2015 16th IEEE International Conference on Mobile Data Management

On the Feasibility of Leveraging Smartphone Accelerometers to Detect Explosion Events

Srinivas Chakravarthi Thandu Pratool Bharti Levi Malott Sriram Chellappan
Department of Computer Science, Missouri University of Science & Technology
Rolla, MO 65401 USA
Email: {stgk4, pbtn4, lmnn3, chellaps}@mst.edu

***Abstract*—In this paper, we investigate the feasibility of leveraging the accelerometer in modern smartphones to detect the triggering of explosion events. By emplacing a static smartphone and a state-of-the-art seismometer in the vicinity of real explosion blasts (conducted at an Explosives Research Lab in a university setting), and comparing their detected event readings, we make several insightful contributions. We find that readings from events in the smartphone and the seismometer are highly correlated in the temporal and frequency domain. We then demonstrate the feasibility of designing an algorithm in the smartphone (executing as an app) to detect the triggering of an explosion based on comparing short term sudden spikes in vibrations due to an explosion event, and long-term dormancy in vibration readings (in the absence of an explosion). To the best of our knowledge, ours is the first work that demonstrates the feasibility of leveraging smartphones for detecting explosion events.**

I. Introduction

In this paper, we attempt to demonstrate the feasibility of leveraging smartphone accelerometer to detect the triggering of explosion events. While there are prior studies demonstrating the feasibility of leveraging smartphones to detect earthquake events [4–6], the issue of detecting explosion events has not been attempted before. In May 2014, we participated in multiple blasting experiments at the Explosives Research Lab (ERL) at Missouri S&T, where explosives are blasted in a controlled environment for training students [2]. We statically emplaced a Samsung Galaxy S4 smartphone at a carefully chosen location right next to a state-of-the-art seismometer, and extracted the accelerometer readings from smartphone and ground truth Geophone data from seismometer, before, during and after blasts. Subsequent processing and comparison of the data, led us to make several insightful contributions.

We demonstrate that the temporal and frequency responses of the explosion event readings in the smartphone and the seismometer are highly correlated over multiple blasts. Specifically, an average correlation value of 0.83 was observed for the temporal responses. We also observed that the frequency responses are correlated with the frequency of vibrations corresponding to the peak amplitudes sensed by both devices being close to one other. Subsequently, we demonstrate the feasibility of designing an algorithm and implementing it on the smartphone (as an app) to detect the triggering of an explosion event. Seismic monitoring devices use a standard average based trigger algorithms [7, 9] for detection of earthquakes. The rationale of our algorithm is to identify appropriate thresholds for the ratio of sudden spikes in vibration during an explosion to the long term dormant vibration readings in the absence of an explosion. Finally, we identify challenges of explosion detection using smartphones in the context of sampling frequencies, device stability, data processing, energy and storage.

II. Problems Addressed in the Paper and Challenges

A. Problems Addressed

Broadly speaking, the overall goal of the paper is to demonstrate the feasibility of leveraging smartphone accelerometers (when the phones are static) to detect explosion events. Within this context, there are two sub-problems that we address in this paper. The first one is to statistically compare the similarity of accelerometer readings from the smartphone with the ground truth, which in this case comes from a state-of-the-art seismometer when both devices are sensing the vibrations emanating as a result of an explosion. Should the detected events from both devices demonstrate similarity, the next problem is to design an algorithm that can be implemented on a smartphone (as an app) to detect the triggering of an explosion in real-time, while also being efficient in storage and energy consumption.

B. Challenges

Processing of accelerometer readings from smartphones for detecting explosions can be challenging. In this section, we highlight the most critical challenges in this realm.

- **Sampling frequency differences:** Different smartphones come with different sampling frequencies for sensing acceleration. For instance LG Nexus-3 has a sampling rate of 50Hz, while the Samsung Galaxy Note-2 has a sampling rate 80Hz, and the the Samsung Galaxy-S4 has a higher sampling rate of 100Hz. However, typical seismometers has a much higher sampling rate of 500Hz, with superior consistency in the sampling rate. To demonstrate correlations of data generated between these two diverse sources requires down-scaling the data from the seismometer for fair comparison. Note that in real-life scenarios where there are numerous smartphones with numerous and possibly inconsistent sampling rates, this challenge is further exacerbated, the addressing of which is out of the scope of this paper.

978-1-4799-9972-9 2015

DOI 10.1109/MDM.2015.66

- **Stability of the smartphone:** Smartphone accelerometers being highly sensitive, generate noise (even when the phone is static) that needs filtering. To address this challenge, we design a mechanism that in real-time checks the $L2$-norm of accelerometer readings across samples, to ensure that the smartphone is stable prior to processing accelerometer data to detect explosions. For the Samsung Galaxy S4 phone we have used in our experiments, we have conducted a series of experiments to determine the appropriate stability thresholds and time-window.
- **Big-Data processing in real-time:** The smartphone could maintain an archive of all the accelerometer readings to detect explosion events, but this results in enormous amount of data flooding the phone's memory. To address this problem, we employ an effective technique that identifies and filters out samples not related to an explosion event, but retains those samples corresponding to an actual explosion event for detection.

III. Experimental Set-up

In this section, we discuss the experimental set-up to address our problems, namely a) Compare detected event readings from smart-phone and a seismometer during explosions, and b) Design and implement an algorithm (as a smartphone app) to detect the triggering of an explosion events. The Explosives Research Lab (ERL) is a unique facility at Missouri S&T, where students are taught fundamental concepts in Explosives Engineering. There is an experimental mine, which is actually a functional limestone mine that serves as the experimental facility for students training on explosions, mine constructions, operations, safety and rescue. Numerous blasting experiments are conducted regularly by ERL in the experimental underground mine at Missouri S&T.

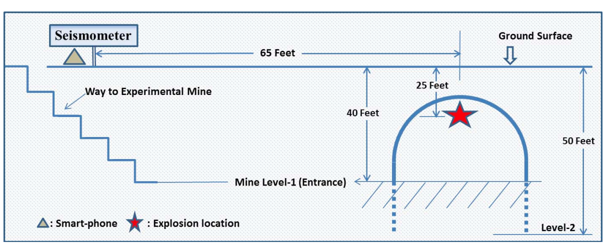


Figure 1. Experimental Setup

A. Blasting Type and Environment

In May 2014, we participated in several blasting experiments conducted by ERL, the set-up of which is described here. Figure 1 shows the experimental set-up. The type of explosive detonated was Dynamite, the details of which are shown in Table I. The explosive material was kept in holes drilled into the mine's ceiling to trigger the blasts. The location of the source of the blasts was 25ft underground, and they consisted of five shots each lasting for 250ms with a delay of 1 second duration between each blast. The seismometer, manufactured by GeoSonics/Vibra-Tech Inc. was installed at ground level, and a smart-phone was placed very close to the seismometer to compare its acceleration data from sensing associated ground vibrations. The team assembled to collect data went through appropriate training procedures prior to visiting the blasting site. Note that when a blast occurs, the seismometer triggers its data acquisition scheme and transmits the event report to a server, from which the data can be recorded. The readings from the smartphone accelerometer are recorded locally on the phone.

Table I
Details of the Explosion at Experimental Mine

Type of explosive	Dynamite (Unimax-TT)
Material blasted	Dolomite Lime
No of holes	15
Explosive sticks	15 Sticks (2 Charge)
Charge weight	15 pounds
Detonating cord	50 grains/foot

B. Smart-phone Selection

Modern smartphones come with a number of sensor arrays including accelerometers, gyroscopes, temperature, acoustic, pressure, humidity sensors and more. For the purposes of this paper, the chief criteria in choosing a smartphone was clearly the performance of the accelerometer, which basically depends on the sampling rate of the sensor, consistency and sensitivity. After reading a number of research blogs, and performing limited experiments with a number of smartphone models, we chose the Samsung Galaxy S4 phone. The accelerometer in the Galaxy S4 phone is manufactured by STMicroelectronics [1]. It has a relatively high sampling rate of 100Hz, is highly consistent and also has very good sensitivity [1].

Table II
Details of the smartphone used during the explosion blasts to collect the accelerometer readings

Smartphone model	Samsung Galaxy S4
Model No.	SAMSUNG-SGH-I337
Operating system	Android-4.4(KitKat)
Accelerometer Model	LIS344ALH
Sampling rate	100 Hz
Dimensions	4x4x1.5mm

In our experiments, the Galaxy S4 phone was statically emplaced very close to the seismometer. An application was installed on the phone to continuously sample the accelerometer sensor. As soon as the application receives the readings from sensors, it tags the values with the time-stamp information and subsequently writes each time-stamp tagged sample as a record to a Comma Separated Value (.csv) file. The output file is stored in the SD card of the smartphone in the form of raw

[1] Needless to say, with appropriate calibration and post data processing, the results of this paper can be applied to other smartphones as well, and is part of our future work.

data, later used for analysis which is further discussed in the Section IV. Critical specs of the Galaxy S4 phone are shown in Table II.

IV. Comparison Results from Statistical Analysis

In this section, we present results from analyzing the ground vibration from the blasts as measured by the accelerometer in the Galaxy S4 smart-phone and geophone in seismometer. The Geophone basically outputs the velocity of ground movement. We converted these velocity readings to acceleration by finding the rate of change of velocity ($a = (v_2 - v_1)/t$) to bring both the events on to same scale. We wish to reiterate that the detonation consisted of five blasts, each lasting for 250ms with a delay of 1 second duration between each blast as discussed in Section-III.

A. Comparison of Temporal Responses

Figures 2(a) and 2(b) demonstrate the accelerometer response for the five blasts, where the Y-axis denotes the absolute resultant value of the linear accelerations in x, y, z directions and X-axis denotes the time. As can be seen visually, the accelerometer in the smartphone correctly detects spikes in ground vibration due to blasts almost exactly when the seismometer detects it. It also can be seen that the absolute values of the smart-phone accelerometer and the one in the seismometer are different, with the seismometer recording higher amplitudes of ground vibration. While this can be explained as a result of calibration issues, our investigations also revealed some insights into the calibration. The output from the smartphone accelerometer is raw sensor data initially and it is ratiometric with respect to the phone's prefixed input voltage. As such, they need to be calibrated to obtain the accurate acceleration reading. However, the results of our paper are directly applicable even if the phone is not calibrated. The calibration effort is part of our on-going work.

To further quantify the fidelity of temporal responses between the accelerometer readings from both instruments, we attempted to correlate them using statistical measures. Unfortunately, doing this is not so straightforward. Most of the correlation algorithms are applied on same number of samples of two given signals. But, we have different sampling rates from smartphone (100Hz) and the seismometer(500Hz). So we had two options to bring seismometer and smartphone on to same scale. One approach is to up-sample the smartphone data and other being down-sampling the seismometer data. If we up-sample the smartphone data by introducing fabricated sample points into the existing signal, then the comparison would be between genuine seismometer data and modified smartphone data which may not give a reliable analysis of correlation. Considering this, we have chosen the latter approach of down-sampling the seismometer data to make the comparison more viable. Recall that duration of an explosion blast is 250ms, which needs to be captured. So even though data from seismometer is down-sampled from 500Hz to 100Hz, we get a sample every 10ms which ensures the capture of explosion trigger, which is of interest to this paper. The down-sampled temporal representation of events from smartphone and seismometer are seen in Figure 2(c).

Correlation analysis was done on the down-sampled data of the seismometer and the smartphone corresponding to the durations of recording of individual blasts. We have used Pearson product-moment correlation method shown in Equation (1), to determine the correlation coefficient,

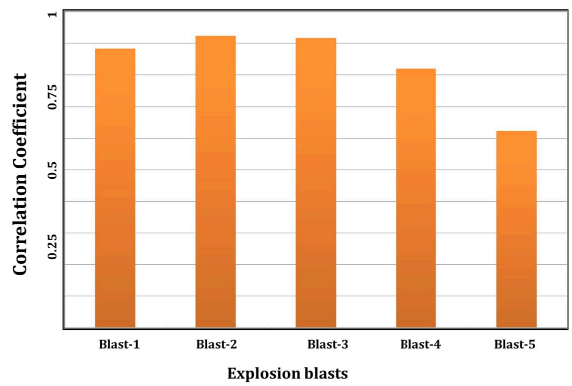


Figure 3. Correlation of individual blasts

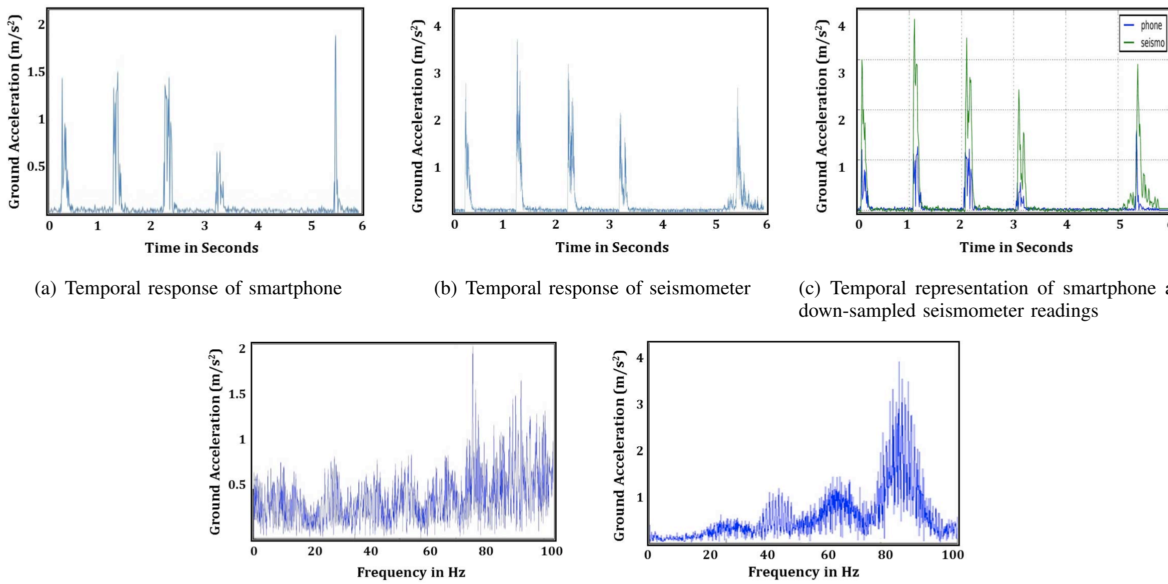
$$r = \frac{\sum(x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum(x_t - \overline{x})^2 \sum(y_t - \overline{y})^2}}, where \qquad (1)$$

x_t: smartphone sample at time t,
y_t: seismometer sample at time t,
$\overline{x}$: sample mean of the smartphone data,
$\overline{y}$: sample mean of the seismometer data.

From the analysis of correlation for the duration of the explosion events, the correlation coefficient was observed in the range 0.62 to 0.92 as shown in Figure 3 with the average correlation being 0.83. Furthermore, to study correlations when the blasts are not taking place (i.e., when the smartphone and seismometer are sensing background noisy vibrations), we also compared accelerometer readings from both devices. We found that the correlations between the two devices when blasts are not taking place is quite poor due to advanced noise filtering algorithms in the seismometer unlike that in the smartphones. This itself does not change any results in our paper, but is nevertheless an interesting insight. Due to space limitations, we do not address this issue further in this paper.

B. Comparison of Frequency Responses

To further derive insights on fidelity of data between the smartphone and seismometer, we also compared their frequency responses. Frequency-domain analysis also helps determine the spectral capability of the smart-phones (and the seismometer as well) to determine the significant frequency component of the explosions. Time-domain data is converted to frequency-domain using Fast Fourier transform (FFT) algorithm to analyze the data from frequency perspective. It can be seen that vibrations corresponding to the peak amplitudes



(a) Temporal response of smartphone
(b) Temporal response of seismometer
(c) Temporal representation of smartphone and down-sampled seismometer readings
(d) Frequency response of smartphone
(e) Frequency response of seismometer

Figure 2. Temporal and Frequency domain responses of events detected from smartphone and seismometer

in seismometer and smart-phone events occurred at 83Hz and 74.6Hz respectively as seen in Figures 2(d) and 2(e). However, the quality of the data from the smartphone accelerometer has more noise when they sense background vibrations, which is relatively easy to fix with filtering algorithms, and is out of scope of the paper.

V. Design, Deployment and Validation of a Smart-phone Algorithm to Detect Explosions

Motivated by the positive results discussed in the earlier section, we now present our contributions in the design, deployment and validation of a smartphone algorithm that can execute in real-time to detect the triggering of explosion events. We first present the motivation of the proposed algorithm. We then present some design challenges, followed by the actual algorithm, its operation, and validation.

A. *Overview of the Algorithm*

An explosion is an event that results in extremely high increases in volume and release of energy. Such an event typically results in the ground vibrations generating high pressure waves, due to the deposition of a large amount of energy in a very small localized volume. The proposed algorithm we design to detect explosion events relies on this principle. Specifically, we take into account, and integrate two critical characteristics of explosion events. The first one is the long term dormancy in vibration readings sensed before and after an explosion event, and the second is the sudden increase in vibrations during the brief explosion event. There have been some prior works focused on this principle for detecting earthquakes and volcanic events using seismometers [3, 8, 10]. However, we are not aware of any work that tailors this methodology to be used in smartphones for explosion detection. A significant challenge is to ensure that any algorithm designed and implemented to be executed on smartphones for detecting explosions must do so with minimal latency (i.e., in near real-time). Ideally, it should execute within the time window between two accelerometer samples sensed by the sensor. This ensures that any incoming data (all of which are critical) is not ignored. Critical other challenges to overcome are ensuring the stability of the phone to prevent false negatives and positives, determining right thresholds for triggering and de-triggering of the event, and ensuring efficient processing of incoming data streams in the realm of storage and energy efficiency, all of which are elaborated when we discuss our proposed algorithm next.

B. *Workflow and Design of the Algorithm*

Algorithm 1 shows the pseudo code of our detection strategy. There are three phases of operation: Sensing, Triggering and Event Recording. In the Sensing phase, the smart-phone will execute tasks related to sensing of ground vibrations, with the core challenge being ensuring the stability of the phone from ambient noise, prior to data processing. In the Triggering phase, the algorithm will execute tasks related to detecting the triggering of an explosion, based on set parameters explained below. Finally, in the Event Recording phase, the phone will perform tasks related to recording of the explosion event, and getting ready to sense again.

A. Sensing Phase: When device starts running the detection algorithm, this is the initial phase it enters into. In this phase, the incoming samples of the accelerometer are processed to ensure the stability. Since the goal of this paper is to demonstrate the feasibility of leveraging static smartphones

Algorithm 1 Pseudocode to Detect Triggering of Explosions

Input: STA Window(S_{TA}), LTA Window (L_{TA}), a_x, a_y, a_z
Output: Triggering status of the accelerometer samples processed

Sensing Phase:
while $True$ **do**
    Compute s_d **(as in Equation** (2))
    if $s_d < 0.01$ **then**
        $goto\ TriggeringPhase$
    end if
end while

Triggering Phase:
x = $(x - \overline{x})$, y = $(y - \overline{y})$, z = $(z - \overline{z})$
Compute S_a, L_a **for time windows** S_{TA}, L_{TA}
if $\frac{S_a}{L_a} > TR_{th}$ **then**
    $goto\ EventRecordingPhase$
else
    Return $to\ TriggeringPhase$
end if

Event Recording Phase:
while $\frac{S_a}{L_a} < DTR_{th}$ **do**
    Add x, y **and** z $into\ E$
end while
Store E **on Phone**
Return $to\ SensingPhase$

for detecting explosions, it is reasonable to conclude that the accelerometer readings sensed by the sensor in the phone will be stable (due to lack of motion). However, our experiments revealed something different. Considering that the phones are small in form factor and weight, they are quite easy to be displaced even with minimal amount of external stimuli. Such movement, even though minor can cause changes in accelerometer readings which can corrupt values sensed under explosion events. As such, we have designed a simplistic model that ensures the stability of the phone prior to executing our algorithm to make sure that any values sensed as a result of ground vibrations during explosions are minimally corrupted by noise.

As soon as the smartphone starts monitoring, for each sample the algorithm determines the $L2 - Norm$ of the difference of acceleration vector from its previous known to its current sample value. We call it as s_d and is shown in Equation (2).

$$s_d = (a_t^x - a_{t-1}^x)^2 + (a_t^y - a_{t-1}^y)^2 + (a_t^z - a_{t-1}^z)^2, where \quad (2)$$

t: time of arrival of current sample,
$t-1$: time of arrival of previous sample,
$a_t = (a_t^x, a_t^y, a_t^z)$: acceleration vector at time t.

Ideally $s_d = 0$, if the device is absolutely stable, which happens when $a_t^x = a_{t-1}^x$, $a_t^y = a_{t-1}^y$ and $a_t^z = a_{t-1}^z$. Unfortunately, the case of perfect stability cannot be achieved in practice, and reaching acceptable stable levels also takes time. If s_d is found to be less than the threshold value for a pre-defined time interval, then we declare the smartphone to be stable. Further, the current average values of the tri-axial acceleration readings in their respective moving windows, denoted as $\overline{x}$, $\overline{y}$, $\overline{z}$ are sent for processing in the next phase of algorithm operation. Figure 4 illustrates an instance of the phone reaching from an unstable to a stable state after 6 seconds with a constant shaking further. Numerous experiments for the Samsung Galaxy S4 phone were conducted and we set the threshold for s_d below which we considered the smartphone to be stable as 0.01. Note that to demonstrate the feasibility and for the sake of simplicity, our algorithm does not correct any stabilization errors, but rather ignores any ground vibrations sensed in an unstable state.

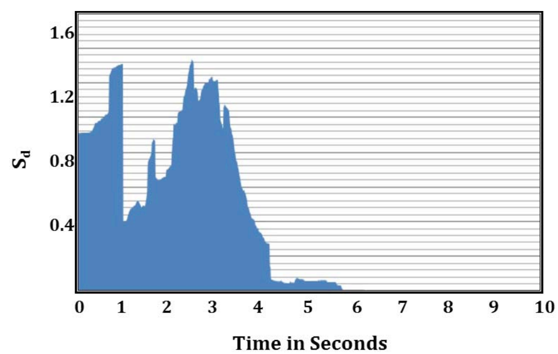


Figure 4. Time taken for the smartphone to stabilize

B. Triggering Phase: The algorithm will move into this phase after the device is confirmed to be stable. This is phase in which the smartphone detects the actual triggering of an explosion event if it meets the criteria for the explosion as discussed below.

The algorithm processes each incoming signal in multiple steps. When an accelerometer sample is received, each component of the measurement vector x, y, and z are processed separately. In the first step, the samples x, y, z are corrected as $(x - \overline{x})$, $(y - \overline{y})$, $(z - \overline{z})$ respectively, using the correction factors evaluated in the sensing phase to measure the absolute change in acceleration. Then, the next step is processing these acceleration values to detect the triggering of an explosion event.

Our idea again here is to leverage the long term dormancy in vibration readings sensed before and after an explosion event, and the second is the sudden increase in vibration readings during an explosion event. To implement this strategy, the acceleration readings from the phone are stored in two separate and sliding time windows in our design. The first window is called the Short Term Averaging (S_{TA}) window, and the second one is called the Long Term Averaging (L_{TA}) window. Essentially, these parameters quantify the durations in which the the rapid increase ground vibrations are sensed during an explosion event, and for how long the increased spikes last until they go down. Let us denote S_a and L_a as the average of the acceleration readings computed in the S_{TA} and L_{TA} windows, and R_a as the ratio of their values, i.e., $R_a = \frac{S_a}{L_a}$.

The algorithm computes the ratio upon every sample received, while sliding the windows. If R_a exceeds a pre-set threshold (TR_{th}), then the algorithm triggers an explosion event. Also the end of event is declared when R_a falls below a de-trigger threshold (DTR_{th}). We point out that in practice, these parameters are sensitive to a number of characteristics of an explosion event, including the chemical composition, duration and intensity of an explosion. As such, setting these parameters in a generalized sense is very challenging and requires significantly more experiments and data sets. To demonstrate the feasibility, and within the context of this paper, we tailor these parameters specifically to the context of explosion event whose properties are listed in Table I.

Our experiments revealed that an S_{TA} window of 0.1 seconds, an L_{TA} window of 10 seconds, TR_{th} = 1.75 and DTR_{th} = 1.5 provides the best discriminator for triggering and de-triggering of an explosion event and can be seen in Figure 5.

C. Event Recording Phase: The final phase of the operation of the algorithm is the Event Recording Phase. This phase is executed when the phone detects the triggering of an explosion event, and its subsequent de-triggering. When that happens, the accelerometer samples are recorded on to the phone along with the pre-event memory as an Event (E). Note that all readings are recorded until the de-triggering condition is met. The phone then returns to the sensing phase and the cycle repeats.

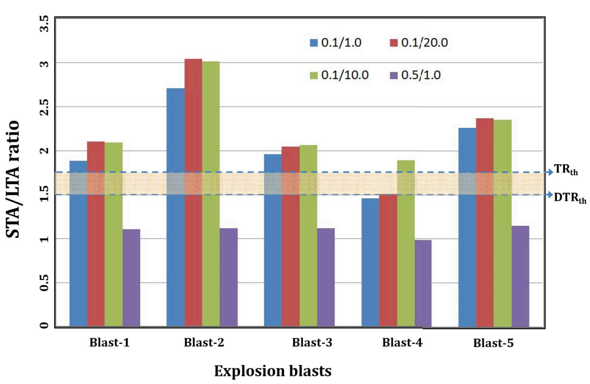


Figure 5. STA/LTA ratios for various combinations of STA and LTA values chosen

C. Deployment and Validation of the Algorithm on a Smartphone

To validate feasibility, we created an Android application (app) that executes our algorithm in real time and it was deployed during another round of blasting experiments (same parameters as Table I). The phone was placed at a location similar to the one placed earlier for the training phase, and it successfully triggered to the explosion event. The application consumes 331KB of memory, 0.173 joules of energy per second and a typical sample processing time of 4ms. This is quite minimal.

VI. CONCLUSIONS

In this paper, we demonstrate the feasibility of leveraging smartphones to detect the triggering of explosion events. Specifically, we demonstrated the similarity of accelerometer readings from smart-phones and ground-truth seismometer readings in the temporal and frequency domains. We also designed an app that can detect explosion events in real-time, and also identified critical performance metrics like energy, storage and execution times.

Our next step is to design algorithms to localize explosive events based on multi-modal data sensed from multiple phones. We also believe that with feasibility of smartphones performing the functionality of seismometers, there are clear applications to citizen aware disaster management applications during detection, rescue and recovery. However, in this realm there are other open issues including handling diversity of smartphones and associated sensor hardware, multi-sensor fusion, communication and networking among devices, energy aware data processing, design of ranging algorithms, and real-time execution due to mission critical nature of the application.

ACKNOWLEDGEMENTS

We thank Professor Paul Worsey, the staff of Explosives Research Lab at Missouri S&T, and GeoSonics/Vibra-Tech Inc. for assisting us with data collection.

REFERENCES

[1] Samsung Galaxy S-4 sensor specs. http://electronics360.globalspec.com/article/94/teardown-what-s-inside-samsung-s-galaxy-s4/, 2012. [Online; last accessed, March 31 2015].

[2] Explosives Research Lab. http://mining.mst.edu/research/depexpmine/, 2014. [Online; last accessed, March 31 2015].

[3] L. D'Auria, F. Giudicepietro, M. Martini, and R. Peluso. Seismological insight into the kinematics of the 5 april 2003 vulcanian explosion at stromboli volcano (southern italy). *Geophysical research letters*, 33(8), 2006.

[4] M. Faulkner, R. Clayton, T. Heaton, K. M. Chandy, M. Kohler, J. Bunn, R. Guy, A. Liu, M. Olson, M. Cheng, and A. Krause. Community sense and response systems: Your phone as quake detector. *Commun. ACM*, 57(7):66–75, July 2014.

[5] Q. Kong and R. Allen. Using smartphones to detect earthquakes. In *AGU Fall Meeting Abstracts*, volume 1, page 04, 2012.

[6] J. Reilly, S. Dashti, M. Ervasti, J. D. Bray, S. D. Glaser, and A. M. Bayen. Mobile phones as seismologic sensors: Automating data extraction for the ishake system. *IEEE T. Automation Science and Engineering*, 10(2):242–251, 2013.

[7] B. Sharma, A. Kumar, and V. Murthy. Evaluation of seismic events detection algorithms. *Journal of the Geological Society of India*, 75(3):533–538, 2010.

[8] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 305–318. ACM, 2009.

[9] A. Trnkoczy. Understanding and parameter setting of STA/LTA trigger algorithm. Technical report, Department of Electrical Engineering, University of Ljubljana, Slovenia, 1999.

[10] M. Withers, R. Aster, C. Young, J. Beiriger, M. Harris, S. Moore, and J. Trujillo. A comparison of select trigger algorithms for automated global seismic phase and event detection. *Bulletin of the Seismological Society of America*, 88(1):95–106, 1998.