Pervasive and Mobile Computing 40 (2017) 185–204

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Leveraging multi-modal smartphone sensors for ranging and estimating the intensity of explosion events

Srinivas Chakravarthi Thandu [a], Pratool Bharti [b], Sriram Chellappan [b], Zhaozheng Yin [a,*]

[a] Department of Computer Science, Missouri University of Science & Technology, Rolla, MO 65401, USA
[b] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA

ARTICLE INFO

Article history:
Available online 20 June 2017

Keywords:
Explosions
Ranging
Machine learning
Smartphones
Participatory sensing

ABSTRACT

Our society, unfortunately, is increasingly becoming exposed to explosion events that could have serious consequences. While explosion events like intentionally triggered bombs cause obvious harm to life and property, other explosions intended for benign purposes in quarries and construction zones may also cause unintended harm as a result of emanating seismic vibrations. As of today, detecting explosions, ranging them, and estimating their intensities are all accomplished only by seismometers that sense the associated ground vibrations and pressure changes as a result of their triggering. Unfortunately, seismometers are bulky, expensive and unsuitable for the ubiquitous use. In this paper, our broad motivation is to demonstrate the feasibility of leveraging the pervasive sensing and processing capabilities of modern smartphones to analyze explosion events. Within this context, we specifically address the problem of ranging and estimating the intensity of an explosion by leveraging the accelerometer and pressure sensors in the smartphone. To do so, we emplaced a number of smartphones in the vicinity of real explosion blasts conducted at a university mining laboratory, where the material blasted was Dynamite with Ammonium Nitrate Fuel Oil (ANFO). We then collected the corresponding accelerometer and pressure readings sensed by the phone. We extracted a number of novel features, and designed a machine learning based algorithmic framework for ranging and estimating the intensity of the explosion event. After an extensive validation, we find that the average-case error in ranging (i.e., estimating the distance to the source of the explosion event) and estimating the intensity of explosive material (in terms of its charge weight) are 8.24% and 7.37%, respectively. We also present perspectives on encoding our algorithm as a smartphone app, identify several critical challenges that will be encountered in real-time data processing of smartphone accelerometers and pressure sensors in the context of pervasive sensing of explosions, and also identify other practical issues like the diversity of smartphones. To the best of our knowledge, our work is pioneering in demonstrating the feasibility of using smartphones to analyze explosion events. We believe there are significant societal benefits emanating from our work.

* Corresponding author.
E-mail addresses: stgk4@mst.edu (S.C. Thandu), pratool@mail.usf.edu (P. Bharti), sriramc@usf.edu (S. Chellappan), yinz@mst.edu (Z. Yin).

http://dx.doi.org/10.1016/j.pmcj.2017.06.012

1. Introduction

As a society, we are increasingly being exposed to explosions that are both man-made and naturally occurring. Unfortunately, these can be dangerous to our life and property. In the recent past, a critical component of such dangers is the triggering of explosives (like bombs) intended to harm the society [1–4]. As we are well aware, the shock and blast waves arising as a result of an explosion being triggered, along with the associated debris can cause significant societal scale damages. On the other hand, triggering explosions without malicious intent like those in quarries and construction sites also create vibrations that could cause damage to buildings in their vicinity as demonstrated in a number of studies [5,6]. Needless to say, devices that can monitor such vibrations and their sources, along with superior technologies to analyze them are important today.

Traditionally, the standard device that has been used to sense vibrations is a seismometer. While the chief utility of seismometers so far has been primarily for studying earthquakes, they have also been used to analyze the intensity of volcanic activities, vibrations of civil engineering structures, and explosion events. A conventional seismometer consists of a spring and a weight and works on the principle of inertia, wherein the motion of the weight is sensitive to the ground vibration. Modern seismometers, on the other hand, sense ground vibrations using a combination of electronic vibration sensors and amplifiers. They are now equipped with sensors to sense the ambient pressure also. This is useful to characterize explosions since the ambient pressure in the vicinity of an explosion rises above the atmospheric pressure due to emanating over pressurization blast waves. Unfortunately though, seismometers are bulky, expensive and not viable for ubiquitous usage.

With advances in MEMS (Micro-Electro-Mechanical Systems) technologies, modern smartphones come with significant pervasive sensing capabilities [7]. Most modern smartphones today like the Samsung GALAXY series phone, the iPhone, Google Nexus phone and more, come packaged with various sensors for measuring environmental and human activity parameters. For instance, a Samsung GALAXY S4 smartphone today has built-in sensors that can measure the acceleration, ambient temperature, pressure, humidity, light intensity, magnetic intensity, sound intensity, and much more with high sampling rates. The *LIS*344*ALH* accelerometer sensor in the Samsung GALAXY S4 phone [8] can consistently sample at the rate of 100 to 110 samples per second and the sampling rate is programmable. This phone is also equipped with a pressure sensor (*LPS*331*AP*) to measure the atmospheric pressure at its location, with a sampling rate of up to 10 samples per second. Furthermore, numerous studies are being conducted today to optimize the performance of smartphone sensors today from the perspective of accuracy, energy efficiency and processing speed [9–11].

There is a clear and tangible reason for the continued innovation in sensing capabilities of smartphones today, and that lies in numerous innovative and societally useful applications leveraging smartphone sensors. While we discuss important ones in Related Work section in more detail, we find that smartphone accelerometers have been used for monitoring vibrations emanating from earthquakes [12], and for activity tracking of human users [13]. The pressure sensor in smartphones has applications in indoor localization [14–16]. Smartphone sensors like ambient temperature, humidity, and light intensity have innovative applications in context-awareness, and pervasive computing applications [17–19].

1.1. Problems addressed in this paper

With increasing instances of explosions (triggered with or without malicious intents) affecting daily life, there is a clear and tangible need to move beyond seismometers to sense explosions, and therein lies the significance of our contributions. In this paper, our goal is to demonstrate the feasibility of leveraging multi-modal sensors embedded in a state-of-the-art smartphone for analyzing explosion events. Specifically, we demonstrate how data sensed by the accelerometer (that senses ground vibrations associated with an explosion) and a pressure sensor (that senses changes in atmospheric pressure as a result of emanating blast waves) can be leveraged for

1. **Ranging:** Estimating the distance from the smartphone to the source of the explosion.
2. **Estimating the intensity:** Estimating the intensity of the explosion event in terms of charge-weight of the explosive material.

1.2. Challenges in analyzing explosion events

We now highlight some of the critical challenges in addressing our problems stated above.

a. Availability of real datasets: Explosions are difficult to be studied for the fact that, they are often inaccessible and difficult to be replicated in the physical environment. Needless to say, it is challenging to gain the access to environments where blasts take place. Furthermore, it is even more challenging to have the environment under control to be able to place smartphones in and around the vicinity of an explosion, and simultaneously obtain high-quality ground truth data for subsequent analysis of explosion events.

b. Lack of practical models to range explosion sources: As mentioned earlier, the challenge in obtaining ground truth data during the explosive blasting, has meant that there is very little body of work in attempting to range explosion sources. While there is some existing work in this realm in [20,21], they all address this problem using measurements from seismometer sensors only. However, there are clear differences between sensors used in seismometers and smartphones in terms of sampling rates, energy consumption, sensing ranges, sensitivities, and deployment environments, which necessitate new techniques for addressing the problems defined in this paper.

1.3. Our technical contributions

a. Generating real explosion datasets: The initial roadblock for our problem is the access to locations of real explosion events, while simultaneously having such explosive events under a safe experimental environment to obtain real-time data. The Explosives Research Lab (ERL) at Missouri University of Science and Technology [22] is one such environment where explosives are blasted in a controlled environment for training students majoring in Explosives Engineering program. Fortunately, we had the access to the Explosives Research Lab (ERL). To the best of our knowledge, such a facility is unique in college campus environments. This provided us with a controlled environment, wherein we could place smartphones to derive an extremely rich source of datasets to devise techniques to address our problems.

In May 2014, we participated in multiple blasting experiments at ERL, and stationarily emplaced multiple Samsung GALAXY S4 smartphones in the vicinity of the explosions. Specifically, the blasting experiments for the purposes of this paper were conducted in an underground experimental mine as part of experiments conducted by the ERL. The explosive material blasted was *Dynamite with Ammonium Nitrate Fuel Oil* with different intensity for each experiment. We then collected the data sensed by the accelerometer and pressure sensor from the smartphones during the explosion events for the subsequent post processing. The sampling rate is set as 100 samples per second for the accelerometer sensor and for the pressure sensor, it is set as 10 samples per second, which is achievable by most of the available smartphones today.

b. Design of a model for ranging and intensity estimation: In this paper, we adopt a machine learning approach to designing a model for ranging and estimating the intensity using data collected from blasting experiments. We extracted a number of features that are distinct for the accelerometer and pressure readings, and also created novel features that integrate them together.

First, we pre-processed the raw-data to separate individual blast event datasets and labeled them. We then extracted a total of 45 features from the acceleration and pressure sensor data collected from the phones. Leveraging these features, we then designed a nonlinear polynomial regression model to estimate the distance to the source of an explosion event and its intensity. In our current model, the average-case error was 8.24% and 7.37% in estimating the range and intensity of explosion events. We also validated the overall consistency of our model using a number of critical statistical metrics and results are quite promising, as we demonstrate in Section 6.

c. Designing a smartphone app: After demonstrating the feasibility of leveraging smartphone sensors to estimate the range and intensity of explosions by the proposed model, the issue of designing smartphone apps become a practical one. However, this aspect is quite challenging from numerous fronts including energy, processing and storage efficiency. In this paper, we report our findings on designing smartphone apps to sense and analyze explosion events, which we believe will provide critical guidelines for practical deployments in the future. Furthermore, we provide critical perspectives on practical issues while encoding our algorithm as a storage and energy efficient smartphone app to sense and analyze explosion events.

The rest of the paper is organized as following. Section 2 discusses the related work. Section 3 details the critical background on explosions and their underlying physics that we leverage while sensing their impacts. Section 4 presents a detailed description of our experimental set-up. The detailed analysis of smartphone data and the model built to range and estimate the intensity of explosion, is presented in Section 5. Section 6 presents experimental validation and performance evaluation of the model along with detailed analysis on designing a smartphone app using the algorithmic framework proposed in the model. Further, we provide some perspectives on practical applications, limitations, and future work along with other critical issues in Section 7, and conclude the paper in Section 8.

2. Related work

In this section, we elaborate some important work related to our research in this paper from the perspective of leveraging smartphone sensors for innovative societal applications. The field is certainly broad with some research works attempting to leverage a smartphone's ability to sense the ambient temperature, humidity, magnetic field and light intensity, for innovative applications like sleep monitoring [23], human activity recognition [24–26], context recognition [27–29], determining wall layout of a building [30] and much more. However, to retain the scope in this paper, we only highlight the important related work on leveraging smartphone accelerometers and pressure sensors for enabling new applications. We then review our prior work on leveraging smartphones for explosion sensing that serves as the initial foundation for our contributions in this paper.

2.1. Leveraging smartphone accelerometers for monitoring ground vibrations from earthquakes

The Community Sense and Response (CSR) system proposed by Faulkner, et al. [12] leverages accelerometer sensors in smartphones and consumer electronics for monitoring earthquakes. The authors have built a seismic network for the smartphone users to contribute a significant amount of event data to a centralized server. To demonstrate the effectiveness of the system, the authors have used real datasets collected from 3000 low-cost accelerometers distributed freely in Los Angeles area, where minor earthquakes are common phenomena. The CSR system demonstrated that around 50 phones should be enough to detect a nearby magnitude 5 or larger earthquake event with a high success rate. The algorithmic challenges of designing, building and evaluating a scalable network for real-time awareness of earthquakes were also presented.

The iShake project designed by Jack, et al. [31] at the University of California, Berkeley resulted in the design of a mobile client back-end server architecture that uses sensor-equipped mobile devices to measure vibrations from earthquakes. iShake provides the general public with a service to contribute a significant amount of data towards earthquake research by automating the data collection and reporting mechanisms via the iShake mobile application. To demonstrate the feasibility, a test procedure called "Shake-table" was devised to verify the quality of the accelerometer recordings in the context of earthquake sensing. The authors have simulated 150 historical ground motion replays for analysis. Two types of mobile devices were used: four 3GS iPhones and three iPod Touches (third generation). The devices were attached to a shaking table that orients the devices at different directions in order to test for biases among axes of the accelerometers. Along with the smart devices, high-quality accelerometers were also attached which served as the ground truth data. Results demonstrate the practical feasibility of smartphone accelerometers to detect earthquake events in real-time.

In [32], accelerometer readings from numerous smartphones were collected over a continuous period of time in Berkeley (CA) area, during which multiple earthquakes affecting the area were monitored. Subsequently, human activities were also been recorded using these smartphones. Using a classification algorithm based on neural networks, accelerometer readings associated with earthquakes were distinguished from that associated with activities of human users with a very high accuracy.

2.2. Leveraging pressure sensors in smartphones for indoor localization

In the recent past, pressure sensors in smartphones have been used for indoor localization, which is an important problem. It is well known that the atmospheric pressure decreases when the altitude increases. With this fact, models have been created to relate altitude/height to pressure. In [14], using the atmospheric pressure determined by the smartphone pressure sensors, the floor level of building is determined for indoor positioning. In [33], an integrated framework was proposed to provide ubiquitous and accurate elevation measurements using pressure sensors in smartphones. Experiments were conducted both indoors and outdoors with different geographical characteristics. It was shown that the system could achieve an error less than 5 m in 90% of the cases when estimating the elevation.

In [34], using pressure sensor values from smartphones, a model was built which can detect floor changes and the mode (elevator, escalator, or stairs) used to change floors. In [16], various characteristics of the pressure sensor data due to door opening/closing events are studied to build a model for recognizing these events. The model could achieve an accuracy close to 99% based on data collected from smartphones carried by people, while entering/exiting in 3 different building environments. The authors claim that it is feasible to build a low-cost ubiquitous door event detection system that enables many in-home monitoring applications without any infrastructure integration which can also work as an augmentation to currently expensive home security systems.

2.3. Our prior work on leveraging smartphones for explosion sensing

In this paper, we are attempting to address the problem of estimating the range and intensity of an explosion event from multi-modal smartphone sensors. We point out that we have done some prior work on analyzing explosion events, but our prior work only focuses on accelerometer sensors, while the work in this paper leverages both accelerometer and pressure sensors in smartphones.

In [35], the problem we addressed was the feasibility of leveraging smartphone accelerometers to detect the triggering of an explosion event. To do so, we emplaced a Samsung GALAXY S4 smartphone near a seismometer to detect vibrations emanating from real explosion events where the blasting material was Dynamite with Ammonium Nitrate Fuel Oil. From data collected, we identified that the temporal and frequency response of the data from the smartphone and seismometer demonstrated a high degree of statistical similarity. With this result, we designed a simple and energy efficient algorithm that computed the ratio of the sudden spikes in acceleration readings (during the explosion) and the long term dormant values in acceleration readings (in the absence of an explosion). We then identified right thresholds in this ratio, which demonstrated a very high degree of accuracy in terms of leveraging smartphone accelerometers to detect the triggering of explosion events in real-time.

In a follow-up work in [36], we attempted to determine the range and intensity of an explosion event using smartphone accelerometers only. Specifically, using real explosion datasets where the blasting material was Dynamite with Ammonium Nitrate Fuel Oil, and four stationary Samsung GALAXY S4 phones, we designed a nonlinear regression model to estimate the distance to the source of explosion event, and the intensity of explosion based on extracting a number of statistical and frequency domain features from the accelerometer readings in the phone. Our model yielded an average-case error of 12.86% for ranging, and 11.26% for estimating the intensity of the explosion event.

Our work in this paper significantly extends our prior work. Firstly, in this paper, we use multi-modal sensors (i.e., accelerometers and pressure sensors) to estimate the range and intensity of explosion events. Several new features fusing these sensing modalities are proposed in this paper as a consequence. Our new model (as we show later) decreases the estimation error and yields superior consistent results. Furthermore, we design an energy efficient smartphone app for analyzing explosion events in real-time. We provide critical perspectives on making the app energy and storage efficient and also identify critical performance specs that will guide the future practical deployment of our contributions in this paper.

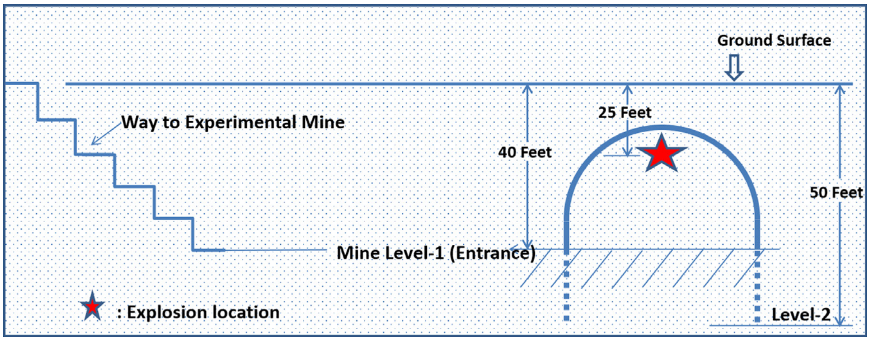


Fig. 1. Experimental setup.

3. Background on explosions

Considering the uniqueness of our work in analyzing explosion events, we provide a very brief background on explosions, and their underlying physics that guides our work in this paper on leveraging smartphone sensors to analyze them.

An explosion is an event that results in extremely high increase in volume and release of energy. Explosions can be caused by cataclysmic nature-created events like volcanic eruptions or man-made ones like bomb blasts, electrical explosions, and fireworks. Man-made explosions of interest to this paper can be classified into two types. High-order Explosives (HEs) and Low-order Explosives (LEs). High-order Explosives are more powerful than Low-order Explosives because detonations from HEs produce a defining supersonic over-pressurization wave that causes more damages. Several sources of HEs include Trinitrotoluene, C-4, Semtex, Nitroglycerin and Ammonium Nitrate Fuel Oil (ANFO). A Low-order Explosive is a material that has an explosion velocity[1] of less than 1000 m/s and hence lacks the over-pressurization wave of HE. Detonations from LEs result in a blast pressure wave that moves slowly, displacing objects in its path. Sources of LEs include pipe bombs, gunpowder, and most pure petroleum-based incendiary bombs such as Molotov cocktails.

The primary outcome of an explosion is a shock wave that rapidly propagates through the medium, creating the vibration impact to the subjects in the vicinity of the explosion. Explosive detonations also create an incident high-pressure blast wave, characterized by a rise from atmospheric pressure to a peak overpressure. The pressure then decays back to the ambient pressure as the intensity of blast wave degrades.

The standard instrument used to detect the triggering of an explosion event is a seismometer, that senses the emanating ground vibrations, and changes in the atmospheric pressure. In this paper, we leverage the accelerometer and pressure sensors in modern smartphones to sense explosion events, and subsequently estimate their ranges and intensities.

4. Experimental set-up

Recall that our problem in this paper is to estimate the range and intensity of an explosion event using data sensed by inbuilt accelerometer and pressure sensors in smartphones. In this section, we first describe the smartphone sensors and then give a detailed description of the blasting explosion environment. We point out that the team assembled to collect data went through numerous training procedures prior to visiting the blasting site, in order to ensure the safety of all participants in the experiments.

4.1. Smartphone sensors

Accelerometer and pressure sensors used in the smartphones are manufactured using MEMS technology that makes them miniaturized and powerful. Smartphone accelerometers are tri-axial sensors sensing the acceleration in lateral, longitudinal and vertical axes corresponding to x, y, and z directions in the vector space in m/s^2. The pressure sensor in smartphones outputs a scalar value which is the atmospheric pressure sensed in hectaPascals (hPa). We have developed an app to collect the raw sensor data. In this app, the sampling rate of accelerometer sensor is fixed to 100 samples per second and that of the pressure sensor to 10 samples per second. This rate is achievable in most modern smartphones today. In each phone, the acceleration and pressure readings were continuously sensed and recorded by the app before, during and after the explosion events. The output file is stored in the SD card of the smartphone device in the form of raw data. After the blasts were completed, the team assembled subsequently collected all the phones for post processing. Subsequently, the values are tagged with the time-stamp information and recorded to a Comma Separated Value (.csv) file.

[1] Explosive velocity, also known as detonation velocity or velocity of detonation (VoD), is the velocity at which the shock wave travels from a detonated explosive.

Table 1
Details of smartphones used during the explosion experiments.

Smartphone brand	Samsung GALAXY-S4
Model no.	Samsung-SGH-I337
Operating system	Android-4.4(KitKat)
Accelerometer	STMicroelectronics LIS344ALH
Pressure sensor	STMicroelectronics LPS331AP
Sampling rate of accelerometer sensor	100–110 Hz
Sampling rate of pressure sensor	10 Hz

Table 2
Details of the explosive material used for experiments at Explosives Research Laboratory (ERL).

Type of explosive	Dynamite with ANFO
Material blasted	Dolomite – Lime
Charge weight	8.05 — 16.05 (in lb)
No. of charges	2, 5, 8
Detonating cord	25 — 50 grains/foot

We have chosen the Samsung GALAXY S4 phones for our experiments.[2] Critical specs of the GALAXY S4 smartphone are shown in Table 1 below.

4.2. Blasting type and explosion environment

Fig. 1 depicts the experimental mine environment where explosion experiments are conducted. The environment wherein we are collecting real experimental data from explosion experiments is actually an operational underground mine. The experimental mine is a functional limestone mine that serves also as an experimental facility for students training on mine constructions, operations, safety, and rescue. In this paper, to retain the scope, we make the following critical assumptions:

- Arrangement, where the explosion happens, is a controlled underground mine environment.
- Smartphones are all stationary and placed on the ground during the experiments and there are no energy/processing constraints affecting the sensing and recording of associated vibrations.
- There are no obstacles (like walls or rocks) between smartphones placed and explosion source.
- The explosive material blasted is known in advance. For this paper, the material is Dynamite with Ammonium Nitrate Fuel Oil (ANFO).

We point out that Dynamite is very commonly used in real explosions [37], and as such, it is a very representative material to consider for our problem in this paper. For another type of explosive materials, if given corresponding training datasets from multi-modal smartphone sensors, the same methodology can be applied to learn corresponding parameters and adapt to new types of explosive event analysis. In each experiment, a different charge weight (in lb) was used for the ANFO material, which determines the intensity of the explosion. The explosive material was kept in holes drilled into the mine's surface to trigger the blasts. The location of the source of the blasts was 25 ft underground. Table 2 presents more specs on the explosive material blasted in our experiments.[3]

A total of four different explosion experiments were conducted for the purposes of this paper. We denote them as $E1$, $E2$, $E3$, $E4$. Also, four smartphones were used in the experiments, denoted by $P1$, $P2$, $P3$, $P4$. Each explosion experiment typically occurs with a specific charge weight and a certain number of charges. The number of charges indicates the actual number of blasts during an explosion experiment. Each blast typically lasts for 250 ms with a delay of 1-second duration between two sequential blasts. In our experiments, the number of charges was 8, 2, 5 and 2 for each explosion experiment, and the corresponding intensities (in lb of the charge weight) were 16.05, 8.05, 11.83 and 8.25 respectively. For each of these experiments, the number of phones emplaced was 2, 4, 4 and 4 respectively to collect data. The phones were emplaced at different distances in each experiment in the vicinity of an explosion event, as shown in Fig. 2. The distances from the smartphones to the explosion source were measured using a laser-distance-measurer with superior accuracy. Critical details of the experiment are summarized in Table 3.

With the above settings, we were able to generate multiple datasets to develop a model for estimating the range and intensity of the explosion events. Specifically, the number of blast event datasets from Experiment-1 was 16 (8 charges and 2 phones), Experiment-2 had 8 datasets (2 charges and 4 phones), Experiment-3 had 20 datasets (5 charges and 4 phones) and Experiment-4 had 8 datasets (2 charges and 4 phones). This resulted in a total of 52 blast event datasets for all the explosion experiments combined.

2 Our techniques proposed in this paper are general and are independent of the type of smartphones.

3 More details on the experimental design are presented in the Appendix.

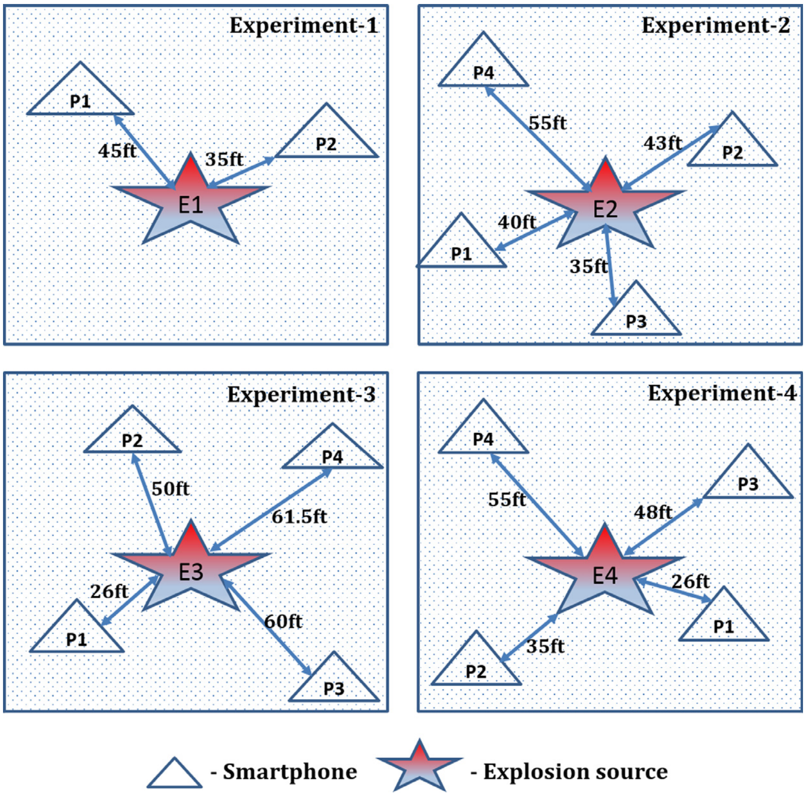


Fig. 2. Layout of phones during various experiments.

5. Model for estimating the range and intensity of explosion events

Broadly speaking, we want to demonstrate the utility of smartphone sensing capabilities to analyze explosion events. Within this broad context, the specific goals of the paper are to determine the *range* (distance from the smartphone to an explosion source) and *intensity* (in terms of charge-weight of explosive material) of an explosion event. In this section, we describe the design of our model and algorithmic framework to estimate the *range* and *intensity* of an explosion event from smartphone acceleration and pressure sensor data. We start with some preliminaries first on smartphone sensitivities, followed by the actual model design.

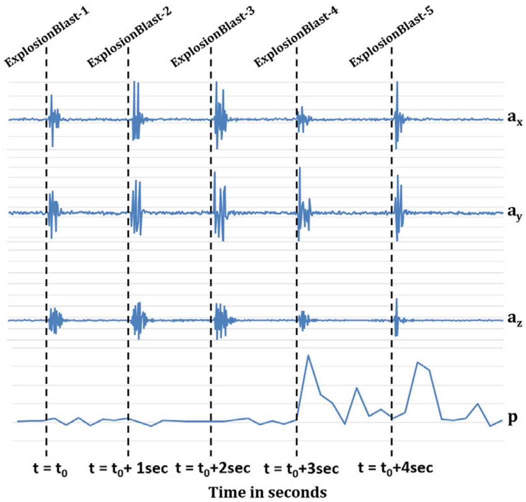
5.1. Preliminaries on the sensitivity of smartphone sensors to explosion events

As we discussed earlier in Section 2.3, we have done some preliminary work comparing the statistical similarity of accelerometer readings from a smartphone to that of a state-of-the-art seismometer. Our results in [35] indicate that there is a high degree of statistical similarity in the temporal and frequency responses of the accelerometer readings in both the smartphone and the seismometer. To illustrate a bit further, we show in Fig. 3, the sample explosion events from both smartphone and seismometer during one explosion experiment which consisted of 5 charges (i.e., Experiment 3 in Table 3). Fig. 3a shows the temporal responses of acceleration in lateral (a_x), longitudinal (a_y), and vertical (a_z) directions, along with the pressure readings (p) sensed by one Samsung GALAXY S4 phone for five explosive blasts. Fig. 3b shows the temporal responses of velocity in lateral (v_x), longitudinal (v_y), and vertical (v_z) directions sensed from the seismometer's geophone sensor, along with pressure readings (s) sensed from its microphone sensor. The perpendicular dotted lines in the figure indicate the time-stamp of the individual blast. As seen from the Fig. 3, the smartphone accelerometer is clearly sensitive to the blasts correlating with the seismometer during each blast.

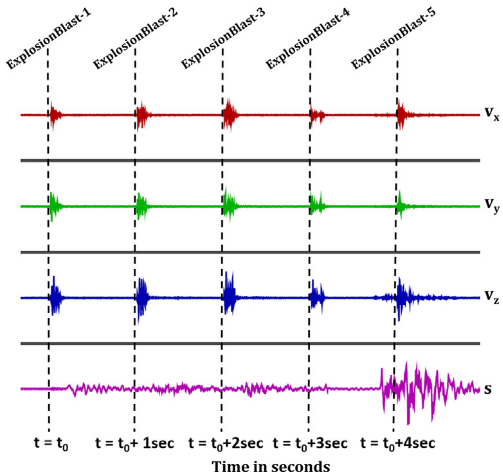
Recall from Section 3, the atmospheric pressure in the vicinity of an explosion event is sensitive to the emanating blast waves. Specifically, the blast wave causes a rise in the atmospheric pressure to a peak overpressure. As the intensity of blast wave degrades, the atmospheric pressure decays back to the ambient pressure and a negative pressure phase occur that is usually longer in duration than the positive phase. The pressure sensor on the smartphone captures this phenomenon, as we see in Fig. 3a. It can also be observed that, the pressure excitation is delayed when compared to that of both acceleration (in Fig. 3a) and velocity (in Fig. 3b) excitations. This is because the blast waves induced by explosions (that affect the ambient

Table 3
Details of the explosion experiments at ERL.

Experiment	Intensity (lbs)	No of blasts	Smartphone	Distance (feet)
E1	16.50	8	P1	45
			P2	35
E2	8.05	2	P1	35
			P2	40
			P3	43
			P4	55
E3	11.83	5	P1	26
			P2	50
			P3	60
			P4	61.5
E4	8.25	2	P1	26
			P2	35
			P3	48
			P4	55



(a) Sample explosion event from a smartphone.



(b) Sample explosion event from the seismometer.

Fig. 3. Sample explosion event containing ground vibration readings in 3 directions and pressure readings, detected from a smartphone and the seismometer.

pressure) travels slower than the shock waves (that cause ambient vibration). Similar patterns of sensitivity were observed for all other phones as well. These findings serve as our foundation in leveraging smartphone accelerometers and pressure sensors for designing our model.

5.2. *Design of a model for ranging and intensity estimation of explosion*

We propose a regression model to range and estimate the intensity of explosion events. The unknown variables we want to estimate are the distance (d) and intensity (i) of an explosion event, and these two variables are called world-state variables in our regression model. The inputs we have are the raw measurements from the smartphone accelerometers in lateral (x), longitudinal (y), and vertical (z) directions along with the pressure sensed (p). There are four steps involved in designing the model as detailed below, and illustrated in Fig. 4.

Step 1: Event separation: This is a pre-processing step on the sensor data. As we know, the duration of the explosion event is very small. But in our explosion experiments, the phones were sensing the readings for an average time of about 65 min from the time they were placed stationarily (before the start of the experiment) to the time they were collected (after the blasting experiments). This is due to safety standards at the laboratory, wherein we are required to place smartphones well before the experiment and allowed to collect them only after the confirmation that blast impacts are not present anymore. The smartphones are pre-programmed to sense the ambient vibrations and atmospheric pressure readings, continuously. This itself consumes very limited energy, and we found it to be 0.26 Joules per second. But because of high sampling rates, the storage overhead grows dramatically. In real-life scenarios, we hence need a technique that will enable a smartphone to continuously perform sensing activities, while being smart enough to filter unnecessary readings, but retain only those that pertain to an explosion event to save the storage space.

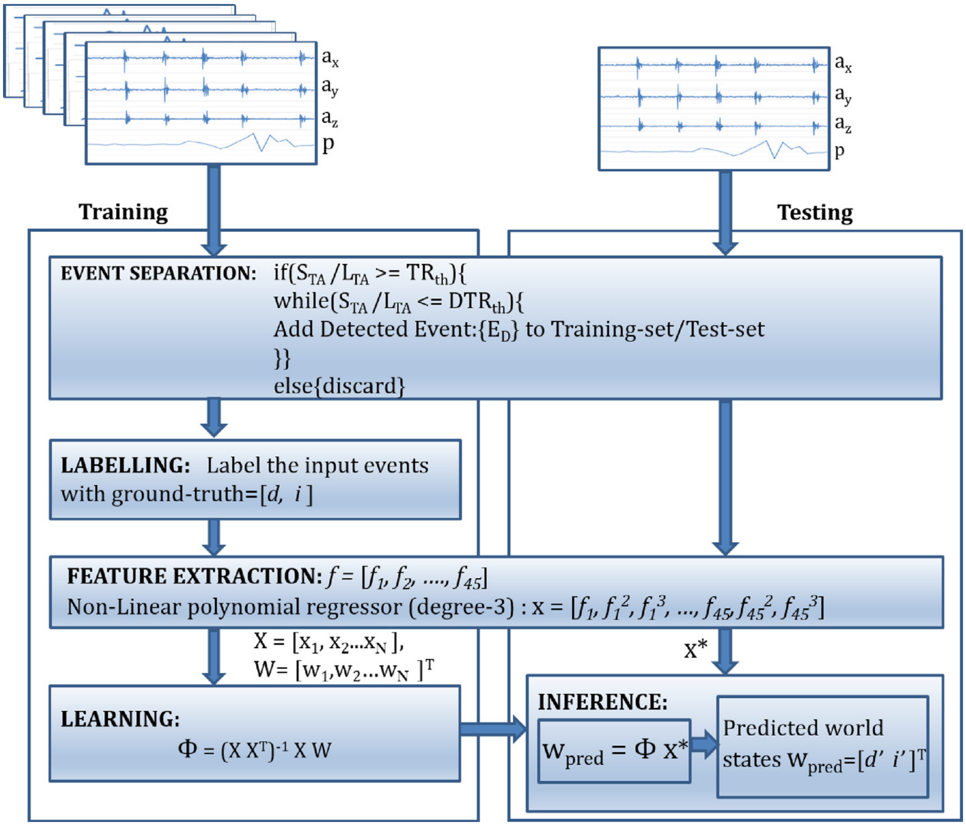


Fig. 4. Flow of the designed estimation model.

We implemented a technique that can run in real-time on the phone towards the above objective. The core rationale of our technique leverages the fact that when an explosion event happens, there is a sudden spike in the sensed vibrations, compared to the long-term dormancy in vibrations sensed by the smartphone in the absence of any explosion event. Leveraging the above insight, we compute on the smartphone in run-time, the ratio of averages of accelerometer readings within a short-term sliding window (denoted as *STQ*) to the long-term sliding window (denoted as *LTQ*). In our prior work in [35], we have identified the triggering threshold (denoted by TR_{th}) and de-triggering threshold (denoted by DTR_{th}) for this ratio which provide the best discriminatory power to isolate an explosion event as $TR_{th} = 1.75$ and $DTR_{th} = 1.5$ and with window sizes of 0.1 s for *STQ* and 10 s for *LTQ*. In our experiments, the smartphone senses the ambient vibration using its accelerometer continually, but only stores those accelerometer and pressure readings that are sensed between the triggering and de-triggering of the explosion event (as identified using the thresholds) while ignoring the rest. Only the retained individual blast event datasets are leveraged to develop the model that will be discussed in the following steps.

Step 2: Labeling: With our experimental datasets obtained from explosion events (presented in Table 3), we then perform the preprocessing step that involves normalizing the raw accelerometer and pressure sensor values in the datasets. By using Z-score normalization or Standardization[4] for sensor value scaling, the diverse smartphones with varying calibrations (sensitivities) for sensor outputs will land in the same scale. Now, we are ready to proceed with designing the model. The distance and intensity are denoted as world-state variables, which we aim to estimate

- d - distance of a smartphone from the explosion source.
- i - intensity of the explosive material blasted.

The inputs available for a detected event are

- a_x - ground acceleration in lateral (x) direction in m/s^2.
- a_y - ground acceleration in longitudinal (y) direction in m/s^2.

[4] This is a phenomenon in which the sensor values are rescaled, so that they will have the properties of standard normal distribution, meaning the values are centered around '0' with a standard deviation of '1'.

- a_z - ground acceleration in vertical (z) direction in m/s^2.
- p - pressure sensed by the pressure sensor as a scalar value in hectaPascals (hPa).

Note that we have recorded the ground-truth values *distance* (from a smartphone to the explosion source) and *intensity* (measured in terms of lb of charge weight of the explosive material) during each experiment conducted, as described in Section 4. Now, we identify the ground-truth values associated with each dataset and tag them with corresponding attributes. To identify them, we have taken into consideration, the fact that each dataset belongs to a smartphone which is placed at a certain distance from the explosion source and also corresponds to a specific intensity (charge-weight) of the explosion blast. With the knowledge of the attributes *distance* and *intensity* for each blast event dataset, we subsequently tag the attributes as labels to the blast event.

Step 3: Feature extraction: After labeling each blast event with its ground-truth data, the next step is feature extraction. Note that we have the accelerometer readings from the smartphones in lateral (x), longitudinal (y), and vertical (z) directions, along with pressure (p), and we could attempt to build a model between these values and the world-state variables, directly. However, such a model will be limited to only these sources of readings. To decide on the type of features that are needed to be extracted, let us look into the physics behind the explosions. If we examine the cause of an explosion, it comprises a material which produces energy, and propagates to the subject transforming the energy into vibrations and pressure. To make a meaningful prediction, the features should be built on the attributes involved in the life cycle of the explosion event, which are sensitive to the physical parameters — distance, intensity, and time. After careful observation, the key attributes which decide the aftereffect of explosion are identified to be frequency and energy-related attributes. Hence, we decided to emphasize on these while extracting features and build a model on them. In order to make the model more accurate, we attempt to extract several statistical features in the temporal domain (i.e. for the duration of the detected blast event), and also features in the frequency domain from the detected event. We first extract features for individual input events (for both sensors), and also combine them to create new features in the feature vector. Upon consideration of the explosion impacts to ambient vibrations, and how they impact corresponding accelerometer readings, the following intuitive features were identified in our study for each acceleration dimension

1. mean
2. median
3. variance
4. minimum
5. maximum
6. duration of event
7. dominant frequency
8. histogram properties: (a) The highest, (b) The second highest, (c) The third highest occurrences of the samples in a bin.

Note that the histogram is built based on the time-domain amplitudes of the explosion blast signals. Now we have these 10 features identified for each dimension, resulting in 30 features for three dimensions of accelerometer readings. The size of the bin is the duration of the blast event.

Before we discuss the feature extraction from pressure readings, let us briefly explain the physics relating to changes in the atmospheric pressure as a result of an explosion. Pressure waves emanating from an explosion propagates much slower than that of ground vibration [38,39]. The blast pressure wave in air typically travels at 330 meters per second while shock waves carrying vibrations travel in the order of 10,000 m per second in solid medium.[5] These studies coupled with our analysis of the experimental data sensed for both sensing modalities, helped us make the following key observations

- The arrival times of the excitations caused by the pressure and acceleration are different, from the perspective of the smartphone. The smartphone senses the acceleration excitation caused by the blast earlier compared to when it senses the pressure excitation.
- The delay between the arrival times of acceleration and pressure excitation of an explosion blast increases with the distance between the smartphone and explosive source, as shown in Fig. 5.

Using the insights from the above observations, we extracted a total of 6 intuitive features from pressure readings towards building our model. Note that the pressure sensor outputs the samples only in one dimension, unlike the accelerometer sensor. The features we extracted are

1. Delay between arrival times of the acceleration and pressure excitation of an explosion blast
2. maximum
3. mean
4. variance
5. histogram properties: (a) The highest, (b) The second highest occurrences of the samples in a bin.

[5] Note that there are some variations based on properties of the medium through which they propagate.

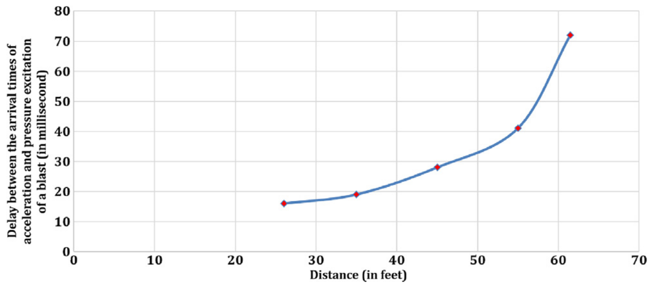


Fig. 5. Delay vs. distance.

To enhance the feature vector, we then create features that explore the correlation between the two sensor modalities. The most intuitive feature in this realm is the product of accelerometer readings and pressure sensor readings. In this paper, we identify a total of 9 additional features as a result of combining features of these two sensing modalities which are described below

1. Product of the acceleration maximum in x dimension and the pressure maximum.
2. Product of the acceleration maximum in y dimension and the pressure maximum.
3. Product of the acceleration maximum in z dimension and the pressure maximum.
4. Product of the highest occurrence in the histogram for the accelerometer reading in x dimension and pressure.
5. Product of the highest occurrence in the histogram for the accelerometer reading in y dimension and pressure.
6. Product of the highest occurrence in the histogram for the accelerometer reading in z dimension and pressure.
7. Product of the second highest occurrence in the histogram for the accelerometer reading in x dimension and pressure.
8. Product of the second highest occurrence in the histogram for the accelerometer reading in y dimension and pressure.
9. Product of the second highest occurrence in the histogram for the accelerometer reading in z dimension and pressure.

From the above, we can see that for any given dataset with accelerometer and pressure readings from the smartphone, we now have 45 features in total resulting in a feature vector of size 45 ($f = [f_1, f_2, \ldots, f_{45}]$).

The next thing to do is to select a model that corresponds to the input feature vector. We have analyzed the relationship between input features (one at a time) with world-state variables (distance and intensity). As discussed earlier in the section, we have identified that the features are sensitive to the world-state variables and hence will be suitable candidates for the selected feature vector. Furthermore, the analysis also revealed the non-linear relationship between the features and world-state variables in most of the cases. Our observation physically translates to - a change of acceleration and pressure measured by a smartphone during an explosion has a nonlinear relationship to the measured distance from the smartphone to the source of an explosion and also to its intensity (which is also true across all phones). As an initial trial, we attempted to build a linear regression model. As such, the model suffered from imprecise fitting problems. In most of the cases, the prediction error was found be close to 50%. This motivated us to employ a nonlinear regression model which resulted in a good fit corresponding to the training data.

Step 4: Learning: The datasets in a given training-set are used to build a model between the world-state variables (i.e., distance and intensity) and the input features extracted. Recall that, we have a total of 52 datasets extracted from 'Event Separation' discussed earlier.

We implemented a nonlinear polynomial regression model to fit the training-set. A nonlinear polynomial regressor can have a degree 2 or higher. We have found that after degree 3, the increase in estimation accuracy is negligible for the context of our current problem. Hence, we set the nonlinearity degree as 3 for our model. This resulted in the final feature vector size growing to 135 for 45 features extracted. For each dataset, an m-dimensional (here $m = 135$) feature vector x is extracted along with a 2 dimensional world-state vector w (ground-truth distance and intensity) as shown below

$$x = [f_1, f_1^2, f_1^3, \ldots, f_{45}, f_{45}^2, f_{45}^3]^T;\ w = [d, i]^T. \tag{1}$$

Aggregating all training samples into a matrix representation, we have

$$\boxed{X = [x_1, x_2, \ldots x_N] \text{ and } W = [w_1, w_2, \ldots w_N]^T.}$$

where each x is an m-dimensional vector ($m = 135$), N is the size of training-set, thus the size of X is $m \times N$ and of W is $N \times 2$ since each w is a 2-dimensional vector. In our case, we have a number of training datasets less than the size of feature vector ($N < m$). This can result in non-invertibility of the input feature vector matrix X. In addition to this, there may be features presenting in our feature vector that will not contribute to the model. The feature-set in its current state containing many features, if used straightaway for training will lead to overfitting problems. So, it needs to be pre-processed which essentially involves feature selection to handle this issue.

In our case, the resulted feature vector is non-sparse (containing very few non-zero real values). We could use Principal Component analysis (PCA) technique for feature selection and also to address the problem of overfitting. But, in our case, considering the non-sparse nature of the feature-vector, PCA just throws away some information and converted to lower dimensional space. We also examined other potential feature selection algorithms including decision tree based feature selection algorithms, which as well demonstrated similar issues related to information loss.

To address this problem, we opted to use Regularization. We use $L-2$ Regularization ($Ridge-Regression$), where it penalizes the coefficients of non-significant features by adding a regularization term. The main reason behind using Ridge Regression is to preserve all significant features, besides addressing the problem of overfitting. With this formulation, we now define the cost function as

$$O(\phi) = \|X^T\phi - W\|^2 + \lambda\|\phi\|^2 \tag{2}$$

where the weight of the regularization term (λ) will be determined by a leave-one-out cross-validation ($LOOCV$) strategy, details of which are given in Section 6. Solving the cost function shown in Eq. (2) leads to the estimated parameter ϕ as shown below in Eq. (3) below. The parameter ϕ is a matrix of size $m \times 2$ (135×2), where each column corresponds to a world-state variables and I is the identity matrix of order $m \times m$

$$\phi = (XX^T + \lambda I)^{-1}XW. \tag{3}$$

It is to be noted that the parameters estimated corresponds to the current experimental settings that include underground mine environment and type of explosive used. As mentioned earlier in Section 4.2, the current blasting material (Dynamite) constitutes majority of real explosions that occur at high intensities in real-world. So, the current parameter estimation will still be relevant with respect to the type of explosive used. The accelerometer sensor in the evolving smartphones are only improving in terms of sampling rates. So, a modern updated sensor will not impact the parameters estimated for the current setting, instead contribute to improvement in prediction accuracies. Generalizing the parameters to all the environments is rather a part of our future work that requires working with datasets collected from various environmental settings.

If we have more and more datasets collected over time, the trained model will become consistent and prediction accuracies will improve relatively. But, if the training data grow regularly in size over the time, we need to re-compute the model parameters iteratively. We can avoid this by adapting sequential learning where we do not have to do the entire training process.[6]

Inference: Given a new explosive event, we perform the Event separation and Feature extraction steps, and then apply the nonlinear transformation to obtain an m-dimensional ($m = 135$) feature vector X^*. Then, we infer the 2-dimensional world-state vector w_{pred} by

$$w_{pred} = \phi^T X^*. \tag{4}$$

we get $w_{pred} = [d' \; i']^T$, where d' and i' are the estimated distance and intensity, respectively.

6. Results and analysis

In this section, we first present our approach for experimental validation of our model, and then we present the quantitative performance evaluations on the model. Thirdly, we compare our results in this paper using multi-modal sensors with our prior work in [36] that leverages accelerometer sensors only. Finally, we present detailed analysis on designing a smartphone app to estimate the range and intensity of an explosion event, and demonstrate the app's performance specifications.

6.1. Experimental validation of our model

We perform the experimentation in the following steps

Step 1: First we split the total datasets into training-set and test-set. In our approach, a test-set will include the datasets from one phone during an experiment, and datasets from all the other phones form the training-set used for building the model. For instance, in Table 4, the 8 blast event datasets sensed by Phone *P1* during Experiment *E1* will form the test-set-1, while the 8 blast event datasets sensed by Phone *P2* from the same experiment (*E1*), along with datasets from all the phones in *E2*, *E3*, and *E4* (a total of 44 datasets) will form the corresponding training-set.

Step 2: Now we build the model (Eq. (3)) using the training-set of size N. We use a coarse-to-fine grid search strategy to select the optimal λ. Initially we search different magnitudes (e.g., 0.001, 0.01, ..., 1, 10, ..., 10^4, 10^5, ...). After we find the optimal interval (e.g., $\lambda \in [0.1, 1]$), we divide the interval by ten and finely search the optimal λ. The process continues until

6 In cloud based machine learning platforms today, capabilities for streaming machine learning and transfer learning are available to support this kind of requirement.

Table 4
Evaluation of our model for distance estimation.

Test-set	Training-size	Test-size	Distance (in feet)		
			Actual	Estimated	Error (%)
1	44	8	45	48.69	8.2
2	44	8	35	34.53	1.34
3	50	2	35	32.78	6.34
4	50	2	40	41.78	4.45
5	50	2	43	44.1	2.55
6	50	2	55	51.67	6.05
7	47	5	26	23.59	9.26
8	47	5	50	56.24	12.48
9	47	5	60	54.82	8.63
10	47	5	61.5	56.46	8.19
11	50	2	26	33.45	28.65
12	50	2	35	34.1	2.57
13	50	2	48	53.27	10.97
14	50	2	55	51.89	5.65

Table 5
Evaluation of our model for intensity estimation.

Test-set	Training-size	Test-size	Intensity (in lb)		
			Actual	Estimated	Error (%)
1	44	8	16.5	16.19	1.87
2	44	8	16.5	16.04	2.78
3	50	2	8.05	8.08	0.37
4	50	2	8.05	8.69	7.95
5	50	2	8.05	9.67	20.12
6	50	2	8.05	7.34	8.81
7	47	5	11.83	10.29	13.01
8	47	5	11.83	10.98	7.18
9	47	5	11.83	12.48	5.49
10	47	5	11.83	12.63	6.76
11	50	2	8.25	7.65	7.27
12	50	2	8.25	7.17	13.09
13	50	2	8.25	8.8	6.66
14	50	2	8.25	8.1	1.81

it converges. To evaluate the performance of each λ during the coarse-to-fine grid search, we use the standard 5-fold cross validation on the training dataset.

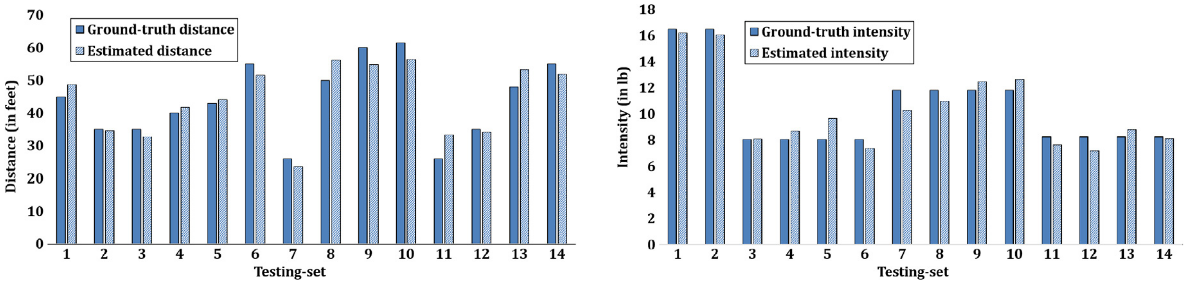
Step 3: Now the resultant model built after tuning the parameter λ, is used to infer world-state variables using the Eq. (4). Note that each dataset within a test-set will have a predicted value for world-state variables (distance and intensity). After validating the model on a test-set, we took mean of the estimated values (mean of distance values, mean of intensity values) of all the datasets within each test-set. This completes estimation of world-state variables for one test-set.

The above process containing three different steps is repeated for all test-sets, and as such, we have a total of 14 test-sets resulted from 52 datasets. As can be seen in Tables 4 and 5, for 14 test-sets the estimated world-state variables are compared against their ground-truth values.

Training time of our model: Recall that our feature vector was of size 135 in our nonlinear regression (degree-3) model. We implemented our model in Matlab environment. With this feature vector, the total training took 11.2 seconds on a Windows machine with 32 GB of RAM, and CPU clock speed of 4.6 GHz.

6.2. Quantitative performance evaluation of our model

Tables 4 and 5 detail the summary of the performance of our model. Each row describes a test-set with attributes such as its training-size, test-size along with estimated, actual (ground-truth) values of distance and intensity along with their error percentage values for estimation. We define 'error' by the relative error (the absolute difference between the ground-truth value and estimated value, divided by the ground-truth value). We observe that the error in estimating the distance is quite low, in fact, it is less than 10% in a majority of instances and this observation holds to the intensity as well. It was shown that the model could achieve an error less than 5 m for distance estimation and an error less than 1.5 lb for intensity estimation in almost 90% of the cases. Figs. 6a and 6b show the bar-plot to visualize the performance of our model. Model is evaluated using statistical metrics also, shown in Table 6, which will be discussed in next subsection. L2-regularization worked better for our case, by penalizing the non-significant features and turned out to be that number of features did not impact over-fitting.



(a) Distance estimation. (b) Intensity estimation.

Fig. 6. Evaluation of our Model.

Table 6
Statistical comparison with prior work.

		Model using both accelerometer and pressure data in this paper	Model using only accelerometer data in [36]
Distance	Average-case error (in %)	8.241	12.86
	RMS-deviation (rmsd)	4.008	7.805
	Normalized-rmsd (nrmsd)	0.112	0.219
	Training error	1.071	2.086
Intensity	Average-case error (in %)	7.374	11.26
	RMS-deviation (rmsd)	0.838	1.429
	Normalized-rmsd (nrmsd)	0.099	0.169
	Training error	0.224	0.381

6.3. *Comparison of our current model with the model in the previous work*

As we pointed earlier, in a prior work in [36], we attempted to build a model to estimate the range and intensity of an explosion event with only the accelerometer from smartphones. In Table 6, we compare our new model using both the accelerometer and pressure sensor, with the previous work using the acceleration sensor data only. The comparison of models is demonstrated using statistical analysis and is summarized in Table 6. The statistical metrics we use to quantify the estimation performance of the models are average-case error, root-mean-square-deviation (rmsd),and normalized-rmsd (nrmsd) are evaluated from the testing along with the training error. For determining 'average-case error', we find the mean of 'error's for all the test-sets. An average-case error of 8.24% was observed for ranging in our experiments. The intensity on the other hand is also accurate in our estimation with an observed average-case error of 7.37%.

As we can see in Table 6, the model in the current paper that integrates multiple sensors (i.e., accelerometer and pressure), consistently outperforms the model in [36] that leverages data from only one sensor (i.e., accelerometer). This provides us further impetus that the techniques fusing multiple sensing modalities enable our model to learn much better for characterizing explosions.

These results being the first of their kind in the realm of estimating the range and intensity of an explosion event from stationarily emplaced smartphones, are quite promising and demonstrate the feasibility for the improvement in future with more experiments. The results demonstrate good consistency, hence validating our model.

6.4. *Designing a smartphone app*

We now report our findings on designing a smartphone app based on our model in this paper to estimate the range and intensity of an explosion event. We implemented the application as a two-module architecture, as shown in Fig. 7. The functionality of the first module is to detect the triggering of an explosion event and handover the detected event to the second module which estimates the distance and intensity of the explosion. The detailed functionality is described in the following:

Functionality of module 1: This module is an implementation of the algorithm proposed in our previous work [35] to sense and retain only explosion events. Note that, in the current implementation we have used time-window values, threshold constants determined in our prior work in [35].

While the app runs continuously in the background, the smartphone senses and processes the acceleration samples in two moving time windows of different sizes. The sample 'S_a' consists of accelerometer readings available in x, y, z directions. The readings in sample 'S_a' are added to two moving time windows, namely a 'short-time-queue' (STQ) of size 0.1 s and 'long-time-queue' (LTQ) of size 10 seconds (simultaneously for all the 3 directions). Then, averages are calculated in the

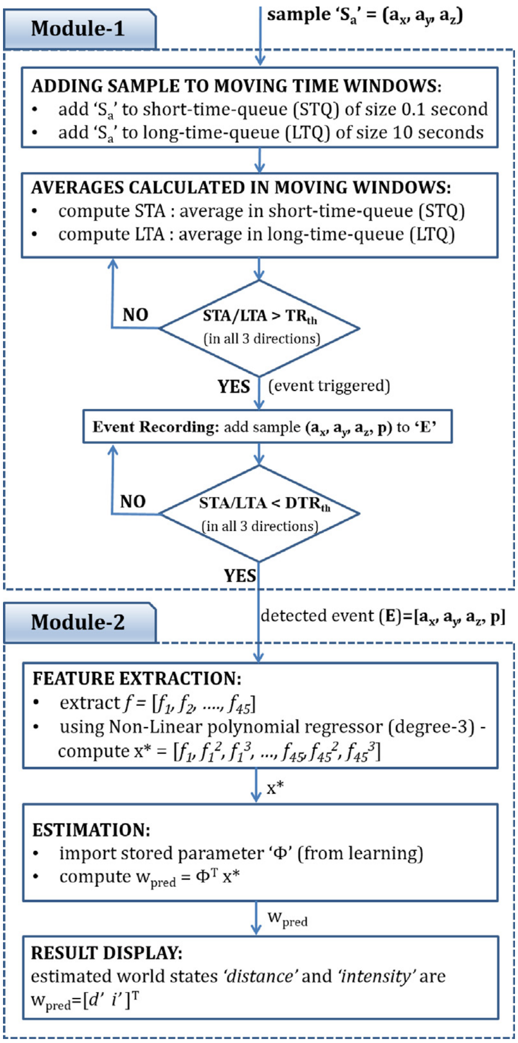


Fig. 7. Workflow of our smartphone app.

moving time windows, and their ratio is computed as *short-time-queue-average/long-time-queue-average (STA/LTA)*. This ratio is compared against the trigger-threshold TR_{th} whose value is 1.75. If the ratio is found to exceed the threshold in all 3 directions, then the algorithm triggers the detection of explosion event and starts recording the explosion event as 'E', by adding accelerometer and pressure sensor readings to it (i.e., 'E' contains readings a_x, a_y, a_z, p). The event is recorded until the ratio value falls below the de-trigger-threshold DTR_{th} whose value is 1.5, in all 3 directions. This completes the functionality of module-1 resulting in the output 'E' = $[a_x, a_y, a_z, p]$, which are the sensor readings corresponding to the detected explosion event.

Note that, the app while collecting the raw sensor values sets the sampling rate of accelerometer sensor to be 100 samples per second and that of pressure sensor to 10 samples per second, to make the algorithm work uniformly across all the smartphone platforms.

Functionality of module 2: This module is an implementation of the model proposed in the current paper for ranging and estimating the intensity of an explosion event detected by module-1.

As soon as the module receives the event 'E' $= [a_x, a_y, a_z, p]$ from the previous module, the feature vector $f = [f_1, f_2, \ldots, f_{45}]$, is extracted for the event. The feature vector is then fed to a nonlinear polynomial regressor of degree-3, which results in a test vector $x^* = [f_1, f_1^2, f_1^3, \ldots, f_{45}, f_{45}^2, f_{45}^3]^T$. The next step is inference, where world-state variables - *distance* and *intensity* are estimated for x^* based on our learned regression model. As described in Section 5.2, the learning from the training datasets results in a parameter ϕ. This learned parameter is a static matrix resulted from training process and will be stored in the application, which will be used for the estimation of *distance* and *intensity* of an explosive event.

The estimation is done as $w_{pred} = \phi^T x^*$, resulting in a vector containing estimated *distance* and *intensity*. These estimations are the outputs of the module which are displayed as the final output of the application.

Specifications of our smartphone app: The overall app that contains the algorithms to sense and store only accelerometer/pressure readings pertaining to an explosion, and the algorithm to estimate the range and intensity of an explosion event is of size 4 MB. This is well within the limits as per smartphone OS standards for app design. Detecting the explosion event, and computing the range and intensity happen almost instantaneously. The energy consumed by the entire application is 0.350 Joules per second on the average. The average RAM memory in the smartphone consumed by the application for processing accelerometer and pressure readings is 2.4 MB which is quite low as well.

7. Discussions

In this section, we briefly highlight some additional perspectives of our work on the feasibility of leveraging smartphones to range and estimate the intensity of explosion events.

a. Practical values of our contributions: An important issue to point out is that in our experiments, the smartphones deployed are stationary. However, our ensuing contributions are still practically useful, since smartphones are not mobile all the time, and they are actually stationary for a significant portion of the time. The issue of experimenting with mobile smartphones (emplaced in moving humans) requires much more careful planning to ensure the subject's safety during the experiments,[7] and is part of our future work based on contributions of this paper. It is also straightforward to extend our work to the case of mobile smartphones if appropriate filtering algorithms can be designed to filter out acceleration readings generated as a result of humans moving with the phones. Such algorithms are feasible to design with a proper training. There are works like [32] that attempt similar objectives. Also, with major initiatives nowadays across the world on instrumenting buildings and living spaces with "smart sensors", we believe that our work is very timely when our algorithms can be integrated with these smart devices (most of which are stationary) for pervasive sensing and analysis of explosion events.

b. Detecting the type of explosive materials: In this paper, we do not address the important issue of detecting the type of explosive blasted. In all our experiments, it was Dynamite (with ANFO), and this is assumed to be known. However, in practice, we may need techniques to identify the material blasted as well. We believe that is doable today with many prototypes on integrating chemical sensors with smartphones [40–42]. Such integrations can dramatically enhance the sensing capabilities of smartphones, and they can now perform initial detection of explosives material blasted, following which our algorithms can adaptively attempt to analyze them based on the type of material blasted. This will significantly expand the practicality of our proposed techniques and is part of our future work.

c. Some perspectives on the app design and smartphone capabilities: In Section 6, we demonstrated the feasibility of designing smartphone apps for sensing and analyzing explosion events. We have also identified and addressed an important problem of diversity of smartphone sensors, which is a very crucial aspect in the realm of participatory sensing applications. While we demonstrated critical specs of the app we designed, and its performance, there are some additional issues in practice. In any smartphone today, there are a number of ancillary services like GPS services, application updates, and notification services running in the background in a phone at all times. However, most of these services are pushed into the background depending on priorities and processing costs of more important applications like calling/messaging. While more experiments are certainly needed from the context of this paper, we believe that our application in this paper for explosion sensing and analyzing incurs very little processing costs, and hence we expect minimal impact to the execution of its services when more important services are processed by the smartphone.

Also, for the kind of applications mentioned in this paper, the key aspect is data processing in real-time. However, this can be very challenging from battery perspective due to quick draining caused by usage of multiple sensors at the same time and heavy usage of resources, gigabytes of RAM for onboard processing. In this paper, our model is to let each smartphone make independent decisions on the range and intensity of the explosion event. While this is still practical, we also want to state that it may be better to let the sensory data be offloaded to a central server like a cloud that can process data from many devices for superior accuracy, and also save battery consumption at the phone end. How to do this in real-time, while minimizing false negatives and positives in identifying if a particular stream of data is explosion related or not is the practical future challenge to overcome. Furthermore, with recent innovations in wearable computing, there are also newer sensor platforms we could integrate with our current model. We also anticipate the future smartphones will handle yet faster read/writes that assist high sensor sampling rates. Along with that, we may see pre-built efficient onboard API's for data preprocessing which will improve the real-time processing on a great extent.

d. Future works: We point out clearly that there are many other challenges in the realm of network bandwidth, trust, security, and privacy of data when numerous smartphones in a real-time sense and transmit data for detecting explosions at societal scales pervasively. However, these challenges are currently outside the scope of the paper, and is part of our future work. There are some critical open issues that we are addressing currently.

7 Currently, in the underground mine at Explosion Research Laboratory (ERL), we are not allowed to carry phones and walk near the explosion site.

First is the integration of more sensing functionalities for modeling. One modality which we could consider is Acoustic. Acoustic signals do provide a rich stream of information in the context of explosives sensing, and features like pitch, timbre, spectral responses, and Mel-frequency cepstrum. We strongly believe that modeling accuracy will improve, but the trade-off is processing efficiency and also privacy, since a scenario involving pervasive sensors (static or otherwise) collecting acoustic information can raise privacy concerns. This issue needs deeper study.

Secondly, we are planning to consider networking of multiple smartphones in real time to report data to the Cloud for superior accuracy and efficiency. Instead of streaming all data that can drain battery, a potential approach could be exporting only those data that meet the criteria for an explosion as determined by processing accelerometer signals via the STA/LTA module in our paper. This approach will save battery resources by preventing unnecessary transmission of sensory data unrelated to explosion events to the cloud. Challenges like data losses and privacy considerations need to be overcome here.

As mentioned earlier, conducting experiments with multiple static sensors, and mobile smartphones from different manufacturers, and also leveraging sensory data from wearable devices are also part of our ongoing work to better understand the utility of our work proposed. With more training data and newer advances in deep learning approaches, we could explore alternate modeling techniques that may reduce false negatives/ positives, while also improving accuracy, and this is also a potential future work.

Conducting experiments in a variety of environmental settings is also a part of our future work for generalizing the explosion prediction capabilities across all environments.

8. Conclusions

In this paper, we designed a model for ranging and estimate the intensity of explosion events by leveraging multi-modal smartphone sensors. To the best of our knowledge, this work is pioneering. Our technique employed a nonlinear polynomial regression model using numerous features extracted from the accelerometer and pressure sensors in smartphones. We showed our work in this paper that integrates accelerometer and pressure sensors in smartphones can improve the estimation accuracy of range and intensity of an explosion event, compared to prior work that only used accelerometer sensors, by comparing performance across several standardized statistical metrics. There can be further improvement with future advancements in smartphone sensors.

Our results are quite promising and demonstrate the clear feasibility of designing smartphone based pervasive and participatory sensing networks for the problem of ranging explosion events, and determining their intensities. We subsequently provided details and specs of our smartphone app to sense and analyze explosion events. We also provided important discussions that will guide the future practical deployments of our technologies in this paper.

Acknowledgments

The authors would like to thank Professor Paul Worsey and the staff of Experimental Mining Laboratory at Missouri University of Science and Technology for their assistance in experiments.

The authors would also like to thank GeoSonics/Vibra-Tech Inc. for providing raw data of seismometer during explosion events that served as the ground-truth for analyzing the capabilities of smartphones to solve the problems in the context of explosions.

This work was partially supported by NSF grant CNS-1205695, and Intelligent System Center at Missouri University of Science and Technology.

Appendix. More details of the blast experiment

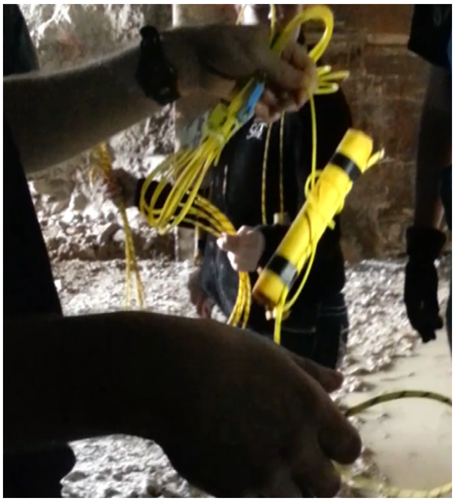
In this section, we attempt to describe steps in blasting experiments at the experimental mine where we emplaced our smartphones. We expect these discussions to assist future experiments by various stakeholders in this realm.

Explosion material: The initial step is choosing the type of explosive material. In all our experiments it was Dynamite (Unimax TT) with Ammonium Nitrate Fuel Oil (ANFO). Fig. 8a shows how it looks like. The total explosive material are of different weights, and also have a varied number of charges fixed to it. As mentioned in Section 4.2, the amount of explosive material used determines the intensity of the explosion, and the number of charges determines the number of individual blasts in an experiment.

Fixing detonating cords to explosive: After choosing the explosive material, the next step is to connect detonating cords to it. Fig. 8b shows a detonating cord being attached.

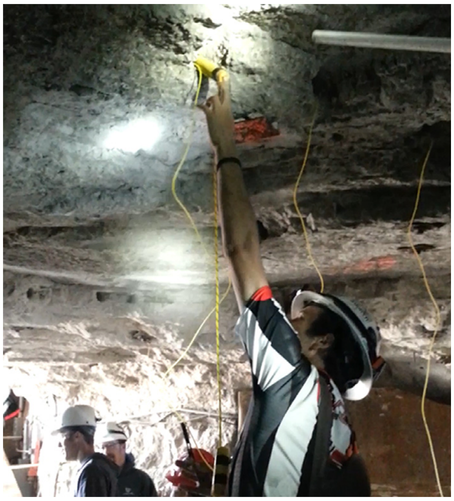
Mounting the explosive material: The material to be blasted in the explosion is selected and holes are drilled on the material to exactly fit the explosive material. In our case, the explosion environment is an underground mine, so holes are drilled and explosive material is fixed as shown in Fig. 8c.

Connecting the detonating cords: All the detonating cords are later connected to a single point as shown in Fig. 8d. An explosion is detonated from a distance through a long detonating cord which is connected to the junction of cords associated to individual explosive material.



(a) Explosive material — Dynamite.

(b) Fixing a detonating cord to the explosive material.



(c) Mounting the explosive material into the mine's surface.

(d) Connecting individual detonating cords together.

(e) Smartphone emplaced (on the ground) near the explosion site.

Fig. 8. Sequence of the setup of explosion experiments at Explosives Research Lab (ERL).

Installing smartphones on the ground: We emplaced a number of smartphones at various distances from the source of the explosion. The smartphones were placed well in advance of detonation, and we were allowed to pick them up only after experts assured us that there were no remnants of the blast. A snapshot of the phone after the blast is shown in Fig. 8e. All data stored were fully recoverable for all the phones.

References

[1] Paris terrorist attacks and suicide bombings of 2015. https://en.wikipedia.org/wiki/November_2015_Paris_attacks/.

[2] Turkey train station bombing of 2015. http://www.cnn.com/2015/10/10/middleeast/turkey-ankara-bomb-blast/.

[3] Nigeria bomb blasts of 2015. http://www.cnn.com/2015/11/18/africa/nigeria-blasts/.

[4] Boston Marathon bombing of 2013. http://www.britannica.com/EBchecked/topic/1924021/Boston-Marathon-bombing-of-2013.

[5] Y. Yan, L. Cheng, Z. Wu, L. Yam, Development in vibration-based structural damage detection technique, Mech. Syst. Signal Process. (ISSN: 0888-3270) 21 (5) (2007) 2198–2211. http://dx.doi.org/10.1016/j.ymssp.2006.10.002. http://www.sciencedirect.com/science/article/pii/S0888327006002226.

[6] D. Hiller, V. Hope, et al., Groundborne vibration generated by mechanized construction activities, Proc. ICE-Geotech. Eng. 131 (4) (1998) 223–232.

[7] R. Bogue, Recent developments in mems sensors: a review of applications, markets and technologies, Sensor Rev. 33 (4) (2013) 300–304.

[8] Teardown: Samsung Galaxy-S4. http://electronics360.globalspec.com/article/94/teardown-what-s-inside-samsung-s-galaxy-s4/.

[9] Z. Zhuang, K.-H. Kim, J.P. Singh, Improving energy efficiency of location sensing on smartphones, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, 2010, pp. 315–330.

[10] A. Misra, L. Lim, Optimizing sensor data acquisition for energy-efficient smartphone-based continuous event processing, in: Mobile Data Management, MDM, 2011, 12th IEEE International Conference on, vol. 1, IEEE, 2011, pp. 88–97.

[11] Y. Chon, E. Talipov, H. Shin, H. Cha, Mobility prediction-based smartphone energy optimization for everyday location monitoring, in: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, ACM, 2011, pp. 82–95.

[12] M. Faulkner, R. Clayton, T. Heaton, K.M. Chandy, M. Kohler, J. Bunn, R. Guy, A. Liu, M. Olson, M. Cheng, A. Krause, Community sense and response systems: your phone as quake detector, Commun. ACM (ISSN: 0001-0782) 57 (7) (2014) 66–75. http://dx.doi.org/10.1145/2622633. http://doi.acm.org/10.1145/2622633.

[13] M. Sousa, A. Techmer, A. Steinhage, C. Lauterbach, P. Lukowicz, Human tracking and identification using a sensitive floor and wearable accelerometers, in: Pervasive Computing and Communications, PerCom, 2013, IEEE International Conference on, IEEE, 2013, pp. 166–171.

[14] B. Li, B. Harvey, T. Gallagher, Using barometers to determine the height for indoor positioning, in: Indoor Positioning and Indoor Navigation, IPIN, 2013 International Conference on, IEEE, 2013, pp. 1–7.

[15] T. Willemsen, F. Keller, H. Sternberg, Concept for building a smartphone based indoor localization system, in: Information Fusion, FUSION, 2014, 17th International Conference on, IEEE, 2014, pp. 1–8.

[16] M. Wu, P.H. Pathak, P. Mohapatra, Monitoring building door events using barometer sensor in smartphones, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2015, pp. 319–323.

[17] K.-M. Yang, S.-B. Cho, A non-gps low-power context-aware system using modular bayesian networks, Proc. Mobility (2014).

[18] Q. Xu, R. Zheng, S. Hranilovic, Idyll: indoor localization using inertial and light sensors on smartphones, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2015, pp. 307–318.

[19] F. Wahl, T. Kantermann, O. Amft, How much light do you get? Estimating daily light exposure using smartphones, in: Proceedings of the 2014 ACM International Symposium on Wearable Computers, ACM, 2014, pp. 43–46.

[20] B.C. Johnson, J.B. Jamieson, R.R. Stewart, Seismic measurement of fracture speed in a weak snowpack layer, Cold Reg. Sci. & Technol. 40 (1) (2004) 41–45.

[21] M. Dobrin, R. Simon, P. Lawrence, Rayleigh waves from small explosions, EOS Trans. Am. Geophys. Union 32 (6) (1951) 822–832.

[22] Experimental Mine, Rolla, MO. http://mining.mst.edu/.

[23] Z. Chen, M. Lin, F. Chen, N.D. Lane, G. Cardone, R. Wang, T. Li, Y. Chen, T. Choudhury, A.T. Campbell, Unobtrusive sleep monitoring using smartphones, in: Pervasive Computing Technologies for Healthcare, PervasiveHealth, 2013, 7th International Conference on, IEEE, 2013, pp. 145–152.

[24] C.A. Ronao, S.-B. Cho, Human activity recognition using smartphone sensors with two-stage continuous hidden markov models, in: ICNC ,2014, pp. 681–686.

[25] N. Roy, A. Misra, D. Cook, Ambient and smartphone sensor assisted adl recognition in multi-inhabitant smart environments, J. Ambient Intell. Human. Comput. (2015) 1–19.

[26] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, D. Anguita, Transition-aware human activity recognition using smartphones, Neurocomputing 171 (2016) 754–767.

[27] D. De, P. Bharti, S.K. Das, S. Chellappan, Multimodal wearable sensing for fine-grained activity recognition in healthcare, Int. Comput. IEEE 19 (5) (2015) 26–35.

[28] R.E. Guinness, Beyond where to how: a machine learning approach for sensing mobility contexts using smartphone sensors, Sensors 15 (5) (2015) 9962–9985.

[29] M. Tateishi, Context recognition on smartphones by categorizing extracted features from acoustic and other sensor data, J. Acoust. Soc. Am. 138 (3) (2015) 1782–1782.

[30] Y. Zou, G. Wang, K. Wu, L.M. Ni, SmartSensing: sensing through walls with your smartphone!, in: Mobile Ad Hoc and Sensor Systems, MASS, 2014, IEEE 11th International Conference on, IEEE, 2014, pp. 55–63.

[31] J. Reilly, S. Dashti, M. Ervasti, J.D. Bray, S.D. Glaser, A.M. Bayen, Mobile phones as seismologic sensors: automating data extraction for the ishake system, IEEE Trans. Autom. Sci. Eng. 10 (2) (2013) 242–251. http://dx.doi.org/10.1109/TASE.2013.2245121.

[32] Q. Kong, R. Allen, Using smartphones to detect earthquakes, in: AGU Fall Meeting Abstracts, vol. 1, 2012, p. 04.

[33] G. Liu, M. Iwai, Y. Tobe, D. Matekenya, K.M.A. Hossain, M. Ito, K. Sezaki, Beyond horizontal location context: measuring elevation using smartphone's barometer, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, pp. 459–468.

[34] K. Muralidharan, A.J. Khan, A. Misra, R.K. Balan, S. Agarwal, Barometric phone sensors: more hype than hope!, in: Proceedings of the 15th Workshop on Mobile Computing Systems and Applications, ACM, 2014, p. 12.

[35] S. Thandu, B. Pratool, L. Malott, S. Chellappan, On the feasibility of leveraging smartphone accelerometers to detect explosion events, in: Mobile Data Management, MDM, 2015, IEEE 16th International Conference on, 2015, pp. 1–6.

[36] S.C. Thandu, S. Chellappan, Z. Yin, Ranging explosion events using smartphones, in: Wireless and Mobile Computing, Networking and Communications, WiMob, 2015 IEEE 11th International Conference on, 2015, pp. 492–499. http://dx.doi.org/10.1109/WiMOB.2015.7348002.

[37] A.M. Online, The basics of blasting vibrations and noise, 2016. http://www.molimestone.com/site_page.cfm?pk_association_webpage_menu=2285{&}pk_association_webpage=6082 (Accessed 21 February 2016).

[38] Controlling adverse effects of Blasting. http://www.osmre.gov/resources/blasting/docs/WYBlasterCertModules/8AdverseEffectsBlasting.pdf.
[39] Risk Managemet Series: Reference Manual to Mitigate Potential Terrorist Attacks Against Buildings. http://www.fema.gov/media-library-data/20130726-1455-20490-7465/fema426_ch4.pdf.
[40] D. Hasenfratz, O. Saukh, S. Sturzenegger, L. Thiele, Participatory air pollution monitoring using smartphones, Mobile Sensing (2012) 1–5.
[41] M. Rossi, D. Brunelli, Ultra low power wireless gas sensor network for environmental monitoring applications, in: Environmental Energy and Structural Monitoring Systems, EESMS, 2012, IEEE Workshop on, IEEE, 2012, pp. 75–81.
[42] R.A. Potyrailo, N. Nagraj, C. Surman, H. Boudries, H. Lai, J.M. Slocik, N. Kelley-Loughnane, R.R. Naik, Wireless sensors and sensor networks for homeland security applications, TRAC Trends Anal. Chem. 40 (2012) 133–145.