when the computer awakes. When the computer awakes, it checks whether or not the current time is close to the defined start time. A large difference could be caused by either an unknown source or user logging into the computer. If it is an unknown source, the computer will go back to sleep. If a user desires to log into the computer, the processing will pause until the user resumes data processing.

After the start time has been determined, the processing checks to see if all the necessary infrasound data is available. An algorithm checks to see if any miniSEED files from datalog match the start and end time of the processing window. The algorithm will attempt to read the data for a given number of tries defined in the configuration file. This allows the most current data from Q330S to be retrieved and written to disk. If no data is present after all the tries, the computer will be put to sleep.

Once the necessary data is present, data processing can begin (Figure 30). The infrasound data between the start and end time is loaded, calibrated, and filtered to the desired bandwidths defined in the configuration file. The Fisher statistic is calculated for the processing window.[21] For each window, the maximum Fisher statistic, with the corresponding back azimuth, apparent velocity, and slowness vector are saved to the Fdata table in the database (Table 6). The power in the frequency domain for five different frequency ranges are also calculated for classification and stored.

Table 6. Fisher Statistic Data in Fdata Table

Field	Description
ID	Integer ID for each event, auto increments
StartTime	Start time of the event window
EndTime	End time of the event window
EventStart	Estimated start time of the event
Duration	Duration in seconds of the event
Classification	Classification of the event
Sent	Whether or not the event has been classified/sent
Backazimuth	Back azimuth for entire event, string with values separated by commas
v_app	Apparent velocity for entire event, string with values separated by commas
F	Fisher statistic for entire event, string with values separated by commas
pvalue	P-value for entire event, string with values separated by commas
Power1	Power1 for entire event, string with values separated by commas
Power2	Power2 for entire event, string with values separated by commas
Power3	Power3 for entire event, string with values separated by commas
Power4	Power4 for entire event, string with values separated by commas
Power5	Power5 for entire event, string with values separated by commas

Event Detection

After the Fisher statistic is calculated within the processing window, event detection is performed. (Figure 30) Detection requires a large amount of data to make up the background model and prior Fisher statistic data is loaded from the database. The p-values are calculated for each time window and are stored in the Detection table of the database (Table 7).

Table 7. Detection Results in Detection Table

Field	Description
date	Date and time of window
pvalue	P-value of the window
h	If pvalue was less than alpha

Event Grouping

Event detection only determines if a single processing window is an event and does not group into larger events. Therefore, the next step is to group the single events into larger groups for classification (Figure 30, Classification). The single events are grouped based on how similar the back azimuth and apparent velocity are of two close single detections. If the two single event properties are similar and are reasonably close in time and may be from the same event, then they will be grouped together. A new event will occur when the back azimuth and apparent velocity are significantly different or there is a large time gap between two single events. Once the single events have been grouped, the new large event is stored in the database (Table 8).

For the classification, padding must be added to the event. Therefore, the EventStart and Duration of the actual event are stored separately from the StartTime and EndTime of the window around the event. The event properties are a string of comma separated values which allows time series data to be stored in the database. Each event gets a unique ID for identification.

Table 8. Event Data in Events Table

Field	Description
ID	Integer ID for each event, auto increments
StartTime	Start time of the event window
EndTime	End time of the event window
EventStart	Estimated start time of the event
Duration	Duration in seconds of the event
Classification	Classification of the event
Sent	Whether or not the event has been classified/sent
Backazimuth	Back azimuth for entire event, string with values separated by commas
v_app	Apparent velocity for entire event, string with values separated by commas
F	Fisher statistic for entire event, string with values separated by commas
pvalue	P-value for entire event, string with values separated by commas
Power1	Power1 for entire event, string with values separated by commas
Power2	Power2 for entire event, string with values separated by commas
Power3	Power3 for entire event, string with values separated by commas
Power4	Power4 for entire event, string with values separated by commas
Power5	Power5 for entire event, string with values separated by commas

Event Classification

Event classification is achieved using a neural network that uses the events stored in Table 8. Once the event has been detected, the statistics collected in the previous steps (F-statistics, p-values and back azimuth) are collected and provided to the neural network as inputs.

The network has been trained with the same statistics from archived avalanches and will output a number based on the inputs and how it is being classified. If the output is close to 1, the event will be treated as an avalanche, otherwise it will be labeled “Not an Avalanche.”

When an avalanche is detected, a small text file is created. The file contains important fields from the Events table like the StartTime, EndTime, EventStart, Duration, Classification, Backazimuth, v_{app}, and F. The file is saved with a time stamp as the file name which is used by the telemetry process to determine new events.

Telemetry

The telemetry is a separate process that runs in parallel to the signal processing and event detection. (Figure 30) The telemetry process looks for the most recent event text files that have not been sent. When a file needs to be sent, the process locks the computer from going to sleep while the file is sent to the Iridium modem. Once the messages have been sent, the lock file is removed and the computer may go to sleep.

Once the file is sent, it is routed to a server waiting for the message. The raw message is inserted into a database table on the server without any processing. Once the raw message is in the table, the message content can be parsed into the fields that were sent, making it easier to view the data on the web application.

System Testing

On August 8, 2014 the prototype version of the real-time infrasound detection system was installed at the 100.5 array located at mile post 100.5. The neural network was trained to detect vehicles, a much easier task in the summer. During testing, the system detected approximately 146 events between August 8th 20:30 UTC and August 9th 08:15 UTC. The system sent out 16 messages with information about the vehicle detections. The events could be viewed on a simple webpage.

The test has been successful in processing the data real-time, detecting events, correctly classifying vehicles, and relaying the information over Iridium satellite modem. However, the test came to an end with a bug in the code which shut down the processing steps. The test brought to light areas to improve the system prior to operational deployment.

Chapter 5
Implementation

This project provides the foundation for a implementation of a real-time avalanche detection system. The record showed that the infrasound system and software can accurately detect avalanches occurring within a highway corridor. The software required for avalanche detection with infrasound was completed at the end of Year 2. Should ITD choose to implement the system developed through this research, it would be necessary to purchase the instrumentation to be installed for real-time application by BSU from additional funding not part of this grant. This cost was estimated at $50,000 when the project started.

Through this research project, BSU has developed a standalone avalanche detection system that can be deployed to other locations (e.g. US12). To implement the system in another location would require deploying the array and building a catalogue of the events to be detected. Depending on the location and frequency of avalanche events, it could take at least a year to create a robust detection system and would require a research technician from BSU to perform the necessary data processing to build the initial catalogue. Beyond avalanches, infrasound can be used to detect rock fall, landslides, and traffic and with the proper event catalogue this system could detect anything that emits an infrasound signal.

Through the results of this research project, BSU recommends that the first array installation be placed at the previous array location at mile post 100.5. This location provided robust results and targets a significant portion of the most frequent avalanche paths. If the system were to expand, BSU recommends locating an array near mile post 96.92 and two arrays between mile post 98 and 99.7.

Personnel Costs

The first year of deployment will be the first year that the entire system will be running in an operational mode. We recommend a research technician from BSU will work part time during the winter of 2014/2015 to deal with any problems that may occur. The research technician will be able to:

1. Provide extra technical support to the ITD avalanche forecasters for web application, event database, or general questions.
2. Perform additional web application development to ensure an excellent and functioning end product.
3. Perform site maintenance to ensure that all hardware is properly working.
4. Maintain event database. A more robust database will allow for significantly better event classification.
5. Fine tune event detection algorithms to minimize missed events.
6. Quality control the infrasound data.

7. Retraining the Artificial Intelligence with the new event database.

8. Take down infrasound array during the summer for testing and storage for next winter.

9. Improve SNOSS results using avalanche timing from infrasound.

Avalanche Forecaster Time

ITD’s avalanche forecasters can expect to spend approximately 1 hour per day reviewing events that occurred the previous day. This daily interaction with the detection system will provide high quality data to later improve the detection and classification of events. As more events are classified, we expect the amount of time needed for reviewing events will decrease. The amount of time required can also be decreased by having a BSU research technician also reviewing events when the ITD’s forecasters are not able to.

Installation Budget

The following outlines an estimated budget for the installation of the infrasound array. (Table 9) The budget includes all the hardware costs, personnel costs for installation, and personnel costs for maintenance during the first year.

Table 9. Array Budget

Item	Quantity	Price	Amount
Array			
Quanterra Q330S 6-ch	1	$ 13,000.00	$ 13,000.00
Electronics Enclosure	1	$ 500.00	$ 500.00
Infrasound Sensor	5	$ 20.00	$ 100.00
Geophone 4.5Hz Short Period	1	$ 150.00	$ 150.00
Infrasound Cable (500’ roll)	1	$ 300.00	$ 300.00
Cable Conduit (500’ roll)	1	$ 600.00	$ 600.00
Sensor Poles	5	$ 50.00	$ 250.00
Sensor Housing	5	$ 50.00	$ 250.00
		Subtotal	$ 15,150.00
Power System at Array			
Tower	1	$ 5,000.00	$ 5,000.00
Solar Panels (80W)	3	$ 300.00	$ 900.00
Charge Controller	1	$ 300.00	$ 300.00
Deep Cycle Battery (50-85 A hr)	4	$ 300.00	$ 1,200.00
Electrical Safety Equipment	1	$ 200.00	$ 200.00
Mounting Equipment	1	$ 1,000.00	$ 1,000.00
Conduit	1	$ 300.00	$ 300.00
Sunwize E4-500 4 Battery Enclosure	1	$ 450.00	$ 450.00
Cables	1	$ 300.00	$ 300.00
		Subtotal	$ 9,650.00
Communications			
Iridium 9603-U from JouBeh Technologies	1	$ 325.00	$ 325.00
Activation Fee	1	$ 40.00	$ 40.00
Monthly Subscription Fee		$ 16.00	
Data Charge per Kbyte		$ 1.40	
		Subtotal	$ 365.00
Signal Processing			
FitPC 2i Running Xubuntu	1	$ 500.00	$ 500.00
Backup Hard Drive (minimum 1TB)	1	$ 200.00	$ 200.00
		Subtotal	$ 700.00
Installation Costs			
Travel To and From Site	1	$ 3,000.00	$ 3,000.00
Additional Labor	20	$ 150.00	$ 3,000.00
		Subtotal	$ 6,000.00
Maintenance & Running Costs			
Maintaining Site for Year 1	1	$ 10,000.00	$ 10,000.00
Maintaining Processing & Databases	1	$ 10,000.00	$ 10,000.00
Tuning NN to New Events	1	$ 10,000.00	$ 10,000.00
		Subtotal	$ 30,000.00
Array			$ 15,150.00
Power System			$ 9,650.00
Communications			$ 365.00
Signal Processing			$ 700.00
Installation			$ 6,000.00
Maintenance & Running			$ 30,000.00
Miscellaneous Expenses			$ 5,000.00
		Total	$ 66,865.00

References

1. **McClung, D. M., and P. A. Schaerer**. *The Avalanche Handbook, 3rd Edition,* Seattle, WA*:* The Mountaineers, 2006.

2. **Kogelnig, A., E. Suriñach Cornet, I. Vilajosana Guillén, J., J. Hübl, B. Sovilla, M. Hiller and F. Dufour**. "On the Complementariness of Infrasound and Seismic Sensors for Monitoring Snow Avalanches." *Natural Hazards and Earth System Science,* Vol*.* 11, No. 8 (Aug. 2011): 2355 - 2370.

3. **Comey, R. H. and T. Mendenhall**. "Recent Studies Using Infrasound Sensors to Remotely Monitor Avalanche Activity." *Proceedings of the 2004 International Snow Science Workshop, Jackson Hole, WY.* 2004.

4. **Bedard Jr. A. J.** "Detection of Avalanches Using Atmospheric Infrasound." *Proceedings of the 57th Western Snow Conference, Fort Collins, CO.* 1989.

5. **Bedard Jr, A. J.** "An Evaluation of Atmospheric Infrasound for Monitoring Avalanche." *In* W. D. Neff, Editor, *Proceedings of 7th International Symposium on Acoustic Sensing and Associated Techniques of the Atmosphere and Oceans, Boulder, CO.* 1994.

6. **Bedard Jr, A. J., G. E. Greene, J. Intrieri, and R. Rodriguez.** "On the Feasibility of Detecting and Characterizing Avalanches by Monitoring Radiated Subaudible Atmospheric at Long Distances." pp. 267 – 275, *In*: *Proceedings of A Multidisciplinary Approach to Snow Engineering Foundation Conference, Santa Barbara, CA.* 1988.

7. **Chritin, V., M. Rossi and R. Bolognesi.** "Acoustic Detection System for Operational Avalanche Forecasting." pp. 129 – 133, *In*: *Proceedings of the 1996 International Snow Science Workshop, Banff, Canada.* 1996.

8. **Scott, E. D. and C. Lance**. "Infrasonic Monitoring of Avalanche Activity." *Proceedings of the International Snow Science Workshop, Penticton, BC:* 2002.

9. **Scott, E. D., C. T. Hayward, R. F. Kubichek, J. C. Hamann, and J. W. Pierre.** "Results of Recent Infrasound Avalanche Monitoring Studies." pp. 696 – 704: *Proceedings of the International Snow Science Workshop, Jackson Hole, WY:* 2004

10. **Scott, E. D., C. T. Hayward, R. F. Kubichek, J. C. Hamann, J. W. Pierre, B. Comey and T. Mendenhall,** "Single and Multiple Sensor Identification of Avalanche-Generated Infrasound." *Cold Regions Science and Technology*, Vol. 47, No. 1-2 (Jan. 2007): 159 - 170.

11. **Ulivieri, G., E. Marchetti, M. Ripepe, I. Chiambretti, G. De Rosa and V. Segor.** "Monitoring Snow Avalanches in Northwestern Italian Alps Using an Infrasound Array." *Cold Regions Science and Technology*, Vol. 69, No. 2-3, (Dec. 2011): 177 - 183.

12. **Cansi, Y.** "An Automatic Seismic Event Procesing for Detection and Location - the P.M.C.C. Method." *Geophysical Research Letters,* Vol. 22, No. 9 (May 1995): 1021 - 1024.

13. **Rost, S. and C. Thomas.** "Array Seismology: Methods and Applications." *Reviews of Geophysics,* Vol. 40, No. 3 (Sept. 2002): 2-1 – 2-27.

14. **Douglas, A. and A. Douglas.** "International Handbook of Earthquake and Engineering Seismology 23 Seismometer Arrays—Their Use in Earthquake and Test Ban Seismology." *International Geophysics Series*, Vol. 81, (2002): 357 – 367.

15. **Karason, H, and R. D. van der Hilst**. "Tomographic Imaging of the Lowermost Mantle with Differential Times of Refracted and Diffracted Core Phases (*PKP*, *P*diff)." *Journal of Geophysical Research: Solid Earth,* Vol. 106, No. B4 (April 2001): 6569 - 6587.

16. **Johnson, J. B.** "Source Location Variability and Volcanic Vent Mapping with a Small-Aperture Infrasound Array at Stromboli Volcano, Italy." *Bulletin of Volcanology,* Vol. 67, No. 1 (Dec. 2004): 1 - 14.

17. **Arechiga, R. O., J. B. Johnson, H. E. Edens, R. J. Thomas, and W. Rison.** "Acoustic Localization of Triggered Lightning." *Journal of Geophysical Research - Atmospheres,* Vol. 116, No. D9 (May 2011).

18. **Havens, S. and H.-P. Marshall.** *Development of an Operational Avalanche Forecasting Model for Highway 12 and 21.* Boise, ID: Boise State University for the Idaho Transportation Department, RP207, 2011

19. **Havens, S.** *Development and Application of Tools for Avalanche Forecasting, Avalanche Detection, and Snowpack Characterization.* Boise, ID: Boise State University, PhD Dissertation, p. 191, 2014.

20. **Melton, B. S., and L. F. Bailey.** "Multiple Signal Correlators." *Geophysics* Vol. 22, No. 3 (July 1957): 565 - 588.

21. **Blandford, R. R.** "An Automatic Event Detector at the Tonto Forest Seismic Observatory." *Geophysics, Vol.* 39, No. 5 (Oct. 1974): 633 - 643.

22. **Smart, E. and E. A. Flinn.** "Fast Frequency-Wavenumber Analysis and Fisher Signal Detection in Real-Time Infrasonic Array Data Processing." *Geophysical Journal International*, Vol. 26, No. 1-4 (1971): 279 - 284.

23. **Arrowsmith, S. J., R. Whitaker, S. R. Taylor, R. Burlacu, B. Stump, M. Hedlin, G. Randall, C. Haywood and D. ReVelle.** "Regional Monitoring of Infrasound Events Using Multiple Arrays: Application to Utah and Washington State." *Geophysical Journal International, Vol.* 175, No. 1 (Oct. 2008): 291 - 300.

24. **Martinez, W. L., and A. R. Martinez.** *Computational Statistics Handbook with MATLAB, 2nd Edition.* Boca Raton, FL: Chapman & Hall/CRC Computer Science & Data Analysis, 2008.

25. **Brachet, N. and J. Coyne.** "The Current Status of Infrasound Data Processing at the International Data Center." pp. 873 - 881, *In*: *28th Seismic Research Review: Ground-Based Nuclear Explosion Monitoring Technologies*, 2006.

26. **Elgammal, A., R. Duraiswami, D. Harwood, and L. S. Davis.** "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance." *Proceeedings of the IEEE*, Vol. 90, No. 7, (July 2002): 1151 - 1163.

27. **Kosko, B.** *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence/Book and Disk.* Vol. 1. Englewood Cliffs, NJ: Prentice Hall, 1992.

28. Fahlman, S. E. "The Recurrent Cascade-Correlation Architecture" *Advances in Neural Information Processing Systems,* Vol. *3,* (1991): 190 – 196.

29. **Marcillo, O., J. B. Johnson, and D. Hart.** "Implementation, Characterization, and Evaluation of an Inexpensive Low-Power Low-Noise Infrasound Sensor Based on a Micromachined Differential Pressure Transducer and a Mechanical Filter." *Journal of Atmospheric and Oceanic Technology,* Vol. 29, No. 9 (Sept. 2012): 1275 - 1284.

30. **Johnson, J. B., R. C. Aster, M. C. Ruiz, S. D. Malone, P. J. McChesney, J. M. Lees, and P. R. Kyle.** "Interpretation and Utility of Infrasonic Records from Erupting Volcanoes ." *Journal of Volcanology and Geothermal Research,* Vol. 121, No. 1–2 (Feb, 2003): 15 - 63.

31. **Ripepe, M., E. Marchetti, and G. Ulivieri**. "Infrasonic Monitoring at Stromboli Volcano During the 2003 Effusive Eruption: Insights on the Explosive and Degassing Process of an Open Conduit System." *Journal of Geophysical Research: Solid Earth,* Vol. 112, No. B9 (Sept. 2007).