Smart Seismic Sensing for Indoor Fall Detection, Location, and Notification

Jose Clemente, Fangyu Li, Maria Valero, and WenZhan Song

***Abstract*—This paper presents a novel real-time smart system performing fall detection, location, and notification based on floor vibration data produced by fall downs. Only using floor vibration as the recognition source, the system incorporates a person identification through vibration produced by footsteps to inform who is the fallen person. Our approach operates in a real-time style, which means the system recognizes a fall immediately and can identify a person with only one or two footsteps. A collaborative in-network location method is used in which sensors collaborate with each other to recognize the person walking, and more importantly, detect if the person falls down at any moment. We also introduce a voting system among sensor nodes to improve person identification accuracy. Our system is robust to identify fall downs from other possible similar events, such as jumps, door close, and objects fall down. Such a smart system can also be connected to smart commercial devices (such as Google Home or Amazon Alexa) for emergency notifications. Our approach represents an advance in smart technology for elder people who live alone. Evaluation of the system shows that it is able to detect fall downs with an acceptance rate of 95.14% (distinguishing from other possible events), and it identifies people with one or two steps in a 97.22% (higher accuracy than other methods that use more footsteps). The fall down location error is smaller than 0.27 m, which is acceptable compared with the height of a person.**

***Index Terms*—Fall detection, seismic sensing, person identification, real-time, in-network system.**

I. Introduction

ELDERY assisted living systems represent an innovative smart technology useful for detecting emergencies. Person identification and fall detection are essential components in smart assisted systems that can help to improve people lives. According to the *Center for Disease Control and Prevention*, one out of five falls causes a serious injury such as broken bones or a head injury [1]. It is clear then, fall detection at home is a problem that must be addressed in a prompt manner. Furthermore, monitoring the person (by identification) is vital to diminish emergencies. Even though some solutions for elderly monitoring are available on the market, most of them are either body-contact equipment [2] or privacy-invasive devices (like cameras or similar) [3].

Manuscript received November 1, 2018; revised March 15, 2019; accepted March 18, 2019. Date of publication April 1, 2019; date of current version February 6, 2020. *(Corresponding author: Jose Clemente.)*

The authors are with the Center for Cyber-Physical Systems, University of Georgia, Athens, GA 30602 USA (e-mail: jclementes@uga.edu; fangyu.li@uga.edu; maria.valero@uga.edu; wsong@uga.edu).

Digital Object Identifier 10.1109/JBHI.2019.2907498

Fig. 1. System detection and location.

Sensor networks play a key role in fall detection and person identification. Privacy and comfort are two important issues to incorporate in the solution. In this paper, we propose a novel *real-time* fall detection, location and notification system that uses floor seismic data produced by fall downs as an only source for recognition. The system is based on the analysis of structural vibration where sensors sense floor vibrations, perform in-situ signal processing, detect fall downs and identify a person. Because the main source of data comes from floor vibrations, the sensors are installed on the floor (for example, in the corners of a room), and there are no cameras or other privacy-invasive devices. Also, our approach is able to discriminate other events in the room and differentiate when it is related to a fall down or a footstep. The system is able to identify a person almost immediately with only one or two footsteps.

The contributions of our system are summarized as follows: (i) A real-time system for recognizing people and their fall downs using structural vibration. Compared with existing works, which requires post-processing of the data, we are presenting an in-network system that use machine learning models in real time to provide information (identification and fall detection) in the same moment that it occurs. Real-time recognition is accomplished by using a computer unit that executes the algorithms inside each sensor and communication. (ii) A person identification method that uses optimized features selection and allows the identification of a person in one or two steps. Previous works required at least five or more steps for accurate recognition. (iii) A real-time and in-network location method, for locating footsteps and fall downs, that does need previous knowledge of the floor for estimations. We include a simple and novel calibration method for floor velocity estimation, which means that unlike other works, we eliminate the assumption of equal velocity in all parts of the floor. (iv) An in-network voting system to improve the accuracy of person identification. A sink

2168-2194 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.

sensor among the ones that detect an event - like a footstep, a fall down, etc. - is dynamically selected, and it estimates the voting weights according to the sensed level of energy broadcasted by each node.

II. Related Works

Regarding **fall down detection**, [4] presented a footfall detection method using seismic signals, which is based on unsupervised learning detection. Authors cluster data to separate noise and footfall events from an unlabeled dataset using Gaussian mixture models (GMM). GMM assumes that the features of both clusters follow a multivariate Gaussian distribution. Even though they were able to detect steps, their work does not identify people according to the footstep. Alwan *et al.* [5] also presented a fall detection method based on floor vibration. The works report 100% of accuracy; however, the main analyzed feature is the amplitude of the signal, and there is not an explicit comparison with other human activities that have similar amplitude (example an object drop, jumping, etc.) Respecting **person identification** via footsteps using vibration, [6] initially presents an in-door person identification method where the system detects signals induced by footsteps, extracts signal features, and applies a hierarchical classifier to identify each registered user. The system takes features from different peoples traces to generate a classification model using Support Vector Machine (SVM), which maximizes the distance between data points and the separating hyperplane. Later, the same group introduced an indoor pedestrian identification through ambient structural vibration [7]. The authors used an Iterative Transductive Learning Algorithm (ITSVM) instead of SVM due to the higher identification accuracy on the low step frequency data. They used a database with step frequencies to increase the accuracy from 90% to 96%. Also, a threshold discards half of the data to get high of accuracy, which means that more than 4 steps are needed to identify a person. Even though ITSVM improves accuracy, the computational time for training and testing is higher than SVM. Concerning **person localization**, Poston *et al.* [8] presented an indoor footstep location method using accelerometers. They analyzed the influence a building's structure has on footsteps sensed by embedded sensors. In this work, the algorithms operate with templates of footstep vibrations and rely on prior information about sensor coordinates that comes from a building's post-construction; the velocity of the floor is assumed constant at each sensor depending on the building material. In 2018, Mirshekari *et al.* [9] proposed an occupancy localization method where there is no need of knowing the floor velocity for estimating the footstep location; they use a heuristic method to estimate the velocity and location by assuming the location is inside some limits. They also use cross-correlation for calculating the arrival time of the events; however, it is well-known that the arrival time is more prone to problems due to ambient noise interference.

Our approach improves the aforementioned work by not only identifying people based on the footstep analysis, but also detecting when a person falls down, in which location the fall down occurs, and which person falls down. Also, we do not assume constant velocity in floor. We want to emphasize that because our system has been designed for in-situ processing, we use sensors with a sampling rate of 1000 Hz as it is lighter for embedded systems and IoT devices. Even though our sampling rate is smaller than previous work (most of then uses sampling rates of 20 KHz or higher), our approach is more accurate and faster as we do not require post-processing.

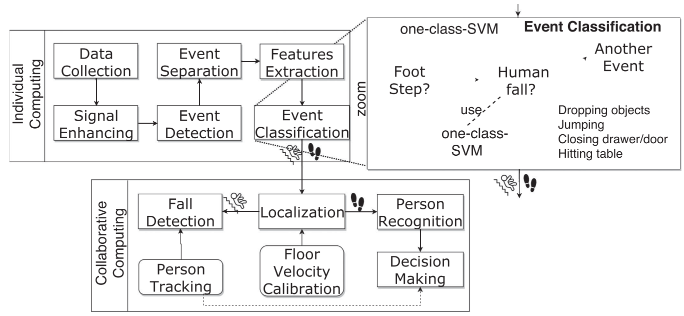


Fig. 2. System Architecture. In event classification workflow in the system two one-class SVM are used to identify first footsteps. Otherwise, another SVM is used to classify the event as either fall down or other events.

III. System Design

The system is designed based on the floor vibration measured by smart seismic sensors attached to the floor. Our seismic sensors are able to measure biometric signatures, like fall downs and/or footsteps, and collaborate among them to locate the fall down place without the need of extra sensor devices like cameras. The system architecture is presented in Fig. 2. Every sensor performs the steps described in the architecture and communicates each other to detect the fall down place in the "localization" step. In this section, we present the details of each step and the logical structure of our system.

A. Data Collection

The vibration measured from the structure is gathered by smart seismic sensors [10]. The detail description of the sensors is specified in Section IV-A. The three-channel seismometer used in our sensors has a sampling rate of 1000 Hz which is wide enough to detect both low and high-frequency components. The data are recorded in an internal database for further analysis if needed; however, the fall detection, footstep recognition, and fall location are done in real time. Then, the sensors in the network need to be synchronized. The synchronization is done via a GPS inside each smart unit. The GPS synchronization is done only the first time the system runs.

B. Signal Enhancing

In any in-door place, there exists background noise due to constant building shaking. This background noise tends to obscure the different events we need to detect, especially the footsteps. To enhance the signal, we attempt to remove the background noise by applying a wavelet denoising technique on the data to suppress non-stationary noises. The denoised signal using

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.

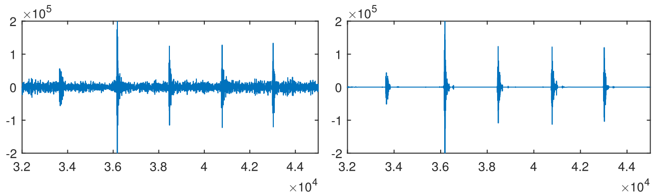


Fig. 3. Signal before (left) and after (right) enhancing process.

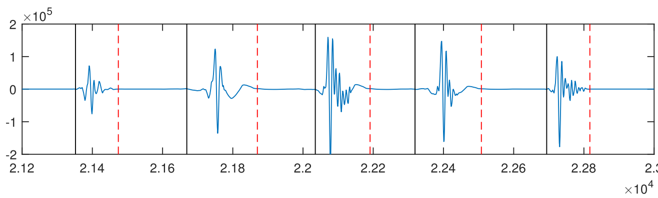


Fig. 4. Example of our event isolation method based on first-order second-moment. Solid lines represent the beginnings of events and dotted lines denote the end of events.

wavelet thresholding can be expressed as [11] $\mathbf{s}_d = \sum_{j,k \in K} \tau(\int s(x)\psi_{j,k}(x)dx)\psi_{j,k}$, where, $(\psi_{j,k})_{j,k\in K}$ denotes the orthogonal basis of wavelets, $\tau(\cdot)$ is the thresholding operator. The enhanced signal in the time domain is shown in Fig. 3. The noise has been removed from the recorded data, and it is suitable for feature extraction.

C. Event Detection and Separation

Once the signal has been enhanced, an event detection technique is applied to separate the events. Fall downs and footsteps are considered events. We apply a signal isolation technique to extract the pulses of footsteps and fall down vibration [12]. This process is important for future determination whether an event represents a fall down, a footstep, or something else. Our isolation is different from those used in other works. For example, in [13], the event detection is based only on the time of arrival (ToA); the event end time is not considered. We implement an on-line changing point detection algorithm to find abrupt variations in data to pick the event. It is based on a probabilistic method - first-order second-moment method - to determine the stochastic moments of a signal. The first-order second-moment method is defined as $\mathbf{m}_2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$, where μ is the mean of the distribution, and N is the windows size of the data to be processed. We use a 50 milliseconds window to find changing points with low latency. An event starts when the moment m_2 is larger than a threshold. We set up the threshold at three times of the standard deviations of the background noise, which is updated constantly with data that does not come from an event. An event culminates when m_2 is below 1.5 times of the standard deviation. Fig. 4 illustrates an event detection and separation example by applying the proposed method.

D. Feature Extraction

The feature extraction procedure is essential for fall down and footstep characterization. As reported in [14], it is possible to distinguish a human footstep vibration signal from background noises by selecting the correct features. Different approaches have been adopted in literature for extracting a set of features that help to obtain signatures of footsteps. For example, features extracted from mean vectors and covariance matrices of spectral coefficients were used in [15]; other works use short-term energy and footstep intervals [16], as well as gait-based information as main features for footsteps representation [17]. Because the vibration and sound pressure responses of human footsteps in buildings can be broadband and frequency-dependent [18], and different vibration signatures from different walking styles can be studied [19], we compute features in both, time and frequency domain. Before obtaining the features, we normalize the signal events to eliminate the distance effect between the sensor location and the event location. The normalization is done by dividing the event by the energy average of its signal, where the energy is defined as $\mathcal{E} = \sum_{n=0}^{N}|x(n)|^2$. We also analyze the duration of the events and the number of peaks in the time of data of detection. These features help to improve the score of the people and detect falls in the studied area. An optimal selection of the features is also adopted to improve the detection, which is explained in Section III-H1. Table I shows the characteristics extracted from the events and the labels used as abbreviations. Figure 5 shows how to calculate the extracted features in time and frequency domain.

TABLE I
EVENT FEATURES

Domain	Label	Feature Name	Label	Feature Name
Time	F1	Event duration	F5	Maximum peak
	F2	Standard deviation	F6	Location of maximum peak
	F3	Entropy	F7	Five values before maximum peak
	F4	First five peak values	F8	Five values after maximum peak
Frequency	F9	Spectra	F12	Location of first five peaks
	F10	Centroid	F13	Number of peaks
	F11	First five peak values		

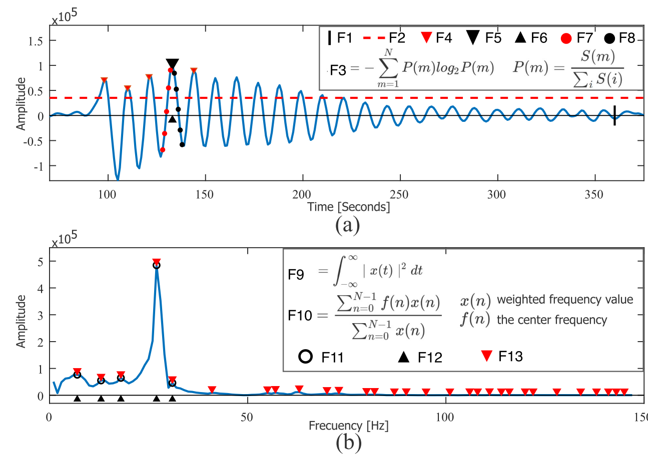
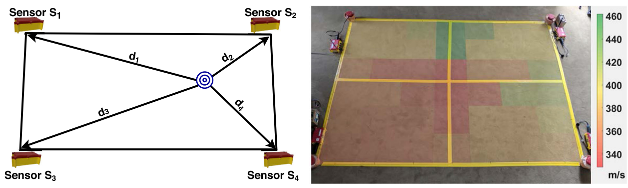


Fig. 5. Representation of the extracted features. (a) Time domain features. (b) Frequency domain features.

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 6. Seismometer-based location system. (a) Seismic sensors cooperatively calculate the event location. (b) Initial velocity map for calibration purposes of our system.

E. Classification

Precise classification is the main point of our system because it works based on two types of events; human fall and footsteps. We use falls to calculate the location and generate an emergency, and footsteps to identify a person. However, there are many events in daily life that generate vibrations that are sensed by our geophones. Many of these events have similar characteristics which must be selected and classified in order to get human falls and footsteps. For example, a jump can generate an event that has an amplitude in time domain similar to that of a fall, but these can differentiate using the duration time and the number of peaks in time domain (both are greater in fall events). Similarly, events such as closing a door, or falling an object have characteristics like human falls. Other events such as hitting a table or closing a drawer have similarities with the steps of a person.

We implemented our classification process using a supervised machine learning algorithm SVM, which transforms the data using a kernel to find the optimal boundary between the different classes. The event classification work-flow is presented in Fig. 2. The inputs are the features extracted from each detected event. We are using two independent classifiers that are placed in two layers. The first classifier is to identify if the event is a footstep. Otherwise, the features are sent to the second layer to identify if it is a human fall. The event is discarded if it does not belong to either of these two classes. The footstep classifier is located at the beginning because it is the most common event present in our system where more than 90% of the events are footsteps.

Both classifiers were implemented using one-class-SVM to be able to discard events of which there is no data for their training. For example, an earthquake is an uncommon event that can be detected by our system and because it is not a footstep, or a human fall is discarded by the classifier.

F. Location

The two types of events, fall and footstep, are located by our system in real-time. The fall location is used to send an alarm with the position where the event is presented, and the footstep location is implemented for person identification and tracking. It allows identifying the person who fell based on the location of previous steps. Also, it reduces false positives that may appear far from the person's location. Fig. 6(a) shows our seismometer-based location system.

Location techniques require prior knowledge of the propagation speed and the seismometer locations. They also need several units to detect the event location. We incorporate four units to reduce the location error margin. In addition, all the seismometers should be synchronized in time. Since the GPS accuracy is lower inside a building and the seismometers are placed too close for the satellite positioning system, we need first manually setup the relative sensor locations. Then, in the defined coordinates, a jump synchronization process is applied to initialize the wave propagation velocity model of the floor. The process involves jumping twice next to each unit. Then, all sensors broadcast the arrival time of the event, and the sensor with the minimum arrival time calculates the speed having the assumption that the jump was made in its location. After eight events the sensors generate the floor velocity model. Our calibration technique allows the estimation of the floor velocity and eliminates the assumption of constant velocity that previous works use. The calibration result of our floor is shown in Fig. 6(b).

We estimate the event location using the time difference of arrivals (TDOA) between all time-synchronized stationary sensors $\mathbf{s_m} = [x_m, y_m]^T$, $m = 1, \ldots, M$ to locate where the event $\mathbf{e} = (x, y)^T \in \mathbb{R}^2$ is generated (source). Every single node determines the event arrival time. $\tau_m = t_0 + \frac{d_m}{v}$ is the TDOA for node $\mathbf{m}$, and it is composed for the time when the event starts $\mathbf{t_0}$ and the transmission time between the source and sensor $\mathbf{m}$. $d_m = \| \mathbf{e} - \mathbf{s_m} \| = \sqrt{(x - x_m)^2 + (y - y_m)^2}$, $m = 1, \ldots, M$, denotes the Euclidean distance between the source and sensor m, and v is a constant speed.

We propose to use an optimization method to estimate a two-dimensional position vector. We eliminate the assumption of constant speed [20] due to the propagation velocity value is not constant. V is the propagation velocity matrix obtained from the calibration process. We use Maximum Likelihood (ML) estimator to calculate the position vector $\mathbf{e}'$ which maximizes the likelihood function or rather minimizes:

$$t' = \underset{t}{\operatorname{argmin}} \sum_{i=1}^{M} \| \widetilde{\tau}_i - \tau_i \|, \quad M > 2, \tag{1}$$

Subject to:

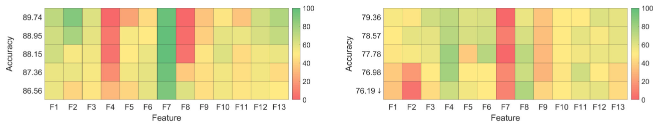
$$\min_{sx} \leq x \leq \max_{sx} \;\; sx \in \mathbf{s_m}[x_m] \; \min_{sy} \leq y \leq \max_{sy} \;\; sy \in \mathbf{s_m}[y_m]$$

We use Nelder and Mead as numerical iterative search in a multidimensional space [21]. The previous event location (step or fall) is set as the initial points of the algorithm, and if there is no previous event registered the points are generated randomly next to the unit with the minimum event arrival time. Based on our experiments we used a threshold of 1.5 milliseconds as a stopping criterion.

G. Fall Detection

When the recognized event has been classified as a fall down (see Section III-E), a system alert is activated to verify if the event was a real fall down or it is a false positive. The method used for doing this fall down recognition is described as follows: (i) a fall down is detected by the classifier; (ii) the area in a radius of 1.5 meters around the location of the last recognized step is analyzing; (iii) if the location of the event falls into

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 7. Feature selection test heat map. (a) Five best accuracy values of the test , (b) Five worst accuracy values; the last value 76.19↓ stands for all accuracies below 76.19%.

this area, the system ensure that the recognized event is, in fact, a fall down. This mechanism improves the accuracy of the system by eliminating other events that can generate false positive recognition such as: dropping an object, closing a door or drawer, hitting a table.

In addition, when the fall down is recognized, an alert is sent to a smart home device. We use GOOGLE assistant library for connecting our nodes with this smart device (GOOGLE HOME). Using this API, the unit sends an internal message to the smart device that provides an instruction to be completed.

H. Person Recognition

We implement a multi-class-SVM [6] as a supervised classification algorithm because it provides two main advantages. First, it has kernels to classify not linearly separable classes. After performing several tests with the features from the floor vibrations induced by the footsteps, we determine that our classes are not linearly separable. Second, although SVM is computationally expensive for training, its search is fast enough to be implemented on a small device as a single computer board. The training is done previously in a machine with good computational resource. The multi-class-SVM fits perfectly in the requirements for our person recognition method.

1) Feature Selection: As shown in Section III-D, we extract thirteen (13) features from our signals that can be used for training our multi-class-SVM. Eleven (11) of these features were used by the state of the art approaches [6], and two (2) were identified by the authors after analyzing the raw data behavior: F1 (event time duration) and F13 (number of peaks in the frequency domain). We noticed that some of these features do not contribute to the classification model to recognize people with good accuracy. For that reason, we implement a numerical analysis procedure to determine which features contribute or not to the model. To do this optimization, we cannot apply linear regression because our kernel is not linear. Instead, we perform the following procedure: (i) We use data from 4 subjects to train the model to avoid over-fitting; to test the model we use new data from these 4 subjects and we add 2 additional subjects. (ii) We train and test the model multiple times with different combinations of the features. Our combinations go from C_5^{13}, C_6^{13}, C_7^{13} ... to C_{13}^{13} where C_n^x means all the combinations of x features respecting the total n features. (iii) With the accuracy of each combination, we construct a matrix with x-axis referred to features and y-axis referred to accuracy value. Accuracy heat maps are shown in Fig. 7, colors of which represent the contribution proportion of the features to obtain the related accuracy. Green color means major contribution, and red color refers to features that do not contribute to a specific accuracy value. Fig. 7(a) shows the top 5 accuracy obtained in the test. It is clear that features F4 and F8 do not contribute in a large proportion to obtain these values. While the two features included, F1 and F13, add information to the SVM model in more than 80% of the cases. In contrast, Fig. 7(b) shows the lower accuracy obtained. In the same way, features F4 and F8 are contributing in more than 70% of cases to obtain these low accuracy. The remark of this optimization test is to remove characteristics F4 and F8 and keep the new two features (F1 and F13). We improve the more than 5% accuracy by the aforementioned procedure.

2) Decision Making: The four units used to locate the event work together to recognize people through the floor vibrations induced by footsteps. The energy signal is different in each unit because the signal degrades during transmission. Normally, the unit closest to the event source is the unit with the highest energy event. The signal degradation causes that some events do not have the correct features for recognizing a person. It produces false positives reducing the system accuracy. For this reason, we make a weighted collaborative decision making or weighted voting system (WVS). The weights are based on the energy event, and the amount of units who detect the event. This method adapts dynamically to any number of units incorporated into the system. After the individual person recognition, each unit broadcasts the event energy and the person id number in the tuple $\Phi_i = [\mathcal{E}_i, \mathcal{P}_i]$ where $i = 1, \ldots, M$.

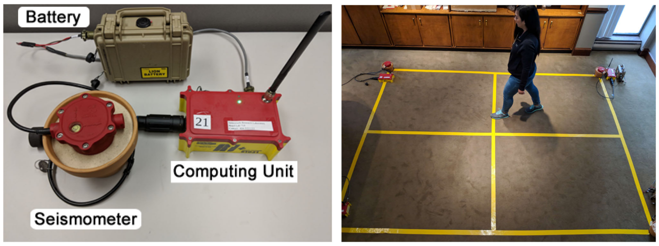
Just the unit with the highest event energy calculates and identifies the person. We use a weighted voting decision making, in which the weight is calculated as $\omega_i = \frac{\mathcal{E}_i}{\sum_{j=1}^{M} \mathcal{E}_j}$.

The selection is then calculated based on the weights obtained by each identified person represented as a subset $H \subset \mathcal{P}$ without duplication. $\sum_{j=1}^{|H|} \sum_{i=1}^{M} \mathcal{S}_j = \omega_i \quad \forall H_j = \mathcal{P}_i$ where $|H|$ is the size of the subset H (e.g. if there are 5 people identified by different unit in the process; then $N = 5$). The selected person is H_x where x is the index of the maximum value in $\mathcal{S}$. Due to the algorithm needs two footsteps to make the final selection, the ids and weights of the two people with greater value in S are broadcasted. This information is used for the next sink unit to make the final selection. It is done by accumulating the voting values of the two iterations and selecting the greatest. After the recognition, the unit communicates the person name to the smart assistant which is in charge of issuing a greeting message, e.g. "Hello Adam".

IV. EXPERIMENTS AND VALIDATION

In this section, we describe the details of the experiments conducted with our in-network footstep and fall detection system. Instead of using simulated data, we use real devices and a seismic testbed to validate our methods. First, we present the hardware for testing our system, and then, the results of each step (fall detection, fall location, person identification) are presented. A discussion of the results is also provided at the end of the section.

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 8. Hardware and testbed. (a) Sensor unit hardware description. Visible at the picture: a battery, a seismometer and a boxed pervasive computing unit. (b) Testbed for the fall down/footstep recognition system.

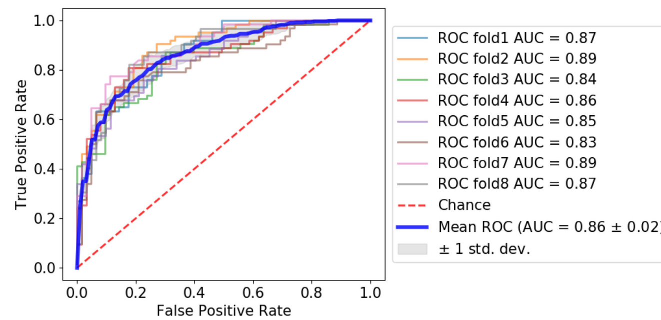


Fig. 9. Receiver Operating Characteristic (ROC) of the cross-validation procedure using 8 folds.

A. Hardware and Software Setup

We use real sensor units for our experiments. Each unit has a three-channel seismometer, a single computer board (Raspberry Pi 3) [22], a lithium battery as shown in Fig. 8(a). The three-channel seismometer detects the velocity of ground movements. Each channel records its own data with respect to its axis X, Y, and Z, which correspond to directions North, East, and Depth. The single-board computer (Raspberry Pi) collects, stores, processes and communicates data. The nodes and the smart device are integrated into a mesh network for communication purposes.

Originally, the hardware was conceived for outdoor experiments [23], [24]; however, its characteristic makes it suitable for our indoor purposes. Currently, we are working toward the design of smaller hardware with the same features for indoor purposes.

The software inside each unit is developed using PYTHON [25]. We save data inside each unit using INFLUXDB [26] database that is suitable for time series monitoring and analytics. The data also is synchronized to a central server for visualization purposes. Results of arrival time and event location (fall downs and footsteps) are also saved using INFLUXDB. The visualization is done using GRAFANA [27]. On the other hand, for SVM purposes, we use the library SCIKIT-LEARN [28]. Two types of SVM are used: (i) one-class SVM to distinguish between footsteps/fall downs and other events, and (ii) a multi-class SVM to differentiate different people. The SVM uses a 2nd degree poly kernel with a penalty parameter C of the error term established in 100.

Our seismic testbed consists of four sensors deployed in an area of 380 square feet approximately.

We design a Graphic User Interface (GUI) to visualize the fall down and footstep location. Fig. 8(b) shows an overview of our testbed and the four units deployed on the corners.

B. Validation

The k-fold cross-validation procedure is applied to evaluate the classification model. We use it to estimate how the model is expected to perform in general when it makes predictions on data not used during the training of the model. The procedure splits the data into k groups and for each group, it (i) takes one as a test data set (ii) takes the remainder as a training data set (iii) fits and evaluates the model (iv) retains the evaluation score. We select 8-folds to cross-validate the model. We use AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) to check and visualize the performance of our classification problem. ROC is a probability curve, and AUC represents the degree or measure of separability. They tell how much model is capable of distinguishing between classes. An excellent model has AUC near to the 1, which means it has a good measure of separability. Fig. 9 shows the AUC for all folds and the AUC average of 0.86.

TABLE II
CLASSIFIERS COMPARISON FOOTSTEPS OF 6 PEOPLE. FIVE RASPBERRY PIS WERE USED TO OBTAIN THE AVERAGE TIMES

	Re	Pr	Ac	f1	Avg. Training Time (s)	Avg Test Time (s)
GP	0.638	0.640	0.842	0.639	4.702	1.937
KNN	0.560	0.567	0.792	0.564	1.494	8.265
SVM	0.737	0.742	0.912	0.739	3.271	0.289

C. Classifiers Comparison

To choose the correct classifier that obtains high levels of accuracy and to have the ability to adapt to the limited computing resources of the single board, we evaluate the accuracy and runtime of three different classifiers using 100 steps of 6 subjects. The classifiers used in the process are Gaussian Process (GP), K-Nearest Neighbors (KNN) and SVM. The data is split on 80 percent for training and 20 percent for testing. The process is conducted on 5 different Raspberry Pis for each classifier. The results are shown in Table II. All classifiers have an acceptable level of accuracy for a person recognition using a single step. However, SVM shows an accuracy greater than 90 percent, and more than 10 and 5 percent compared with KNN and GP respectably. The training times are acceptable for all of them because they will be executed just one time. This time increases if more classes are added. Finally, SVM outperforms the classification run time. The run time increases 6.7X times when GP classifier is used, and 28.6X times using KNN. Although SVM performs better than the other classifier, to achieve the final accuracy shown in our results, the system needs two footsteps and the weighted voting system proposed.

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.

TABLE III
CONFUSION MATRIX OF EVENT CLASSIFICATION

Ground Truth			Evaluation			
First SVM	Footstep	Non	Re	Pr	Ac	f1
Footstep	417	23	**0.948**	**0.893**	**0.924**	**0.920**
Non	50	464				
Second SVM	Fall	Other	Re	Pr	Ac	f1
Fall	43	7	**0.860**	**0.246**	**0.730**	**0.382**
Other	132	332				

TABLE IV
CONFUSION MATRIX OF PERSON IDENTIFICATION USING **one** FOOTSTEP. SUBJECTS ARE NAMED WITH LETTERS FROM A TO F

Ground Truth							Evaluation			
	A	B	C	D	E	F	Re	Pr	Ac	f1
A	75	4	0	4	0	17	0.750	0.815	0.930	0.781
B	17	65	0	5	0	13	0.650	0.575	0.862	0.610
C	0	0	81	0	19	0	0.810	0.764	0.923	0.786
D	0	11	8	76	3	2	0.760	0.844	0.937	0.800
E	0	0	17	0	83	0	0.830	0.791	0.935	0.810
F	0	33	0	5	0	62	0.620	0.660	0.883	0.639
Average							**0.737**	**0.742**	**0.912**	**0.739**

D. Event Classification

We measure the accuracy of our SVM classifier to distinguish between a footstep, a fall down and other events. Table III presents the results in a confusion matrix. It shows four classification metrics. The first metric gives the proportion of actual positive events that are correctly identified as positives by the classifier called Recall ($Re = TP/(TP + FN)$).[1] Precision, that is the second metric, reflects the proportion of events classified as positive that are actually positive ($Pr = TP/(TP + FP)$). The third metric, accuracy, is number of correct predictions divided by the total number of predictions ($Ac = (TN + TP)/(TP + TN + FP + FN)$). The last metric called f1 is used to seek the balance between recall and precision ($f1 = 2((Pr * Re)/(Pr + Re))$). The first SVM differentiates between the footsteps and other events with an accuracy of 92.35% and a precision of 89.30%. The events that are not recognized as footsteps are used for the second SVM to detect fall downs with and accuracy of 73%. The precision obtained is low due to the amount of fall downs events applied for training. However, it is improved by using the acceptance area of 1.5 meters from the last footstep detected. These results are shown on Section IV-H. Note that the percentage of recognition of footsteps is very high due to the training set contains much more footsteps events than other types of events.

E. Person Identification Accuracy

To measure the accuracy of our person identification method, we test our system with six people, who are represented by letters from A to F. For this test we used just one unit in an area of 2.5 × 3.5 meters. The first confusion matrix (Table IV) shows the recall, precision and accuracy measurements of the person identification method using only one footstep. As it is shown, the accuracy is 91.23%, which is somehow acceptable.

[1]TP = True Positive. FP = False Positive. TN = True Negative. FN = False Negative.

TABLE V
CONFUSION MATRIX OF PERSON IDENTIFICATION USING **two** FOOTSTEPS. SUBJECTS ARE NAMED WITH LETTERS FROM A TO F

Ground Truth							Evaluation			
	A	B	C	D	E	F	Re	Pr	Ac	f1
A	93	0	0	0	0	7	0.930	0.949	0.980	0.939
B	5	84	0	3	0	8	0.840	0.771	0.932	0.804
C	0	0	89	0	11	0	0.890	0.967	0.977	0.927
D	0	6	0	94	0	0	0.940	0.931	0.978	0.935
E	0	0	3	0	97	0	0.970	0.898	0.977	0.933
F	0	19	0	4	0	77	0.770	0.837	0.937	0.802
Average							**0.890**	**0.892**	**0.963**	**0.891**

Fig. 10. The three structural types and the sensors configuration for each of them.

However, the precision is low, only 74.16%. This precision may introduce a high number of false positives.

To improve the accuracy and precision of the person identification, we allow the system to analyzed two footsteps instead of just one. The process is the same explained in the workflow, but in the voting decision-making step, two steps are counted for voting, which means that two tuples are sending for decision ($\Phi_i(t)$ and $\Phi_i(t-1)$). Table V shows the result using two footsteps. Note that the accuracy was improved in more than 5% and the precision was increased to 89.21%, which significantly reduce the number of false alarms.

A detailed analysis of the results and a comparison with the physical characteristics of the subjects in the study show that the false positives we still have are due to similarities in the physical features of the subjects. For instance, subject B and F are both men with almost the same height and an approximate same weight. Those subjects are easy to misidentify. However, footstep features are different enough to get an acceptable accuracy and precision measurements.

F. Decision Making Based on Weighted Voting System

We compare our weight based voting decision making with a simple majority based voting system. Using our weighted method, accuracy, precision and most importantly recall are improved. Table VI shows the results of both methods. Note that

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.

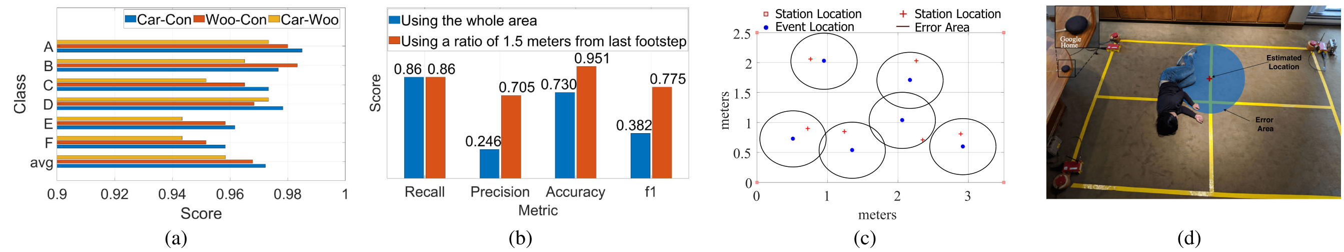


Fig. 11. (a) Accuracy obtained in different types of floors. (b) Comparison of recall, precision, and accuracy using all detected events and the events in the area of 1.5 m radius from the last step. (c) Six different events location and the comparison between real event location and estimated location obtained with our method. Black circles represent the error area. (d) Fall down location and exemplification of the person inside the error area. Smart device alerts the fall detected.

TABLE VI
COMPARISON OF PERSON IDENTIFICATION USING THE WEIGHTED VOTING SYSTEM (WVS) AND SINGLE VOTING (SV)

	Re	Pr	Ac	f1	No Possible Decision
1 Footstep SV	0.583	0.686	0.851	0.630	13.8 %
1 Footstep WVS	0.735	0.704	0.906	0.719	0 %
2 Footsteps SV	0.792	0.881	0.947	0.891	7.7 %
2 Footsteps WVS	0.917	0.923	0.972	0.920	0 %

with a single voting decision making there are some no possible decisions since we are using four units in the experiment, for example, when two units identify the person as subject A, and the other two as subject B. With our weighted voting, we ensure a decision even in these scenarios.

G. Robustness Regarding Floor Types and Sensor-Footstep Distance

To measure the system robustness, we evaluated the performance in different types of floors and placing the sensor at different distances. First, we measure the accuracy of the system in three different types of floors and 6 participants. The three floors types and their configurations are shown in Fig. 10. The first structure is 8.75 square meters of the carpeted-concrete floor from our lab testbed. The second structure is 60.5 square meters of the wooden-concrete floor from a living room of a residential house. The third structure corresponds to 18 square meters of carpeted wooden floor from a second-floor of a residential house. The accuracy results are shown in Fig. 11(a). The best classification accuracy is observed in the carpeted-concrete floor, and the lowest was obtained on the carpeted-wooden. We realized that it is due to the carpeted-wooden introduces more noise. In overall, a good performance was obtained in all types of structures with an accuracy average greater than 95%.

To evaluate the influence of sensors distance in the performance, we also carry out a sensitivity test on the three types of floors to identify the maximum distance between sensors where the accuracy is acceptable. The results are shown in Table VII. The sensors were able to detect events in all floors in distances between 0 and 4 meters. In larger distances, some few events were not be recognized by the system. The greater the distance, the greater the number of events that are not recognized. The maximum distance between two sensors to avoid losing any footsteps was 8 meters. On the other hand, the signal quality to get the features for the classifier also depends on the distance between the steps and the sensors, but the proposed weighted voting system gives more weight to the events with greater accumulated energy. Then, we conclude the system is able to identify people on different floors with an accuracy greater than 95% when the sensors are located at distances up to 11 meters. These results demonstrate that the system is robust in different types of floors and with sensors at different distances.

TABLE VII
PERCENTAGE OF EVENTS DETECTED IN DIFFERENT TYPES OF STRUCTURES AT DIFFERENT DISTANCES IN METERS

	0 to 2	2 to 4	4 to 6	6 to 8	8 to 10	11+
Car-Con	100%	100%	-	-	-	-
Woo-Con	100%	100%	98.8%	95.7%	91.5%	88.2%
Car-Woo	100%	100%	99.6%	-	-	-

TABLE VIII
COMPARISON OF EVENT CLASSIFICATION NUMBER AS FALL DOWN AND/OR OTHER EVENTS. RESULTS ARE SHOWN USING THE WHOLE AREA VS. A 1.5 M RADIUS FROM THE LAST RECOGNIZED FOOTSTEP

	Using the whole area		Using a 1.5 m radius from the last footstep	
	Fall	Other	Fall	Other
Fall	43	7	43	7
Jumping	41	57	8	90
Dropping object	46	64	8	102
Closing drawer	25	69	2	92
Closing door	9	59	0	68
Hitting table	11	83	0	94

H. Accuracy of Fall Down Detection and Location

The second one-class SVM is executed once an event has been detected as other class different from footstep. If the SVM is used without taking into account the person who is walking (using the whole area of study) the number of events misidentified is very high. However, when we introduce the method explained in Section III-G, the number of misidentified events is reduced drastically. The method consists of analyzing a radius of 1.5 meters from the last recognized footstep. Results are shown in Table VIII. The Fig. 11(b) shows the improvements in terms of recall, precision, and accuracy once the 1.5 m radius area is applied.

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.

To test our fall down location method, we use 31 different fall down events inside our testbed to measure the error between the real and estimated location from our method. The average error is 0.27 meters with an standard deviation of 0.15 meters. An example of six different events is shown in Fig. 11(c). Note that our estimated location is near to the event location and inside of the calculated error area. The error area is calculated according to our sampling rate (1000 Hz) and our initial velocity model. The error area is approximately 0.47 meters due to the sampling rate and the average of the velocity model (470 m/s). All estimated events are inside this area. This estimated area is acceptable since when a person falls down, his body occupies an area of at least 1.5 meters. This means that it will be always a part of the body inside the error area.

An example of the described situation can be found in Fig. 11(d). The four units are located at corners. A 1.65 meters tall person falls inside the sensing area. The estimated location of the fall down (red +) is reported to be near to the real fall down location. Hence, the method has an acceptable accuracy respecting the height and proportions of the person.

V. Conclusion

In this paper, we introduced a novel real-time non-intrusive fall detection and person identification system that can be used in smart homes for many purposes (e.g. elder assisted living). The system is based on the floor vibration generated by fall down and footsteps. After data collection with sensor-smart devices and signal enhanced, we performed a feature extraction technique that characterizes fall downs and footsteps. The feature selection was improved with an optimization technique that indicates which features contribute more to the accuracy of our method. We use different types of SVM to classify the features and recognize the person who is walking. All these steps are performed in-network and in real time. A in-network location and calibration method is proposed to localize the footsteps and most importantly the fall downs. The method includes communication and independent cooperation between sensor nodes. A voting decision making is then performed to accurately identify people. Finally, an integration with commercial smart home devices is provided to generate system alerts when it is needed. Experimental results in real seismic testbeds are presented. To the best of our knowledge, we are presenting for the first time an in-network fall down detection and location with structural vibration data. We show our method outperformed other existing methods that need to use more footsteps sensing to recognize people.

References

[1] D. A. Sterling, J. A. O'connor, and J. Bonadies, "Geriatric falls: Injury severity is high and disproportionate to mechanism," *J. Trauma Acute Care Surg.*, vol. 50, no. 1, pp. 116–119, 2001.

[2] G. Diraco, A. Leone, and P. Siciliano, "A radar-based smart sensor for unobtrusive elderly monitoring in ambient assisted living applications," *Biosensors*, vol. 7, 2017, Art. no. 55.

[3] M. Kepski and B. Kwolek, "Fall detection on embedded platform using kinect and wireless accelerometer," in *Proc. Int. Conf. Comput. Handicapped Persons*, 2012, pp. 407–414.

[4] S. Anchal, B. Mukhopadhyay, and S. Kar, "UREDT: Unsupervised learning based real-time footfall event detection technique in seismic signal," *IEEE Sensors Lett.*, vol. 2, no. 1, pp. 1–4, Mar. 2018.

[5] M. Alwan *et al.*, "A smart and passive floor-vibration based fall detector for elderly," in *Proc. Inf. Commun. Technol.*, 2006, vol. 1, pp. 1003–1007.

[6] S. Pan, N. Wang, Y. Qian, I. Velibeyoglu, H. Y. Noh, and P. Zhang, "Indoor person identification through footstep induced structural vibration," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl.*, 2015, pp. 81–86.

[7] S. Pan *et al.*, "FootprintID: Indoor pedestrian identification through ambient structural vibration sensing," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, 2017, Art. no. 89.

[8] J. D. Poston, R. M. Buehrer, and P. A. Tarazaga, "Indoor footstep localization from structural dynamics instrumentation," *Mech. Syst. Signal Process.*, vol. 88, pp. 224–239, 2017.

[9] M. Mirshekari, S. Pan, J. Fagert, E. M. Schooler, P. Zhang, and H. Y. Noh, "Occupant localization using footstep-induced structural vibration," *Mech. Syst. Signal Process.*, vol. 112, pp. 77–97, 2018.

[10] F. Li and W. Song, "Automatic arrival identification system for real-time microseismic event location," in *Proc. SEG Tech. Program Expanded Abstracts*, 2017, pp. 2934–2939. [Online]. Available: http://dx.doi.org/10.1190/segam2017-17667176.1

[11] S. Durand and J. Froment, "Artifact free signal denoising with wavelets," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 6, pp. 3685–3688.

[12] F. Li, J. Rich, K. J. Marfurt, and H. Zhou, "Automatic event detection on noisy microseismograms," in *Proc. SEG Tech. Program Expanded Abstracts*, 2014, pp. 2363–2367.

[13] S. Pan, A. Bonde, J. Jing, L. Zhang, P. Zhang, and H. Y. Noh, "Boes: Building occupancy estimation system using sparse ambient vibration monitoring," in *Proc. Sensors Smart Struct. Technol. Civil, Mech., Aerosp. Syst.*, 2014, vol. 9061.

[14] P. Anghelescu, G. V. Iana, and I. Tramandan, "Human footstep detection using seismic sensors," in *Proc. 7th Int. Conf. Electron., Comput. Artif. Intell.*, 2015, pp. AE-1–AE-2.

[15] U. O. Ofoegbu, A. N. Iyer, R. E. Yantorno, and B. Y. Smolenski, "A speaker count system for telephone conversations," in *Proc. Int. Symp. Intell. Signal Process. Commun.*, 2006, pp. 331–334.

[16] W. Chen, M. Guan, L. Wang, R. Ruby, and K. Wu, "FLoc: Device-free passive indoor localization in complex environments," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[17] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Statist. Soc.: Ser. B (Statist. Methodol.)*, vol. 63, no. 2, pp. 411–423, 2001.

[18] A. Ekimov and J. M. Sabatier, "Vibration and sound signatures of human footsteps in buildings," *J. Acoust. Soc. Am.*, vol. 118, no. 3, pp. 2021–768, 2006.

[19] J. M. Sabatier and A. E. Ekimov, "A review of human signatures in urban environments using seismic and acoustic methods," in *Proc. IEEE Conf. Technol. Homeland Security*, 2008, pp. 215–220.

[20] M. Mirshekari, S. Pan, P. Zhang, and H. Y. Noh, "Characterizing wave propagation to improve indoor step-level person localization using floor vibration," in *Proc. Sensors Smart Struct. Technol. Civil, Mech., Aerosp. Syst.*, 2016, vol. 9803.

[21] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[22] E. Upton, "Raspberry Pi 3," 2016. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b

[23] J. Clemente, F. Li, and W. Song, "Optimal data task distribution for balancing energy consumption on cooperative fog networks," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2018, pp. 693–697.

[24] M. Valero, F. Li, X. Li, and W. Song, "Imaging subsurface civil infrastructure with smart seismic network," in *Proc. 37th IEEE Int. Perform. Comput. Commun. Conf.*, 2018.

[25] Python Software Foundation, "Python," 2018. [Online]. Available: https://www.python.org/

[26] "InfluxDB." [Online]. Available: https://www.influxdata.com/

[27] Grafana Labs, "Grafana," 2018. [Online]. Available: https://grafana.com/

[28] Machine Learning in Python, "Scikit-Learn," 2018. [Online]. Available: http://scikit-learn.org/stable/

Authorized licensed use limited to: Hohai University Library. Downloaded on October 18,2023 at 03:25:06 UTC from IEEE Xplore. Restrictions apply.