Integrity Monitoring of Vertical and Horizontal Structures via Visualization and Statistical Inspection of a Mesh Sensor Network

Ervin Lester G. Lu, Lessandro Estelito O. Garciano, Fritz Kevin S. Flores, Marnel S. Peradilla, Clement Y. Ong, and Arnulfo P. Azcarraga

***Abstract*—Studies have been conducted to understand how mild but frequent tremors affect the integrity of built structures. These have been done through the analysis of data gathered using tremor sensors. However, to be able to interpret the generated seismic graphs, current tools that are used to analyze such sensor data usually require in-depth knowledge of seismic data and expertise in structural engineering. Not only are the data difficult to interpret, the systems being used to gather and process these data also tend to be very expensive and rely on major seismic activities. This study presents a novel approach to monitoring and analyzing the structural integrity of buildings through the use of a mesh of sensors that are sensitive even to small movements of buildings. Data collected from these sensors are analyzed to identify specific areas of built structures that may have some structural defect. These identified anomalies may then be the subject of a more thorough investigation of the structure. Anomaly detection in the structures is done through the use of unsupervised machine learning techniques that estimate the expected movement readings of each areas as provided by the mesh of tremor sensors. By employing statistical tests, specifically the Kruskal-Wallis test based on the chi-square statistic, specific locations in a building are assessed as to whether there is likely to be a structural anomaly. Experiments were conducted using the actual 2013 earthquake from Bohol, Philippines, which were applied on simulated healthy and damaged buildings that were constructed using the ETABS simulation software.**

Manuscript received July 26, 2021; revised March 28, 2022.

This research was supported by the DOST-Philippine Council for Industry, Energy and Emerging Technology Research and Development (DOST PCIEERD Project No. 08625) in cooperation with the De La Salle University - University Research Coordination Office (DLSU URCO Project No. 08-IR-1TAY20-1TAY21).

Ervin Lester G. Lu is a postgraduate researcher of De La Salle University, Manila 0922 Philippines. (email: ervin_lester_lu@dlsu.edu.ph).

Lessandro Estelito O. Garciano is a professor of Civil Engineering Department, De La Salle University, Manila 0922 Philippines. (email: lessandro.garciano@dlsu.edu.ph).

Fritz Kevin S. Flores is an assistant professor of Computer Technology Department, De La Salle University, Manila 0922 Philippines. (email: fritz.flores@dlsu.edu.ph).

Marnel S. Peradilla is an associate professor of Computer Technology Department, De La Salle University, Manila 0922 Philippines. (email: marnel.peradilla@dlsu.edu.ph).

Clement Y. Ong is an assistant professor of Computer Technology Department, De La Salle University, Manila 0922 Philippines. (email: clement.ong@dlsu.edu.ph).

Arnulfo P. Azcarraga is a professor of Software Technology Department, De La Salle University, Manila 0922 Philippines. (email: arnulfo.azcarraga@dlsu.edu.ph).

***Index Terms*—Mesh Sensor Network, Statistical Inspection, Structural Health Monitoring, Structural Integrity, Visualization**

I. Introduction

SEVERAL studies were conducted in the past to help mitigate damages caused by earthquakes. However, the current earthquake detection methods do not predict the occurrences of earthquakes [1] and are designed only to detect and record moderate to large earthquakes [2]. Low magnitude tremors are often masked as seismic noise [3], [4]. However, although these low magnitude tremors may not have immediate impact on structures, the accumulated stress and strain over the years tend to weaken the structures [5].

There are two approaches of mitigating seismic risk. One is by predicting the occurrence and possible effects of earthquakes using previous historical data on present structures so as to prevent similar destruction, and another is by assessing the structural health prior to occurrences of earthquakes. The first approach of mitigating seismic risk is to understand the causes of tremors (natural or human-induced) by properly detecting and cataloguing earthquakes [2] with the use of the Geographical Information System (GIS) analysis [6] or with the use of waveform autocorrelation [2]. This approach aims to predict the occurrence of earthquakes by estimating the earthquake intensity, simulating several scenarios with different magnitudes, and identifying possible sources of earthquakes such as locating fault lines near the area of study [7]–[9].

After inspecting the geographical data about the area of study, the gathering previous seismic data in the area and identifying repeating earthquake patterns can be done to create a dataset of signal patterns to look out for in a full length continuous time series data [10], [11]. However, the detection capability of pattern matching using waveforms is restricted to the available data [12]. Numerous studies have been made to improve earthquake detection and location such as using clustering [13], artificial neural networks (ANN) [14]–[17], autoencoders [18], various deep learning methodologies [19]–[21], and convolutional neural networks [2].

The second approach is done by assessing the structural health of built structures prior to any occurrence of major earthquakes with the use of Internet of Things (IoT) sensors that provide real-time seismic health monitoring. Previous studies have explored the use of global positioning system

(GPS) [22] along with accelerometers [23] to keep track of displacement of floor segments during an earthquake. The data from the sensors are then used for computing the drift ratios which are the relative distances between two consecutive floors divided by the elevation of the two floors [24]. After computing for the drift ratios, the values are correlated to a force-deformation curve to estimate the state of damage of a structure. This approach could be adjusted to record ambient vibrations or micro-tremors instead of strong magnitude earthquakes.

In 2015, the Department of Public Works and Highways (DPWH) of the Philippines implemented guidelines and implementing rules on earthquake-recording instrumentation for buildings [25] which require all buildings included in the categories listed to install a number of sensors. This is to monitor significant changes in building response especially after a major event such as the "Big One". These guidelines along with several researches being conducted by civil and structural engineers are aimed to ensure the safety of building occupants by determining the conditions of buildings, and by collecting earthquake related-data during major seismic activities.

This paper describes the design and implementation of a tool for the structural health monitoring (SHM) of vertical and horizontal structures based on visualization and statistical tests of the signals recorded by a mesh of motion sensors. The full blown system entails deployment of a huge number (organized as a mesh) of small and cheap tremor sensors covering a structure. For now, while awaiting the deployment of such a mesh of sensors, the data are extracted from an earthquake simulator that is also used to create building scenarios, i.e. healthy and damaged buildings. The data are visualized to track even the minute structural movements under different magnitudes of stress. These time-series data are then analyzed using a combination of autoencoders (machine learning models) and statistical tests to identify probable anomalies in the structure that may need to be further investigated.

II. Structural Health Monitoring

Several studies have been made with regard to monitoring the structural integrity of built structure such as of buildings, bridges, and flyovers [26]–[29]. These studies have made use of earthquake sensors and different methods have been used to collect and interpret information from these sensors. Most of the previous studies made use of combinations of accelerometers, temperature sensors, humidity sensors, and piezoelectric sensors for analyzing structural health [30]–[33]. There were also several studies which explored wireless methodologies to assess structural health such as using acoustic emission monitors which make use of sound waves, and using IoT sensors.

A. Data Collection Methodologies

A study by Syafrudin et al. [34] discusses the use of IoT sensors for providing real-time information for better understanding and performance of environment condition monitoring, smart buildings, and healthcare applications. In addition, Qing et al. [30] discusses different methods of implementing structural health monitoring on aircrafts and have used piezoelectric materials for their SHM system. The said SHM system consists of both mounted and embedded sensors which monitors the structural state of the materials or the member of the structure where the sensors are attached by retrieving information such as load and temperature data. As with most IoT implementations, the numbers and types of sensors, and the sensor positioning critically affects the sensitivity and performance of the SHM system. One way of minimizing this issue is applying the Stanford Multi-Actuator-Receiver Transduction (SMART) layer technology [35] developed at Stanford University. The SMART layer technology is a simple and efficient way of integrating large sensor networks onto structures with high reliability and lasts throughout the service life of the structure.

Saoudi et al. [31] demonstrates the use of different data mining techniques that could be applied to wireless sensor networks (WSN) for early damage detection specifically for forest fire detection. The system involves a clustered WSN where each sensor node individually decides on detecting fire using a classifier of data mining techniques. The system makes use of a simulation software called CupCarbon Simulator to design, visualize, debug, collect data, simulate various scenarios, and validate the different algorithms. The simulator is commonly used for conducting studies about Smart Cities and IoT WSN. The system is based on measuring and combining real data from different sensors and using the Naïve Bayes classifier applied to the dataset for fire detection. A node detects fire locally by itself, discards normal data, and transmits only abnormal or out of the ordinary (OOTO) values to a central node for further analysis. Based on their results, the different data mining techniques are able to reduce data size, delete redundancy, improve WSN speed, and decrease network traffic.

B. Existing Structural Health Monitoring Approaches

Cai, Cheng, and Liu [27] compares the traditional ultrasonic testing methods and the proposed nonlinear electromagnetic acoustic testing method for tensile damage evaluation. The traditional ultrasonic testing methods make use of linear theory and depend on measuring several parameters such as size, orientation, location of cracks, acoustic velocity, attenuation, transmission, and reflection of ultrasonic waves to detect damage. However, this approach is not sensitive to early stage degradation and microcrack of materials. The previous nonlinear approach to this problem would make use of nonlinear piezoelectric methods which are wired connections of sensors, while the proposed approach by the researchers makes use of nonlinear electromagnetic acoustic waves. This provides a wireless approach to the problem which is also not susceptible to varying surface conditions. Both approaches are used to assess the relative nonlinearity parameters of the damage. The nonlinear method which make use of acoustic waves is considered to be more robust as compared to the traditional ultrasonic method as it can match the performance of the traditional method as well as characterize microstructural features in materials; and quantifies them with the measured acoustic nonlinearity parameter (ANP) which are caused by the interaction of sinusoidal waves and microstructural features like microcracks.

Ding, Shen, and Du [28] analyzed the strain data from various sensors to study the stress and fatigue experienced by a structure across its operational lifespan. Based on their study, temperature-induced strain contributes little to the overall stress of a structure. However, conventional sensors cannot differentiate which data are of lesser significance than others which is why the researchers proposed an empirical mode decomposition method to remove noise and temperature-induced strain from the dataset, thus leaving only the dynamic strain response needed for strain analysis. Based on the gathered data, the frequency characteristic is obtained from the dynamic and raw strain data, and a statistical analysis of the data is performed with the prior assumption that the extracted dynamic stress peaks and valleys are normally distributed. The results from their experiments show that the expected maximum values from the statistical analysis are near the measured maximum values at different heights. As such, the proposed method of Ding, Shen, and Du [28] has significant advantages over traditional parametric-based methods and wavelength-based methods as it is adaptive; and timescale-based decomposition properties can be incorporated in the analysis and can be used for nonlinear and nonstationary applications.

As far as tremor sensors are concerned, Kong et al. [32] used a low-cost sensor network approach in monitoring structural health by making use of off-the-shelf smartphones as substitute for accelerographs. The smartphones were used to collect movement data during a simulated earthquake and were compared to the data collected by Episensor, which is a commercial data logger for earthquake tremors that sends data to the Southern California Seismic Network. The results from their experiments demonstrate that the proposed system is able to collect similar data as compared to those gathered by Episensor. The proposed system may not have as accurately recorded low power readings as those of Episensor, but the system was able to match accurately the modal peak readings.

Another study on the design of a system of tremor sensors, Uy et al. [36] developed a system for structural health recording system called Universal Structural Health Evaluation and Recording (USHER) system. The system is intended for buildings to comply with the local regulations in the Philippines prior to issuance of a occupancy permit [25]. The proposed system is composed of an accelerograph sensor and web-portal system, displaying the real-time readings of the sensors in a structure through its web-portal. However, the system lacks the necessary analysis and feedback to users. Instead, analysis and feedback are done by partner structural engineers by manually interpreting the data gathered by each sensor.

C. Machine Learning for Structural Health Monitoring

According to the study of Chang and Lin [37], advancement in IoT web technologies, and wireless integrated sensor devices paved the way for real-time monitoring on various structural behaviors which can be used for damage warning. Their study explores the use of a visualization tool where users can create a three dimensional (3D) model of a structure where sensors would be placed so as to visualize the dynamic movement of the columns under different magnitudes of ground shaking. The 3D displacement values are measured at every timestamp by the sensors, which are located at each node of the structure. The sensor positions on the 3D model are then updated depending on the translational values from the sensors. The assumption is that columns of the structure are rigid bodies (there are no deformation during rotation and translation), and the floors or beams can extend along with the moving direction without lateral deformation or bending. Their approach also has the capability to zoom in and out, and rotate the structure to further inspect the effects of the movements on the structure.

SHM indeed involves 1) continuous monitoring of a structure over time based on sensor data that represent dynamic structural responses, 2) signal or data processing and analysis, and 3) proper decision making to infer the current health state of the structure [38]. Vibration-based SHM is one of the popular approaches to understanding SHM because vibration is naturally available, be it naturally occurring or human induced. The assumption with most vibration-based SHM is that small changes in a structure cause corresponding changes in the structural dynamics which may indicate possible damage.

To interpret the vibration data gathered by the sensors, statistical time series methods are usually applied to the dataset. Statistical time series methods present themselves as data-based models rather than physics-based. This offers several advantages compared to other forms of analysis [39], [40] such as: a) no need for analytical models such as finite element (FE), b) not needing structural models as some applications may just need a single pair of excitation and vibration response signals for creating the statistical model, c) inherent accounting for uncertainties (measurement, environmental, operation, etc.) through statistical tools, d) statistical decision making with specified performance characteristics, e) effective operation even in low frequency range, and most importantly f) effective use of natural obtained random vibration signals which does not disrupt the structure’s normal operation.

Statistical time series methods, however, do have some limitations which include being able to detect only the damage to the extent allowed by the model used, and would require more training data to cover as much behavior as possible for a “normal” or healthy building state.

There are multiple studies conducted related to the use of low cost accelerometers and machine learning for structural health monitoring. A report by Acevedo [41] discusses an implementation of Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN) in detecting residual displacement using displacement sensors and accelerometers. Experiments show that FFNN and RNN are able to predict ground motions for low magnitude tremors (magnitude 1 to 6) but fail for higher magnitudes (greater than magnitude 7).

Meanwhile, Kusumo et al. [42] studied about differentiating human activities and earthquake vibrations from smartphone accelerometers using Long Short-Term Memory (LSTM). Their experiments focused on collecting accelerometer data from both human activities and earthquakes, then using LSTM to be able to classify and identify the kind of activity or if an actual earthquake is occurring. Although their data is quite imbalanced due to the small amount of earthquake data available, their results show that the system is able to recognize every kind of vibration with an accuracy of 97% on the 20% test data. The findings are significant as it implies that human activities and natural earthquakes have different signatures and that they are distinguishable using only smartphone accelerometers.

Kong et al. [32] also used machine learning in seismology, having implemented various machine learning techniques (classification, regression, and clustering) in the different areas of seismology which includes earthquake detection and phase picking, early warning, and ground-motion prediction. Their study also made use of the accelerometers in smartphones then applied the different machine learning techniques to the generated dataset. They concluded that a hybrid approach which combines machine learning methodologies and traditional physical modeling would be viable for assessing SHM as machine learning alone may have difficulties due to poor sampling, and noisy and incomplete geophysical data.

The research by Liu et al. [43] implemented a generative adversarial active learning for unsupervised outlier detection. The study focused on being able to determine outliers from a dataset, without any prior knowledge or information. The study demonstrates the possibility of differentiating out-of-the-ordinary situations, such as seismic activities, from natural occurring vibrations. A similar study was conducted by Mousavi et al. [44] which implemented an earthquake detector using both convolutional neural networks (CNN) and recurrent neural networks (RNN). The detector consisted of a combination of convolutional layers and LSTM trained with seismograph data which recorded 3D movements for a specified duration.

III. Design Considerations

Due to the absence of open access data that can be used for the study, we opted to generate own synthetic but realistic data using real recorded earthquake data from the Philippine Institute of Volcanology and Seismology (PHIVOLCS). The earthquake data is from the October 15, 2013 Bohol earthquake which was a 7.2 magnitude earthquake on the Richter scale and an intensity VII earthquake based on the PHIVOLCS Earthquake Intensity Scale. The earthquake resulted in the damage of over 79,000 structures which included homes, roads, churches, schools, and public buildings. 14,500 of the said damaged structures were completely destroyed. The earthquake recording lasted for 400 seconds. The earthquake was parallel to the x-axis of the building thus had the greatest effect on the said values and would be the focus for the experiments.

The sample structure is created using Extended Three-dimensional Analysis of Building Systems (ETABS), a civil engineering simulation software commonly used to design, model, and test structures according to different building codes [45]. ETABS has been used by several studies [46]–[50] to simulate various scenarios and building designs and has been comparable to real world setting. The structure that was configured using ETABS is a 20-floor regularly shaped building created with waffle slabs (reinforced concrete) made to resemble modern buildings. Each floor is assumed to have eight sensor locations from where the tri-axial displacements are extracted. Fig. 1a shows the 3D building plan of the sample structure and Fig. 1b shows the floor plan of the structure.

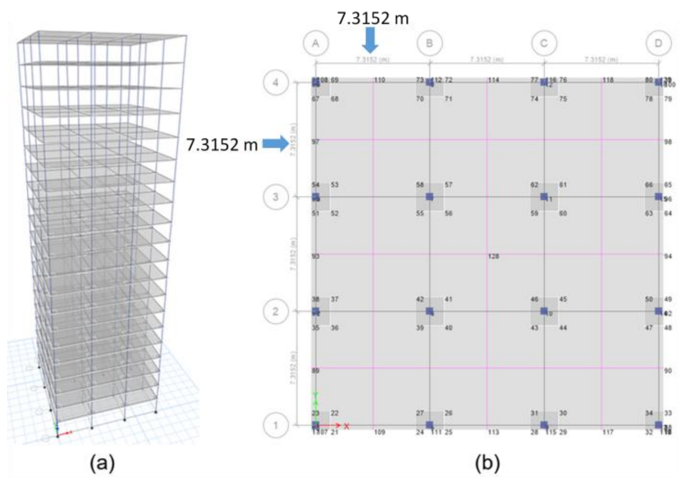


Fig. 1. a) Building three-dimensional plan. b) Building floor plan.

For the experiments and visualization discussed in this paper, three sample buildings were constructed using ETABS. One healthy building was constructed with uniform materials all throughout the floors, and two damaged buildings were configured that had some weakened columns to simulate the weakening or damage of materials either due to faulty design or to the effect of natural phenomena leading to building damage such as tremors, as well as weather elements like rain, wind, and sun. Damage is simulated by lessening the column stiffness of some portion of the building to 25% of the normal stiffness value, thus making them more vulnerable to tremors. For this research, the damaged portions on the two damaged buildings are located at the southwest corner of the building with one located at the 10th floor and the other at the 5th floor. Indeed, the possibility of experimenting with different design and structural defect scenarios is an important advantage of using a simulator like ETABS.

IV. Civil Engineering Analysis

Using the built-in analysis module of ETABS, it can be seen from Fig. 2 that the maximum storey-drifts of the two damaged structures show anomalies at the damaged floor marked by the green circles. These anomalies are manifested as discontinuities in the drift readings going from the ground floor all the way to the top-most floor of the building.

The discontinuities in the storey-drift reading at the bottom floor of all three buildings, including the healthy building, is due to how the buildings were modeled and how ETABS interprets them. Drift readings are indeed a good basis for detecting possible structural anomalies, but for the rest of the experiments discussed in this paper, the focus will be on the actual x, y, and z-axis displacements of the different parts of the building, at a more granular level to include the different points in a given floor of the building; instead of just the drift readings of the entire floor. This is so that whenever possible anomalies are detected, the actual location of the likely source of the structural anomaly can be clearly identified.

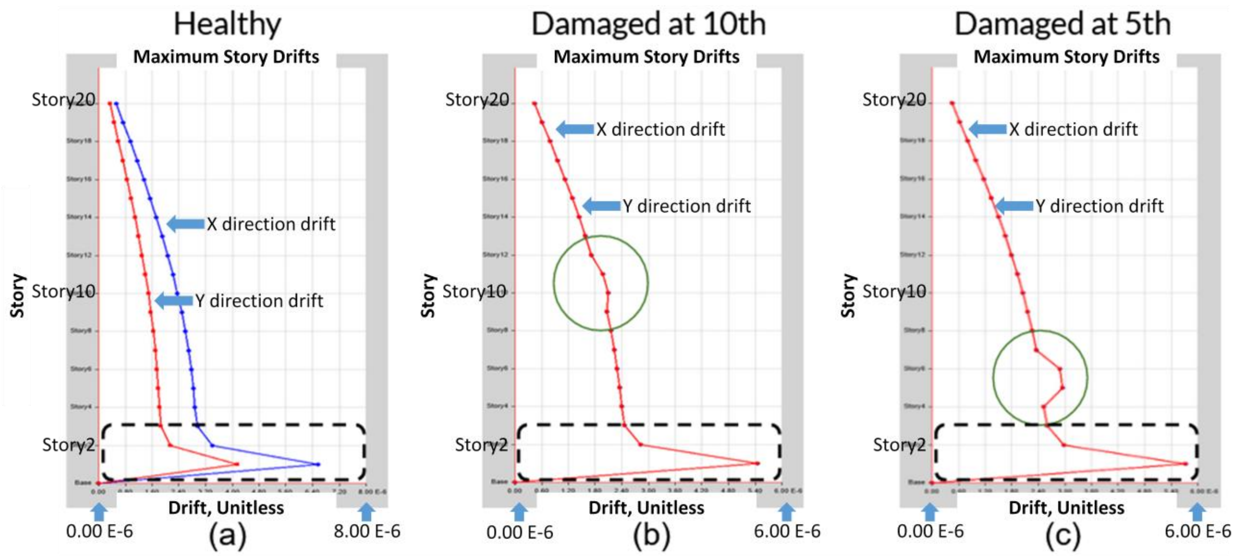


Fig. 2. a) Maximum storey drift of the healthy building. The blue plot is the x-direction drift and the red plot is the y-direction drift. b) Maximum storey drift of the damaged building where damage is located at the tenth floor. c) Maximum storey drift of the damaged building where damage is located at the fifth floor.

V. Preliminary Data Visualization

For this study, the proxy used for the health of buildings is the smoothness of the displacement curves or absence of discontinuities in the movements of the numerous location points of a solid structure. Since the ETABS configuration was implemented in such a way that the earthquake was positioned parallel to the x-axis of the building, most of the analyses are conducted on the x-displacement values. Fig. 3 shows the x-displacements plotted against the different floors of each of the buildings at a certain time instance. Looking closely at the plots, one would notice that a healthy building shows a relatively smooth curve as compared to the damaged buildings. The damaged buildings have a hardly noticeable deviation happening somewhere at the tenth floor for the orange plot, and somewhere at the fifth floor for the green plot.

Fig. 4 shows the zoomed-in view at the tenth floor which reveals that there is indeed a slight deviation that can be seen occurring at the tenth floor mark for the damaged building as highlighted by the widening gap from the floors below the damaged portion. Similarly, Fig. 5 shows the zoomed-in view at the fifth floor which reveals that there is also indeed a slight deviation at the fifth floor mark for the damaged building.

To better observe the lateral and vertical displacements, we plot the values in a 3D chart with the x and y displacements on the x and y-axis, and showing the different floors of the building on the z-axis of the chart (cf. Fig. 18).

To see how vibrations propagate from the ground floor up to the top floor, Fig. 19 displays the x-displacements of the different floors (signified by the shifting colors: violet for the first floor, red for the 19th floor) during the early stages of the earthquake. From the plots, all three buildings exhibit expected behavior of low displacement values for the lower floors that gradually increase as the floor level increases.

However, the plots of Fig. 20, for the time instance right after the highest peak of the tremor, reveal that the movements of damaged structures' tremors last for a longer period (signified by the wider wavelengths) as compared to the healthy building. The red vertical lines show the arbitrary marks indicating the position of the healthy building. We then investigated whether troughs or amplitudes are actually longer for damaged buildings. Longer amplitudes reflect the larger displacement values as seen in previous plots.

Fig. 21 shows that, indeed, amplitudes are longer for damaged buildings. It can be seen from Fig. 21a that the peak of the highest floor has a value of around negative 1.9 millimeters. While looking at Fig. 21b and 21c, it can be seen that two and three floors respectively were yet to complete their peak values. The red vertical lines are the same as the ones found in Fig 20. These show that the damage on the structures (5th floor or 10th floor) affected the wavelengths and amplitudes of the building's response to the tremor.

We then take the first derivative (i.e. to compute for the velocity) of the displacements as this emphasizes the difference between the buildings, compared to simply plotting the raw displacements over time. Fig. 22 highlights the densities of the velocities of each building. Highlighted by the blue rectangles in the figure, it can be seen that the damaged buildings (Fig. 22b and Fig. 22c) have denser clusters compared to the healthy building (Fig. 22a). A dense cluster indicates a long period without a change in displacement values. This supports the previous explanation of damaged buildings having wider displacement wavelengths.

Fig. 23 highlights the comparison between the peak velocities of each building. Highlighted by the blue circles in the figure, it can be seen that the highest floor (signified by the red plot from the rainbow colored plots) of the healthy building (Fig. 23a) already peaks at a value of negative 2.5 millimeters per second which is slower compared to the damaged buildings (Fig. 23b and Fig. 23c) that have yet to reach their peak velocities. The peak velocities reflect the maximum displacement amplitudes because the greater the displacement over a period of time, the faster or larger the velocities. Looking closer at Fig. 23b and Fig. 23c, it can be seen that there are still light colors (shades of yellow and orange) near the peak values of the healthy building. This supports the previous explanation of several floors in the damaged buildings that were yet to reach their peak amplitudes.

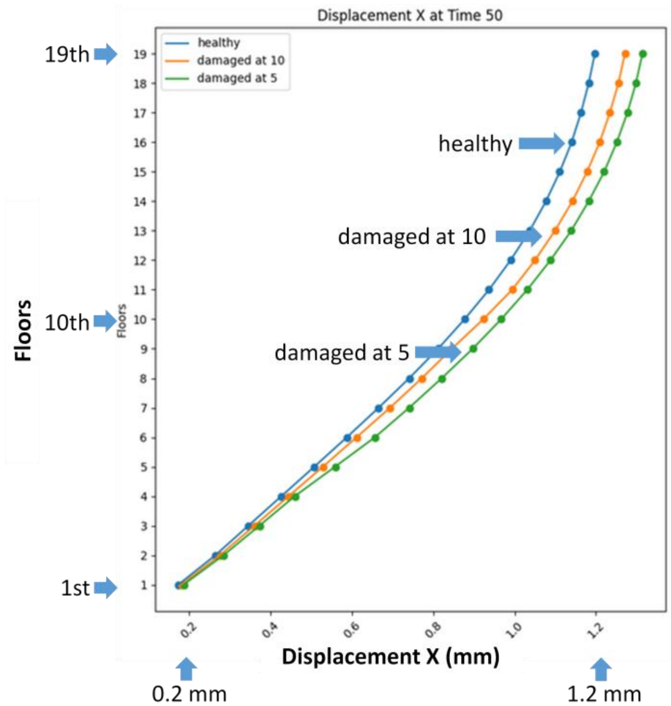


Fig. 3. X-displacements plotted against the different floors of each of the buildings at earthquake timestamp of 50th second.

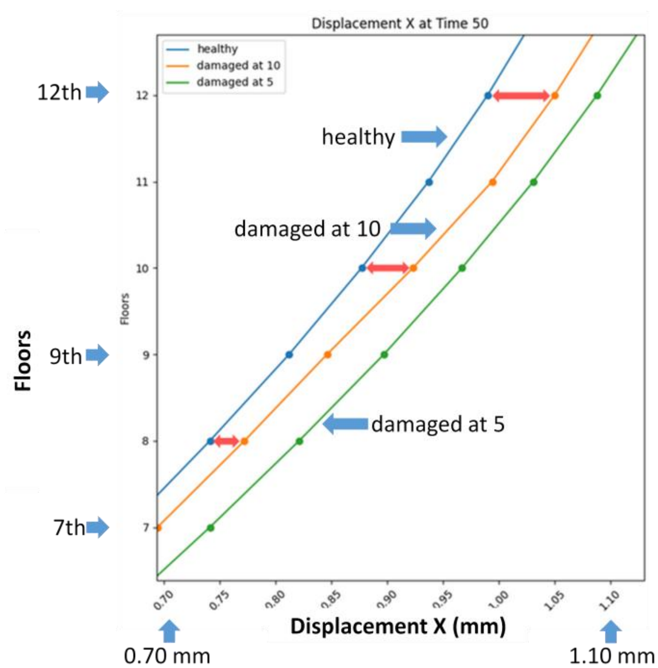


Fig. 4. Zoomed view of the tenth floor and neighboring floors.

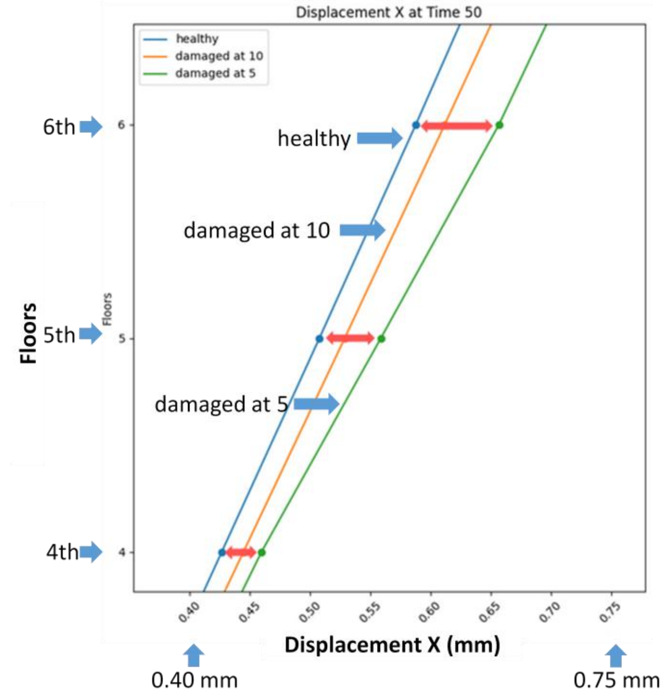


Fig. 5. Zoomed view of the fifth floor and neighboring floors.

VI. Robust Data Visualization

It must be emphasized that visual evidence is not sufficient to conclude anything about the general structural health of a building. A statistical analysis is needed to quantify the findings, and indeed, would also be able to ascertain whether any noted deviations are statistically significant (or may just be attributable to chance).

The preliminary visualizations were useful for visualizing the nature of the data. However, making use of expected values to which the observed values could be compared using statistical tests would give a more solid basis for determining the health of a built structure. This robust data visualization (i.e. guided by statistical tests) makes use of expected values, mesh inferred values, and statistical tests to quantify the observations.

A. Naïve Expected Value

Since the health of a building is manifested by the smoothness of transitions from floor to floor, we proceed by computing for the deviations between the observed displacement values and the expected values based on the observed values of the neighboring points in the mesh. Note that the tremor sensors are systematically positioned all throughout the building and are regularly placed. The assumption is that if the observed value differs significantly from the expected value, then there is a probable problem at that location of the building. Since we are dealing with rigid, physical structures, it is reasonable to assume that the displacement readings of a given floor can be expected to lie in between those of the floors below and above a given floor. The floor boxed in red in Fig. 6a is the target floor for which the expected value is being computed using the readings of its neighboring floors. The simplest way of computing an expected value of a target floor is by getting the average reading from floor above and floor below. This expected value is referred to as the *naïve expected value*.

However, upon looking at the plots on Fig. 18, it can be seen that floors tend to move in groups of three as highlighted by the red circle. So computing the *naïve expected value* could then be expanded into getting the average of averages of three floors above and below a target floor as shown in Fig. 6b.

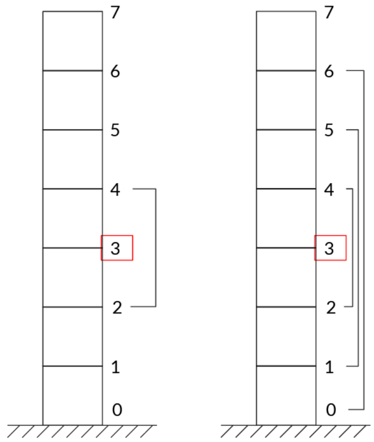


Fig. 6. A visual representation of computing an expected value for a target floor using the observed values of its neighboring floors. The red box highlights the target floor for which the expected value is computed for.

A general equation of computing the *naïve expected value* is shown in (1) where n is the number of floors above and below to be considered, E_i is the expected value of the target floor i, O_{i-k} is the observed or raw sensor reading of k floors below the target floor, and O_{i+k} is the observed or raw sensor reading of k floors below the target floor.

$$E_i = \frac{1}{n}\sum_{k=1}^{n}\frac{O_{i-k} + O_{i+k}}{2} \tag{1}$$

We then plot the *naïve expected values* with the observed raw values to see if the *naïve expected values* can truly be the basis for identifying damaged portions of buildings. Fig. 24 shows the different *naïve expected values* computed for the different buildings. We superimpose the observed values over the expected values computed with two different n values using (1). As can be seen from the plots, the damaged portions of the buildings are not identifiable just by getting the averages of the readings from neighboring floors.

Because of the limitation of the *naïve expected value* to indicate the presence of structural anomalies, we then formulate another method that makes use of autoencoders to generate a more detailed expected value – still based on the notion of a mesh of tremors wrapped around a structure.

B. Autoencoder

An autoencoder is a type of neural network used to learn data codings in an unsupervised manner that aims to reproduce an output as close as possible to its input. An autoencoder has three main parts, encoder, latent code, and decoder. The encoder takes in inputs and encodes them into a reduced dimension, referred to as the latent code. The latent code is a lower dimensional representation of the input for which the decoder tries to decode from to get back the original input.

The main use for autoencoders is to minimize reconstruction errors between the input and output for the autoencoder to learn important features in the dataset. Fig. 7 shows a basic block diagram of an autoencoder.

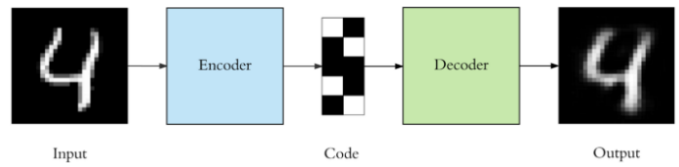


Fig. 7. Basic Autoencoder Block Diagram.

In this study, the autoencoder is used as an associative memory where the autoencoder interpolates information from the input data based on the closest pattern that the model is aware of. To do this, the model is trained with all the "healthy" building data that are currently available so that the machine can learn key features from the healthy dataset. It is important to train the model with only healthy data so that when the autoencoder evaluates the inputs, it will be able to produce an output based on what it expects the values to be *if the inputs were from a healthy building*. Fig 8 shows the block diagram of the autoencoder being used as an associative memory. The expected value generated by the autoencoder is referred to as the *mesh inferred value* (MIV).

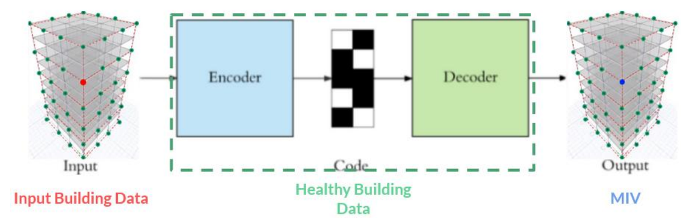


Fig. 8. Block diagram of the trained autoencoder (enclosed with the green dashed rectangle) used as an associative memory. MIV or mesh inferred value refers to the expected value generated by the autoencoder.

For this study, a finer distribution of sensors on a building forming a mesh is envisioned to help localize the problem on a structure. However, due to the limitations of the simulation software used to generate the dataset, we are only able to generate values gathered on a floor and not on the walls of the structure. This implies that we need to interpolate some of the neighboring data of a target sensor.

It is assumed that getting the reading on the walls of the structure forming a mesh is better rather than getting readings from corner-to-corner or floor-to-floor as they may be too far apart and values may differ by a greater magnitude. Fig. 9 shows the naming convention used to label data points and how the mesh of sensors is envisioned to wrap around a structure.

Since we are using a simulator to generate data from real earthquake signals for training our model, we are limited by the capabilities of the tool. The ETABS simulator returns the triaxial readings of each floor at each labeled point as listed in Fig. 9a. However, since the building is a rigid structure, it can be assumed that the point between two floors should follow the same behavior as the floor above and below it. By getting the midpoint between the readings of the floor above and the floor below the target point will result in a hypothetical value which we could use to estimate the possible readings as if they were also extracted from the simulator. An example would be getting the point between Floor 1 and 2 from Fig. 9b. The midpoint value can be calculated by getting the values from a certain axis (x-, y-, and z-axis) one at a time from Floor 1 and 2 and getting their average. This process is repeated for the other two axes to complete the interpolation. This problem is eliminated once actual sensors are deployed on existing structures to periodically gather data as we will be using real mesh data to train the autoencoder.

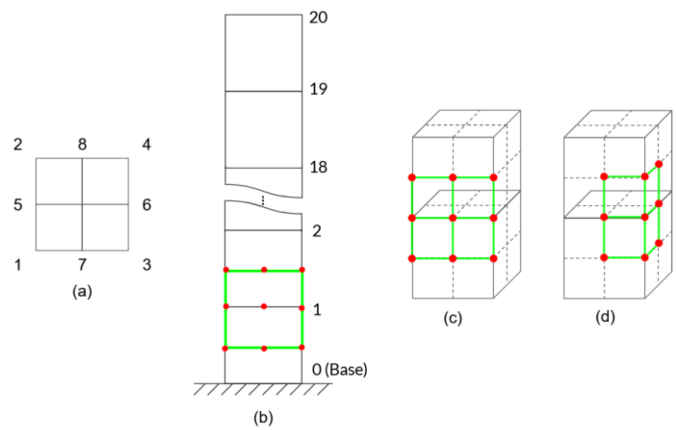


Fig. 9. Naming convention used to label the data points. a) Floor view of the structure. Each number represents a sensor location that data could be extracted from. b) Floor numbers. The base of the building was given the floor number of zero. c) Mesh covering a wall where the center node is the target node. d) Mesh covering a wall where the center node is located at the corner of the structure.

The PyTorch library is used for implementing the autoencoder [51]. The implementation is straightforward where inputs to the model are mapped according to the features of the dataset. The most important component in the implementation is determining the size of the latent code (bottleneck). Several architectures were considered based on the size of the latent code and number of input features.

Architectures considered based on the size of the latent code includes *undercomplete* and *overcomplete* autoencoders. An *undercomplete* autoencoder is an autoencoder where the latent code dimension is less than the input dimension which captures the most features of the training data. The *overcomplete* autoencoder on the other hand has a larger latent code dimension compared to the input dimension which can be useful for uncovering key information for some applications. However, *overcomplete* autoencoders have the tendency of copying and memorizing the inputs without actually learning.

Architectures considered based on the size of the input dimension includes the use of input size ranging from 3 up to 27. The autoencoder which made use of three inputs only had the three raw triaxial displacement values. The autoencoder which made use of nine inputs had the triaxial values of the target floor, the floor above it and floor below it. The autoencoder which made use of 27 inputs had the triaxial values of eight neighboring sensors and the triaxial values of the middle or target sensor node. In other words, the autoencoder with an input size of 27 contains the interpolated values of the wall area between two corners of a given floor, in anticipation of real-data that would come from a mesh of tremor sensors once they are fully deployed.

The mean squared error (MSE) losses from training the models are calculated to evaluate the different autoencoder architectures. The mean squared error is defined as the average of the square of the difference between actual and estimated values. Fig. 10 shows the MSE loss plots of the different autoencoder architectures considered. The closer the MSE loss value is to zero, the better the performance as it has the best success in reproducing the input training set. As it can be seen from the plots, the 27-7-27 autoencoder performed the worst among the different architectures, and the 27-14-27 autoencoder performed the best. However, as mentioned earlier that autoencoders have a tendency to memorize information and not learn useful features from the training dataset when the latent code is still big, the 27-14-27 autoencoder may be performing better as it may have been simply outputting what it had previously known from training. With these said, it seems that only the 27-3-27 or the 27-14-7-3-7-14-27 autoencoders are the viable models. Fig. 11 shows the MSE losses of the two architectures considered. As can be seen from the plots, the autoencoder with more hidden layers did not offer much improvement in performance. As such, the simpler 27-3-27 autoencoder is used for the succeeding experiments discussed in this paper.

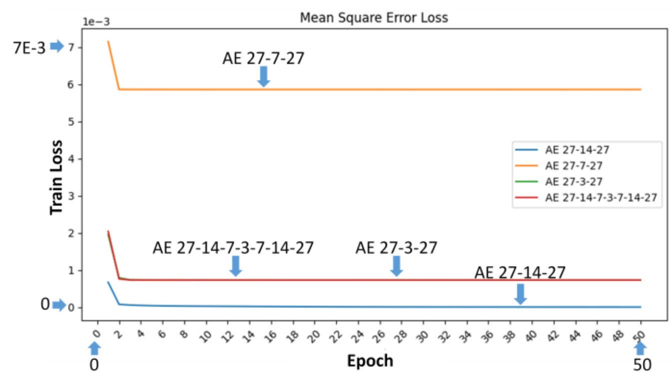


Fig. 10. Mean square error losses of the different autoencoder architectures considered for the study with only a single hidden layer.

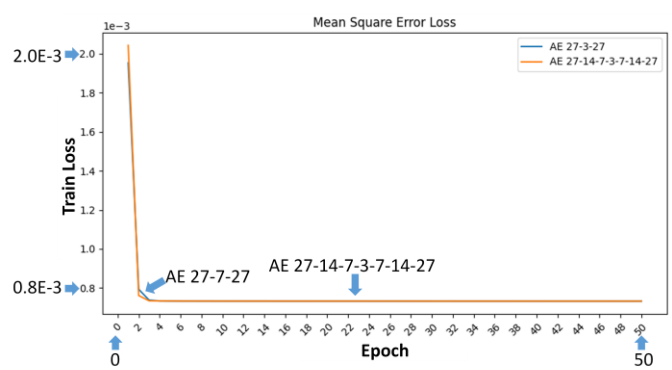


Fig. 11. Mean square error losses of the last two autoencoders considered for the study with varying number of hidden layers.

The use of *mesh inferred value* seeks to pinpoint possible locations of structural anomaly, if any, by comparing the actual observed reading of a given target location to its MIV, which in turn is based on the actual observed readings of the specific points surrounding the target location.

Fig. 25 shows the results of the autoencoder compared to the observed values. We superimpose the observed values over the *mesh inferred values*. As can be seen, the *mesh inferred values* are able to identify the damaged part of the buildings as the discontinuity is more conspicuous compared to those of the *naïve expected values* discussed earlier. Given that the use of the MIV is able to locate possible anomalies in the structure, the *mesh inferred values* are then used for the succeeding experiments discussed in this paper which involve the use of more stringent statistical tests that will give some basis for the confidence in tagging a specific location as probably defective due to some structural defect that is not present in the areas surrounding this tagged spot.

C. Statistical Tests

Nonparametric tests are the preferred statistical approach used in this study as they do not assume a normal distribution or a Gaussian distribution for the samples. This is important as the nature of the data is not fixed and varies depending on the disturbances that a building experiences during the data collection period [52].

Several statistical tests are considered for the study, such as the chi-square statistic, Mann-Whitney test, Wilcoxon signed rank test, Kruskal-Wallis test, and the Kolmogorov-Smirnov test.

The test statistics are computed for each floor comparing the observed and mesh inferred values. The test statistics are then plotted to see if they are able to isolate the damaged portion of the building by measuring how near or far the computed statistic is from the mean of the group. For the succeeding figures, the "blue" vertical line indicates the mean of the test statistic values, the "green" vertical lines indicate half standard deviation away from the mean, and the "red" vertical lines indicate one standard deviation away from the mean.

The first statistic used is the chi-statistic. The chi-square statistic is used to quantify the difference between a set of observed values and a set of expected values. A low value statistic indicated a high correlation between the two sets of data. Equation (2) shows the general formula in computing the chi-square statistic. In the equation, X^2 refers to the chi statistic, O_i refers to the ith observed value in the sample, and E_i refers to the ith expected value in the sample.

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2}$$

The chi-statistic indicates whether the observed values are statistically different from the mesh inferred values from the autoencoder. Fig. 26 to 29 shows the results of computing the chi-statistic at different time instances of the earthquake. Fig 26 shows the chi-statistic for the different floors before the major peak of the earthquake. Clearly, the healthy building (Fig. 26a) has a smoother curve compared to the damaged buildings. Fig. 27 shows the chi-statistic for the different floors during the peak of the earthquake. As can be seen, the curve of the healthy building (Fig. 27a), compared to those of the damaged buildings (Fig. 27b and Fig. 27c), is relatively smoother compared to the curves with clear discontinuity at the 5th and 10th floors of the damaged buildings. Fig. 28 shows the chi-statistic after the peak of the earthquake. The floors would have fully reacted to the tremor of the earthquake at this point in time. As with Fig. 26, the healthy building exhibits a smooth curve compared to the jagged curve of the damaged buildings. A similar pattern is also observed with Fig. 29, which is when the tremor is beginning to wane.

The next statistic considered is the Kruskal-Wallis statistic. The Kruskal-Wallis statistic is the nonparametric equivalent of the one-way analysis of variance (ANOVA) for comparing two or more independent samples and determining whether they have the same distributions or at least one sample is different from the rest. Like with ANOVA, the test statistic only indicates whether medians of the samples are equal or not. Locating the samples that lead to the rejection of the null hypothesis of the test statistic is done by computing the Mann-Whitney over two samples at a time. Equation (3) is used to compute the test statistic. In the equation, H is the test statistic, N is the combined number of values from all the samples, R_i is the sum of the ranks from a particular sample, and n_i is the number of values from the corresponding rank sum.

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1) \tag{3}$$

Similar to the chi-statistic, the aim for this statistic is to see if the observed values are statistically different from the mesh inferred values from the autoencoder. Fig. 30 to 33 shows the results of computing the test statistic at different time instances of the earthquake. Fig. 30 shows the statistic before the peak of the earthquake. It can be seen from the figure that discontinuity on the curve determined the damaged floors in the structure. This discontinuity is further highlighted during the peak of the earthquake as seen in Fig. 31.

VII. Three Dimensional Structure Visualization

A 3D visualization tool is also developed to understand the micro-tremor readings. It incorporates the needed information that engineers want to see from the sensor reading, such as the acceleration, velocity, and displacement of a given point, and a 3D rendering of the sensor nodes in a 3D space. The tool allows the user to pan, move, rotate, and zoom around the structure for better visualization of the movements. The tool was created using Pygame [53] and PyOpenGL [54] libraries which are libraries used for game development in Python.

The tool can be easily modified to accept pre-designed buildings or structures from a CSV file containing the sensor's x-, y-, and z-coordinates. For this implementation, we developed a function that automatically creates a regular or rectangular building with four sensor locations per floor for faster prototyping. The function accepts a structure's height and width ratios, the number of floors, the actual floor height for the drift ratio computation, and a magnitude multiplier to see how the building will move if the sensor readings were a magnitude greater than the original values. Fig. 12 shows a proportionally designed building where the height and width are equal. Fig. 13 shows a tall and slender building where its height is larger than its width similar to buildings with high ceilings. Fig. 14 shows a wide building where its width is larger than its height.

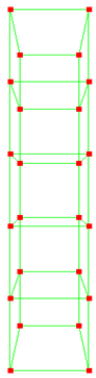


Fig. 12. Sample proportional building designed using the structure visualization tool.

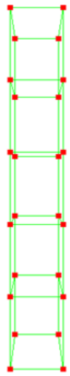


Fig. 13. Sample tall and slender building designed using the structure visualization tool.

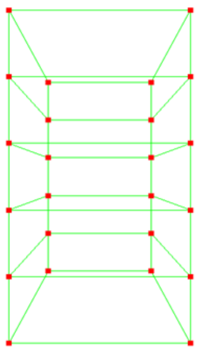


Fig. 14. Sample small and wide building designed using the structure visualization tool.

A graphical user interface (GUI) is also incorporated which allows structural engineers to view the sensor readings (i.e. acceleration, velocity, and displacement) within the tool. The graphical user interface was created using the Tkinter library which is a library that allows the creation of a graphical user interface form within the Python environment. The GUI allows the selection of the target floor and group with dropdown menus for easy navigation. The plots displayed can be zoomed in or out as needed as it uses the Matplotlib library for plotting the sensor readings. Fig. 17 shows a sample plot of Group 1 Floor 1 sensor.

The visualization tool also allows the visualization of the actual movements of the sensor nodes. This is an important feature as engineers want to see the modal shape or behavior of the building under different kinds of excitation force. This is implemented by updating each point of the structure based on the individual CSV files of each sensor node containing the x, y, and z-displacements per time instance. By pressing down on the spacebar on the keyboard, the tool reads and updates each point sequentially resulting in a dynamic visual representation of the sensor data. Fig. 15 shows how the raw data values manifest on the structure, while Fig. 16 uses the magnitude multiplier with a value of three which shows what happens when movements three times the original movements based on a certain were acted on the building.

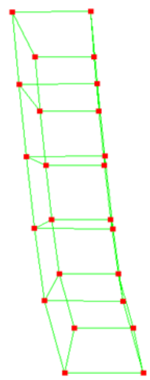


Fig. 15. Sample dynamic visualization of how the raw values manifest on the structure.

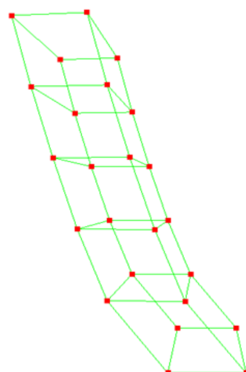


Fig. 16. Sample dynamic visualization of how the building moves if the displacement values are three times the original raw values.

The last component to be incorporated within the tool is the computation of drift ratios. The drift ratio is the difference of displacements of the floor above and below the floor of interest and normalized by the height of each floor. The threshold levels used in the implementation was based on Table I [23]. The computed drift ratio is outputted on the Python console along with the simulation of the moving building. A sample run of this is shown in Fig. 34 and Fig. 35. Fig. 34 shows a "low activity" drift ratio or normal behavior for the building as the floors move in the same direction, while Fig. 35 shows a "medium activity" where it can be seen from the accompanying simulation that a small bump occurred in a corner of the building.

TABLE I
DRIFT RATIO THRESHOLDS

Threshold Stage	1	2	3
Suggested Typical Drift Ratios (%)	0.2 – 0.3	0.6 – 0.8	1.4 – 2.2

VIII. CONCLUSION

We present the design and implementation of a set of tools for the structural health monitoring (SHM) of vertical and horizontal structures based on visualization and statistical tests of the signals emitted by a mesh of motion sensors. To accomplish this, we implemented a tool that structural engineers can use to visualize the individual sensor readings as a graph or by looking at the three dimensional rendering of the structure. The tool allows the users to visually see how the individual readings affect the movement of the structure by simulating how a force acts on a structure based on sensor readings.

We also explored the use of multiple sensors placed on each floor to be able to localize the problem. This may not be possible using the current methods of analyzing structural health monitoring data. By combining the readings from multiple sensors that are wrapped around a structure, it is feasible to locate the damaged portions of the buildings. To quantify the findings as a supplement to visualization, nonparametric statistical tests are used. The Kruskal-Wallis statistical test was able to uncover certain trends between the healthy and damaged buildings given that the time series data is sampled at different time windows of the live earthquake on which the simulation was based.

Indeed, experiments were conducted using actual earthquake data which were applied on sample buildings that were constructed using the ETABS simulation software. The actual sensors are currently being developed and will be deployed on cultural heritage structures (i.e. churches), lifeline systems (i.e. flyovers, bridges, and elevated railways), and modern structures to periodically gather data for analysis. There is a huge potential for this mesh of tremor sensors to identify underlying structural damage that would call for necessary maintenance or intervention before the condition worsens.

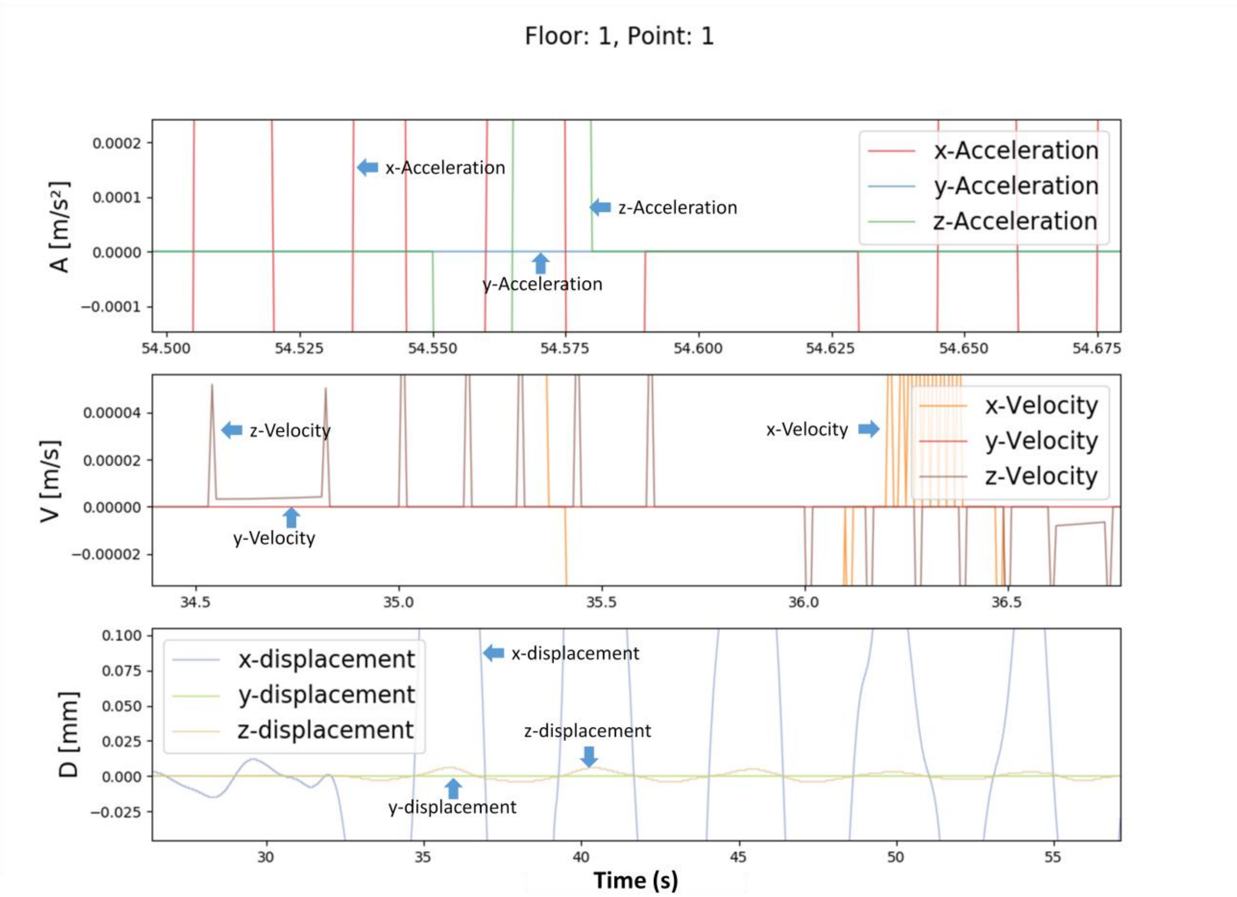


Fig. 17. Sample data plot for the target floor and group.

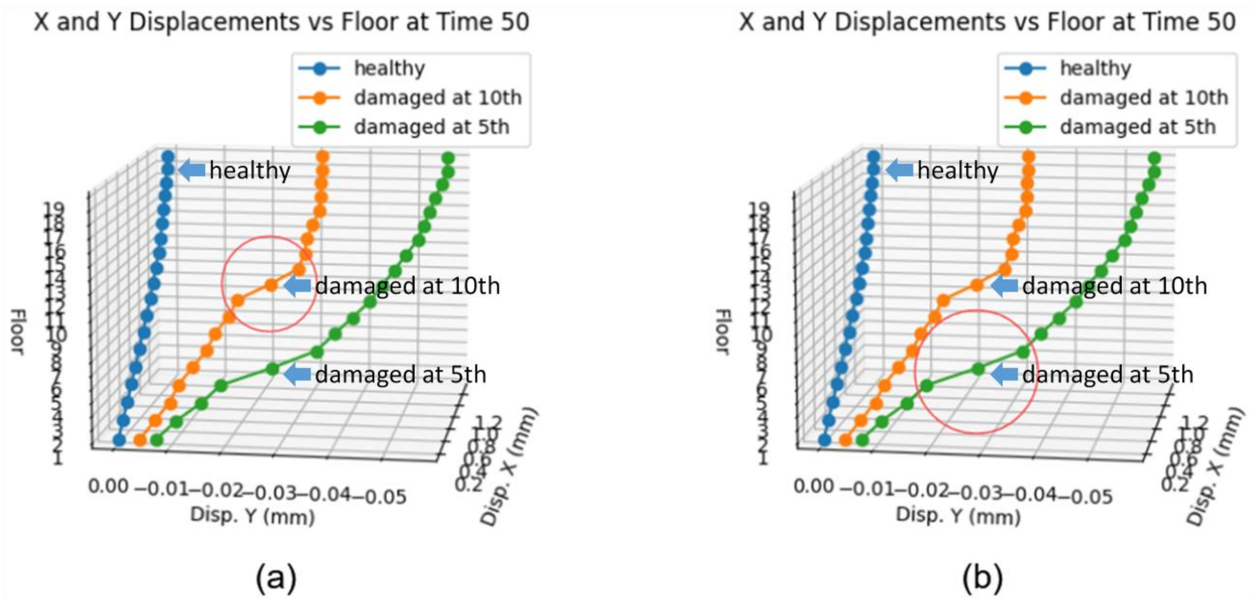


Fig. 18. X and Y displacements per floor plotted against the floor values at a certain point in time. The circle highlights the damaged portions of the buildings: a) tenth floor, b) fifth floor.

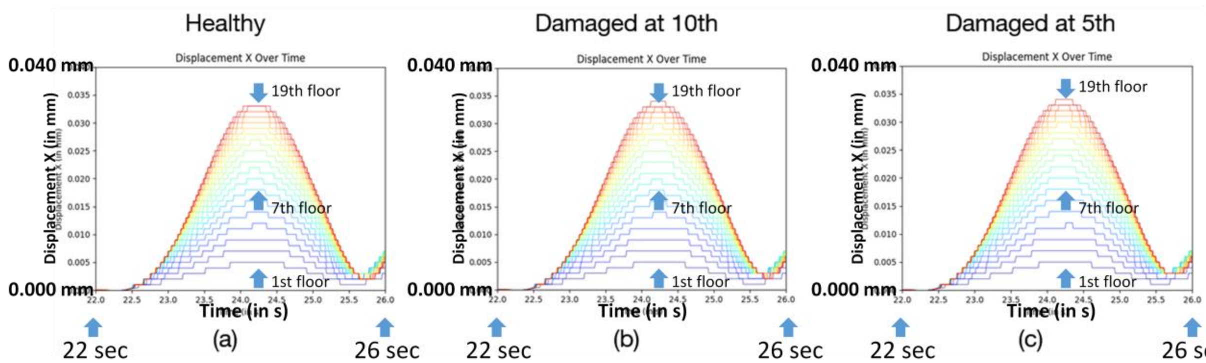


Fig. 19. X-displacements of different floors before a major tremor occur. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

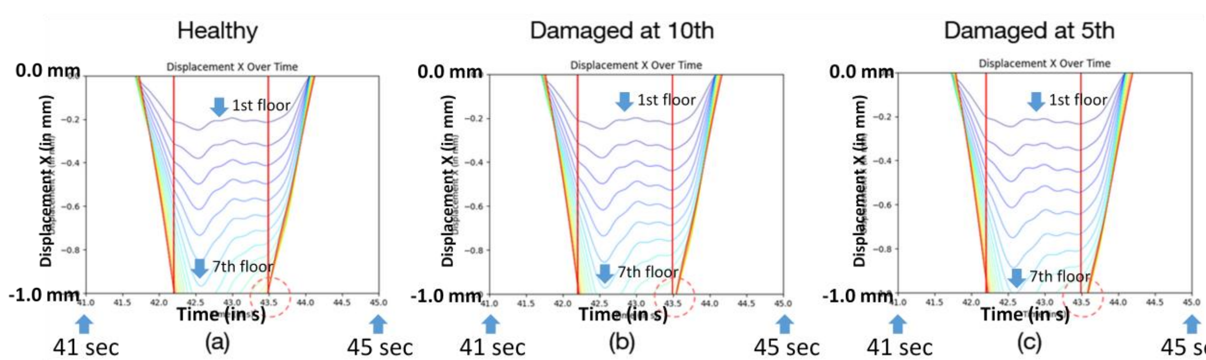


Fig. 20. X-displacements of different floors after a major tremor occurred. The circle highlights the area that shows the difference between the different structures. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

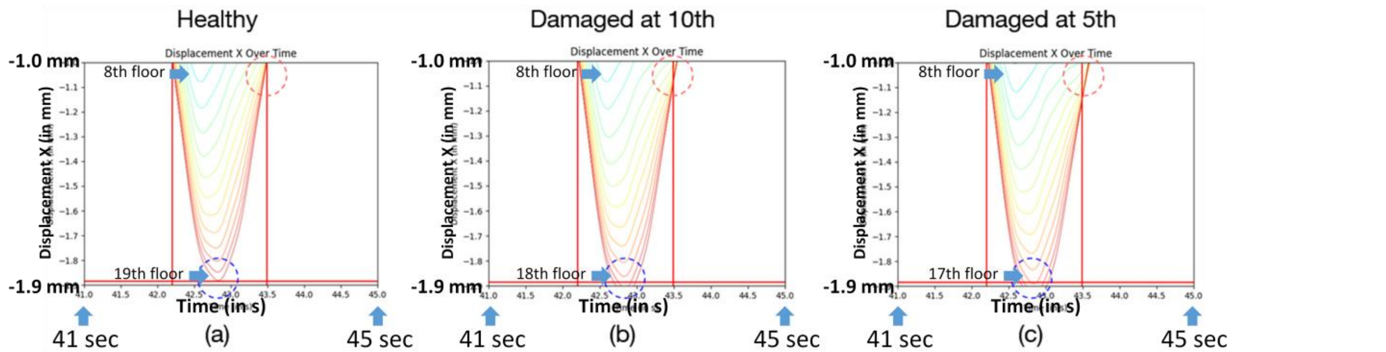


Fig. 21. x-displacements of different floors after a major tremor occurred. The plots compare the amplitudes of the displacement values of different floors. The red circle is the same mark from Fig. 9. The blue circle highlights the area that shows the difference in amplitudes between the different structures. a) healthy building. b) damaged at tenth floor. c) damaged at fifth floor.

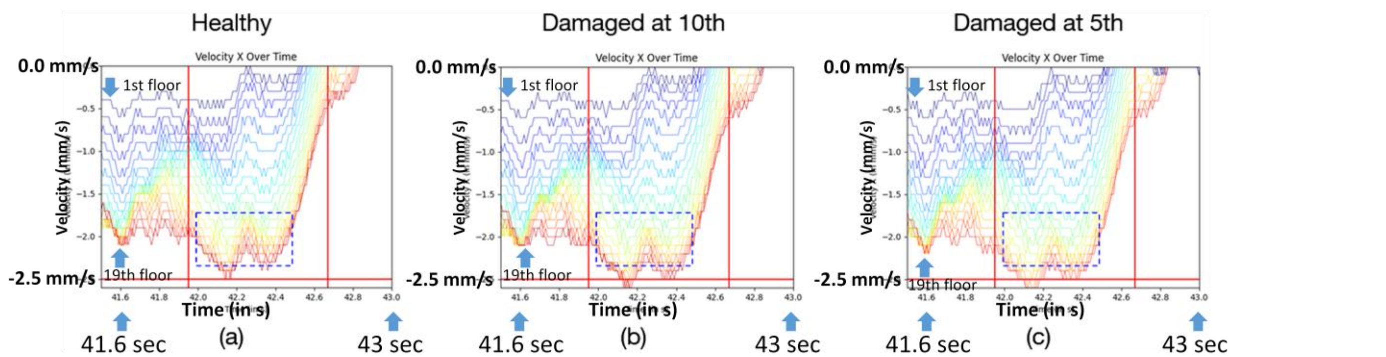


Fig. 22. Velocities along the x-axis of different floors after a major tremor occurred. The rectangle highlights the area that shows the difference between the different structures. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

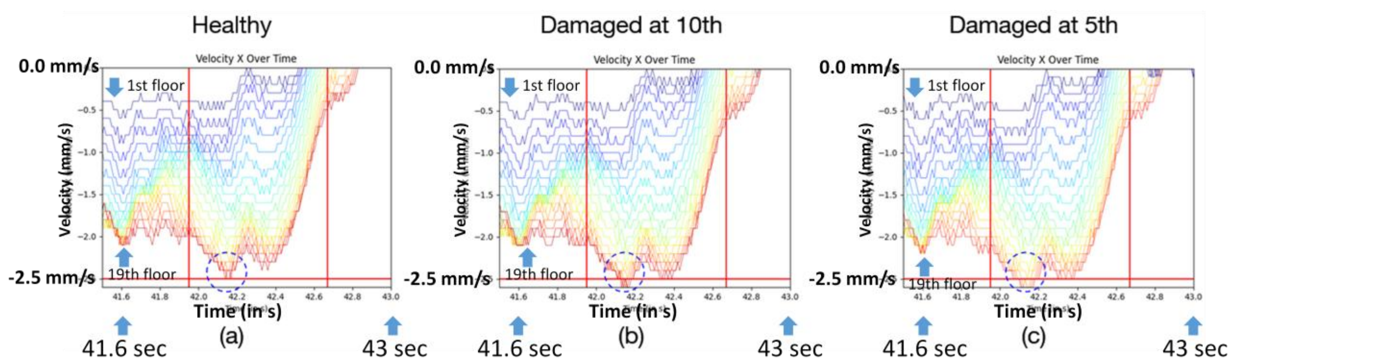


Fig. 23. Comparing the peak velocities along the x-axis of different floors after a major tremor occurred. The circle highlights the area that shows the difference between the different structures. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

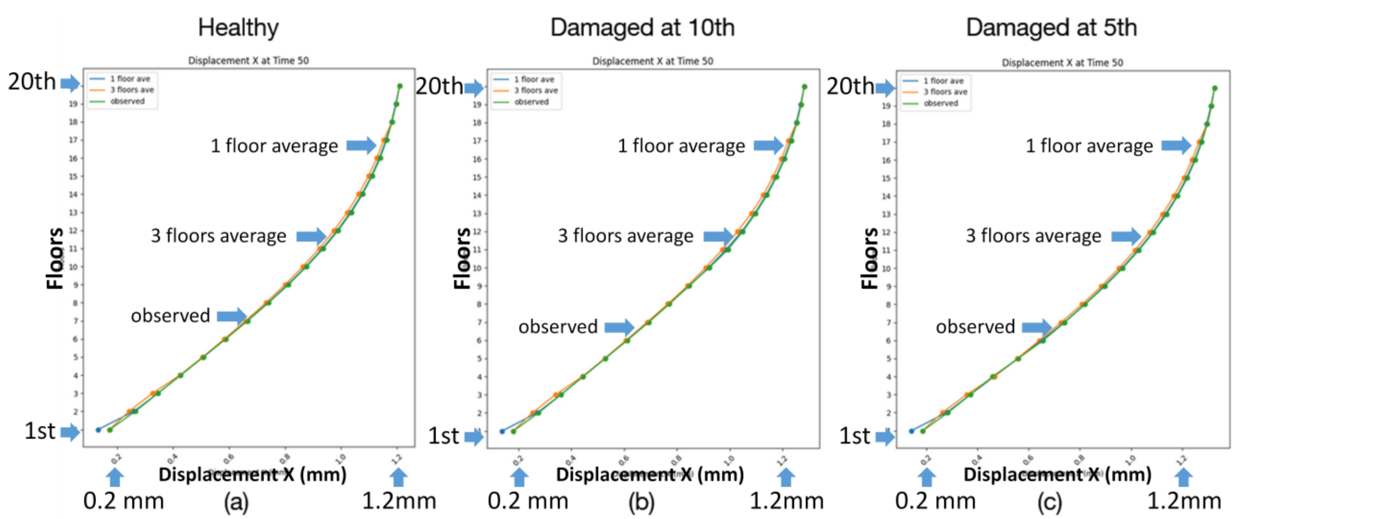


Fig. 24. Comparison of the naïve expected values with the observed values. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

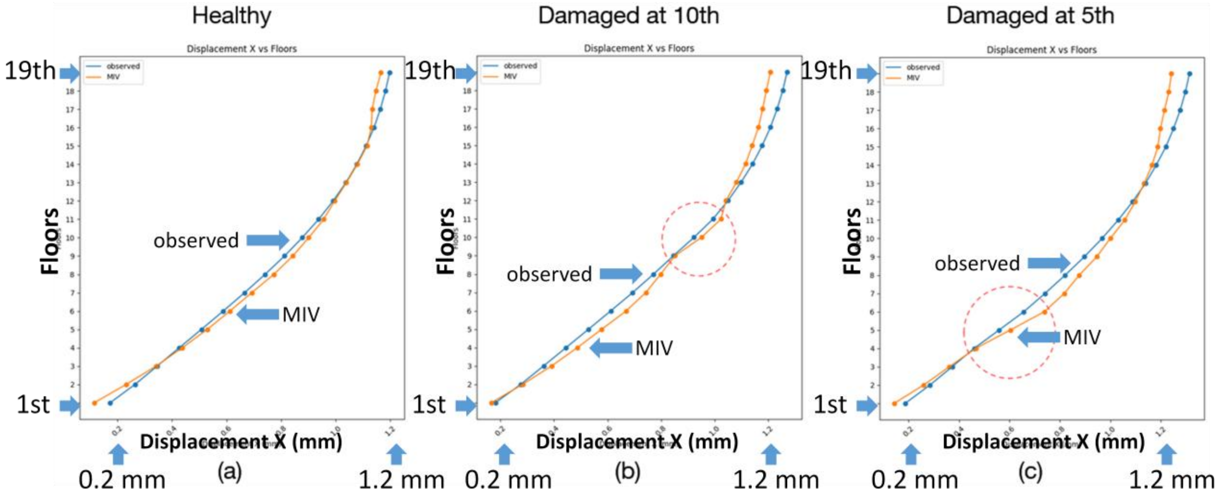


Fig. 25. Comparison of the mesh inferred values with the observed values. The circle highlights the damaged portions of the buildings. a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

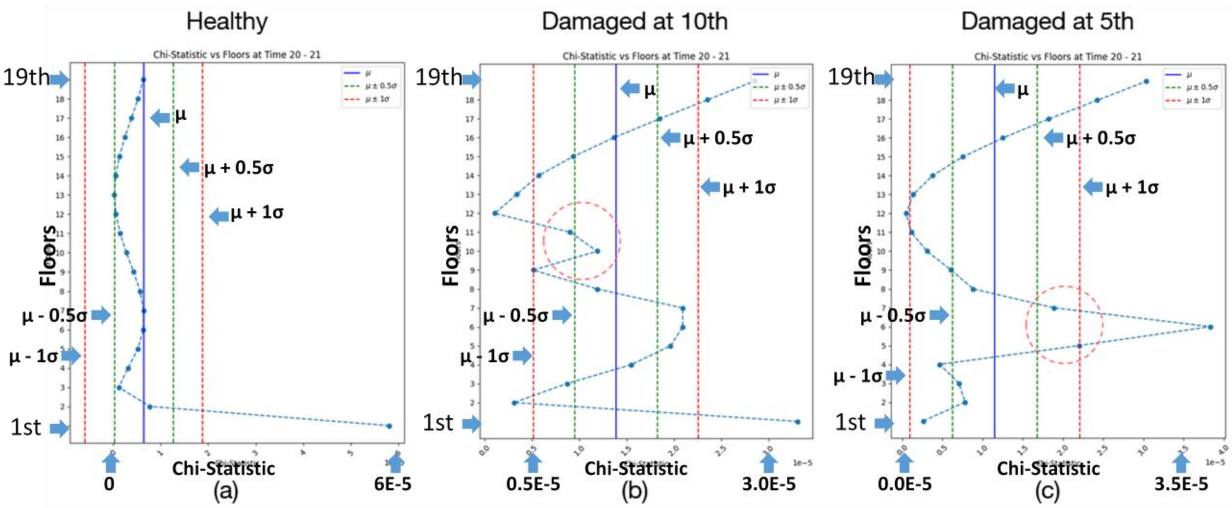


Fig. 26. Chi-statistic before the peak of the earthquake occurred (time 20 seconds to 21 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

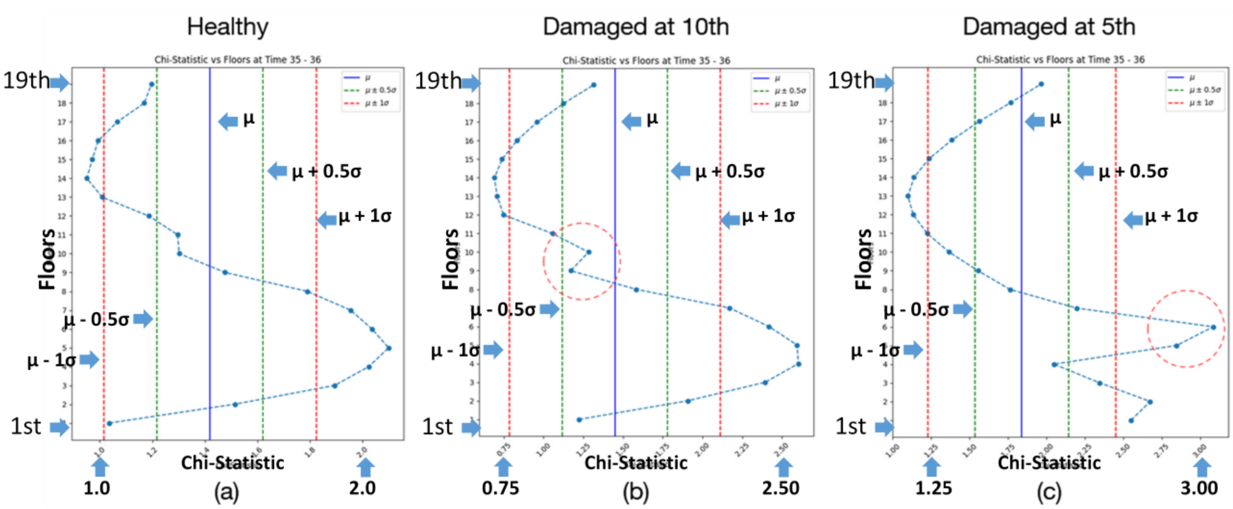


Fig. 27. Chi-statistic during the peak of the earthquake (time 35 seconds to 36 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

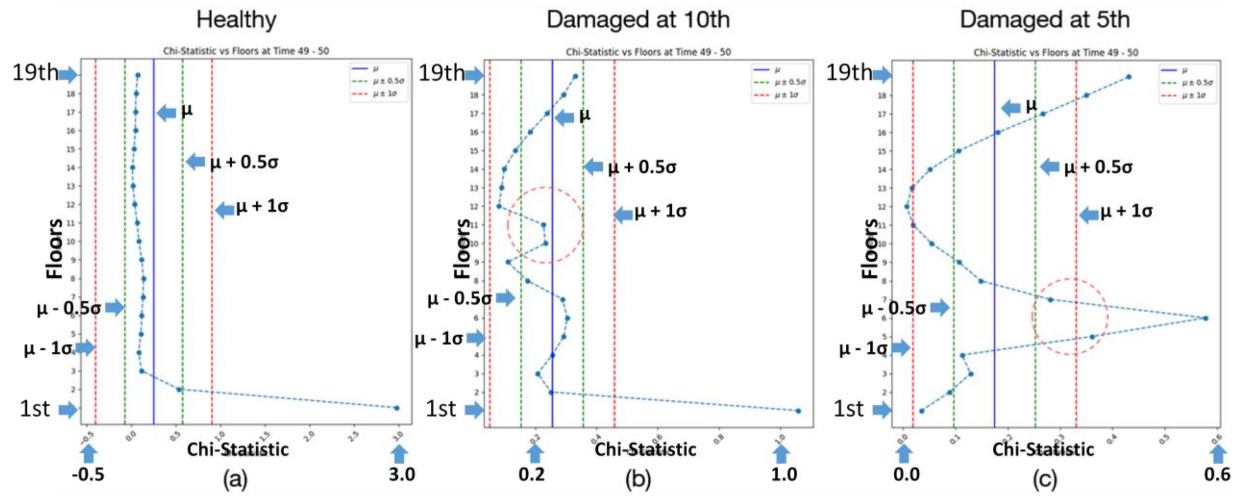


Fig. 28. Chi-statistic after the peak of the earthquake (time 49 seconds to 50 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

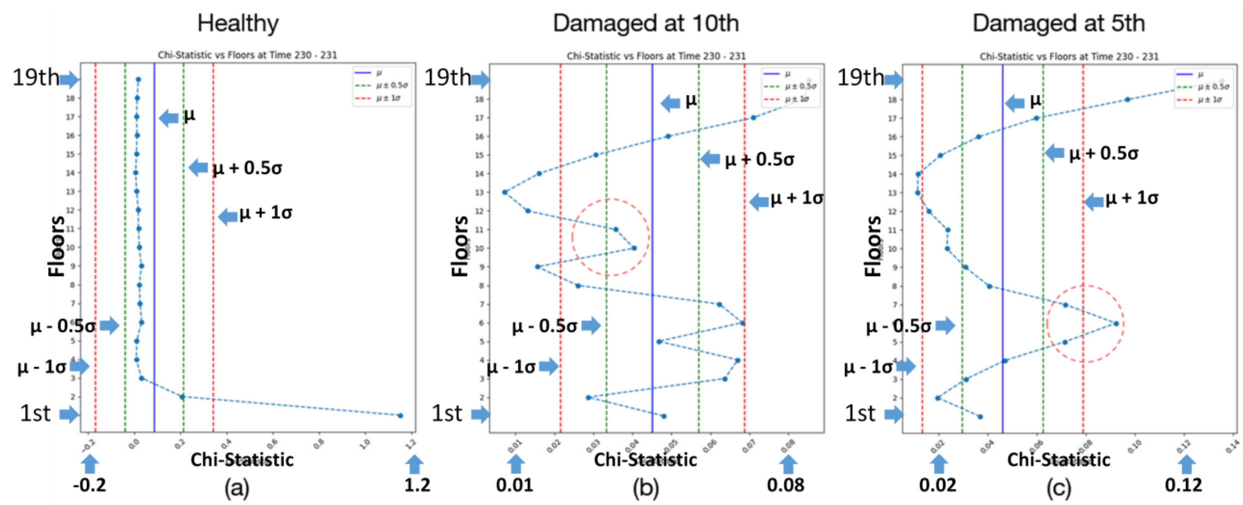


Fig. 29. Chi-statistic during the waning period of the earthquake (time 230 seconds to 231 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

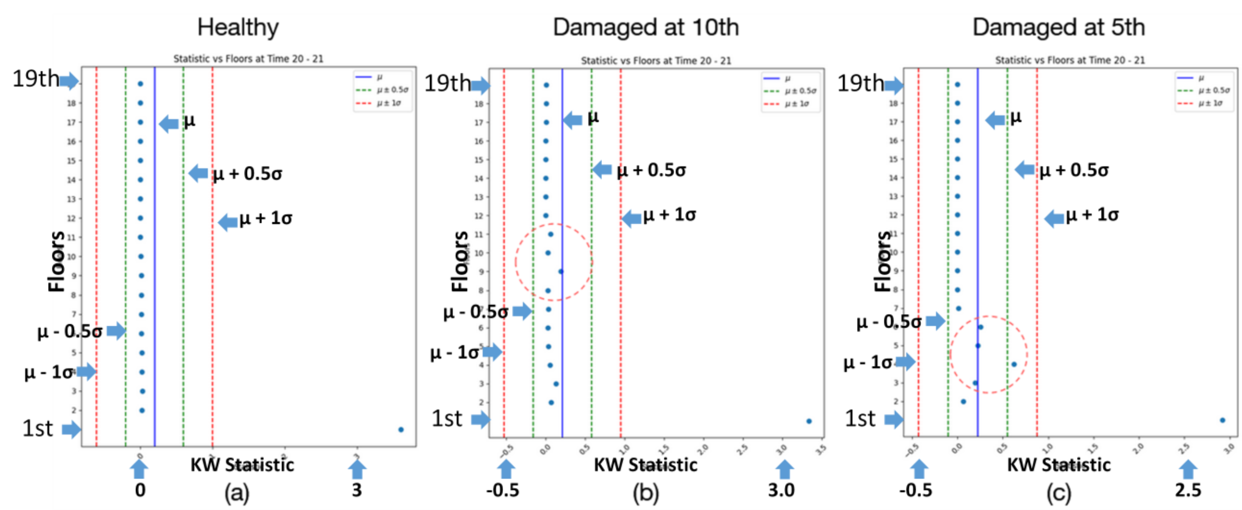


Fig. 30. Kruskal-Wallis statistic before the peak of the earthquake occurred (time 20 seconds to 21 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

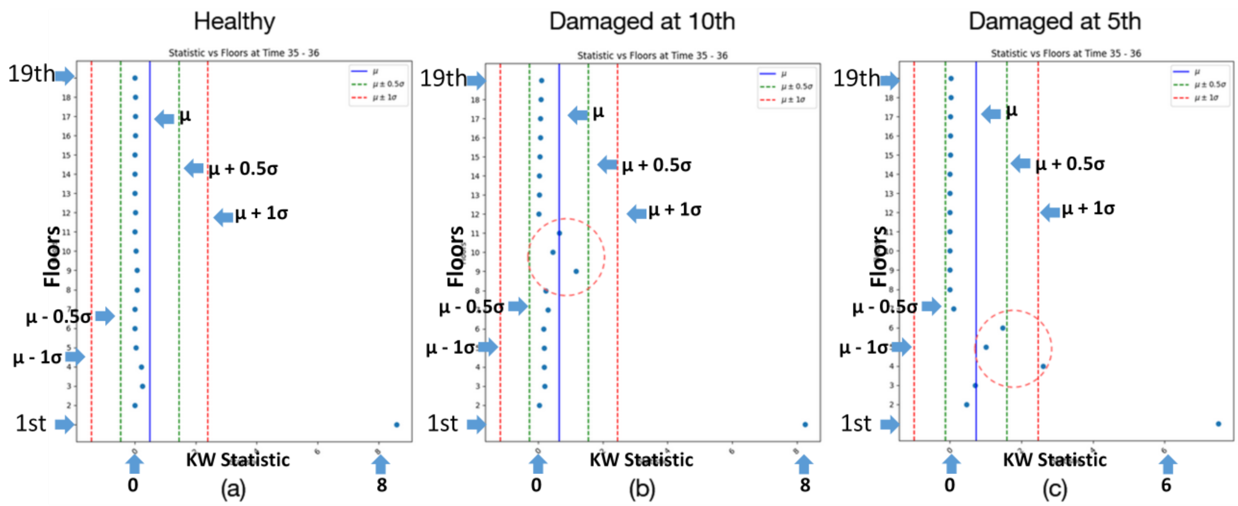


Fig. 31. Kruskal-Wallis statistic during the peak of the earthquake (time 35 seconds to 36 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

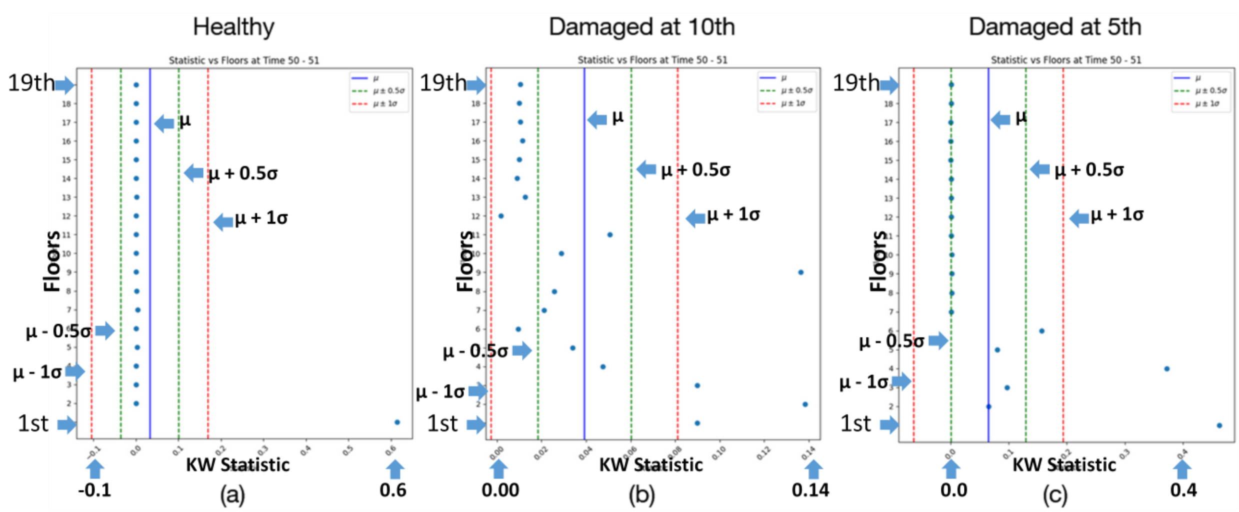


Fig. 32. Kruskal-Wallis statistic after the peak of the earthquake (time 50 seconds to 51 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.

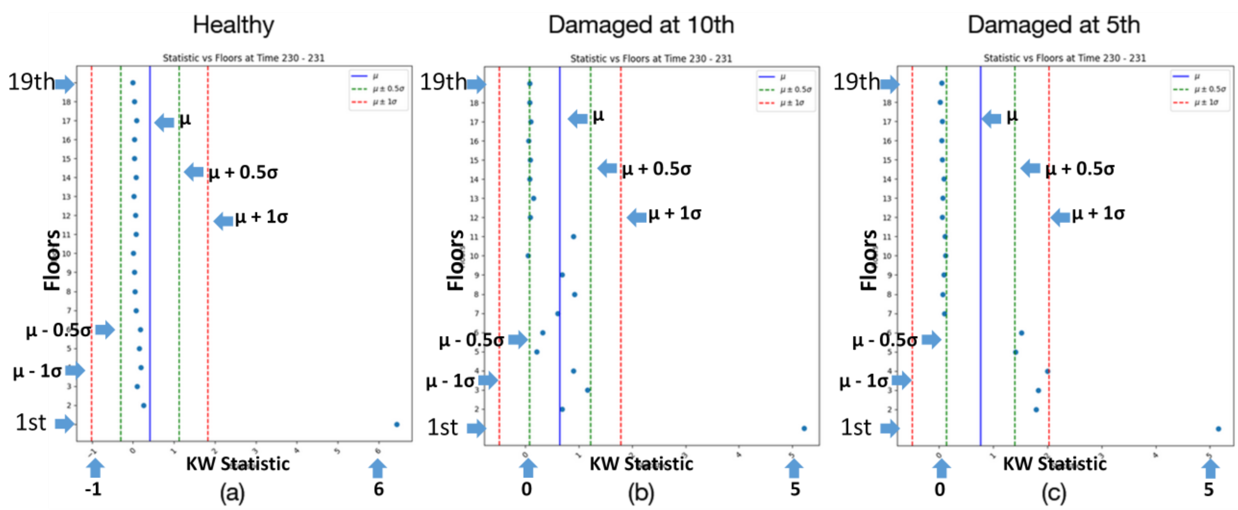
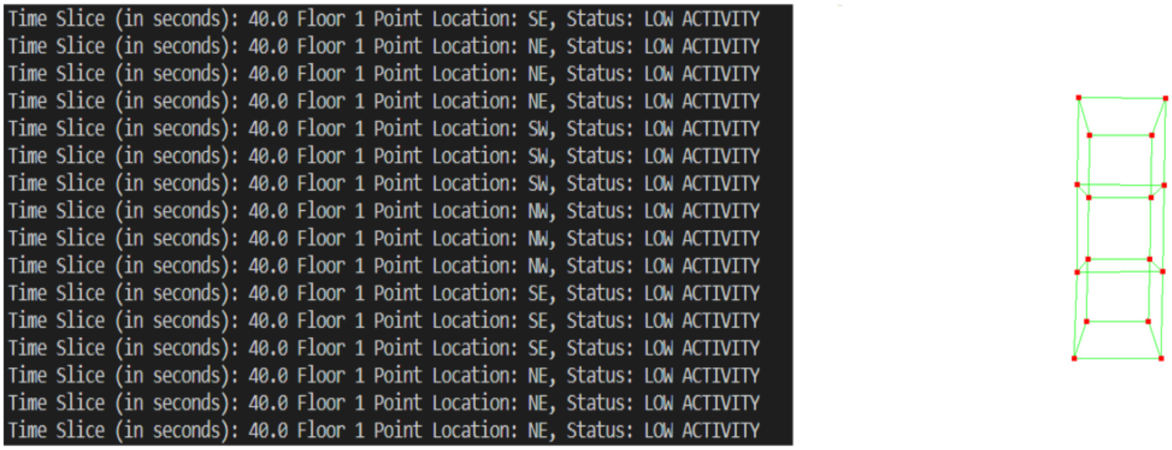
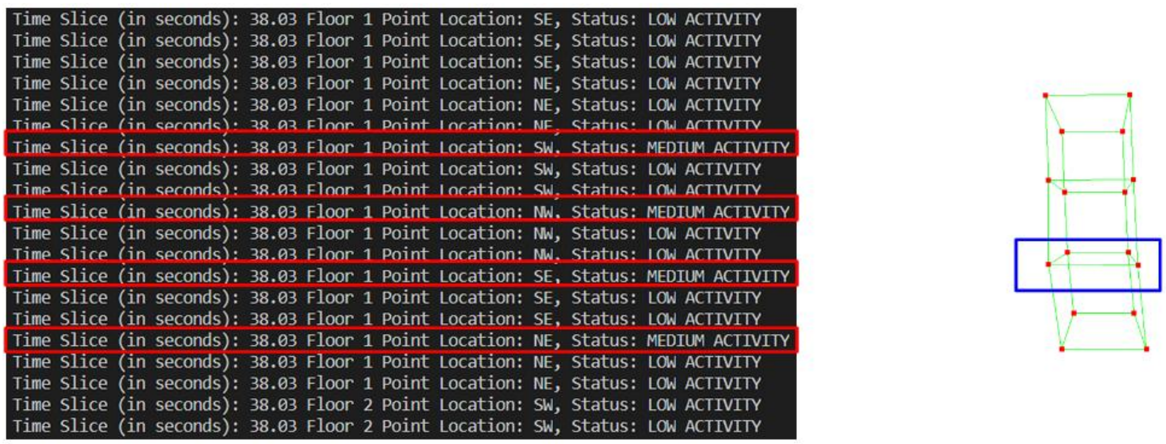


Fig. 33. Kruskal-Wallis statistic during the waning period of the earthquake (time 230 seconds to 231 seconds). a) Healthy building. b) Damaged at tenth floor. c) Damaged at fifth floor.



(a) (b)

Fig. 34. Building movement with low activity drift ratio. a) Console output displaying the drift ratio status. b) 3D visualization showing the dynamic movement of the building.



(a) (b)

Fig. 35. Building movement with medium activity drift ratio. a) Console output displaying the drift ratio status. b) 3D visualization showing the dynamic movement of the building.

References

[1] Sarah Minson, Men-Adrin Meier, Annemarie Baltay, Thomas Hanks, and Elizabeth Cochran, "The limits of earthquake early warning: Timeliness of ground motion estimates," *Science Advances*, vol. 4, 2018.

[2] Thibaut Perol, Michaël Gharbi, and Marine Denolle, "Convolutional neural network for earthquake detection and location," *Science Advances*, vol. 4, 2018.

[3] Rex Allen, "Automatics Phase Pickers: Their Present Use and Future Prospects," *Bulletin of the Seismological Society of America*, vol. 72, pp. S225-S242, 1982.

[4] Mitchell Withers et al., "A Comparison of Select Trigger Algorithms for Automated Global Seismic Phase and Event Detection," *Bulletin of the Seismological Society of America*, vol. 8, pp. 95-106, 1998.

[5] The Montana Department of Natural Resources and Conservation, Teton County All Hazards Mitigation Plan, 2015.

[6] Ibnu Rusydy et al., "A GIS-Based Earthquake Damage Prediction in Different Earthquake Models: A Case Study at the University of the Philippines Los Baños, Philippines," *Philippine Journal of Science*, vol. 147, no. 2, pp. 301-316, June 2018.

[7] H. Miura, S. Midorikawa, K. Fujimoto, B.M. Pacheco, and H. Yamanaka, "Earthquake damage estimation in Metro Manila, Philippines based on seismic performance of buildings evaluated by local experts' judgements," *Soil Dynamics and Earthquake Engineering*, vol. 28, pp. 764-777, 2008.

[8] M Hashemi and AA Alesheikh, "A GIS-based Earthquake Damage Assesment and Settlement Methodology," *Soil Dynamics and Earthquake Engineering*, vol. 31, no. 11, pp. 1607-1617, 2011.

[9] S Karimzadeh, M Miyajima, R Hassanzadeh, R Amiraslanzadeh, and B Kamel, "A GIS-based Seismic Hazard, Building Vulnerability and Human Loss Assessment for The Earthquake Scenario in Tabriz," *Soil Dynamics and Earthquake Engineering*, vol. 66, pp. 263-280, 2014.

[10] Steven Gibbons and Frode Ringdal, "The detection of low magnitude seismic events using array-based waveform correlation," *Geophysical Journal International*, vol. 165, no. 1, pp. 149-166, April 2006.

[11] Robert Skoumal, Michael Brudzinski, Brian Currie, and Jonathan Levy, "Optimizing multi-station earthquake template matching through re-examination of the Youngstown, Ohio, sequence," *Earth and Planetary Science Letters*, vol. 405, no. 1, pp. 274-280, November 2014.

[12] David Shelly, "A 15 year catalog of more than 1 million low-frequency earthquakes: Tracking tremor and slip along the deep San Andreas Fault," *Journal of Geophysical Research: Solid Earth*, vol. 122, no. 5, pp. 3739-3753, May 2017.

[13] Aiping Guo, Ajuan Jiang, and Zhangyu Cheng, "A Hybrid Clustering Method for Bridge Structure Health Monitoring," in *World Congress on Engineering and Computer Science 2018 Vol I*, San Francisco, 2018, pp. 161-165.

[14] Jin Wang and Ta Liang Teng, "Artificial Neural Network-Based Seismic Detector," *Bulletin of the Seimological Society of America*, vol. 85, no. 1, pp. 308-319, February 1995.

[15] Jin Wang and Ta Liang Teng, "Identification and Picking of S Phase Using an Artificial Neural Network," *Bulletin of the Seismological Society of America*, vol. 87, no. 5, pp. 1140-1149, October 1997.

[16] Qing Kai Kong, Richard Allen, Louis Schreier, and Young Woo Kwon, "MyShake: A smartphone seismic network for earthquake early warning and beyond," *Science Advances*, vol. 2, no. 2, February 2016.

[17] Reni Suryanita and Azlan Adnan, "Application of Neural Networks in Bridge Health Prediction based on Acceleration and Displacement Data Domain," in *International MultiConference of Engineers and Computer Scientists 2013 Vol I*, Hong Kong, 2013.

[18] Andrew Valentine and Jeannot Trampert, "Data space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data," *Geophysical Journal International*, vol. 189, no. 2, pp. 1183-1202, May 2012.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436-444, May 2015.

[20] Aaron van den Oord et al. (2016, September) WaveNet: A Generative Model for Raw Audio. Available: https://arxiv.org/abs/1609.03499

[21] Alex Choy, "Structural Health Monitoring with Deep Learning," in *International MultiConference of Engineers and Computer Scientists 2018 Vol II*, Hong Kong, 2018.

[22] M. Celebi and A. Sanli, "GPS in Pioneering Dynamic Monitoring of Long-Period Structures," *Earthquake Spectra: Journal of EERI*, vol. 18, no. 1, pp. 47-61, February 2002.

[23] M. Celebi, A. Sanli, S. Sinclair, S. Gallant, and D. Radulescu, "Real-Time Seismic Monitoring Needs of a Building Owner—and the Solution: A Cooperative Effort," *Earthquake Spectra: Journal of EERI*, vol. 20, no. 2, pp. 333-346, May 2004.

[24] M. Celebi, "Health monitoring of Buildings Using Threshold Drift Ratios — Now an Established Method," in *International Conference on Structural Health Monitoring*, Vancouver, 2007.

[25] Department of Public Works and Highways, Guidelines and Implementing Rules on Earthquake Recording Instrumentation for Buildings, 2015.

[26] G. T. Webb, P. J. Vardanega, P. R. A., and C. R. Middleton, "Analysis of Structural Health Monitoring Data from Hammersmith Flyover," *Journal of Bridge Engineering*, 2014.

[27] Zhichao Cai, Hao Cheng, and Chengcheng Liu, "Nonlinear Electromagnetics Acoustic Testing Method for Tensile Damage Evaluation," *Journal of Sensors*, 2018.

[28] Huaping Ding, Qinghong Shen, and Sidan Du, "Structural Condition Assessment of the Herringbone Middle Pylon of the Taizhou Bridge Using SHM Strain Data," *Journal of Sensors*, 2017.

[29] Ka Veng Yuen, Siu Kui Au, and James Beck, "Two-Stage Structural Health Monitoring Approach for Phase I Benchmark Studies," *Journal of Engineering Mechanics*, 2004.

[30] Xin Lin Qing, Wen Zhuo Li, Yi Shou Wang, and Hu Sun, "Piezoeectric Transducer-Based Structural Health Monitoring for Aircraft Applications," *Journal of Sensors*, 2019.

[31] Massinissa Saoudi, Ahcene Bounceur, Reinhardt Euler, and Tahar Kechadi, "Data Mining Techniques Applied to Wireless Sensor Networks for Early Forest Fire Detection," , 2016.

[32] Qing Kai Kong, Richard Allen, Monica Kohler, Thomas Heaton, and Julian Bunn, "Structural Health Monitoring of Buildings Using Smartphone Sensors," *Seismological Research Letters*, 2018.

[33] W. Flores-Fuentes, M. Rivas-Lopez, O. Sergiyenko, J. Rivera-Castillo, and D. Hernandez-Balbuena, "Analog Signal Processing in Light Scanning Sensors for Structural Health Monitoring Accuracy Enhancement," in *World Congress on Engineering and Computer Science 2013 Vol II*, San Francisco, 2013.

[34] Muhammad Syafrudin, Ganjar Alfian, Norma Latif Fitriyani, and Jongtae Rhee, "Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing," *Sensors*, vol. 18, no. 9, September 2018.

[35] Xinlin P. Qing et al., "Stanford Multiactuator–Receiver Transduction (SMART) Layer Technology and Its Applications," in *Encyclopedia of Structural Health Monitoring*.: John Wiley & Sons, Ltd., 2009.

[36] F. A. Uy, C. Zablan, D. Santiago, and M. Baylon, "USHER: Universal Structural Health Evaluation and Recording System," Mapua University Manila, 2019.

[37] Hung Fu Chang and Tzu Kang Lin, Real-time Structural Health Monitoring System Using Internet of Things and Cloud Computing, 2019.

[38] S. D. Fassois and F. P. Kopsaftopoulos, "Statistical time series methods for vibration based structural health monitoring," *New trends in structural health monitoring*, pp. 209-264, 2013.

[39] J. S. Sakellariou and S. D. Fassois, "A functional pooling framework for the identification of systems under multiple operating conditions," , 2007, pp. 1-6.

[40] M. Basseville, L. Mevel, and M. Goursat, "Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ratios," *Journal of Sound and Vibration*, vol. 275 (3-5), pp. 769-794, 2004.

[41] Cristian Acevedo, "Implementing Machine Learning in Earthquake," Stanford University, 2016.

[42] B. S. Kusumo, A. Heryana, E. Nugraheni, A. F. Rozie, and B. Setiadi, "Recognizing Human Activities and Earthquake Vibration from Smartphone Accelerometers using LSTM Algorithm," , 2018, pp. 88-92.

[43] Y. Liu et al., "Generative adversarial active learning for unsupervised outlier detection," , 2019.

[44] S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Bersoza, "CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection," *Scientific Reports*, vol. 9(1), pp. 1-14, 2019.

[45] Computers and Structures Incorporated. (2021) ETABS Building Analysis and Design. Available: https://www.csiamerica.com/products/etabs

[46] Mrittika Hasan Rodela and Ayesha Binta Ali, "Comparative Study between Manual Analysis and Computer Software Analysis (ETABS) of a Residential Building in Dhaka City," in *International Conference on Structural Engineering Research*, Dhaka, 2019.

[47] Mohammad Kalim, Abdul Rehman, and B. S. Tyagi, "Comparative Study on Analysis and Design of Regular Configuration of Building by Staad.Pro and Etabs," *International Research Journal of Engineering and Technology*, vol. 5, no. 3, pp. 1793-1797, March 2018.

[48] S. Vijaya Bhaskar Reddy and V. Madhu, "Comparative Study on Design Results of a Multi-storied Building using STAAD PRO and ETABS for Regular and Irregular Plan Configuration," *International Journal of Applied Engineering Research*, vol. 13, no. 15, pp. 12194-12201, 2018.

[49] Isha Bedi, Girish Sharma, and Abhishek Gupta, "Comparative study of RCC Frame Structures using Staad.Pro, ETABS, and SAP," *International Journal of Advance Research in Science and Engineering*, vol. 6, no. 9, pp. 1005-1013, September 2017.

[50] P. Prashanth, S. Anshuman, R. K. Pandey, and Herbert Arpan, "Comparison of Design Results of a Structure Designed Using STAAD and ETABS Software," *International Journal of Civil and Structural Engineering*, vol. 2, no. 3, 2012.

[51] Facebook's AI Research Lab. (2021) PyTorch. Available: https://pytorch.org/

[52] Ali Serhan Koyuncugil and Nermin Ozgulbas, "Statistical Roots of Machine Learning, Deep Learning, Artifical Intelligence, Big Data Analytics and Data Mining," in *World Congress on Engineering and Computer Science 2019*, San Francisco, 2019.

[53] Pygame Community. (2021) pygame. Available: https://www.pygame.org/news

[54] PyOpenGL Community. (2021) PyOpenGL. Available: https://pypi.org/project/PyOpenGL/