sensors

MDPI

Article

Earthquake Detection in a Static and Dynamic Environment Using Supervised Machine Learning and a Novel Feature Extraction Method

Irshad Khan [1], Seonhwa Choi [2] and Young-Woo Kwon [1,*]

[1] School of Computer Science, Kyungpook National University, Daegu 41566, Korea; irshad.khaan@gmail.com
[2] National Disaster Management Research Institute, Ulsan 44538, Korea; shchoi33@korea.kr
* Correspondence: ywkwon@knu.ac.kr; Tel.: +82-53-950-7566

Received: 12 December 2019; Accepted: 30 January 2020; Published: 1 February 2020

Abstract: Detecting earthquakes using smartphones or IoT devices in real-time is an arduous and challenging task, not only because it is constrained with the hard real-time issue but also due to the similarity of earthquake signals and the non-earthquake signals (i.e., noise or other activities). Moreover, the variety of human activities also makes it more difficult when a smartphone is used as an earthquake detecting sensor. To that end, in this article, we leverage a machine learning technique with earthquake features rather than traditional seismic methods. First, we split the detection task into two categories including static environment and dynamic environment. Then, we experimentally evaluate different features and propose the most appropriate machine learning model and features for the static environment to tackle the issue of noisy components and detect earthquakes in real-time with less false alarm rates. The experimental result of the proposed model shows promising results not only on the given dataset but also on the unseen data pointing to the generalization characteristics of the model. Finally, we demonstrate that the proposed model can be also used in the dynamic environment if it is trained with different dataset.

Keywords: earthquake detection; time-series features; Internet of Things; machine learning

1. Introduction

Due to the nature of earthquakes, significant research efforts have been made to develop real-time earthquake detection systems for disaster management. Earthquake fatal levels of motion can cause fatalities and damage in populated regions [1]. Because typical human structures are unable to resist large magnitude earthquakes, possible ways to overcome such fatalities are to build earthquake-resistant buildings or to take advantage of an Earthquake Early Warning (EEW) system that provides seconds to minutes of warning in advance, thereby allowing people to move to safe areas or shut down dangerous machinery. However, it is not only costly to construct earthquake-resistant structures but also difficult to build a highly accurate nationwide EEW system.

In recent years, emerging computing technologies such as mobile computing and Internet-of-Thing (IoT) systems equipped with various MEMS (Micro Electro Mechanical Systems) sensors (e.g., accelerometers, gyroscopes, GPSs), Wi-Fi, bluetooth, etc., have been widely adopted in the following areas: smart healthcare, intelligent transportation systems, smart buildings, and earthquake early warning systems [2–5]. In particular, the MyShake project [6] leverages mobile technologies to develop an earthquake early warning system that combines a seismic method and a machine learning (ML) technology. The system is installed on a volunteer's smartphone and then detects earthquakes using an Artificial Neural Network (ANN). It is the first global earthquake detection system using a smartphone and machine learning technique.

Based on the available literature, we can divide IoT-based earthquake detection into two parts by applicability. A mobile-based earthquake early warning system uses low-cost MEMS sensors in a smartphone or an IoT device as a seismic sensor in a dynamic environment, while the stationary sensor-based early warning system uses a dedicated device as a seismic sensor in static (i.e., fixed) environment. The non-earthquake data in a static environment include internal and external noises. The source of internal noises mainly come from a sensor in which an accelerometer continuously captures some vibratory signals. The external noises come from outside of a sensor because of constructions, heavy-traffic roads, etc., near the installed sensor. In a dynamic environment, the variety of human activities become a major part of the non-earthquake data, which significantly affects the system performance, and thus the earthquake detection task using a low-cost sensor is very challenging. In this environment, training a machine learning algorithm is critical because of the activities whose frequency and amplitude patterns look like earthquakes.

In traditional earthquake early warning systems, because acceleration data recorded from seismic sensors installed nationwide are sent to a centralized server for earthquake detection, network and processing delays are inevitable. Because there are a few seconds between a P-wave and an S-wave (e.g., 10 s [7]) depending on the distance from an hypocenter [8,9], to reduce the blind area of earthquake early warning, on-site or standalone earthquake detection devices have been recently introduced [10,11,24]. However, because of the real-time processing requirement and resource constraints of a detection device, heavy computational methods and deep neural networks cannot be applied at the sensor side. Nevertheless, the final detection can be performed at the server-side through advanced detection algorithms, a simple detection algorithm with a few features that require light computations at a client-side to complete the detection procedure as soon as possible is required. Because an earthquake detection device can be operated in either a static or a dynamic environment, trivial statistical amplitude and frequency features are not suitable for such environments.

As a result, our focus is to improve a machine learning model for an earthquake alert device that we developed in our prior work to detect earthquakes in static and dynamic environments [10,11]. The device not only detects an earthquake but also sends an alert with earthquake response information to nearby smart devices such as smartphones, smart TVs, etc. As the device operates independently, without any Internet connection or collaborations with other alert devices, it needs a highly accurate earthquake detection algorithm. Because traditional methods to detect earthquakes such as STA/LTA have high false alarm rates, it is risky to use only one earthquake detection method for a standalone device. Thus, we use both traditional earthquake detection methods and emerging technologies together to decrease the chance of false alarms and increase the overall earthquake detection ability. In this article, we systematically compare different earthquake features and datasets representing static and dynamic environments for the earthquake alert device, and then, based on our experimental results, we propose a new earthquake detection model that can be used in both static and dynamic environments.

The rest of the article is structured as follows. Section 2 introduces our prior work and compares relevant research efforts. Section 3 explains the methodology used in the proposed work, while section 4 discusses (in detail) the experimental work done. Finally, Section 5 concludes this article.

2. Prior Work and Related Work

In this section, we introduce our prior work for earthquake detection using emerging technologies and then compare our work with related research efforts.

2.1. Prior Work

In our prior work [10,11], we developed an earthquake alert device that includes a 32 bit processor, Wi-Fi, bluetooth, a buzzer, an LED light, etc as shown in Figure 1; its hardware system is described in Table 1. To detect an earthquake, the earthquake alert device uses a machine-learning-based algorithm

and then sends out an alert message to nearby devices such as smartphones, smart watches, AI speakers and home automation devices, using Bluetooth or Wi-Fi.

Table 1. Specifications of the developed earthquake alert device.

HW Component	Specification
CPU	1 GHz single-core ARMv6
Memory	512 MB
Networks	802.11n wireless LAN, Bluetooth 4.0
Accelerometer	LIS3DHH

Figure 1. The developed earthquake alert device.

The detection algorithm that we developed for the earthquake alert device is based on an artificial neural network (Artificial Neural Network) [12], which is a simple machine learning technique widely used in the last several decades. The used ANN model has three neurons in the input layer, five neurons in the hidden layer, and one neuron in the output layer as shown in Figure 2.

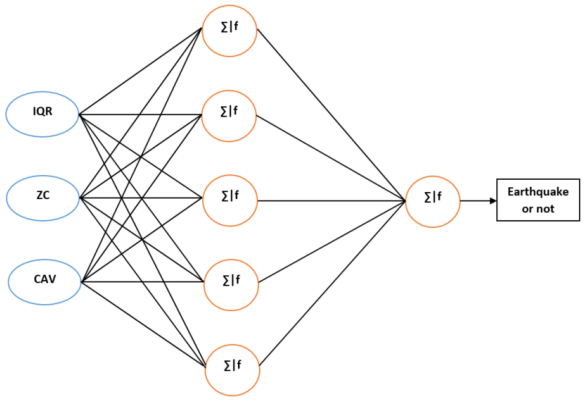


Figure 2. The ANN model with three inputs with one hidden layer.

The detection algorithm consist of four phases including feature extraction, pre-processing, training, and testing of a machine learning model. To detect earthquakes, we use three features including inter-quartile range (IQR), zero crossing rate (ZC), and cumulative absolute velocity (CAV), which are the same features used in MyShake [6]. IQR is the amplitude between 25% and 75% of the acceleration vector sum. ZC is a frequency measure that indicates the number of time that the signal changes its sign. CAV is a cumulative measure of amplitude of the vector sum of three component

acceleration. Then, we use 2 s of a sliding window with a 1.5 s overlap window on the acceleration data to calculate these three features in real-time.

After the extensive experiments, we installed devices in 29 locations of three different cities and operated them for two months. Even though the model showed a high accuracy of over 95% in our experiments, we found a few false alarms throughout the real operation. Therefore, in this article, we carefully assess the performance of the earthquake detection model and test its added features to determine the best features for earthquake detection in our operational environments. In the rest of the article, we present our efforts on the development of various features for earthquake detection and experimental results.

2.2. Related Work

Various monitoring systems leveraging mobile technologies have been proposed, such as eWatch, smartphones, and MEMS [13]. In particular, extensive research has been done on wearable IoT in healthcare. For example, eWatch [14] is an online activity recognition system that embeds four different sensors, i.e., an accelerometer, alight sensor, microphone, and thermometer. The system is very responsive and needs no wireless communication. Similarly, Kao et al. [15] have used a tri-axial accelerometer in a portable device that can be placed on the user's dominant wrist to detect human activities, such as running, working, swinging, brushing teeth, knocking, walking, and hitting. The accelerometer of a smartphone has also been used for human activity recognition, such as walking, running, walking (fast and slow), climbing stairs (up and down), and excercising aerobatics [16]. In the literature, there are many applications that used a sensor-based monitoring system; however, these are beyond the scope of this article. Instead, we deal with the binary classification problem, and our goal is to detect earthquakes from the accelerometer data in which the rest of the data is the non-earthquake class, whether that includes human activity or noise.

Traditional seismic detection involves computational methods such as Short-Term Average/Long-Term Average(STA/LTA), cross-correlation, and template matching [17–21]. These methods are useful but have certain limitations. For example, STA/LTA can detect earthquakes without prior knowledge of the event but can also produce false positives when the situation is more challenging, such as when it involves a low signal to noise ratio (SNR), overlapping events, and some cultural noise. Similarly, cross-correlation detects earthquake signals but is computationally expensive, while template-based matching is a powerful computational method but requires prior information. The above methods are mostly operational in the central system. Moreover, the computational methods do not exhibit any intelligent behavior and operate only on the fixed threshold values.

Recently, there have been research efforts to use MEMS-based sensors as seismic sensors due to their low computational power and cost. Specifically, the NetQuakes project developed by the United States Geological Survey (USGS) installed MEMS sensors around the world but mostly in California [22] and began to collect seismic data from them. Similarly, the following projects developed around the world use MEMS sensors; Home Seismometer Network (HSN) developed by Japan Meteorological Agency (JMA), Palert system developed by NTU(National University Taiwan), Community Seismic Network and Quake-catcher Network (QCN) developed by California Institute of Technology and Standford University, respectively [23–26].

IoT systems for public safety are widely adopted, where the intelligence behavior of such sensors as MyShake, which combines machine learning with traditional STA/LTA algorithm, limit or exclude human intervention [27]. To our knowledge, this is the first globally used smartphone-based earthquake early warning system used in a dynamic environment. Besides, deep learning approaches have also been adopted to detect earthquakes offline or online at the server-side, such as searching seismic data, mining undetected earthquakes in the data archives, and finding the earthquake location [28,29]. In this article, our first goal is to improve the existing earthquake detection model's performance in the static environment. The second goal is to evaluate the machine learning algorithms

and feature sets (both existing and proposed) for sensor-side in the dynamic environment with a variety of human activities.

3. Proposed Methodology

This section will discuss the feature extraction and machine learning methodology. The proposed work follows the supervised machine learning methodology. The steps involved in our proposed methodology are feature extraction, preprocessing, training, testing, and validation.

3.1. Feature Extraction

In the context of ML-based earthquake detection, amplitude and frequency are the two key pieces of information among different statistics of the accelerometer signal. Therefore, based on these two statistics, we extracted features from X, Y, and Z components in the time and frequency domains. Time domain features include features used in MyShake and our proposed features. The MyShake features are the following.

- IQR (Interquartile Range): IQR is the interquartile range $Q3$–$Q1$ of the 3 component vector sum VS;

$$VS = \sqrt{X^2 + Y^2 + Z^2} \tag{1}$$

  where X, Y, and Z are the acceleration components.
- CAV (Cumulative Absolute Velocity): CAV feature is the cumulative measure of the VS in the time window and is calculated as

$$CAV = \int_0^s |VS(t)|dt \tag{2}$$

  where s is the total time period of the feature window in seconds, and t is the time. In this work, we used a two-second feature window.
- ZC (Zero-Crossing): ZC is the maximum zero-crossing rate of X, Y, and Z component and the zero-crossing rate of component X can be calculated as:

$$ZC_X = \frac{1}{N-1} \sum_{t=1}^{N-1} 1_{\mathbb{R}_{<0}}(X_t X_{t-1}) \tag{3}$$

  where N is the total length of the signal X and $1_{\mathbb{R}_{<0}}$ is indicator function.

IQR and CAV are the amplitude features, while ZC is the frequency feature, and these are proposed in [6,30]. These features detect earthquakes and can discriminate non-earthquake data, but through exhaustive experimental evaluations and also its implementation in the static environment as given in our previous work, we found that in a noisy environment (noisy sensors or external events), its performance can be degraded. Moreover, a dynamic environment—in which the variety of human activities that include some challenging activities whose signal patterns are similar earthquake patterns—can also degrade the performance of the model trained on these features. We observed that among the three features, zero-crossing is more sensitive to noise and creates false alarms even if there is wavering involving only one component. This is due to the fact that it counts the feature value for each component and then selects the maximum one. Hence, if there is a count at only one component, then it will select that value and discard the zero-crossing information of the other two components. We observe that earthquake motion has a zero-crossing rate at more than one component simultaneously, while other data—particularly noise data—have zero-crossing rates at only one component most of the time, as given in Table 2. Two-second feature window with a 1-second sliding window is used to count ZC in both earthquake data and noise data, where, for the earthquake data, we selected 3 s of the strongest portion of the earthquake. Further details about datasets are given in the results section.

Table 2. ZC count at only one component.

Class	Total Instances	One Comp: ZC Counts
Earthquake	730	14
Noise	19,813	18,489

This sensitivity issue not only affects the performance of the machine learning model in a dynamic environment (when the sensors are assumed to be smartphones used in daily life) but also affects the model performance while in a fixed-sensors environment. Therefore, to overcome this issue, we tested different variations and statistical features of the amplitude and frequency characteristics of the signal. After extensive experiments, we proposed some variants of the zero-crossings, which are the following.

- Max ZC: Counts for that component whose maximum absolute amplitude value is greater than the other two components when there is more than one zero-crossings at a particular time t. Otherwise, it will behave like the ZC feature.
- Min ZC: Counts for the minimum one, which has lowest absolute amplitude value among the three, if there are zero-crossings in more than one component.
- Max Non ZC: This feature counts the maximum absolute amplitude component for non-zero-crossings when there is more than one non-zero-crossings simultaneously at a particular time.

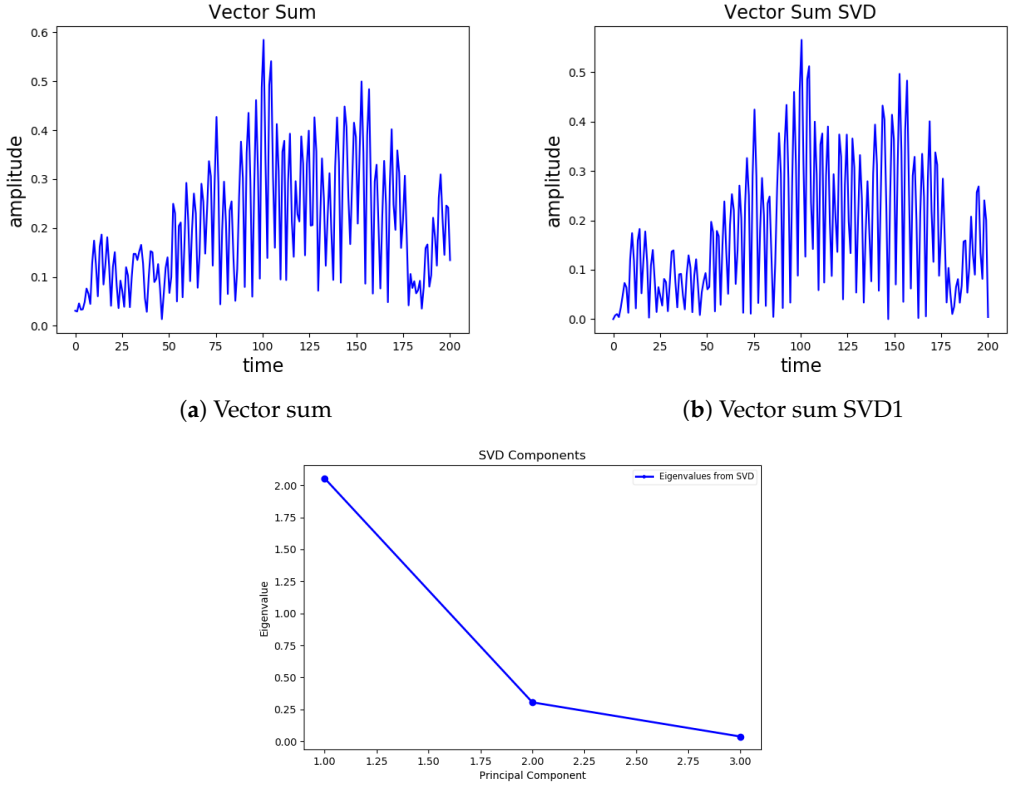
These features are also based on the frequency and amplitude information of the signal; however, these are more specific and consider the other statistics, like multi-component zero-crossings and the frequency information, when there is no zero-crossing. The non-zero-crossing statistic is also important, because if the occurrences of ZC indicate the probability of an earthquake situation, then this feature indicates the probability of a normal situation. Similarly, the multi-component property of these features is also helpful to discriminate human activities from earthquake samples more efficiently.

Apart from the proposed features, we also tested features from the frequency domain, i.e., FFT (Fast Fourier Transform) [31]. In order to consider only one component of FFT, we used a Singular Value Decomposition (SVD) method to decompose multi-dimensional data into one dimension [32]. The SVD of an accelerometer matrix A of three components, X, Y, and Z.

$$A = USV^T \tag{4}$$

where, A is an M x N matrix, where M represents two-second points, i.e., 200, and N is 3. SVD provides three new vectors $U_{MxM}, S_{MxN}, and V^T_{NxN}$, which, if linearly combined, give back the approximated original vector; where U is the set of singular vectors with singular values in vector S, V^T is the primary direction. The new vectors are ordered, and the first vector explains most of the original acceleration amplitude and frequency information, as shown in Figure 3. Figure 3a depict almost the same structure; therefore, we select the first vector as a primary vector $U[:,0]$ from the given SVD's, along with the first value $S[0]$ of S, which is a scaling factor (give amplitude information of the given vector). We extracted the following three additional features.

- FFT: FFT of the given vector $U[:,0]$ is calculated, and we selected the frequency bin as a frequency feature that has the peak amplitude, as shown in Figure 4.
- SVD_Scale: The $S[0]$ is considered the average amplitude feature.
- SVD_ZC: We also computed the zero-crossing rate of the primary vector i.e., $U[:,0]$.



(**a**) Vector sum

(**b**) Vector sum SVD1

(**c**) Scree plot

Figure 3. A two-second window of the strongest portion of the earthquake; (**a**) The vector sum of X, Y, and Z; (**b**) vector sum of the primary vector of SVD (center); (**c**) scree plot of the three components.

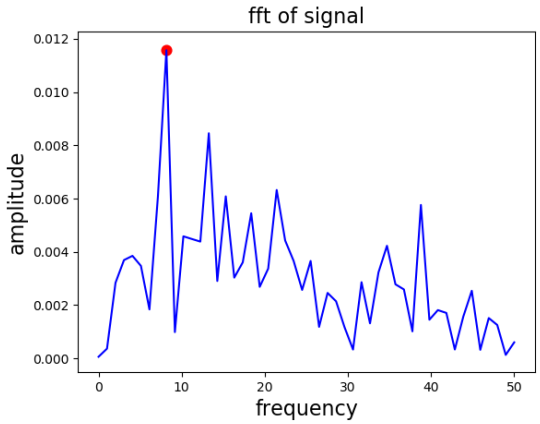


Figure 4. FFT of the primary vector of the SVD extracted from the two-second strongest portion of the earthquake window.

We also analyzed the tsfresh [33], a time series feature-extraction python package for searching the computationally low and effective features such as c3, cid-ce, entropy, mean, and count-above-mean, etc. However, the feature space visualization was not more promising than the abovementioned features. Therefore, we selected only the above features for model training and testing.

3.2. Pre-Processing

In our methodology, the pre-processing involved balancing the dataset and scaling the features to range from 0 to 1. Balancing is required because the imbalanced datasets greatly affect the performance of the machine learning model [34]. In our case, the non-earthquake dataset (noise and human

activities) is much larger than the earthquake dataset. Hence, we used the K-mean clustering algorithm to balance the non-earthquake dataset [35]. Using the K-Mean, clusters of the non-earthquake data are created according to the total number of earthquake data points, and we used centroids of the clusters to represent the non-earthquake data. As shown in Figure 5, centroids represent the original data points in the IQR, ZC, and CAV feature space.

Moreover, to improve the prediction performance and decrease the training time of the model, we also scaled data point d to the range of 0 to 1 using the min-max scaler as follows:

$$d_{scaled} = \frac{d - d_{min}}{d_{max} - d_{min}} \tag{5}$$

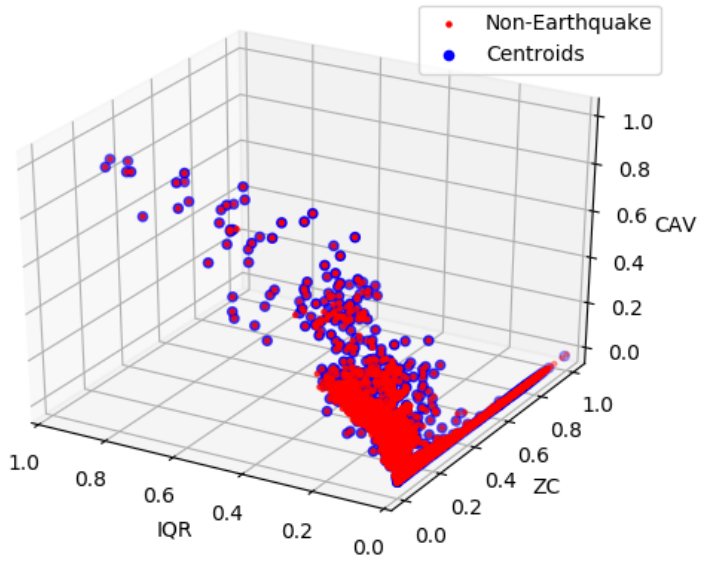


Figure 5. Centroids of the non-earthquake data (noise with some human activity).

3.3. Machine Learning Model

The ANN (Artificial Neural Network) algorithm is designed to accomplish the detection task using both the existing and proposed features [12]. We used an X, 5, 1 layer network architecture for the training and testing of the ANN algorithm, as shown in Figure 6, where X is the number of features input to the model. We kept the same five nodes of the hidden layer as proposed in [6], because the number of features input to the model is 3, 4, or 5, and through experimental results, the 5-node hidden layer is still good for the given number of features. For training the models, we used a multi-layer perceptron (MLP) with the stochastic gradient descent solver [36–38]. For the hidden layer and output layer, the inputs from the previous layer to each node will be first summed and then fed into an activation function as follows:

$$y = \phi(\sum_{i=1}^{n} w_i d_i + b) \tag{6}$$

Here, w denotes the weights vector, d is the input vector, b is the bias, y is the output of the given node, and ϕ is the non-linear activation function. The logistic sigmoid function is used as the activation function for hidden and output layers, which is defined on input d as

$$\phi(d) = \frac{1}{1 + e^{-d}} \tag{7}$$

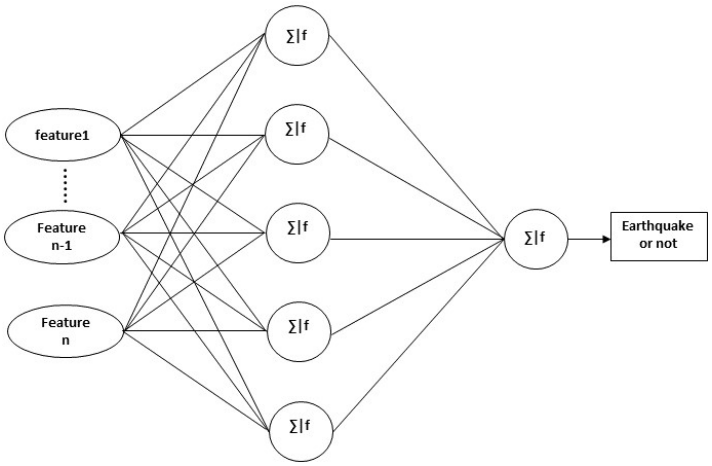


Figure 6. The general structure of the ANN model.

4. Results and Discussion

To obtain a comprehensive comparison, we compared the proposed features with the existing features in both the static and dynamic environments. Accordingly, we trained ANN models with different non-earthquake datasets to distinguish the environments.

4.1. Dataset

The dataset that we used for training and testing the ANN models contains two classes of label data. One class of data is the time series earthquake dataset, which was download from the National Research Institute of Earth Science and Disaster Prevention (NIED) and USGS (United States Geological Survey) database [39,40]. A total of 385 earthquakes events with magnitudes ranging from 4 to 8, recorded between April 2009 and May 2019, were selected from the NIED database. Moreover, 120 stations' data of three earthquakes, i.e., Tottori (2000) (magnitude 6.61), Niigata (2004) (magnitude 6.63) and Chuetsuoki (2007) (magnitude 6.8) were downloaded from the USGS database. The NIED earthquake data were pre-processed and converted into the unit (g). The sampling rate of all the earthquake data is 100 Hz. The data are presented in three columns titled EW, NS, and UD, respectively, where EW (East-West) and NS (North-South) are horizontal components, and UD (Up-Down) is a vertical component.

The second class of data is the time series of non-earthquake dataset recorded on mobile phones for several hours. In the experiments, we used two types of non-earthquake data, i.e., human activity data and noise data. Human activity data includes such activities as bicycle, bus, and car (in hand) riding, jump rope, running (hand, pocket), desk shaking (while mobile on top), climbing stairs (up-down) (bag, hand, pocket), walking (bag, hand, pocket), standing still, and working. Contrarily, noise data contain floor noises (e.g., different degrees of elevations) and machinery noises. These noise data are the external source data; and hence, to include sensor noise data, we also include the tail data of earthquake signals.

The models' generalization characteristics are validated on the third dataset, which is earthquake data collected during shake table tests using different accelerometers (i.e., ADXL355 [41], LIS3DHH [42], MPU9250 [43], and MMA8452 [44]), which have different HW specifications and costs. Sensors were placed on the shake table located in Pusan National University to record two realistic earthquakes including Pohang [45] and El Centro [46], and we collected acceleration data from such low-cost accelerometers.

4.2. Performance Metrices

Different machine learning algorithms are evaluated with different performance matrices. The classification performance metrics are based on the confusion matrix [47], which gives a table of TP (True Positive), TN (True Negative), FP(False Positive), and FN (False Negative) as shown in Figure 7.

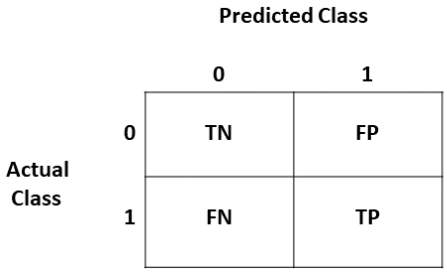


Figure 7. Confusion Matrix.

Common performance measures of the classification such as accuracy, precision, and recall, are calculated from the confusion matrix. Accuracy is computed as

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Accuracy is the ratio of predictive observations to the total observations; it is an intuitive measure to show the overall performance of the model.

Precision is computed as

$$\text{Precision} = \frac{TP}{TP + FP}$$

The precision measure determines how accurate the model is out of those predictive positives. In other words, how many of them are actually (correctly) positive among all predictive positives. High precision means a low false-positive rate.

Recall is computed as

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall determines the sensitivity of the model in terms of how many times it detects earthquakes from all the instances of the earthquakes.

The F1 score is a single score of precision and recall which is the harmonic mean of both. It takes both false positive and false negatives into account.

$$\text{F1 score} = \frac{2}{1/precision + 1/recall}$$

Finally, the classification model performance false and true positive rates can be visualized through a receiver operating characteristics (ROC) curve [48].

4.3. Evaluation

The evaluation is done in static and dynamic environments. In the static environment, the sensors are fixed (stationary) and, therefore, training a model with varieties of human activities is not required. Still, to train the model properly for the static environment, we used some instances of human activities like walking and waiting. This is because the model converges too quickly in the presence of only noisy data and thus cannot learn the underlying patterns of the data, especially earthquake patterns. We evaluated models based on different features and then, for the dynamic environment, we tested the model that showed the best results in the static environment to evaluate its full implementation applicability.

4.4. Static Environment

During the model evaluation in the static environment, we used a combination of different features discussed above. We trained the model using amplitude features combined with frequency features. Here, from the sets of different models, we will discuss six models, beginning with the existing MyShake model, i.e., IQR, ZC, and CAV (Model 1). The remaining five models with feature sets are given below.

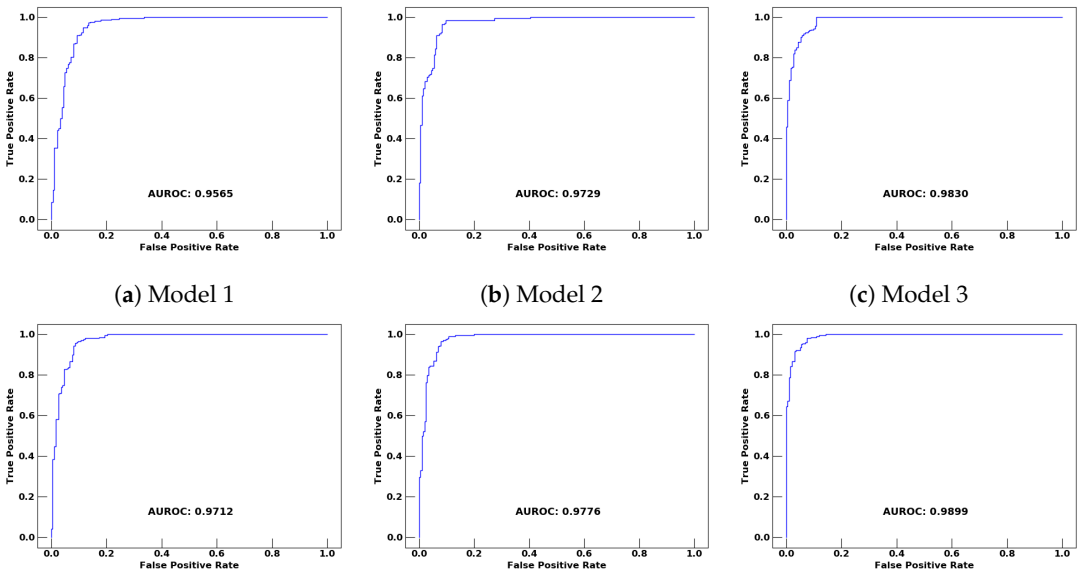
- Model 2: IQR, FFT, CAV
- Model 3: IQR, ZC, FFT, CAV
- Model 4: IQR, SVD_ZC, CAV
- Model 5: IQR, ZC, CAV, SVD_Scale
- Model 6: IQR, CAV, MAX_ZC, MIN_ZC, MAX_NON_ZC

Through the experimental search, the five-nodes hidden-layer structure was used for training all the models, and the input layer nodes were determined according to the feature set. Table 3 provides details of the dataset used for training and testing the models in the static environment.

Table 3. Dataset used for models training and testing in a static environment.

Class	Total Instances
Earthquake	1010
Noise	20,116
Walk and Wait	4000

For training and testing the models, we split the data (earthquake and centroids of the non-earthquake) into 80% and 20%, respectively. In terms of testing the models, we first tested each model on the remaining 20% of the centroids (experiment 1), and then, in the second experiment (experiment 2), the models were tested on the original data (all the instances of the earthquake and non-earthquake class). For the receiving operating characteristics curves of the ANN models, 20% of the remaining data are shown in Figure 8. All the models showed good results, where Model 6 shows the high AUROC of 0.9899 and rapid climb, which is close to the ideal case.



(**a**) Model 1 (**b**) Model 2 (**c**) Model 3

(**d**) Model 4 (**e**) Model 5 (**f**) Model 6

Figure 8. ROC curves of the models on 20% test data.

Despite the fact that all the models showed good results, other performance measures of the models should also be considered. Table 4 gives the performance score of the models on the two test datasets; the first test dataset is the remaining 20% of the centroids data and the second test dataset is the original non-earthquake data. Among all the models, Model 6 successfully classified the earthquake instances in both experiments (i.e., centroids and original data) with high accuracy, F1 score, and a low number of false positives. The false positive in the second experiment is comparatively high as compared to experiment 1 because there are more data points for a particular non-earthquake category with variations in the data. Still, the accuracy score of both experiments was very good. The accuracy and recall of the second experiment are slightly better than experiment 1, which indicates that the model is also trained well for unseen data to deal with the over-fitting problem. As a single frequency feature, FFT standalone cannot provide assistance to the model, as shown in the results of Model 2. However, with the ZC feature, it showed some improvement in the performance of Model 1, as seen in the Model 3 row. The new features of ZC_SVD and SVD_Scale can be used as a frequency feature and amplitude feature as suggested by Models 4 and 5, respectively.

Table 4. Performance evaluation of the ANN models on test data.

Model	Test Data	TP	TN	FP	FN	Accuracy	Precision	Recall	F1
Model 1	Centroids	203	164	20	17	90.84	91.03	92.27	91.65
	Original	962	22,556	1560	48	93.60	38.14	95.24	54.47
Model 2	Centroids	178	192	13	21	91.59	93.19	89.45	91.28
	Original	925	20,515	3601	85	85.33	20.44	91.59	33.18
Model 3	Centroids	210	171	21	2	94.31	90.91	99.06	94.81
	Original	993	22,797	1319	17	94.68	42.95	98.32	59.78
Model 4	Centroids	195	180	16	13	92.82	92.42	93.75	93.08
	Original	949	22,214	1902	61	92.19	33.29	93.96	49.16
Model 5	Centroids	193	185	21	5	93.54	90.19	97.47	93.69
	Original	984	22,537	1579	26	93.61	38.39	97.43	55.08
Model 6	Centroids	192	191	11	10	94.80	94.58	95.05	94.82
	Original	959	23,552	564	51	97.55	62.97	94.95	75.72

The FP counts of Model 1 indicate that the model is sensitive to the noise due to the frequency feature of ZC, as discussed earlier. However, Model 6 has three different statistics for the frequency information, which gives more information to the machine learning model and contributes to the improved results of the model. It has been observed that, for the amplitude information, IQR and CAV are still good features, but the frequency feature is the most sensitive one since it is not only affected by the noise but also by the difference of the sampling rates.

4.4.1. Models Validation

As described in the dataset section, to further validate and compare these models on the unseen data, we used Pohang and El Centro earthquakes data. Here, we used 100% and 50% scale of both the earthquakes, where a 100% of Pohang and El Centro respectively represents approximate magnitudes of 5.4 and 6.9 earthquakes. Similarly, the 50% scale earthquake data represents moderate and low amplitude earthquake data, which allows us to evaluate the model performance on these low scale data.

During each test, these 100% and 50% scale data of both the earthquakes are input into the trained models. The duration of each earthquake data is 70–80 s, and the features are extracted from each sensor's data according to the model feature set; for example, for Model 1, we extracted IQR, ZC, and CAV features. A two-second sliding window with a one-second overlap window was used on the raw acceleration to extract features.

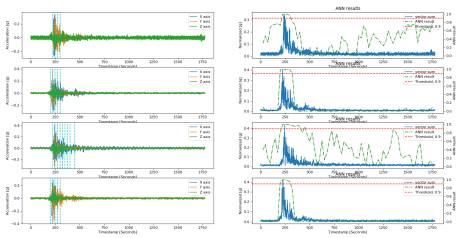
The test results of the models on the Pohang and El Centro data are given in Figures 9 and 10, respectively. Figure 9a shows the results of the detection process for Model 1 (right panel) on the normalized vector sum of three axes, along with the raw acceleration (left panel) of X, Y and Z components of the Pohang earthquake on the 100% scale. The sampling rate of the signal is 25 Hz and the threshold value is kept at 0.9 for the ANN models to detect earthquake triggers. The reason for choosing this threshold value is to decrease the FP rate of the model in the system implementation. As given in Table 5, FP counts of the models' on the original test data decreased, as a result, the precision of the models increased but the recall is also decreased. Here, we can see that the F1 score describes the overall performance of the models, and again Model 6's F1 score is high compared to other models when the threshold value is set to 0.9. The cyan vertical line in the left panel represents the earthquake trigger when the ANN probability (wavy green line in the right panel) meets the threshold value (red line in the right panel).

Model 1 detected earthquakes in both the 100% and 50% scales of the Pohang earthquakes, and did so at peak acceleration with very low false alarms, as shown in Figure 9a,b. The ANN probability graph shows very smooth and stable probabilities in the LIS3DHH and MPU9250 cases during the non-earthquake portion, while the other two cases (i.e., ADXL355 and MMA8452 results, show high peaks due to sensor noise). In our previous publications [10,11], we observed that ADXL355 has noise on one component while MMA8452 has noise on all the components, which confused the model due to the zero-crossing feature. Model 1 produced similar results for the El Centro earthquake data across different scales, as shown in Figure 10a,b. In the El Centro case, as we can see its signal pattern is different from the Pohang pattern, which resulted in more false triggers generated by Model 1.

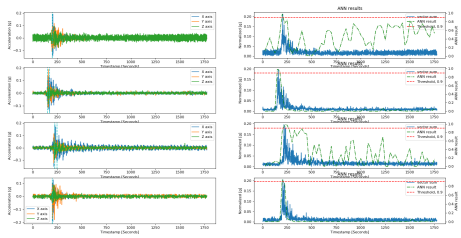
Compared to Model 1, the proposed Model 6 showed promising results on the validation data of Pohang and El Centro earthquakes of different scales and sensors. as shown in Figures 9k,l and 10k,l. The proposed Model 6 shows better performance on the accelerometer sensor data, where Model 1 produced high peaks in the probability graphs. Further, in the case of the ADXL355 accelerometer sensor, Model 6 was able to with the problem of noise on one component due to its extensive information about zero-crossing frequency. Multi-component noises can be a challenge for model performance, as the MMA8452 case reveals.

Model 2 shows poor results due to the lack of frequency information in the time domain, as shown in Figures 9c,d and 10c,d. Despite having tested the FFT feature to provide the frequency information, through the experimental work, we observed that a single FFT feature is not enough to train the model. Then, although combine with the ZC feature it showed slightly better results than Model 1 in 20% test, the results of the Pohang and El Centro test data were poorer than those of Model 1, as shown in Figures 9e,f and 10e,f. Similarly, the results of model 5 were also insignificant, as shown in Figures 9i,j and 10i,j.

To Summarize, Models 1 and 6 show outstanding and consistent results in the unseen data, whereas other models show different performance results in different experimental scenarios. Therefore, we chose to test Models 1 and 6 in a dynamic environment.

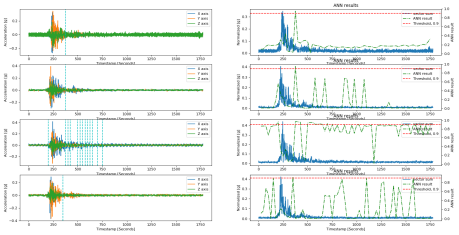


(**a**) Model 1 results on 100% scale

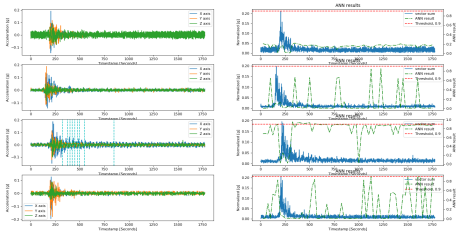


(**b**) Model 1 results on 50% scale

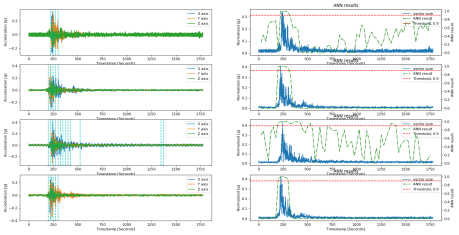
Figure 9. *Cont.*



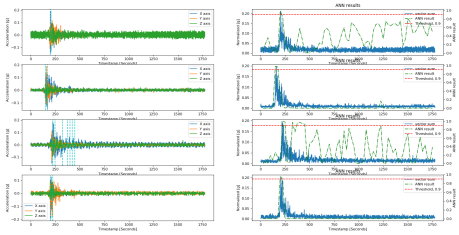
(**c**) Model 2 results on 100% scale



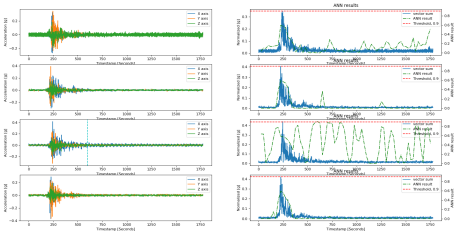
(**d**) Model 2 results on 50% scale



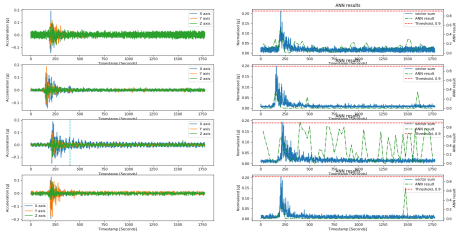
(**e**) Model 3 results on 100% scale



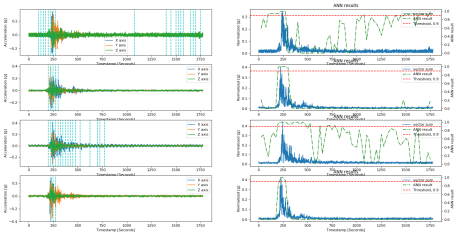
(**f**) Model 3 results on 50% scale



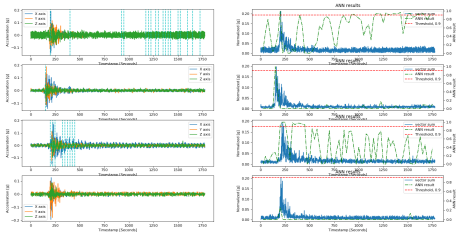
(**g**) Model 4 results on 100% scale



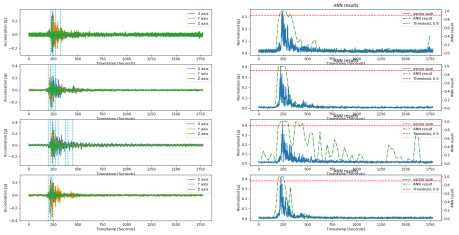
(**h**) Model 4 results on 50% scale



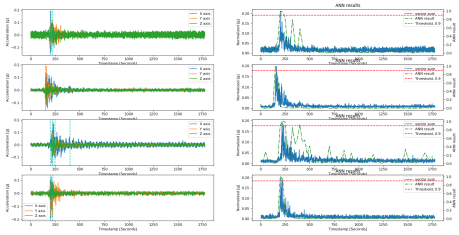
(**i**) Model 5 results on 100% scale



(**j**) Model 5 results on 50% scale

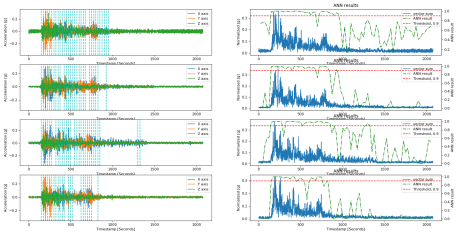


(**k**) Model 6 results on 100% scale

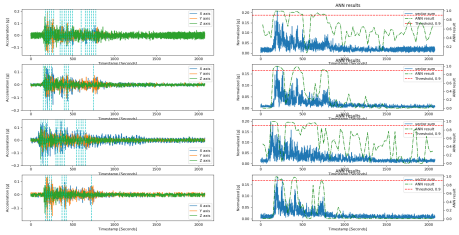


(**l**) Model 6 results on 50% scale

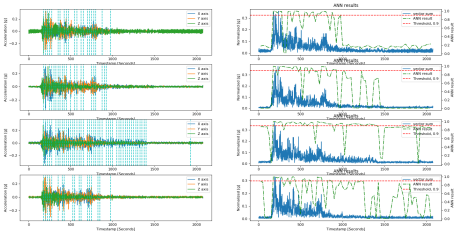
Figure 9. Models' test results on the Pohang earthquake at a scale of 100% (left) and 50% (right); the results are ordered from top to bottom for each subfigure's sensor, namely ADXL355, LIS3DHH, MMA8452, and MPU9250.



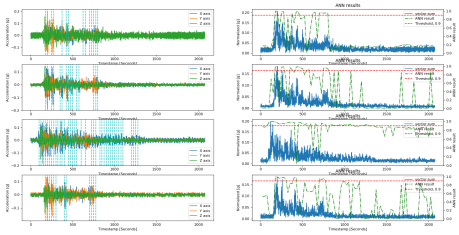
(**a**) Model 1 results on 100% scale



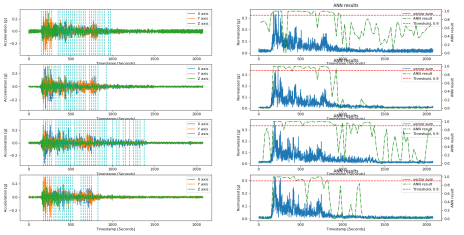
(**b**) Model 1 results on 50% scale



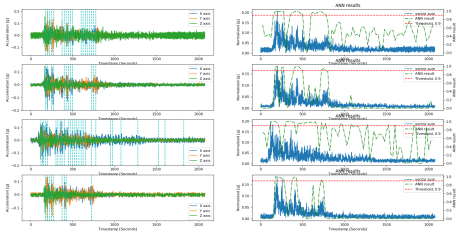
(**c**) Model 2 results on 100% scale



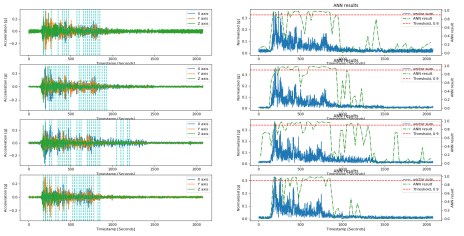
(**d**) Model 2 results on 50% scale



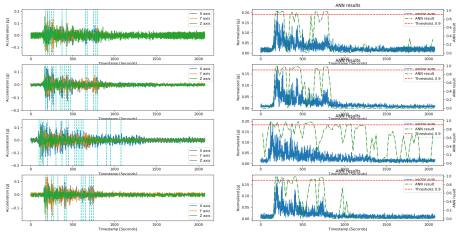
(**e**) Model 3 results on 100% scale



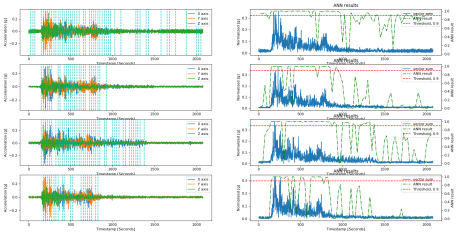
(**f**) Model 3 results on 50% scale



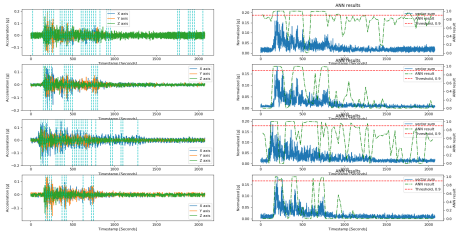
(**g**) Model 4 results on 100% scale



(**h**) Model 4 results on 50% scale

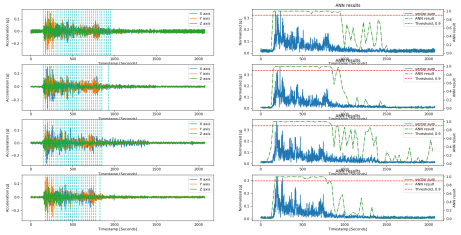


(**i**) Model 5 results on 100% scale

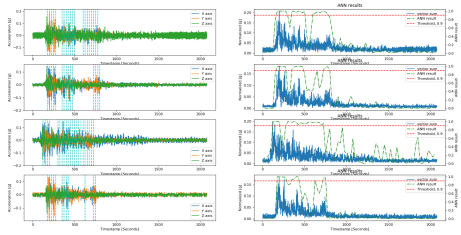


(**j**) Model 5 results on 50% scale

Figure 10. *Cont.*



(**k**) Model 6 results on 100% scale



(**l**) Model 6 results on 50% scale

Figure 10. Models' test results on the El Centro earthquake at a scale of 100% (left) and 50% (right); the results are ordered from top to bottom for each subfigure's sensor, namely ADXL355, LIS3DHH, MMA8452, and MPU9250.

Table 5. Performance evaluation of the ANN models on the original test data using a 0.9 threshold value.

Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1
Model 1	698	23,987	129	312	98.24	84.40	69.11	75.99
Model 2	543	24,112	4	467	98.13	99.27	53.76	69.75
Model 3	815	23,536	580	195	96.92	58.42	80.69	67.76
Model 4	685	23,932	184	325	97.97	78.83	67.82	72.91
Model 5	781	23,814	302	229	97.89	72.11	77.33	74.63
Model 6	807	24,059	57	203	98.96	93.40	79.90	86.13

4.5. Dynamic Environment

In the dynamic environment, we evaluate Model 6, which showed very good results as compared to other models, including the model used in MyShake. Therefore, to evaluate the model with the proposed features in the dynamic environment, we considered all the human activities recorded on smartphones for several hours. Due to the increase of non-earthquake data, we also include more earthquake data to keep the balance between earthquakes and non-earthquakes and then test the model performance on the larger datasets. The trained model is referred to as Dyn-model 6, and to compare the model with the state-of-the-art Myshake features, we also train Model 1 using the dynamic dataset and referred to it as Dyn-model 1. Table 6 provides details of the dataset used for training the models in the dynamic environment.

Table 6. Datasets used for model training and testing in a dynamic environment.

Class	Total Instances
Earthquake	2464
Non-earthquake	44,094

We perform the same methodology used for the static data; that is, we first extracted features from the non-earthquake data then calculated centroids equal to the earthquake data points. To train the model, we split the data (earthquake and centroids) into 80% and 20% for training and testing, respectively. The best test results from a number of experiments are given in Table 7. We can see that the accuracy of the Dyn-model 6 on the original data is higher than that of the centroids data, whereas the Dyn-model 6 accuracy on 20% is similar to its accuracy in the static test (i.e., approximately 94%). However, its accuracy on the original data in the dynamic test is lower than the static test due to the variation in the non-earthquake data. In the original data, the Dyn-model 6 falsely detects 1804 non-earthquake instances as earthquakes. We further investigate the Dyn-model 6 results on each

activity and found that the FP of the Dyn-model 6 was mostly produced due to human activities such as bus riding, desk shaking, and walking, with the accuracy of 90.3%, 93.23%, and 91.65%, respectively.

Like Dyn-model 6, Dyn-model 1 accuracy was also decreased due to the activities that can result in earthquake-like signals. Moreover, Dyn-model 6 results are better than those of Dyn-model 1 results due to the increasing frequency features and different recalculation of ZC to make it conditional to the maximum amplitude.

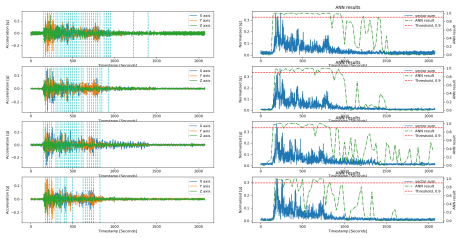
Table 7. Performance evaluation of the ANN models on the test data in the dynamic environment.

ANN Model	Test Data	TP	TN	FP	FN	Accuracy	Precision	Recall	F1
Dyn-model 1	Centroids	460	442	56	28	91.48	89.15	94.26	91.63
	Original	2315	39,935	4159	149	90.07	35.76	93.95	51.80
Dyn-model 6	Centroids	474	453	35	24	94.02	93.12	95.18	94.14
	Original	2336	42,290	1804	128	95.85	56.43	94.81	70.8

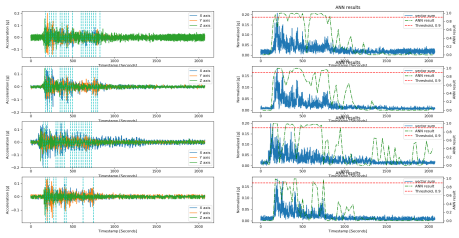
4.5.1. Model Validation

Similar to the validation for the static environmental model, we validated Dyn-model 6 on the unseen earthquake datasets generated during shake table tests on different sensors, i.e., El Centro and Pohang, as shown in Figure 11. This was conducted because our ultimate goal was to see whether the trained model can detect earthquakes when the device is in a steady-state, as compared to the proposed Dyn-model 6, which was trained for the static environment and then retrained for the dynamic environment. The newly trained Dyn-model 6 showed almost the same detection results, but with fewer earthquake detection triggers during the earthquake windows for both the scales of the El Centro and Pohang earthquakes. The proposed Dyn-model 6 also detected some false triggers as can be shown in Figure 11a. In particular, the proposed Dyn-model 6, when trained with dynamic data, showed far fewer detection triggers on the Pohang earthquakes; rather than detecting the earthquake, it only showed a peak below the given threshold of 0.9 for the data recorded on the MPU9250 scale 50%, as shown in Figure 11d. These results indicate that the model learned differently, and the challenging non-earthquake data can affect the model performance, which can result in a false alarm, whether false positive or false negative.

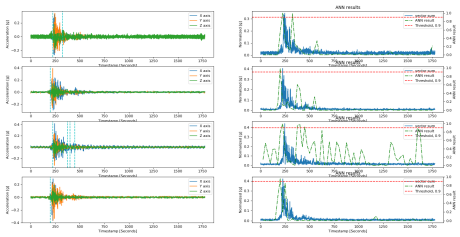
The validation results of Dyn-model 1 on the Pohang and El Centro datasets are given in Figure 12. This time, Dyn-model 1 results are not as promising as those of the static environment in all the windows. In particular, it failed to detect the earthquake in the Pohang 50% scale. It showed the same probability peaks in the ADXL355 and MMA8452 sensors in all the windows, which support our claim discussed with regard to the static environment. Moreover, compared to the proposed model that is Dyn-model 6, again the performance of the existing model that is Dyn-model 1 is below in the dynamic environment too.



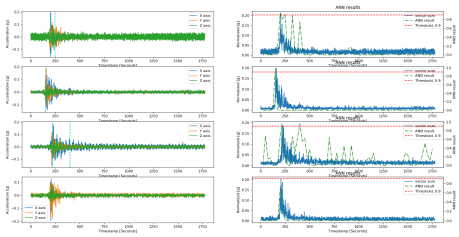
(**a**) Dyn-model 6 results on 100% scale El Centro



(**b**) Dyn-model 6 results on 50% scale El Centro

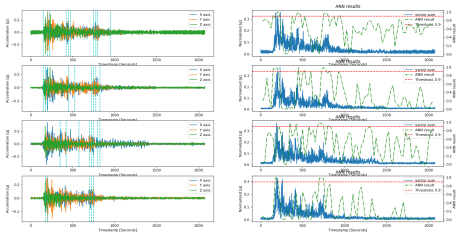


(**c**) Dyn-model 6 results on 100% scale Pohang

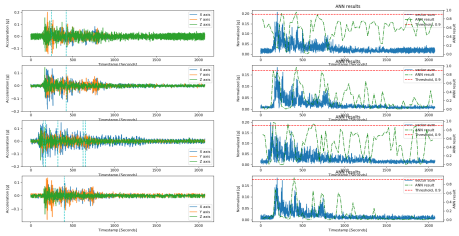


(**d**) Dyn-model 6 results on 50% Scale Pohang

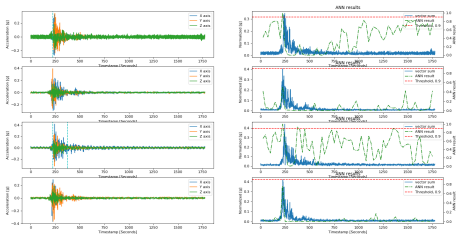
Figure 11. Dyn-model 6 test results on the El Centro earthquake of scale 100% (left-top) and scale 50% (right-top); Model test results on the Pohang earthquake of scale 100% (left-bottom) and scale 50% (right-bottom); the results are ordered from top to bottom for each subfigure's sensor, namely ADXL355, LIS3DHH, MMA8452, and MPU9250.



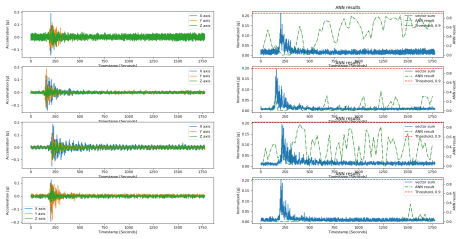
(**a**) Dyn-model 1 results on 100% scale El Centro



(**b**) Dyn-model 1 results on 50% scale El Centro



(**c**) Dyn-model 1 results on 100% scale Pohang



(**d**) Dyn-model 1 results on 50% Scale Pohang

Figure 12. Dyn-model 1 test results on the El Centro earthquake of scale 100% (left-top) and scale 50% (right-top); Model test results on the Pohang earthquake of scale 100% (left-bottom) and scale 50% (right-bottom); the results are ordered from top to bottom for each subfigure's sensor, namely ADXL355, LIS3DHH, MMA8452, and MPU9250.

4.6. Threats to Validity

The experimental results are subject to the following validity threats. Even though we deal with heterogeneous data recorded on different sensors, the models were trained on the data which were mostly recorded on the seismic sensors but we used low-cost accelerometers for validating their performance. Therefore, the experimental result may be different if the models are properly

trained using earthquake data recorded on low-cost accelerometers. Furthermore, the datasets that we collected have different sampling rates. For instance, the models were trained with the earthquake data at a 100 Hz sampling rate and different sampling rate of non-earthquake data ranging from 50 to 100 Hz. Also, the sampling rate for the validation was 25 Hz. Despite above validity threats, the model showed outstanding performances, but the accuracy measurements may vary for different datasets and sampling rates.

5. Conclusions

In this article, we categorized seismic detection mechanisms into the static and dynamic environments and then evaluated different features using the ANN model in the static environment, which include new features and the existing features used in previous studies. Based on the experimental results performed in the static environment produced, the proposed features demonstrated more improved results than the existing features. For the dynamic environment, we used the same model tested for the static environment and then trained it with different datasets, which included various human activities. The selected model showed promising results with a lower possibility of false alarms than other models. As a result, our approach can be used for both a static and a dynamic environment without changing its model and features. As a future research direction, we will explore new features and models that require less computational power while maintaining a high detection ability against the challenging non-earthquake datasets.

Author Contributions: Conceptualization, Y.-W.K.; Data curation, I.K.; Funding acquisition, Y.-W.K.; Methodology, I.K.; Project administration, S.C. and Y.-W.K.n; Software, I.K.; Supervision, S.C. and Y.-W.K.; Validation, I.K.; Writing—original draft, I.K.; Writing—review & editing, Y.-W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Disaster Management Research Institute (2019-02-02), Basic Science Research Program through the National Research Foundation of Korea(NRF) grant funded by the Ministry of Education(NRF-2017R1C1B5075658), and the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holzer, T.L.; Savage, J.C. Global earthquake fatalities and population. *Earthq. Spectra* **2013**, *29*, 155–175. [CrossRef]
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
3. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
4. Komninos, N.; Philippou, E.; Pitsillides, A. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1933–1954. [CrossRef]
5. D'Alessandro, A. Tiny accelerometers create Europe's first urban seismic network. *Eos* **2016**, *97*, 16–20. [CrossRef]
6. Kong, Q.; Allen, R.M.; Schreier, L.; Kwon, Y.W. MyShake: A smartphone seismic network for earthquake early warning and beyond. *Sci. Adv.* **2016**, *2*, e1501055. [CrossRef]
7. Seismograms from the Cooperative New Madrid Seismic Network at Saint Louis University. Available online: http://www.eas.slu.edu/Outreach/seismograms.pdf (accessed on 12 December 2019).
8. Minson, S.E.; Meier, M.A.; Baltay, A.S.; Hanks, T.C.; Cochran, E.S. The limits of earthquake early warning: Timeliness of ground motion estimates. *Sci. Adv.* **2018**, *4*, eaaq0504. [CrossRef]
9. Sims, J. *The No-Nonsense Guide To Earthquake Safety (Enhanced Edition)*; Lulu.com: Morrisville, NC, USA, 2015.
10. Lee, J.; Khan, I.; Choi, S.; Kwon, Y.W. A Smart IoT Device for Detecting and Responding to Earthquakes. *Electronics* **2019**, *8*, 1546. [CrossRef]

11. Lee, J.; Kim, J.S.; Choi, S.; Kwon, Y.W. A Smart Device Using Low-Cost Sensors to Detect Earthquakes. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–4.
12. Bishop, C. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, NY, USA, 1996.
13. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 1192–1209. [CrossRef]
14. Maurer, U.; Smailagic, A.; Siewiorek, D.P.; Deisher, M. *Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions*; Technical Report; Carnegie-Mellon Univeristy, School of Computer Science: Pittsburgh, PA, USA, 2006.
15. Kao, T.P.; Lin, C.W.; Wang, J.S. Development of a portable activity detector for daily activity recognition. In Proceedings of the 2009 IEEE International Symposium on Industrial Electronics, Seoul, Korea, 5–8 July 2009; pp. 115–120.
16. Bayat, A.; Pomplun, M.; Tran, D.A. A study on human activity recognition using accelerometer data from smartphones. *Procedia Comput. Sci.* **2014**, *34*, 450–457. [CrossRef]
17. Withers, M.; Aster, R.; Young, C.; Beiriger, J.; Harris, M.; Moore, S.; Trujillo, J. A comparison of select trigger algorithms for automated global seismic phase and event detection. *Bull. Seismol. Soc. Am.* **1998**, *88*, 95–106.
18. Allen, R. Automatic phase pickers: Their present use and future prospects. *Bull. Seismol. Soc. Am.* **1982**, *72*, S225–S242.
19. Dales, P.; Audet, P.; Olivier, G.; Mercier, J.P. Interferometric methods for spatio temporal seismic monitoring in underground mines. *Geophys. J. Int.* **2017**, *210*, 731–742. [CrossRef]
20. Skoumal, R.J.; Brudzinski, M.R.; Currie, B.S.; Levy, J. Optimizing multi-station earthquake template matching through re-examination of the Youngstown, Ohio, sequence. *Earth Planet. Sci. Lett.* **2014**, *405*, 274–280. [CrossRef]
21. Gibbons, S.J.; Ringdal, F. The detection of low magnitude seismic events using array-based waveform correlation. *Geophys. J. Int.* **2006**, *165*, 149–166. [CrossRef]
22. Luetgert, J.; Oppenheimer, D.; Hamilton, J. *The NetQuakes Project-Research-Quality Seismic Data Transmitted via the Internet from Citizen-Hosted Instruments*; AGU Fall Meeting Abstracts: 2010; S51E-03. Available online: https://ui.adsabs.harvard.edu/abs/2010AGUFM.S51E..03L/abstract (accessed on 31 January 2020).
23. Horiuchi, S.; Horiuchi, Y.; Yamamoto, S.; Nakamura, H.; Wu, C.; Rydelek, P.A.; Kachi, M. Home seismometer for earthquake early warning. *Geophys. Res. Lett.* **2009**, *36*. [CrossRef]
24. Wu, Y.M. Progress on development of an earthquake early warning system using low-cost sensors. *Pure Appl. Geophys.* **2015**, *172*, 2343–2351. [CrossRef]
25. Cochran, E.; Lawrence, J.; Christensen, C.; Chung, A. A novel strong-motion seismic network for community participation in earthquake monitoring. *IEEE Instrum. Meas. Mag.* **2009**, *12*, 8-15. [CrossRef]
26. Clayton, R.W.; Heaton, T.; Chandy, M.; Krause, A.; Kohler, M.; Bunn, J.; Guy, R.; Olson, M.; Faulkner, M.; Cheng, M.; et al. Community seismic network. *Ann. Geophys.* **2011**, *54*, 738–747.
27. Kong, Q.; Lv, Q.; Allen, R.M. Earthquake Early Warning and Beyond: Systems Challenges in Smartphone-based Seismic Network. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*; ACM: New York, NY, USA, 2019; pp. 57–62.
28. Rong, K.; Yoon, C.E.; Bergen, K.J.; Elezabi, H.; Bailis, P.; Levis, P.; Beroza, G.C. Locality-sensitive hashing for earthquake detection: A case study of scaling data-driven science. *Proc. VLDB Endow.* **2018**, *11*, 1674–1687. [CrossRef]
29. Perol, T.; Gharbi, M.; Denolle, M. Convolutional neural network for earthquake detection and location. *Sci. Adv.* **2018**, *4*, e1700578. [CrossRef] [PubMed]
30. Kong, Q.; Kwony, Y.W.; Schreierz, L.; Allen, S.; Allen, R.; Strauss, J. Smartphone-based networks for earthquake detection. In Proceedings of the 2015 15th International Conference on Innovations for Community Services (I4CS), Nuremberg, Germany, 8–10 July 2015; pp. 1–8.
31. Gentleman, W.M.; Sande, G. Fast Fourier Transforms: For fun and profit. In *Proceedings of the November 7–10, 1966, Fall Joint Computer Conference*; ACM: New York, NY, USA, 1966; pp. 563–578.
32. Wall, M.E.; Rechtsteiner, A.; Rocha, L.M. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*; Springer: Berlin, Germany, 2003; pp. 91–109.
33. Christ, M.; Braun, N.; Neuffer, J.; Kempa-Liehr, A.W. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing* **2018**, *307*, 72–77. [CrossRef]

34. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
35. Bock, H.H. Clustering methods: A history of k-means algorithms. In *Selected Contributions in Data Analysis and Classification*; Springer: Berlin, Germany, 2007; pp. 161–172.
36. Hecht-Nielsen, R. Theory of the backpropagation neural network. In *Neural Networks for Perception*; Elsevier: Amsterdam, The Netherlands, 1992; pp. 65–93.
37. Haykin, S.; Network, N. A comprehensive foundation. *Neural Netw.* **2004**, *2*, 41.
38. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the COMPSTAT'2010*; Springer: Berlin, Germany, 2010; pp. 177–186.
39. National Research Institute for Earth Science and Disaster Prevention. Available online: http://www.kyoshin.bosai.go.jp (accessed on 31 January 2020).
40. Peer Ground Motion Database, Pacific Earthquake Engineering Research Center. Available online: https://ngawest2.berkeley.edu/ (accessed on 31 January 2020).
41. ADXL355 Available online: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_355.pdf (accessed on 31 January 2020).
42. LIS3DHH Available online: https://www.st.com/resource/en/datasheet/lis3dhh.pdf (accessed on 31 January 2020).
43. MPU9250 Available online: https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf (accessed on 31 January 2020).
44. MMA8452 Available online: https://www.nxp.com/docs/en/data-sheet/MMA8452Q.pdf (accessed on 31 January 2020).
45. Available online: https://www.nature.com/articles/d41586-018-04963-y (accessed on 31 January 2020).
46. Neumann, F. The analysis of the El Centro record of the Imperial Valley earthquake of May 18, 1940. *Eos Trans. Am. Geophys. Union* **1941**, *22*, 400–400. [CrossRef]
47. Powers, D.M. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*; Bioinfo Publications: Pune, India, 2011.
48. Metz, C.E. Basic principles of ROC analysis. In *Seminars in Nuclear Medicine*; Elsevier: Amsterdam, The Netherlands, 1978; Volume 8, pp. 283–298.