ANOMALY DETECTION IN THE HOME WITH SEISMIC SENSORS

by

Siraphat Boonchan

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Data Science and Artificial Intelligenc

Examination Committee:	Prof. Matthew N. Dailey (Chairperson) Dr. Mongkol Ekpanyapong Dr. Attaphongse Taparugssanagorn

Nationality:	Thai
Previous Degree:	Bachelor of Engineering in Electrical Engineering Kasetsart University Thailand
Scholarship Donor:	The Royal Thai Government

Asian Institute of Technology
School of Engineering and Technology
Thailand
August 2021

ABSTRACT

Text here

Keywords: keyword1, keyword2.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Globally, injuries after a fall is a significant public health problem. Each year, approximately 37 million falls requires medical attention, and approximately 684,000 individuals die from falls. Falls are the second leading cause of unexpected injury death, after road traffic injuries. According to the World Health Organization (WHO, 2018), the highest death rate from falls in all regions around the world was faced among adults who are over the age of 60 years. The frequency of falling down increases with age and weakness level. In the future, injuries caused by falls will affect more civilians as the population ages, and fall deaths are expected to double by 2030. According to Fuller (2013), The elderly, who represent 12 percent of the population, account for 75 percent of those who die from falls.

In addition, the Ministry of Public Health in Thailand (ThaiNCD.com, 2019) says that one-third or greater than 3 millions of Thailand's people fall in their homes every year. Approximately 66% of the cases involve slippery floors, stumblings and missing a step on the same ground level. They report an average of 140 calls to local ambulances per day, and on average, 2 people die each day. More than 55% of falls occur inside the home environment (Pynoos, Steinman, & Nguyen, 2010), most frequently in the bathroom, kitchen and dining room. Therefore, when victims fall, if nobody knows about the accident, and nobody takes care of the victim immediately, it can result in more serious injury, long term impairments, and even death.

From the statistics mentioned above, developing any technology able to help decrease or mitigate false will be useful. I am specifically interested in artificial intelligence approaches to detection of fall events that can also immediately alert caretakers or assistants.

1.2 Statement of the Problem

There has been great deal of research on fall detection. Researchers try to find the best methods to detect and mitigate falls. Each approach has pros and cons, depending on the situation and the environment as following:

1. User-activated fall alert with a pendant: Although manually-activated fall alarms are simple and low cost, they are only successful when a user who has fallen activates the alarm button by himself or herself manually. This system is ineffective if the person is not wearing the pendant because he or she refuses to press the emergency button, forgets it, or cannot press it. Elders may hesitate to push an emergent button for several reasons such as concern about bothering others and privacy.
2. Automatic Wearable Devices (Degen, Jaeckel, Rufer, & Wyss, 2003; Yang & Hsu, 2010; Rihana & Mondalak, 2016): This solution is popular because it is uncomplicated and provides high accuracy. Devices in this group are based on inertial measurement units (IMU)s, which contain an accelerometer and gyrometer. A significant disadvantage of this solution is that the user has to wear the device all the time, which can lead to discomfort, and if the device cannot be wore in the shower, the device will miss the period in which individuals have the highest probability of falling. Moreover, a wearable may even cause injury when people fall down.
3. Cameras (Tsai & Hsu, 2019; Ramirez et al., 2021; Taufeeque, Koita, Spicher, & Deserno, 2021): Many researchers have developed camera-based systems to detect falls, since cameras can track residents, and falls can be detected based on image processing algorithms trained to identify abnormal activity. However, the drawbacks of cameras are that residents may feel uncomfortable and concerned about privacy, even if the images are not leaked. Moreover, when a victim falls in a place out of view of the camera, e.g. an aed occluded by furniture, the method cannot alert caretakers. Also, cameras cannot be installed in the toilet or bathroom, again missing some of the highest risk periods of time.
4. Vibration analysis (Alwan et al., 2006; Liu, Jiang, Su, Benzoni, & Maxwell, 2019; Clemente, Li, Valero, & Song, 2020): This approach has not been explored as much as the others. Vibration has several limitations in terms of data collection:
    - Vibration sensors: The general sensors popular in the commercial market have low sensitivity. When the floor is concrete, it is quite difficult to detect vibrations with a general sensor. Madarshahian, Caicedo, and Arocha Zambrana, (2016) use a high-sensitivity piezoelectric sensors, but this sensor requires embedding in the ground, making it difficult to install. In addition, when the area is large, more sensors are required, which increases cost and complexity of the system.
    - Sample fall data: White falls can be simulated to get data for IMU or

camera sensors, vibration data from a fall have specific characteristics depending on the type of floor, the weight of the subject, and the distance of the sensor to the locus of the event. Realistically, real falls on concrete and other hard surfaces are too dangerous to simulated.

Despite these limitations, the benefits of the vibration signals for fall detection does overcome the drawbacks associated with all previous methods. As vibration signals have been analyzed further to include human activity and peoples' heart rates (Jia, Howard, Zhang, & Zhang, 2017), using vibrational signals to detect falls may significantly advance the technology available in this area, and it help mitigate the elderly fall problem.

Research on vibration data has thus far used supervised classification models including k-nearest-neighbors (Shao et al., 2020), support vector machines (S. Wang, Chen, Zhou, Sun, & Dong, 2015; Kasturi & Jo, 2017; Liu et al., 2019), and neural networks (Sultana, Deb, Dhar, & Koshiba, 2021). Others have used unsupervised learning methods such as k-means (Shao et al., 2020) and simple amplitude thresholds to classify fall events (Alwan et al., 2006; Charlon, Bourennane, Bettahar, & Campo, 2013; Britto Filho & Lubaszewski, 2020). Classification with supervised data requires collecting real fall data, which, as mentioned above, is dangerous, because faking a fall can lead to serious injury if we make a mistake while doing an experiment. Liu et al., (2019) solve this problem using dummy humans, but realistic dummies are expensive.

As falls occur infrequently and diversely, and there also are several types of falls such as forward falls, backward falls and lateral falls (El-Bendary, Tan, C. Pivot, & Lam, 2013), any attempt to exhaustively train a supervised classifier can lead to a lack of sufficient data for training. Although, falling events occurring during different activities such as walking, standing, sleeping, or sitting share some characteristics in common, they also have significant differences (X. Wang, Ellul, & Azzopardi, 2020). It is difficult to anticipate all possible patterns in advance. Furthermore, as fall events rarely occur in daily life, if we train a model with an imbalanced dataset, it can result in bias.

Anomaly detector methods may be the key to addressing all of these issues. I will apply anomaly detection methods to detect adverse event such as falls indirectly. The main advantage of anomaly detection beside addressing the diversity of fall is that anomaly detection will not only detect falls but also detect other abnormal activities such as fighting and any other activities the model is not trained on.

1.3 Research Questions

The purpose of this paper is to develop an robust automated anomaly detection system capable of detecting falls and other anomalies by combining knowledge from signal processing, embedded systems, machine learning, and edge devices. The study aims to answer the following questions:

1. Can a seismic sensor and an embedded system detect human activity on the surface of a typical concrete floor in the home?
2. What are the best methods for detecting anomaly events such as falls using seismic sensors?
3. Can a system be designed and implemented that identifies falls in daily human activities in real time?
4. Can the system thus designed be deployed in real home environments?

1.4 Objectives of the Study

The main objective of this study is to alert caretakers immediately when an anomalous event such as a fall occurs in the home. To fulfill this main objective, I will take the following specific steps:

1. Design and build a filter, amplifier, and embedded system to digitize and analyze signals from seismic sensors characterizing human activities.
2. Collect data on daily human activities by many subjects.
3. Build an anomaly detection and alerting system for detecting anomaly patterns.
4. Deploy the model in the dining room in my home.
5. Evaluate the deployed model in terms of its accuracy in identifying unusual events.

1.5 Scope and Limitations

The scope and limitations of this study are as follows :

1. The study will focus on concrete floor because most household floors in Thailand are concrete material covered with tile.
2. I assume the home has only a single elderly person.
3. Accuracy may suffer if multiple people are present and active at the same time.

CHAPTER 2

Literature Review

Nowadays, Falls are concernable problem around the world. Fall detection is an interested topic that researchers prefer to receive the best accuracy. Several methods have tried to overcome this problem, but they have suffered with a lot of constrains. Nonetheless, using vibration signal to detect fall actions may highly modernize in order to mitigate senile fall problem.

There are several knowledge related fields which start from vibration untill artificial intelligence model, and every section of this system as software and hardware are equally important. Thus, we have to explore and deeply understand in each branch in order to build the best system.

2.1 Fall

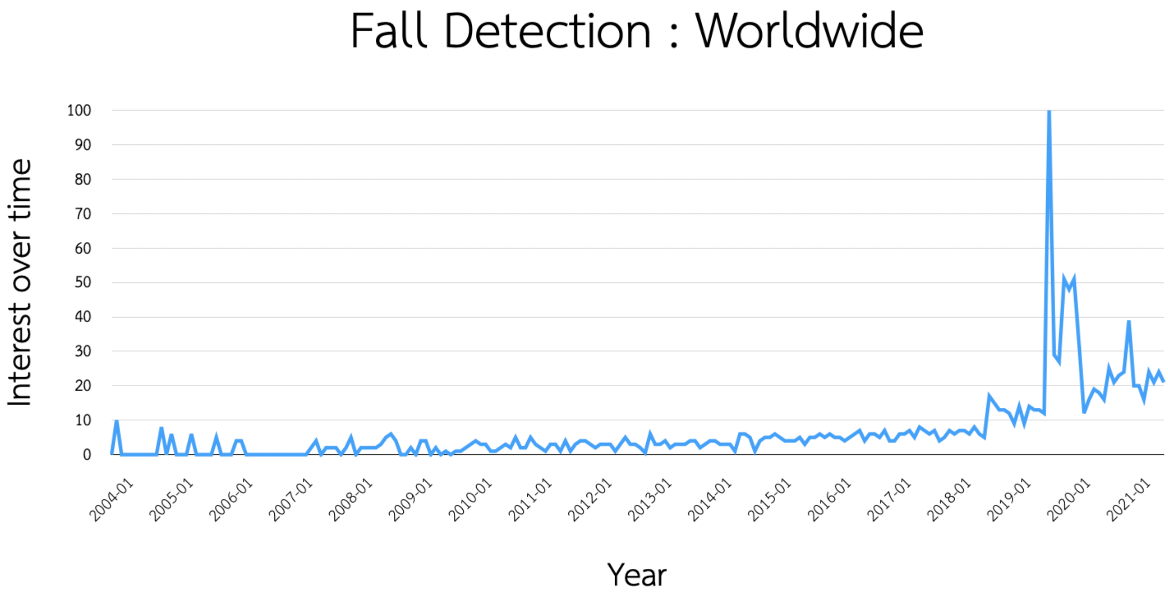
Falls happen to people of all ages, but older people have a high probability of being harmed and are more likely to fall, especially if they have an abnormal health conditions or balance problems. Falls are a common but often disregarded cause of injury. According to NHS (2019) one in three adults over 65 and half of the people over 80 have at least one fall per year. Most falls do not result in serious injury, but there is always a risk that a fall could lead to broken bones, and it can cause the person to have paralysis. In addition, the level of injury depends on the timeliness of the assistance. Unintentional falls can cause severe injuries and even death, especially if no immediate assistance is given.

2.1.1 Fall Detection

Global trends in fall detection are illustrated in Figure 2.1. The data are downloaded from Google Trends with the search topic “Fall Detection”. Fall detection has gotten increasingly more attention over time and significantly increased in 2019. The values are indexed to be 100, where 100 is the maximum search interest for that period of time with specific location. Researchers have developed systems using a variety of different sensors and methods depending on their proposes and technological industry. Consequently, we can conclude that this topic is of interest and is becoming increasingly popular.

Figure 2.1

Interest in “Fall Detection” over time from 2004 to present according to Google Trends.



2.1.2 Fall detection by using vibration sensors

Table 2.1 shows the evolution of fall detection from floor vibration. Most researches use classifier models to detect fall events with training performed on simulated fall data that were not real falls. Furthermore, none of these researchers have deployed their system in real environments, so the real world performance of the models is not convincing. To overcome these weaknesses, I will apply anomaly detection to compensate for the rarity of events such as real falls and other or unseen patterns, and send alerts to the caretaker or an assistance who can take care of a victim who is alone as soon as possible.

Table 2.1
Summary of literature review for fall detection from floor vibration.

Authors	Data Collection	Sensors	Algorithms	Alarm
Alwan et al. (2003)	Simulated by people.	N/A	Threshold	N/A
Alwan et al. (2006)	Simulated by people and dummies.	Piezoelectric	Threshold	Send messages to a pager
Litvak et al. (2008)	Simulated by people and dummies.	Microphone Accelerometer	Gaussian model Sequential forward floating selection (SFFS)	N/A
Davis et al. (2011)	Simulated by people.	N/A	Threshold	N/A
Yazar et al. (2014)	N/A	Pyroelectric infeared (PIR) Vibration sensor	Support vector machine (SVM)	N/A
Shao et al. (2020)	Simulated by 3d-printed skeleton	Accelerometer on smartphone	K-nearest-neighbor (KNN)	N/A
Liu et al. (2019)	Simulated by people and dummies.	Seismic	A multi-features semi-supervised support vector machines (MFSS - SVM)	N/A
Clemente et al. (2020)	N/A	Seismic	One-class SVM	N/A
Mukherjee and Zhang (2020)	N/A	Motion sensor Heat sensor Vibration sensor	Threshold	N/A

2.2 Human Activity

To create training data for anomaly detection in the home, it is important to cover all of the typical activities that people are expected to engage in in the home.

Schrader et al. (2020) say there is no common definition or description of human activities because human activity is highly diverse. Nonetheless, the most fundamental activity in home is clearly walking, since a resident needs to move several inside the home to perform any other activities (Oukrich, 2019). There also are other general activities that every person does. I summarized activity catagories proposed in the literature on human activity in home in Table 2.2. Each paper in the table includes experiments on different activities of interest, and there are some common activities across most of the studies such as sitting, walking, standing and lying.

Table 2.2
Summary of literature review on human activity.

Authors	Objective of study	Related Sensors	Identified Activities
Roggen et al. (2010)	Collect complex activity datasets in home	Microphone Accelerometers Gyroscope Magnetometer Inertial sensor	Sitting Walking Standing Lying
(Chen & Xue, 2015)	Classify human activity by single accelerometer	Accelerometer	Walking Standing Lying Running Rope jump Vacuum cleaning Downstairs Upstairs
(Reiss & Stricker, 2012)	Published a new public dataset for physical activity	Gyroscope Magnetometer	Sitting Step walking Walking quickly Falling Jumping Running Downstairs Upstairs
Ugolotti et al. (2011)	Detect and classify human activities	Camera Accelerometer	Sitting Walking Standing Lying Get up Fall Rise
(Abbate et al., 2012)	Detect fall events	Accelerometer on smartphone	Sitting Walking Lying Running Jumping Hitting the sensor

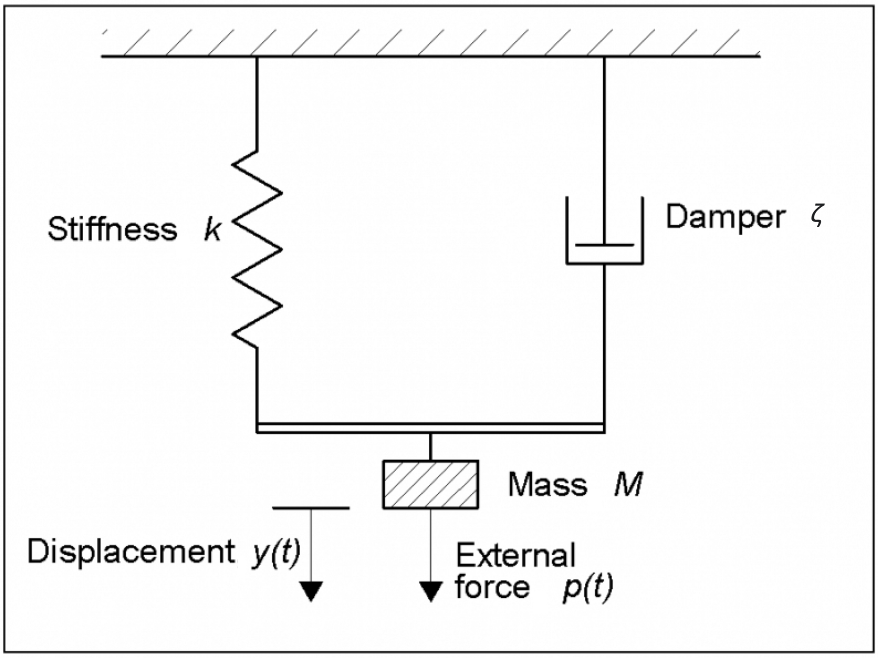
2.3 Floor Vibrations

Movement in a building by residents during their normal activities causes floor vibration. This vibration is normally vertical (SteelConstruction, 2016). Floor vibrations are generated by dynamic loads caused directly by people (e.g. walking, dancing, jumping) or machinery, or they may be generated indirectly by the external environment (e.g. traffic). Theoretically, vibrations are cyclic motions with two significant attributes, frequency and amplitude. In practice, floor vibrations are quite complex dynamic systems with unlimited vibrational modes. Ljunggren (2006) summarises the parameter that influence the dynamic system of a floor:

- Stiffness (k): Stiffness controls the springiness of the floor. Higher stiffness can decreases the vibrational amplitude occurring due to a force.
- Damping (ζ): This factor depends on the material making up the surface. It is extremely difficult to obtain an exact damping value.
- Mass (M): Higher mass surfaces have reduced vibrational amplitudes. Lower mass is desirable if we want to observe vibrations. However, when the mass is too little, the resulting strong vibrations may disturb residents.
- Fundamental frequency: Floor vibrations are assumed to be occur mainly at a natural frequency, which depends on the stiffness and the mass. Higher frequencies are usually less annoying to residents than lower frequencies.

The complexities of the dynamic system can be modeled as a series of simple mass and spring models with a single degree of freedom (Gavin, 2015). The characteristics of a vibration model are illustrated in Figure 2.2.

Figure 2.2
Single degree of freedom system mass - spring model for floor vibration.
Reprinted from SteelConstruction (2016).



2.4 Time Series

A time series is a sequence of measurements of a particular random variable at specific sequence of discrete points in time. Generally, the data should be sampled at a constant interval expresssed in as seconds, minutes, hours, days, months, and/or years. In time series analysis, we would generally like to predict a traget variable at particular time lags given a window of previous measurements. This is unusual in that in ordinary supervised classification or regression, the target at time t is not used as a feature at a later time, but in time series analysis, this is often the case.

There are many diverse techniques for analyzing sequential data. The simplest techniques are a special case of regression analysis in which we want to capture four different elements as following (Dash, 2020):

- Seasonal variations: Repeating shape or appearance occuring during a specific period such as daily, weekly, monthly, or seasonally.
- Trend: Possible trends are upwards, downwards, or constantard can be linear or nonlinear.
- Cyclical variations: Movement that follows a specific cyclic period such as business cycles. Cyclical variations are similar to seasonal variations but have different

underlying cases specific to the particular problem.

- Random variations: The variation remaining after the first three types of predictable variation are accounted for.

2.4.1 Autoregressive (AR)

Autoregressive models, the simplest time series models, which are used for predict or forecasting proposes, operate under the assumption that each new value depends on some or all of the the past values. The generative model for a linear of autoregressive process is shown below:

$$Y_t = \varphi_1 Y_{t-1} + \cdots + \varphi_p Y_{t-p} + \varepsilon_t$$

where $\varepsilon \sim N(0, \sigma^2)$. p is the order of the model, which we write as AR(p). For example, AR(1) means the observation at time t depends only on the observation at time $t-1$ plus noise, whereare. AR(2) means y_t depends on the previous two values as well as a noise sample.

2.4.2 Time series classification

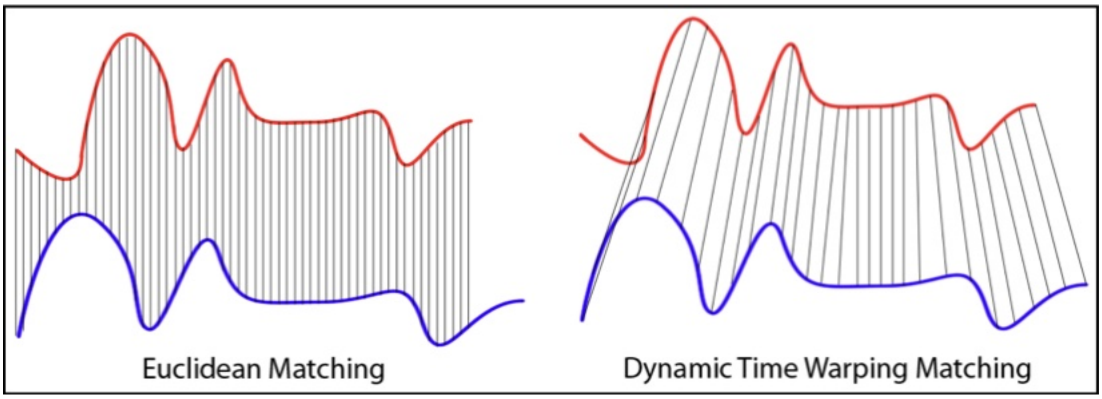
Over the last two decades, one of the most challenging problems in data mining is a classification of time series (Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019). Several methods have emerged for time series classification. The naive algorithm is Euclidean maching, which is not normally effective without some modification. On the other hand, dynamic time warping (DTW) is an outstanding baseline, and the current state of the art would in the most cases be represented by deep learning classifiers. Dynamic time warping is based on an alignment cost computed between two data sequences that can be stretched or shrunk to accommodate variations along the time axis (Müller, 2007; Toyoda & Sakurai, 2012). Consider two sequences, $X = (x_1, x_2, ..., x_n)$ of length n and $Y = (y_1, y_2, ..., y_m)$ of length m. The DTW distance $D(X,Y)$ is defined as:

$$D(X,Y) = D(m,n)$$

$$D(i,j) = (x_i - y_j)^2 + \min \begin{cases} \text{D(i-1, j)} \\ \text{D(i-1, j-1)} \\ \text{D(i, j-1)} \end{cases}$$

where $D(0,0) = 0$, $D(i,0) = D(j,0) = \infty$, $i = (1,2,...,n)$ and $j = (1,2,...,m)$.

Figure 2.3
Euclidean maching versus DTW matching.
Reprinted from Dynamic time warping (Wikipedia, 2021).



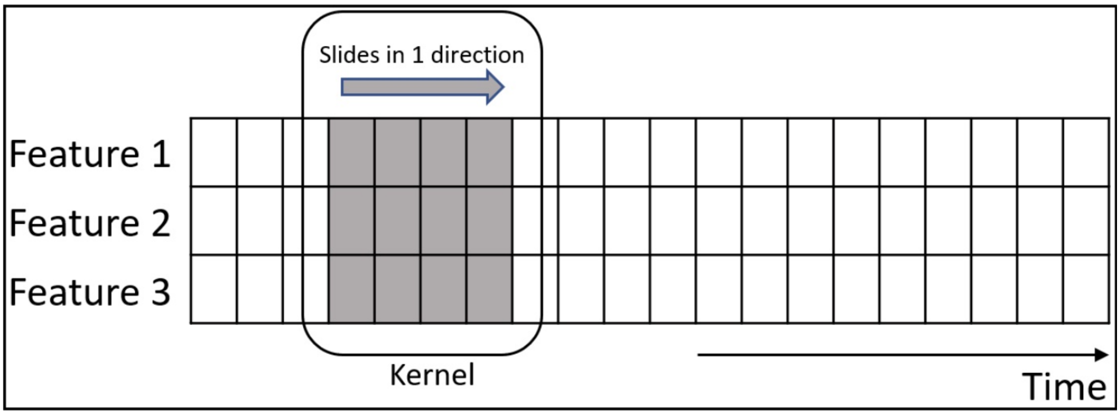
DTW can be used not only for pattern matching or classification, but also for anomaly detection. If the distance between a new signal and each signal in a gallery of historical signals is higher than a set threshold, we can conclude the new signal is an anomaly. The main weaknesses of dynamic time warping is its long processing time. Some more effective learning-based approaches are explained in the following sections.

2.4.3 Convolutional Neural Network (CNN)

A Convolutional neural network is a deep learning model whose input can be an image, video, spatial data, or any multidimensional tensor with locality. One-dimensional CNNs can be used on general data types including text tokens and other types of time series data. CNNs capture spatial and temporal dependencies in a dataset through convolutional filters. A convolution kernel is local linear filter that is slid over the imput tensor along one or more dimensions to obtain a feature map as shown in Figure 2.4. The general method for temporal CNN layer with a nonlinear activation function is

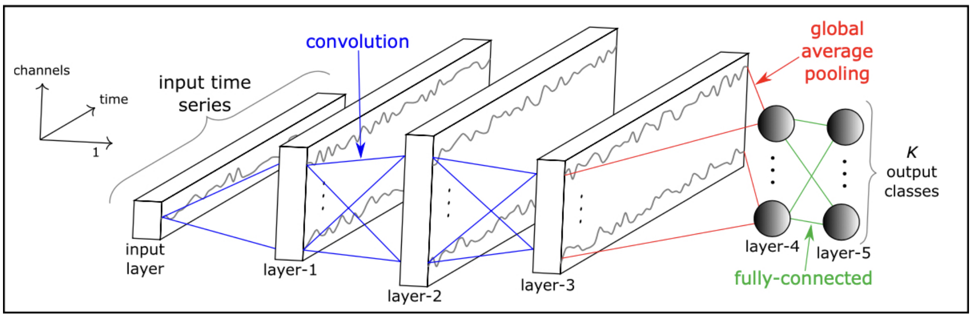
$$C_t = f(W \cdot X_{t-l/2 \to t+l/2} + b) | \forall t \in [1, T],$$

Figure 2.4
Convolving on univariate input time series
Reprinted from Ismail Fawaz et al. (2019).



where C_t is the result of the convolution operation at time t on time series X of length T with a filter W of length l, a bias parameter b, and a final non-linear function f. It can be noticed that the same filter values W and bias b are used at every timestep, a very significant and useful property called weight sharing. When a series of convolutions are completed, the resulting feature maps would typically be fed though fully-connected layer as in the simple neural network architecture shown in Figure 2.5.

Figure 2.5
Typical temporal convolutional neural network architecture.
Reprinted from Ismail Fawaz et al. (2019).

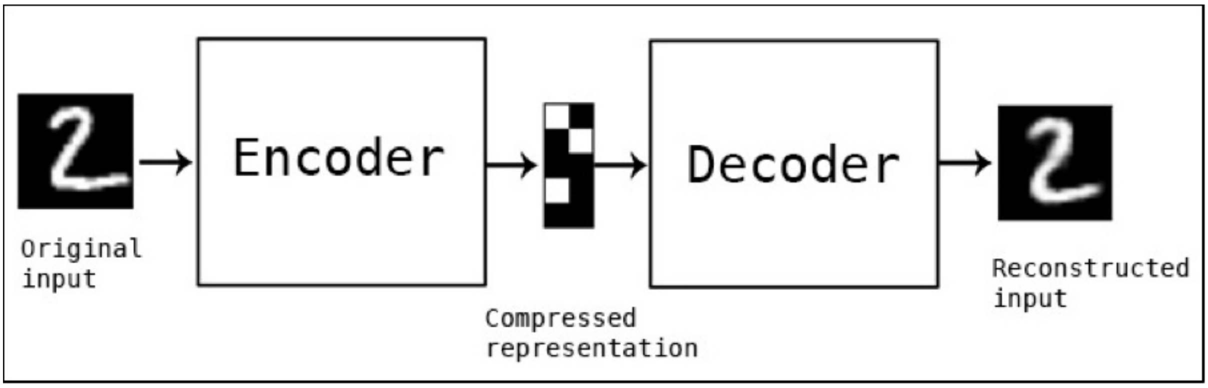


2.5 Autoencoders

An autoencoder is a neural network able to compress data similar to what it was trained on. Autoencoders do not require labeled data for training since they utilize unsupervised learning. We just feed the raw input into the model. Figure 2.6 illustrates the intuition of how an autoencoder works.

Besides compression, an autoencoder can be used for denoising by training the autoencoder to reproduce an original noiseless input given a noisy input. This allows the autoencoder to be flexible in the presence of white noise capturing only useful patterns in the data (Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010).

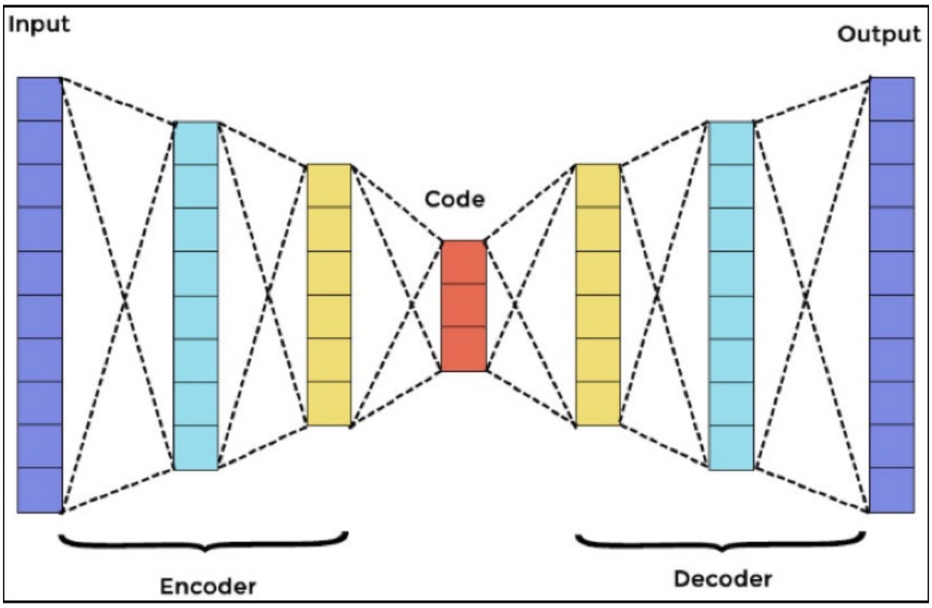
Figure 2.6
An autocoder workflow.
Reprinted from Chollet (2016).



An autoencoder has three main components (Badr, 2019): the encoder, the code or bottleneck, and the decoder, as shown in Figure 2.7.

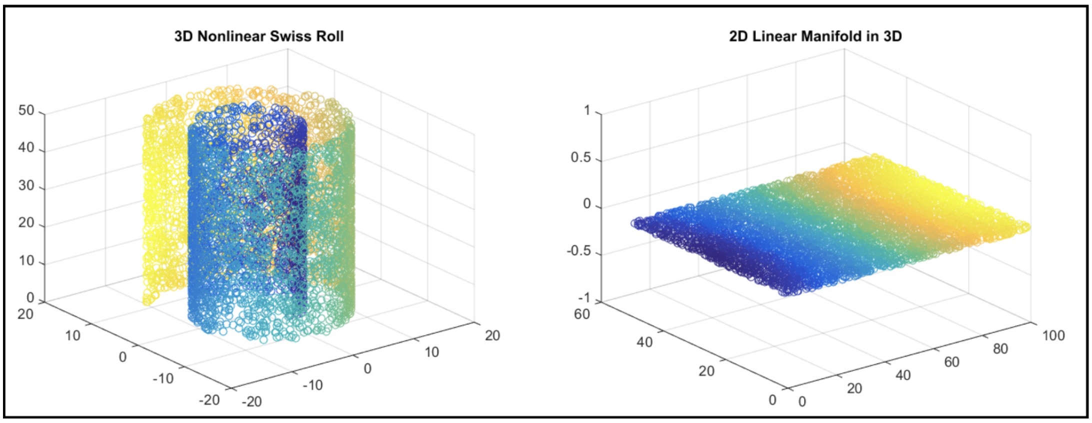
- Encoder: Learns how to reduce input dimensionality, compressing the input data into an encoded representation.
- Bottleneck: The layer that contains the compressed representation of the input data. This code space is also called the latent space.
- Decoder: Learns how to reconstruct as closely as possible the input pattern from the encoded representation.

Figure 2.7
The autoencoder architecture.
Reprinted from Pedamkar (2019).



The indirect benefits of this model is that it can be used for dimensionality reduction (Rajan, 2021). The bottleneck has the fewest units of any layer. An example of the kinds of compression an autoencoder can achieve is shown in Figure 2.8. This model reduces the three dimensional input to two dimensions.

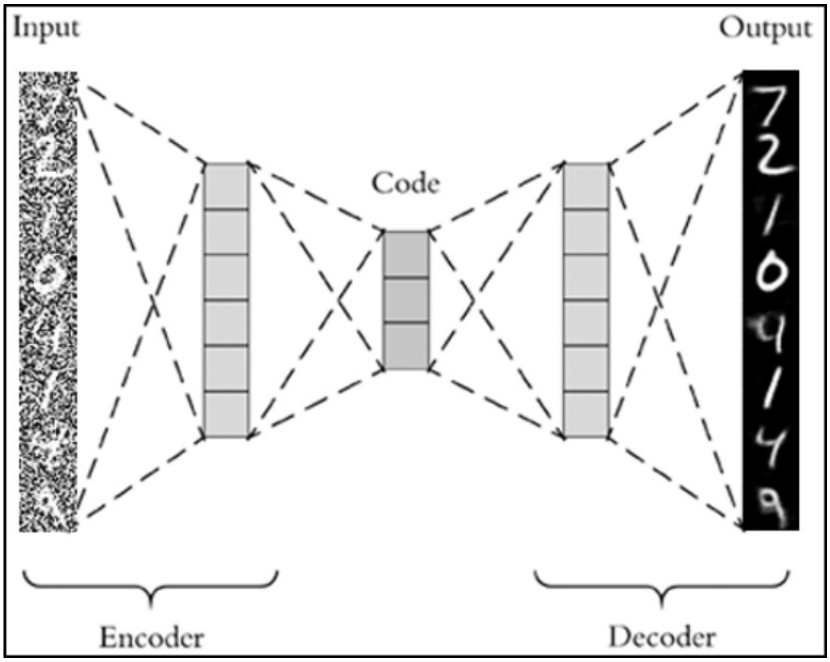
Figure 2.8
Visualization of dimensionality reduction using autoencoders.
Reprinted from Johns Hopkins University (2015).



Besides compression, an autoencoder can be used for denoising by training the autoencoder to reproduce an original noiseless input given a noisy input, as shown in Figure 2.9. This is because the optimizer encodes the inputs it was trained on as much as possible.

Vincent, Larochelle, Bengio, and Manzagol (2008) found that the robustness of the code at the bottleneck was improved by adding noise to the original input. This allows the autoencoder to be flexible in the presence of white noise capturing only useful patterns in the data (Vincent et al., 2010).

Figure 2.9
An autoencoder trained on "clean" images can correct noisy input.
Reprinted from Rosebrock (2020).

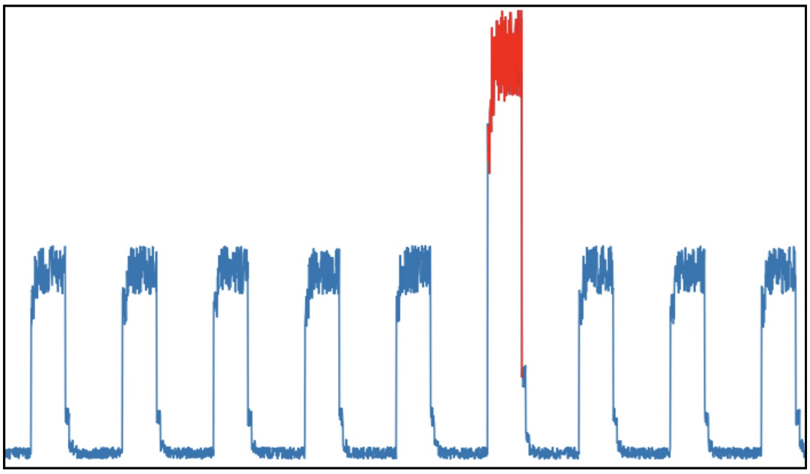


Beside the simple feedforward layers described previously, autoencoder can be combined with long short term memory networks (LSTMs) or convolutional neural networks (CNNs) depending on the type of input. In my thesis, the input, vibration from human activities, is a sequential time series. Therefore, an combination of autoencoder with LSTM networks may be suitable for my purpose.

2.5.1 Autoencoders for Anomaly Detection

Autoencoders are extremely useful as methods of typicality. Consider a person who does the same things every day. Suppose that one day, an unusual event occurs. An autoencoder trained on the usual daily activities will map the new situation to something similar in the training, as shown in Figure 2.10. The reconstruction error in abnormal causes should be high. A model trained on one type of data (the normal activities) will fail when facing abnormal data it has never seen before. The simple autoencoder-based anomaly detection algorithm is shown in Algorithm 1.

Figure 2.10
An autoencoder capable of detecting anomalous events in time series.
Reprinted from pavithrasv (2020).



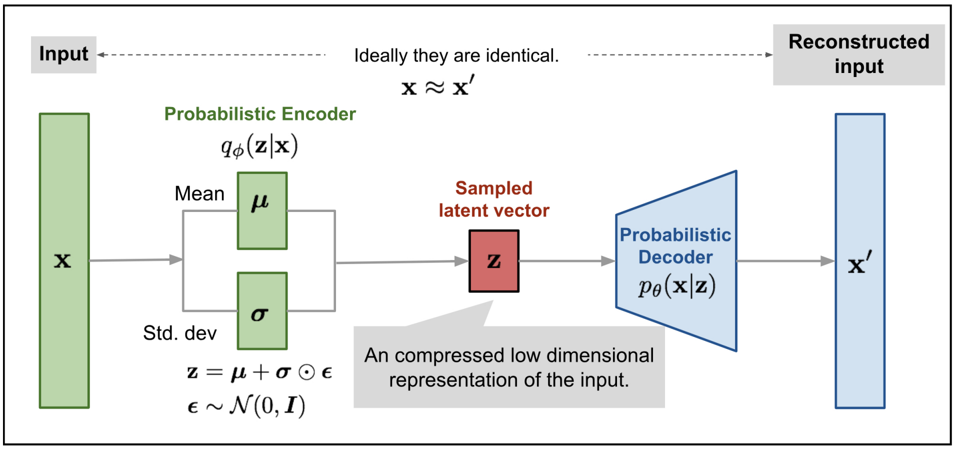
Algorithm 1 Autoencoder-based anomaly detection

Input: Normal dataset: $X^{(i)}(i = 1,\ldots,m)$, abnormal dataset: $x^{(j)}$ $(j = 1,\ldots,n)$, threshold: α
Output: Reconstruct data: $\hat{X}$
  Reconstruction error: $\|X - \hat{X}\|$
  Train an autoencoder using the normal dataset X $\rightarrow L^* = \underset{L}{\operatorname{argmin}} \sum_{i=1}^{m} \|X^{(i)} - \hat{X}^{(i)}\|^2$
  Testing an autoencoder:
  for j = 1 to n **do**:
    if reconstruction error $< \alpha$:
      $x^{(j)}$ is a nomal.
    else:
      $x^{(j)}$ is an anomaly.

2.5.2 Variational Autoencoder

The variational autoencoder (VAE) is a generative model like the ordinary autoencoder, it encodes and decodes data in the training set, but it also attempts to model the probability density over the input space of the examples emitted by the data source, by transforming, e.g., Gaussian distributied latent vectors to elements of the input space. For example, if model is trained with traffic images, the decoder, when passed a sample from the code space, would have a high probability of emitting vehicle images object related to traffic. Other data would have a low probability of being emitted. By sampling in the latent space reconstructing, the variational autoencoder can also generate new examples that look similar to those from the original dataset (Roger, 2021). In the other words, a variational autoencoder is an encoder which is trained to be regularized at the bottomneck in order to guarantee that latent space is a good source for the generative process (Rocca, 2020). The architecture of a variational autoencoder is illustrated in Figure 2.11. The latent space of a variational autoencoder is easy to sample from.

Figure 2.11
Architecture of a variational autoencoder.
Reprinted from Weng (2018).

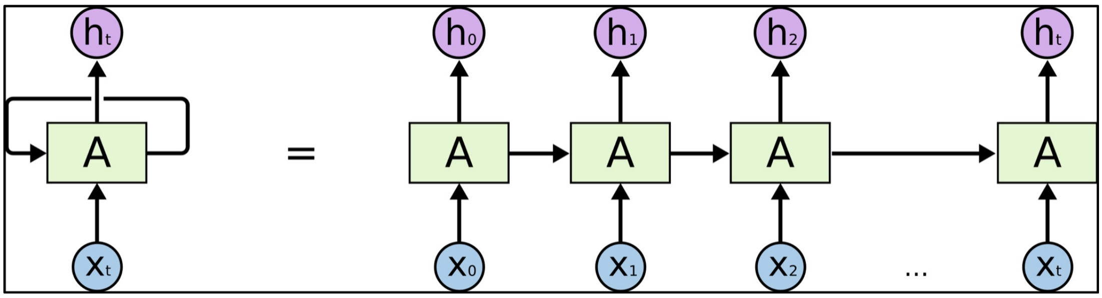


The principals mentioned above do not mean that a variational autoencoder always has better performance than general autoencoders in anomaly detection tasks (Agmon, 2021), since the objective of the variational autoencoder is as a generative model for new data.

2.6 Recurrent Neural Network (RNN)

The main idea of Recurrent Neural Network (RNN) is to apply sequential data such as video (sequence of images) or text (sequence of word). For example, when people are reading a book, it is a sequence of words because we read a book from left to right. That we can know what the sentence we are reading is about. We take the story from what we have read in the past, it is a hidden state, and mix it with the words we just read (input data or the words we are reading at that time). RNN uses the same principle, which is to modify the format of the old neural network so that the previous state or knowledge can be added to the new input data to understand something in a sequential time series (Donges, 2019). A key attribute of recurrent neural networks is their ability to persist information, or cell state, for use later in the network. There are 2 significant components of RNN as hidden state and input data.

Figure 2.12
The Recurrent Neural Network architecture.
Reprinted from Olah (2015)



where X_t is input data at time t, A is Hidden layer, and h_t is an output from RNN at time t shown in Figure 2.12. The main benefit of this loop is to bring back the previous hidden state, or simply say that RNN is a Neural Network with more memory to store the previously calculated hidden state.

The main problem of RNN is its gradient. For those who have experienced in neural networks would clearly know that to update weights we use a backpropagation (Arnx, 2019), which calculates the gradient of the loss function (E) to update the weights which is shown in Figure 2.13, but RNN is a bit more complicated, because getting the output h_t is not only from the interval $t = t$, but also from $t = t - 1, t - 2, ..., t = 1$. Therefore, backpropagation has to be included in all calculations from $t = 1$ to $t = t$. Then If the gradient value is less than 1, long continuous multiplications like this will cause the gradient to decrease as the length of its sequence. In explicit, the RNN still has a problem with the data that the sequence is too long.

Figure 2.13
The concept of optimization in a feed-forward neural network.
Reprinted from Donges (2019)

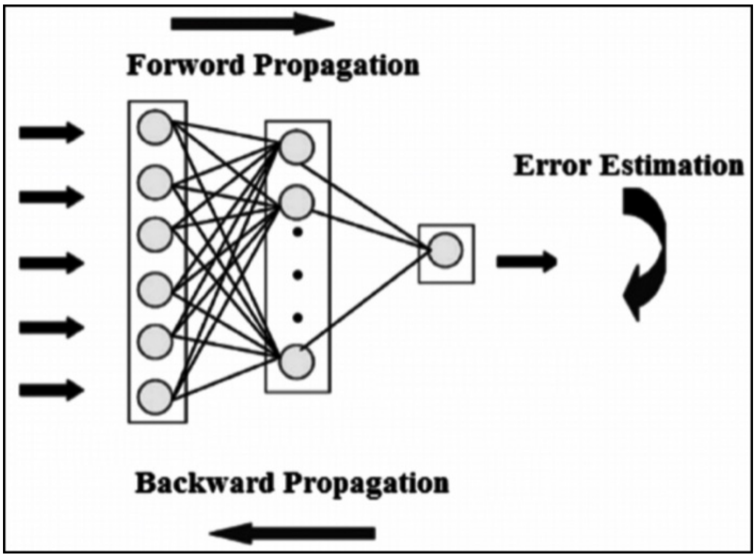
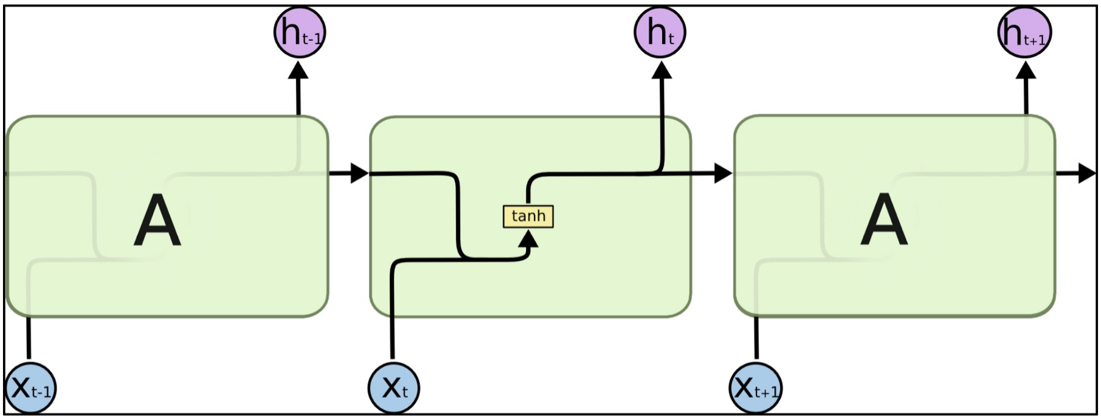


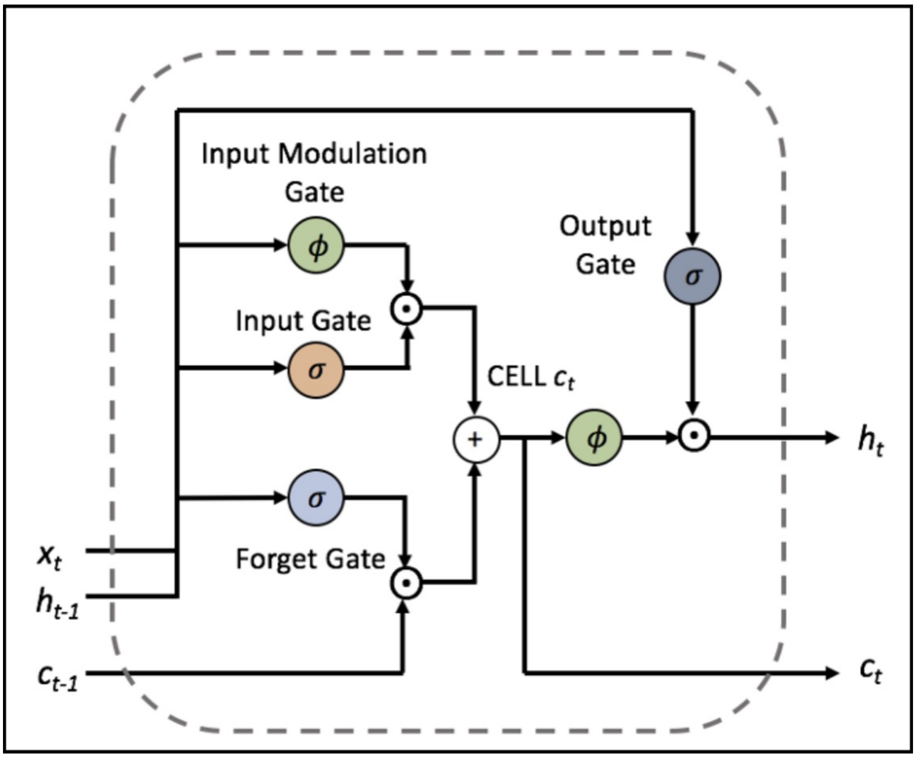
Figure 2.14
The repeating module in a standard RNN contains a single layer.
Reprinted from Olah (2015)



2.7 Long Short-Term Memory (LSTM)

Long short-term memory networks are an extension for recurrent neural networks, which basically extends the memory. Therefore it is well suited to learn from important experiences that have very long time lags in between (Donges, 2019; Olah, 2015). In addition, memory can also have a descriptor when should write, forget (delete) or read as shown in Figure 2.15.

Figure 2.15
The procedure in the LSTM.
Reprinted from Tangruamsub (2017)



Before getting into the working of LSTM, there are some variables which should be known as following (Tangruamsub, 2017):

- Cell state: Store the memory state of memory cell the LSTM
- Gate: Control the flow of data, (i.e. analog values) that control when should write to allow data flow in, read to allow data flow out or forget.

To be more clear, we will explain each functional gate one by one as following :

Forget : Forget is like clearing the old cell state, and preparing to clear memory for the new input. The person who decides whether to delete or not delete is a rule of forget gate. If the forget gate returns 0, then delete the previous cell state. If the forget gate returns 1, the model is going to store this cell state further. To create this forget gate, the model is going to look at the incoming input data with the previous hidden state (according to the RNN formula) for making decisions. The sigmoid function is used as shown in the equation below.

$$f_t = \sigma(W_{x^f} x_t + W_{h^f} h_{t-1} + b_f)$$

Write : When the new input is fed to the model, it will raise up 2 possible questions. Firstly, should the model update its cell state? This action is controlled by an input gate which still uses the sigmoid function. This computation which is shown below uses the incoming input data value and the previous hidden state.

$$i_t = \sigma(W_{x^i} x_t + W_{h^i} h_{t-1} + b_i)$$

Secondly, If the model really updates, what value should it update? It is called "Input modulation date" to handle. The equation which is shown below is similar to the input gate, but uses a tanh function instead.

$$g_t = \tanh(W_{x^c} x_t + W_{h^c} h_{t-1} + b_c)$$

Update cell state : Currently, we get information from the forget gate, input gate and input modulation gate which are enough to update cell state. The equation of update cell state will be shown below.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$$

First part, If the forget gate wants to delete the old cell state (f_t is 0), the model will not let c_{t-1} to update cell state anymore. But if f_t is 1, the model can still keep c_{t-1} to be considered. Second part, this section will update the cell state from the new data. Now that model has the values to be updated and waited from the input modulation gate or g_t. If i_t is 1, then use g_t to update. Otherwise, g_t is overlooked.

Read : From the original RNN, what the model needs to produce is the hidden state at time t or h_t. At the time of $t+1$, this LSTM takes this h_t to be calculated. Therefore, the word "read" means to allow outsiders to read the h_t or not. Or it will not pass the h_t value. We have an output gate to help the model decide as shown in the equation below.

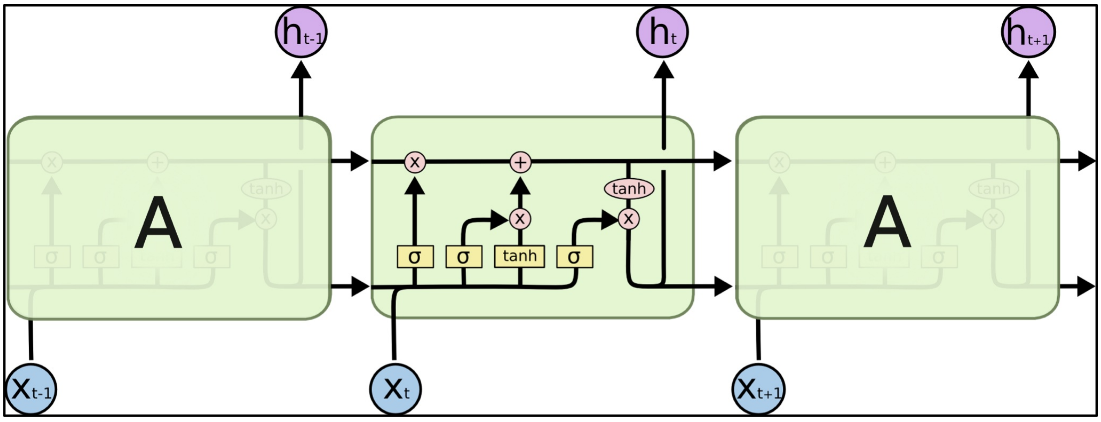
$$o_t = \sigma(W_{x^o} x_t + W_{h^o} h_{t-1} + b_o)$$

And the output will be h_t for the next sequence

$$h_t = o_t \cdot \tanh(c_t)$$

If the output gate provides o_t with 0 value, then h_t is 0 (meaning nothing is sent). Meanwhile, if o_t is 1, the model computes h_t and sends it outside or simply says that it allows others to see the h_t value.

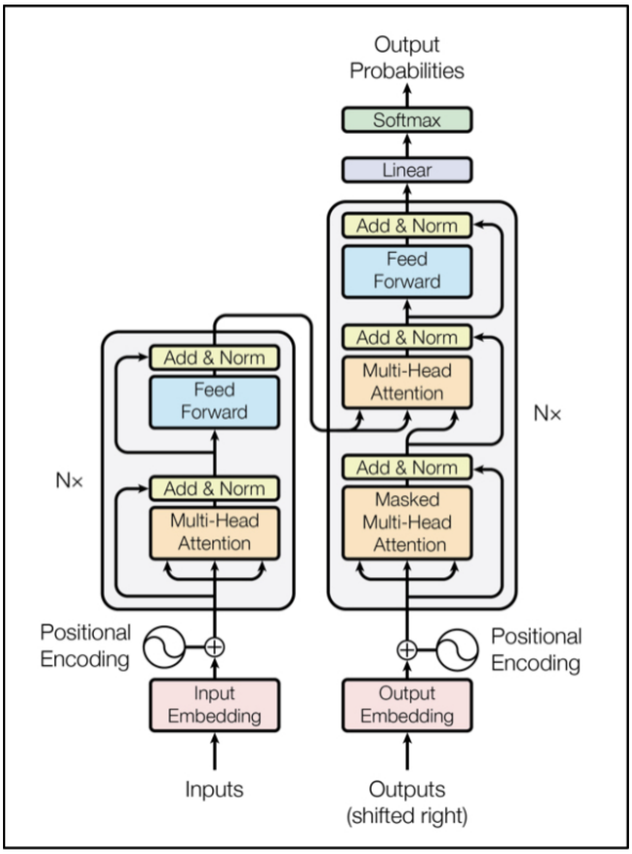
Figure 2.16
The repeating module in a LSTM contains four interacting layer.
Reprinted from Tangruamsub (2017)



2.8 Transformer

The Transformer was proposed in the paper Attention is All You Need by Google (Vaswani et al., 2017). This paper proposes a new architecture that replaces RNNs with attention called Transformer as shown in Figure 2.17. Transformer architecture has continued to beat benchmarks in many domains. Explicitly, it has revolutionized the Natural Language Processing (NLP) field particularly on the machine learning task. This model contains 2 significant parts as an encoder and decoder which can work as similar as an autoencoder. Thus, it can be used for anomaly detection purposes as well.

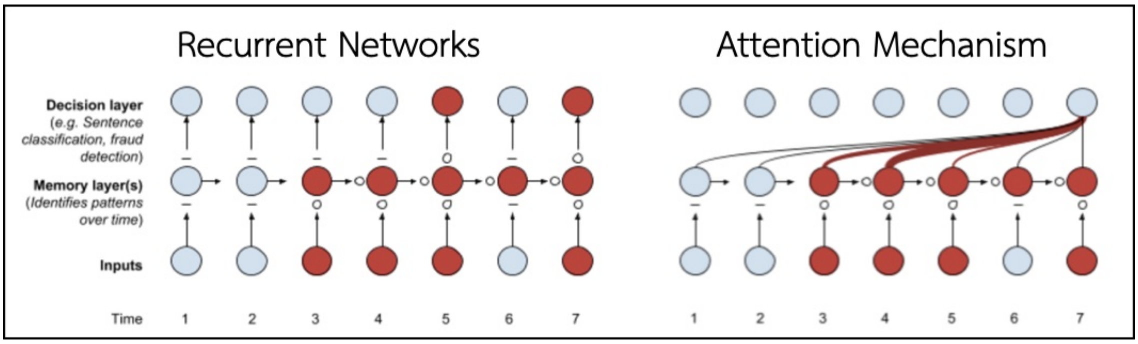
Figure 2.17
The Transformer architecture.
Reprinted from Vaswani et al. (2017)



Let's compare RNNs and attention, RNNs include every information that they had known about a sequential data into the final hidden state of the network. Thus, the decision layer can access only the memory layer which is related to that time step. It means that at every time step, it focuses on different positions on the other RNN. On the other hand, an attention mechanism regards the input from several time steps and sets different weights to each input to know which input should be focused in order to make one prediction. In Figure 2.18, this image is going to provide simple intuition of both methods.

Figure 2.18

Comparison RNNs and Attention.



2.8.1 Attention

In psychology, attention is a concentration of mind on a single object or thought, especially one preferentially selected from a complex, with a view to limiting or clarifying receptivity by narrowing the range of stimuli. Similarly, attention was specifically designed to focus on only the most important subsets of long sequences which are related to completeness of a given task (Alammar, 2018, 2019; Klingenbrunn, 2021). It actually consists 3 main steps as following:

1. Create the Query, Key, and Value vectors for each path and each input token by multiplying by weight matrices as W^Q, W^K and W^V as shown in Figure 2.19.
2. For each input token, use its query vector to get a score against all the other key vectors by multiplying the current Query vector with all the Key vectors as shown in Figure 2.20.
3. Sum up the Value vectors after multiplying them by their associated scores. The more transparent means lower value as shown in Figure 2.21.

Therefore, If the model does the same operation for each input token, it is going to finish with a vector which represents the appropriate context of each token as shown in Figure 2.22. And these vectors are going to the next sub layer in the transformer block which must be fed into the forward neural network.

Figure 2.19
Create the query, key and value vector.
Reprinted from Alammar (2018)

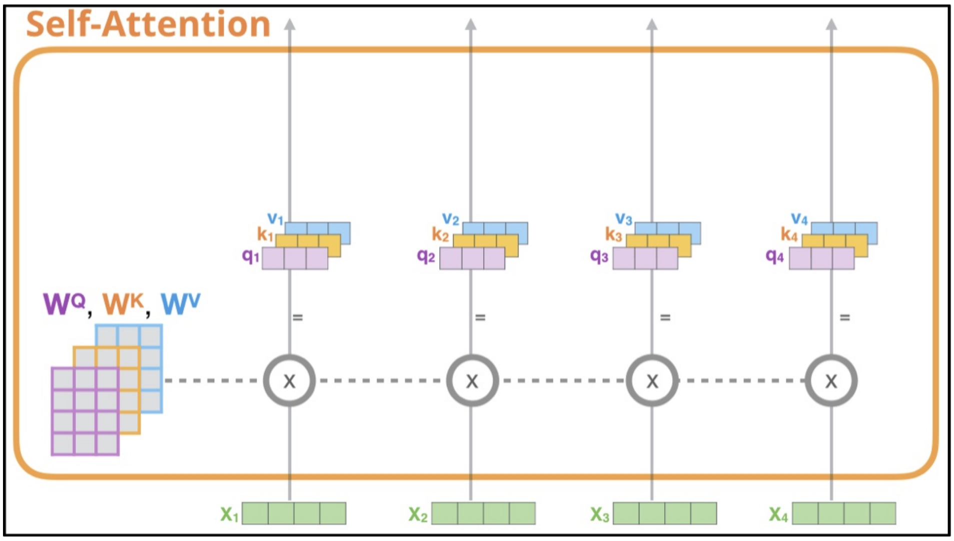


Figure 2.20
Get score of how they match.
Reprinted from Alammar (2018)

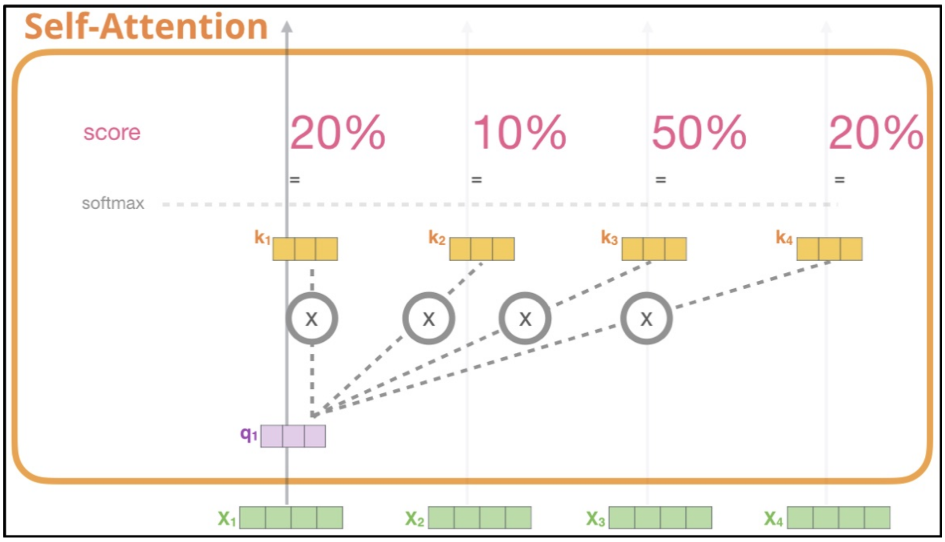


Figure 2.21
Sum up the value vectors.
Reprinted from Alammar (2018)

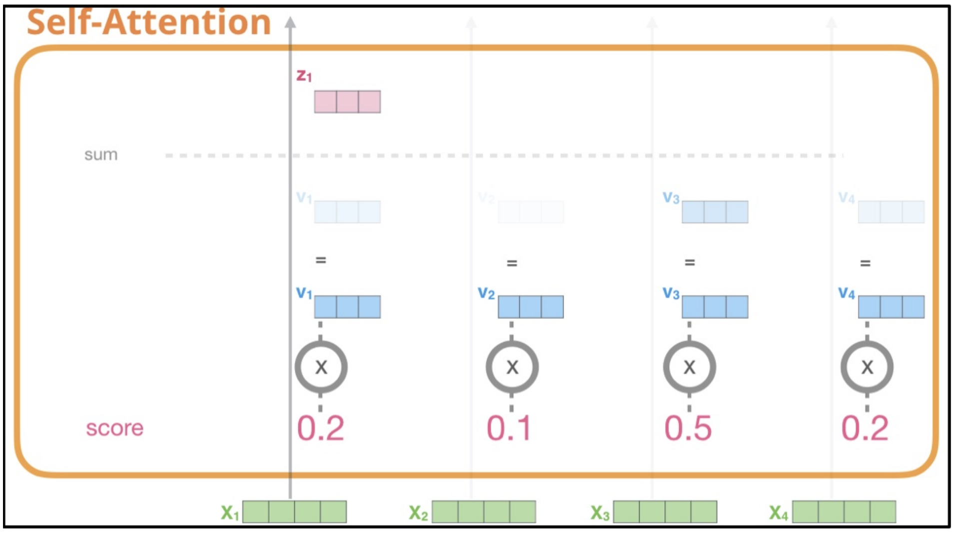
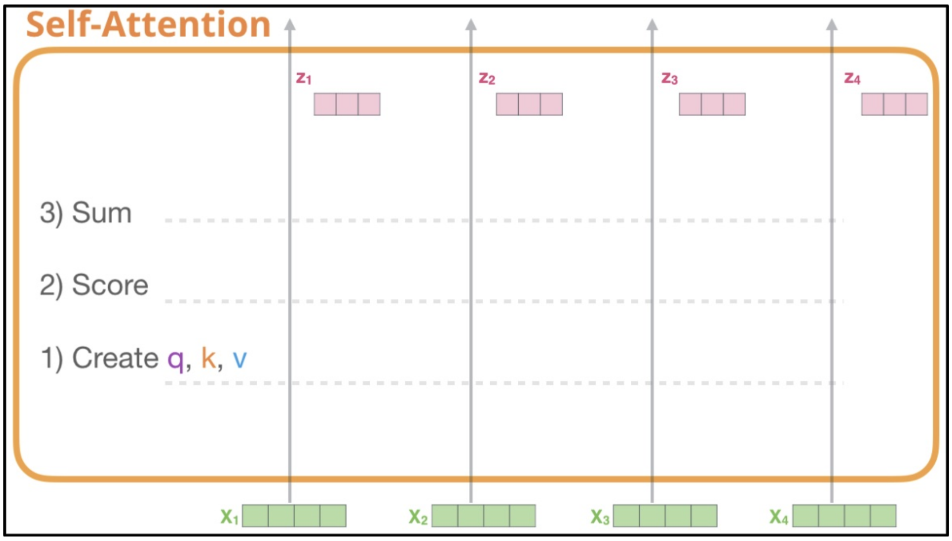
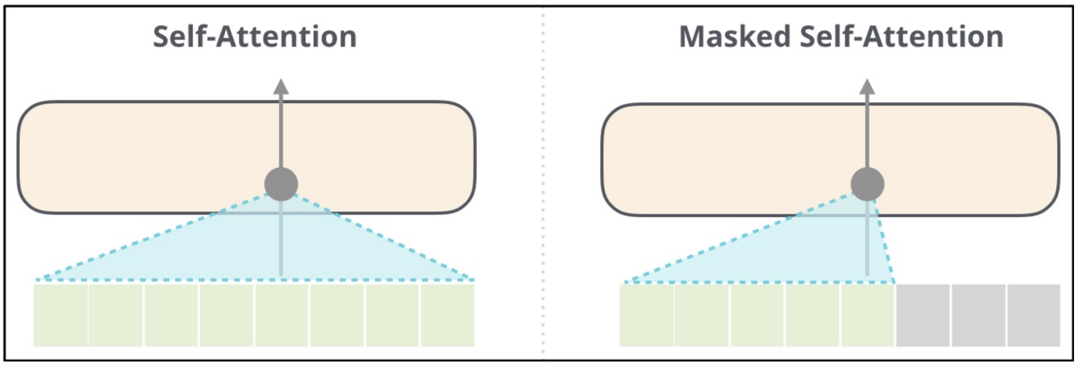


Figure 2.22
The outcome after finishing the Attention process.
Reprinted from Alammar (2018)



In addition, it can be noticed that in the decoder, it contains Masked Self-Attention. It is very important that the difference between Self-Attention and Masked Self-Attention is quite clear when you look at Figure 2.23. A Self-Attention allows each position to attend to all positions from input but Masked Self-Attention only considers the previous position and including that position in order to preserve the auto-regressive property.

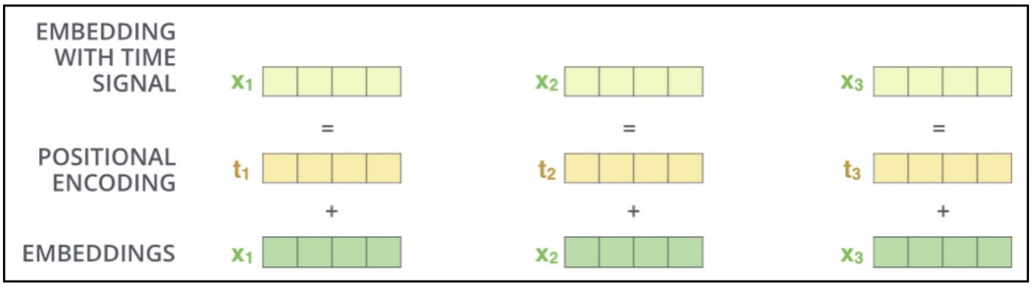
Figure 2.23
The difference of Self-Attention and Masked Self-Attention.
Reprinted from Alammar (2019)



2.8.2 Positional Encoding

For sequence to sequence model, orders and position are important. Since the model does not have recurrence and convolution, we have to add some information to make use of the order in the sequence. You can see in Figure 2.24.

Figure 2.24
The dimension of each positional encoding and embeddings.
Reprinted from Alammar (2018)



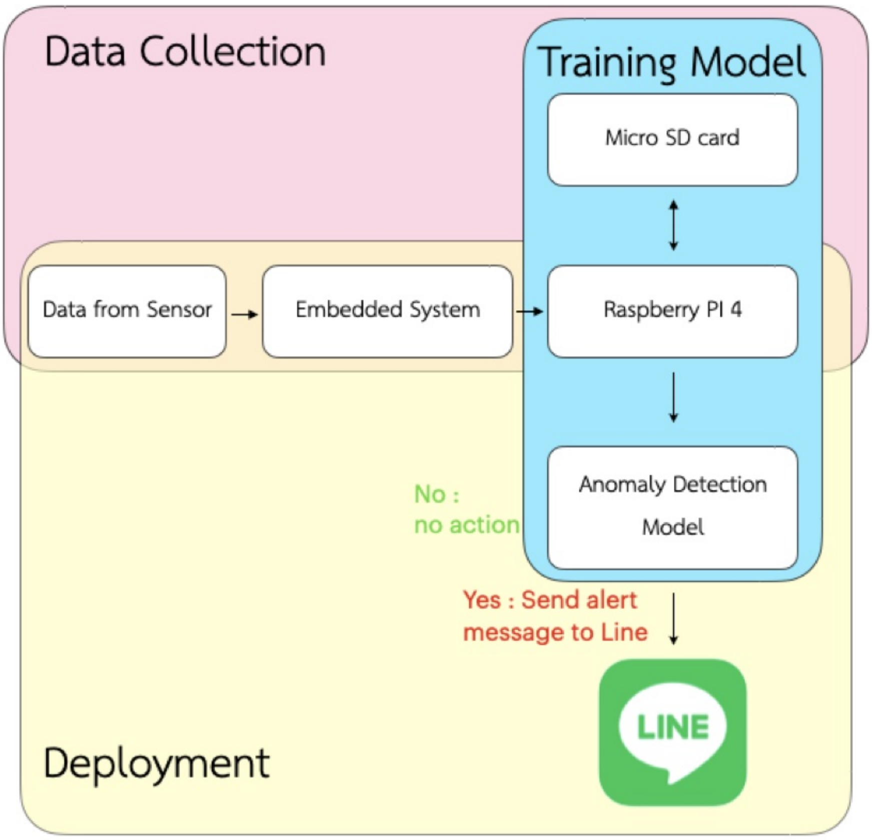
CHAPTER 3

Methodology

The main methodology of proposed study can be separated into four main processes as following and is shown in Figure 3.1:

1. Design and build filter, amplifier and embedded system for seismic sensor in order to measure vibration
2. Collect normally event such as human activities
3. Build anomaly detection models for detecting anomaly event as fall
4. Deploy this system in the real environment

Figure 3.1
The overview system architecture.



3.1 Data Collection

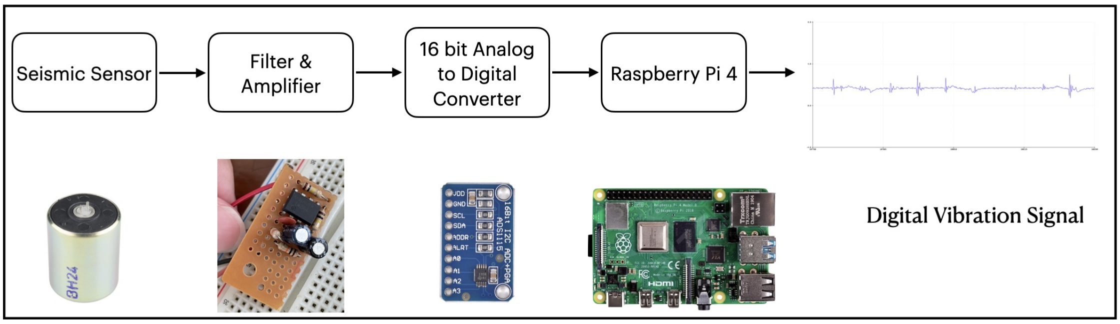
To collect raw data, we need to build our own embedded system because in order to detect human fall, the system must have the ability to detect human activities and objective drop.

And then, we have to build an algorithm for collecting the vibration signal as well.

3.1.1 Hardware

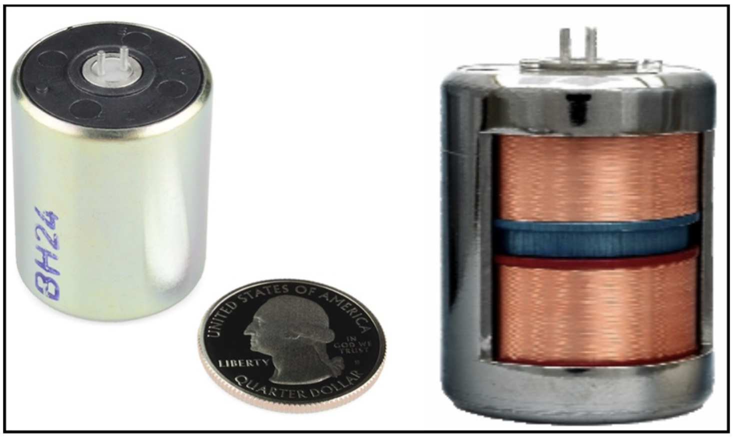
There are 4 significant components as shown in Figure 3.2. Each component has their own proposed.

Figure 3.2
The required hardware to receive raw data.



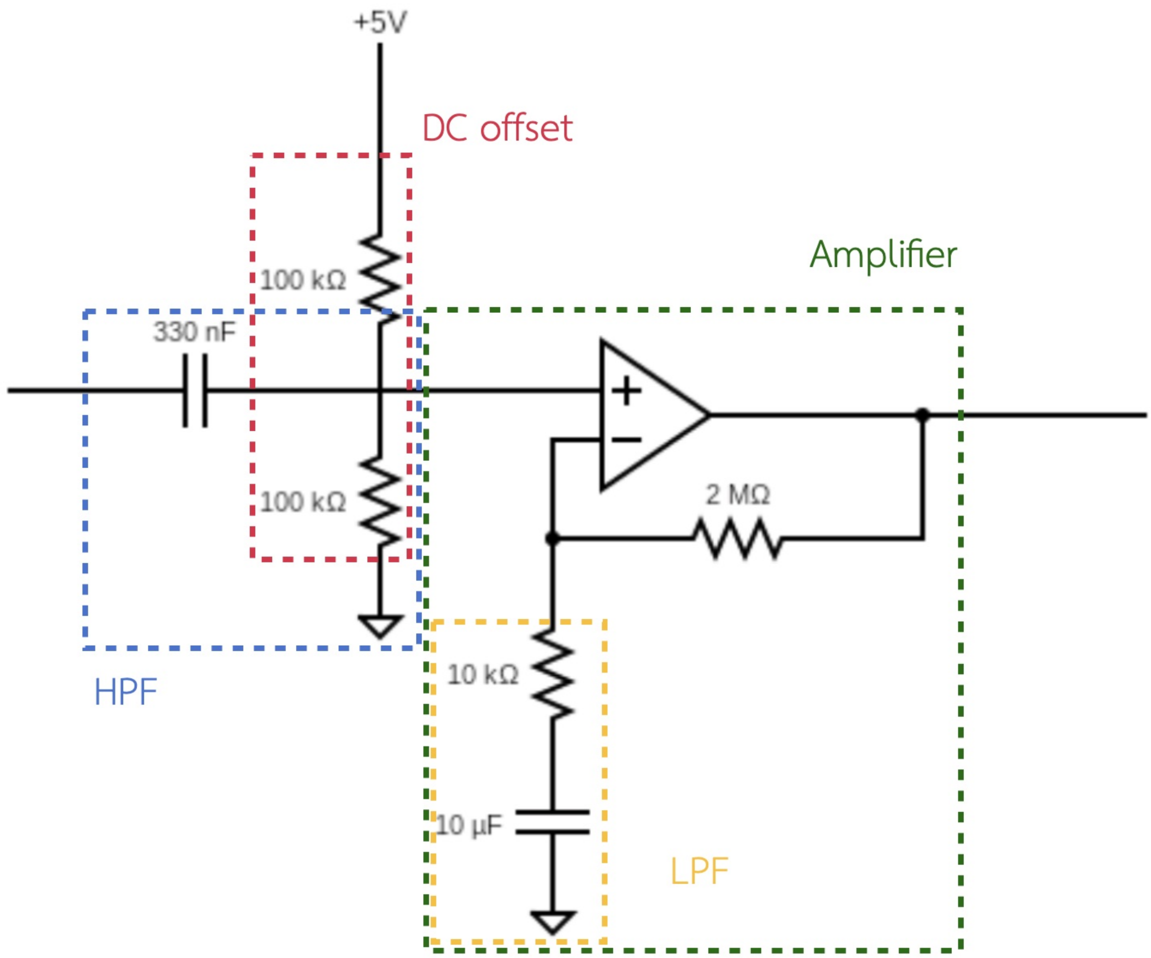
Firstly, a seismic sensor or geophone in Figure 3.3, A geophone is a device that converts ground vibration (velocity) into voltage. It has historically been passive analog devices and typically comprise a spring-mounted wire coil moving within the field of a case-mounted permanent magnet to generate an electrical signal. The reason that I decided to use this model (Geophone - SM-24) is because it has a small size similar to a coin and is easy to install just laying it on the ground. However, it cannot connect to the microcontroller directly because it can generate voltage up to $28.8V/m/s$. Therefore, we have designed an embedded system to convert voltage into range $0-5$ Volts.

Figure 3.3
A geophone SM-24 and its inside elements.
Reprinted from its brochure



Secondly, The analog circuit of filter & amplifier , which is shown in Figure 3.4, has 4 significant component as a dc offset, a high-pass filter (HPF) followed by an amplifier with including low-pass filter (LPF). The dc offset is designed for setting the reference signal as $2.5V$. The HPF is designed to filter any frequencies that are outside the frequency range of interest. It has an ideal cutoff frequency around $100Hz$. The amplifier provides the high voltage gain around $200V/V$ needed for preparing the data to be sampled at high resolution at the analog to digital converter (ADC). Lastly, the low-pass filter has cut-off frequency at $100kHz$.

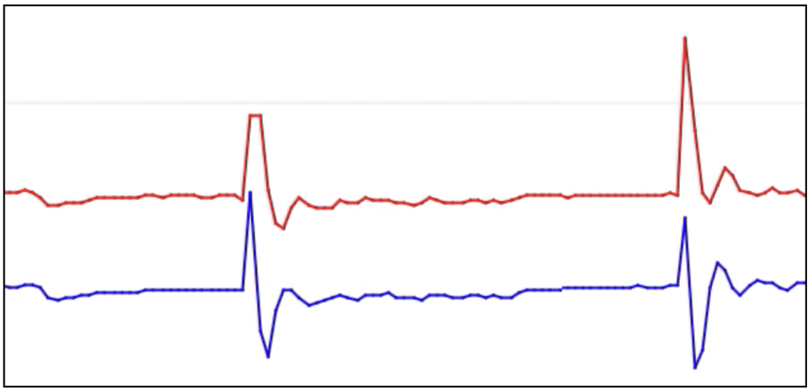
Figure 3.4
An analog circuit which was designed to be suitable with human activity.



Thirdly, analog to digital converter (ADC), the higher bit means that it can contain more information. In Figure 3.5, you can get more intuitive what is the difference between 10-bit and 16-bit ADC. The red line show the signal by using original adc pin on the Arduino mega which has only 10 bit and the blue line represent the signal by using 16-bit ADC. After the raw analog signal is filtered and amplified, it needs to be sampled at a rate of

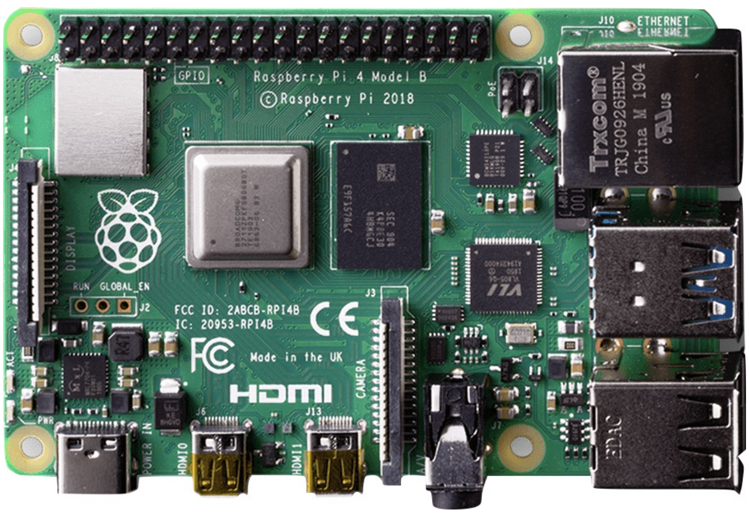
500 samples/second and quantized at a resolution of 16 bits/sample in order to prepare it for digital transmission. A 16-bit ADC is capable of distinguishing 65536 (2^{16}) different voltage levels within a narrow voltage range from $0 - 5$ Volts. It means that each level represents approximately $76.3uV$ which is enough to capture signals of seismic sensor.

Figure 3.5
This is different between 16-bit and 10-bit ADC.



Fourthly, in Figure 3.6, the used micro-computer is an Raspberry Pi 4 model B, which is a small size, low price, and several useful functions, and can do machine learning. The digital signal is fed from ADC to ESP8266 via I2C which is a synchronous serial communication interface specification used for short-distance communication. And then these raw signals are directly collected into mircro sd card.

Figure 3.6
Microcomputer - Raspberry Pi 4.



3.1.2 Experimental Setup

To collect the raw data, experiments are performed in the living room of my house in Bangkok, Thailand which was built from reinforced concrete structure and on top with tile as show in Figure 3.7. Actually, my system has detectable range around 3 meters and this room also has dimension 3.5 x 3.5 $meter^2$. The hardware should be installed near the corner in order to be as suitable for the application as possible.

Figure 3.7
The living room where is used for experiment.

3.1.3 Experimental Event

There are several events which should occur during the daily event. However, the author would like to only collect often activities such as walking, sitting, standing and lying as shown in Table 3.1. Sincerely, due to covid-19 situation, I cannot invite strange volunteers to come in my house in order to collect data. However, If the covid situation in Thailand is better, I will invite approximate 5 - 10 friends to join this experimental event. Therefore, I plan to collect these activity with 4 subjects who are family member as my father, my mother, my older brother and me, and the detail of them is shown in Table 3.2.

Table 3.1
The detail of each activity and its number of action.

Human Activity	Number of action
Walking	2,500
Sitting	400
Standing	400
Lying	400

Table 3.2
The detail of each participant and their details.

Subject	Sex	Age	Weight (kg)
1	M	23	58
2	M	25	70
3	M	55	70
4	F	58	75

3.2 Anomaly Detection Models

I choose two candidate models as autoencoder with LSTM architecture in Figure 3.8 and Transformer in Figure 3.9 , which are quite distinguished in this field.

Figure 3.8
The autoencoder with LSTM architecture

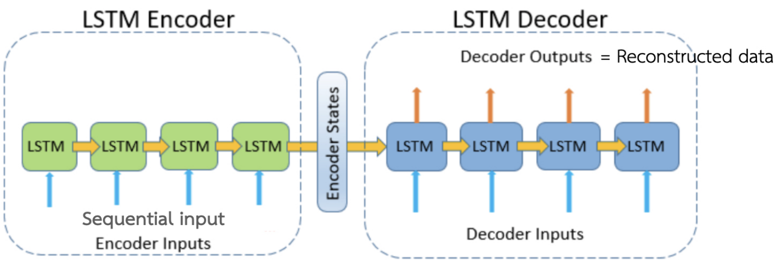
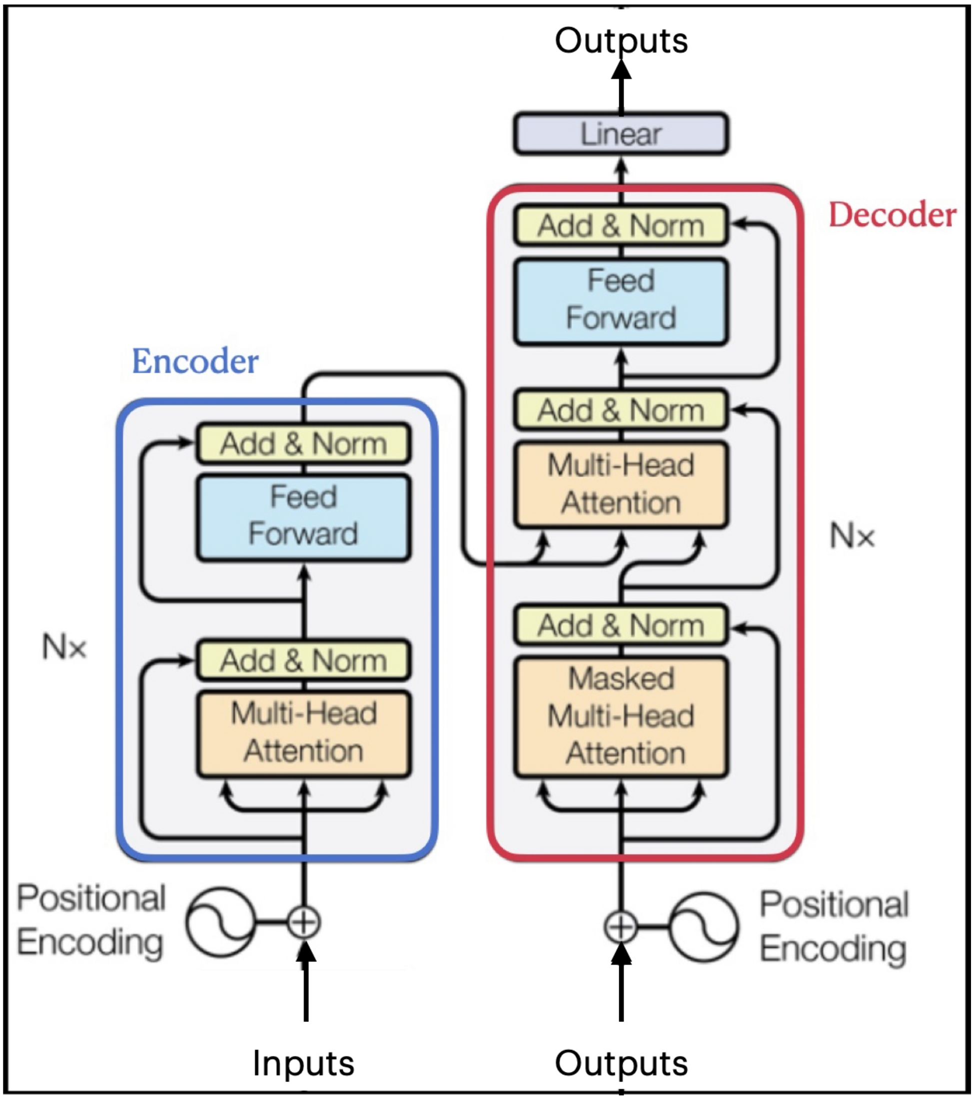


Figure 3.9
The Transformer architecture.
Reprinted from work of Vaswani et al. (2017)



3.2.1 Evaluation Plan

To assess the performance of the model, The model will be tested with untrained activities such as jumping and dropping the ball in different locus around the installed system. The prefered accuracy should be greater than 75 percents of true positive and 100 percents of true negative. (Not sure may be the number need to be changed)

3.3 Deployment

The desirable system should be plug & play. Therefore, every component such as seismic sensor, embedded system and Raspberry Pi must be integrated into the small box, which require only power adapter as shown in Figure 3.10. When anomaly activities are occuring, an alert messages is going to be sent via Line application to me. The dining room in my house will be the tested location of this study shown in 3.11.

Figure 3.10
The complete system.

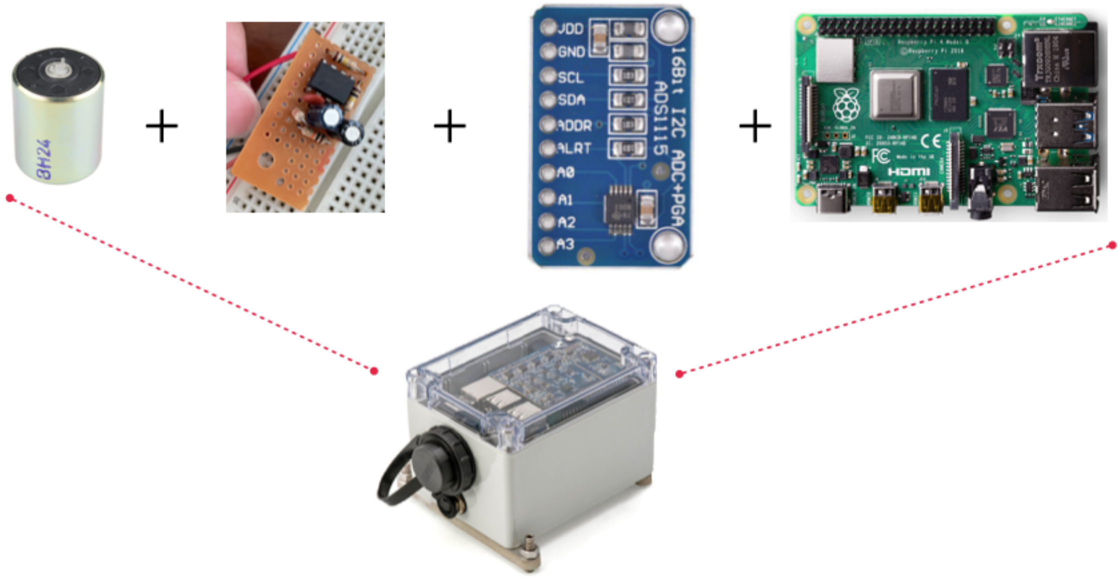


Figure 3.11
The dining room in my home.

CHAPTER 4

EXPECTED OUTCOMES

This chapter describes the expected results of this study based on objectives. The expected results are:

For the object 1: Design and build filter, amplifier and embedded system.

- The analog circuit and embedded system which can detect vibrational signal of human activity on the concrete floor

For the object 2: Collect normally human activities.

- The csv file that contains raw sequential data of each activity, name of activity and subject.

For the object 3: Build anomaly detection model for detecting anomaly event as fall

For the object 4: Deploy this system in the dining room

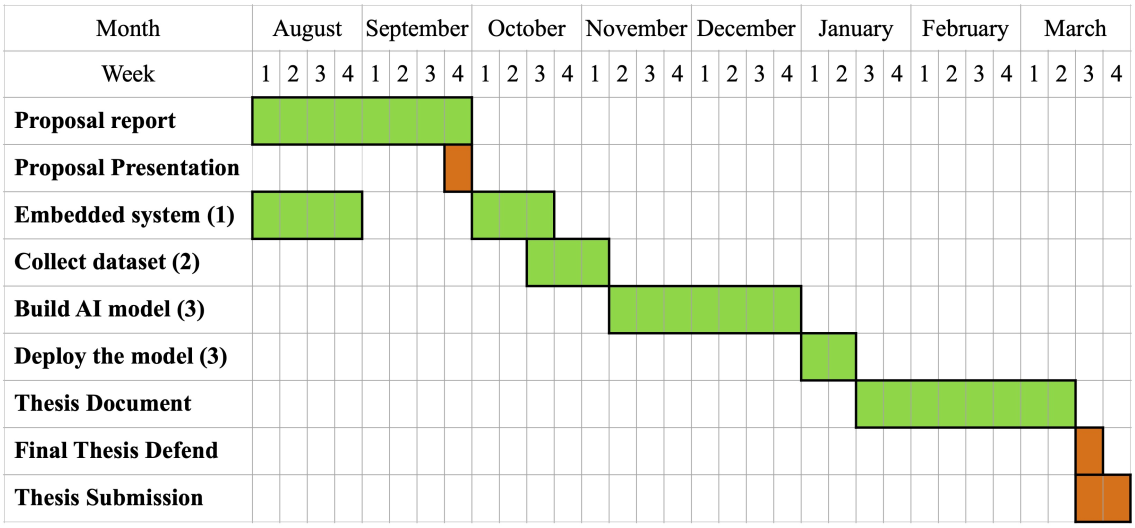
- Can send alert message to the registered user though Line application when fall occur immediately.

CHAPTER 5

WORK PLAN

In Figure 5.1, it illustrates the working plan of this study in the year of August 2021 to March 2022.

Figure 5.1
The schedule working plan of this study.



REFERENCES

Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., & Vecchio, A. (2012). A smartphone-based fall detection system. *Pervasive and Mobile Computing*, *8*(6), 883–899. doi: 10.1016/j.pmcj.2012.08.003

Agmon, A. (2021). *Hands-on anomaly detection with variational autoencoders.* Retrieved from `https://towardsdatascience.com/hands-on-anomaly-detection-with-variational-autoencoders-d4044672acd5`

Alammar, J. (2018). *The illustrated transformer.* Retrieved from `https://jalammar.github.io/illustrated-transformer/`

Alammar, J. (2019). *The illustrated gpt-2 (visualizing transformer language models).* Retrieved from `https://jalammar.github.io/illustrated-gpt2/#part-2-illustrated-self-attention`

Alwan, M., Dalal, S., Kell, S., & Felder, R. (2003). Derivation of basic human gait characteristics from floor vibrations.

Alwan, M., Rajendran, P., Kell, S., Mack, D., Dalal, S., Wolfe, M., & Felder, R. (2006). *A smart and passive floor-vibration based fall detector for elderly* (Vol. 1). Retrieved from `https://ieeexplore.ieee.org/document/1684511` doi: 10.1109/ICTTA.2006.1684511

Arnx, A. (2019). *First neural network for beginners explained (with code).* Towards Data Science. Retrieved from `https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf`

Badr, W. (2019). *Auto-encoder: What is it? and what is it used for? (part 1).* Retrieved from `https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726`

Britto Filho, J. C., & Lubaszewski, M. (2020). A highly reliable wearable device for fall detection. *2020 IEEE Latin-American Test Symposium (LATS)*. doi: 10.1109/lats49555.2020.9093673

Charlon, Y., Bourennane, W., Bettahar, F., & Campo, E. (2013). Activity monitoring system for elderly in a context of smart home. *IRBM*, *34*(1), 60–63. Retrieved from `https://www.sciencedirect.com/science/article/abs/pii/S1959031812001509` doi: 10.1016/j.irbm.2012.12.014

Chen, Y., & Xue, Y. (2015). A deep learning approach to human activity recognition based on single accelerometer. *2015 IEEE International Conference on Systems, Man, and Cybernetics*. Retrieved from `https://ieeexplore.ieee.org/document/7379395/` doi: 10.1109/smc.2015.263

Chollet, F. (2016). *Building autoencoders in keras.* Retrieved from `https://blog.keras.io/building-autoencoders-in-keras.html`

Clemente, J., Li, F., Valero, M., & Song, W. (2020). Smart seismic sensing for indoor fall detection, location, and notification. *IEEE Journal of Biomedical and Health Informatics*, *24*(2), 524–532. doi: 10.1109/jbhi.2019.2907498

Dash, S. (2020). *An overview of time series forecasting models part 1: Classical time series forecasting models.* Retrieved from `https://shaileydash.medium.com/an-overview-of-time-series-forecasting-models-part-1-classical-time-series-forecasting-models-2d877de76e0f`

Davis, B. T., Caicedo, J. M., Langevin, S., & Hirth, V. (2011). Use of wireless smart sensors for detecting human falls through structural vibrations. *Civil Engineering Topics, Volume 4*, 383–389. doi: 10.1007/978-1-4419-9316-8_37

Degen, T., Jaeckel, H., Rufer, M., & Wyss, S. (2003). Speedy:a fall detector in a wrist watch. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*. doi: 10.1109/iswc.2003.1241410

Donges, N. (2019). *Recurrent neural networks 101: Understanding the basics of rnns and lstm.* Retrieved from https://builtin.com/data-science/recurrent-neural-networks-and-lstm

El-Bendary, N., Tan, Q., C. Pivot, F., & Lam, A. (2013). Fall detection and prevention for the elderly: a review of trends and challenges. *International Journal on Smart Sensing and Intelligent Systems*, *6*(3), 1230–1266. doi: 10.21307/ijssis-2017-588

Fuller, G. F. (2013). Falls in the elderly. *American Family Physician*, *61*(7), 2159. Retrieved from https://www.aafp.org/afp/2000/0401/p2159.html

Gavin, H. (2015). *Vibrations of single degree of freedom systems.* Retrieved from https://people.duke.edu/~hpgavin/cee201/sdof-dyn.pdf

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*. doi: 10.1007/s10618-019-00619-1

Jia, Z., Howard, R. E., Zhang, Y., & Zhang, P. (2017, Apr). Hb-phone: a bed-mounted geophone-based heartbeat monitoring system. *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. doi: 10.1145/3055031.3055042

Kasturi, S., & Jo, K.-H. (2017). Classification of human fall in top viewed kinect depth images using binary support vector machine. *2017 10th International Conference on Human System Interactions (HSI)*. doi: 10.1109/hsi.2017.8005016

Klingenbrunn, N. (2021). *Transformer implementation for time-series forecasting.* Retrieved from https://medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820

Litvak, D., Zigel, Y., & Gannot, I. (2008). Fall detection of elderly through floor vibrations and sound. *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. doi: 10.1109/iembs.2008.4650245

Liu, C., Jiang, Z., Su, X., Benzoni, S., & Maxwell, A. (2019). Detection of human fall using floor vibration and multi-features semi-supervised svm. *Sensors*, *19*(17), 3720. doi: 10.3390/s19173720

Ljunggren, F. (2006). *Floor vibration: dynamic properties and subjective perception* (Unpublished doctoral dissertation). Luleå tekniska universitet.

Madarshahian, R., Caicedo, J. M., & Arocha Zambrana, D. (2016). Benchmark problem for human activity identification using floor vibrations. *Expert Systems with Applications*, *62*, 263–272. doi: 10.1016/j.eswa.2016.06.027

Mukherjee, A., & Zhang, Z. (2020). Multisense: A highly reliable wearable-free human fall detection systems. In *Sensornets* (pp. 29–40).

Müller, M. (2007, 01). Dynamic time warping. *Information Retrieval for Music and Motion*, *2*, 69-84. doi: 10.1007/978-3-540-74048-3_4

NHS. (2019). *Overview - falls.* Retrieved from https://www.nhs.uk/conditions/Falls/

Olah, C. (2015). *Understanding lstm networks – colah's blog.* Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oukrich, N. (2019). *Daily human activity recognition in smart home based on feature selection, neural network and load signature of appliances* (Unpublished doctoral dissertation).

pavithrasv. (2020). *Timeseries anomaly detection using an autoencoder.*

Pedamkar, P. (2019). *Autoencoders — main components and architecture of autoencoder.* Retrieved from https://www.educba.com/autoencoders/

Pynoos, J., Steinman, B. A., & Nguyen, A. Q. (2010). Environmental assessment and modification as fall-prevention strategies for older adults. *Clinics in Geriatric Medicine*, *26*(4), 633–644. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6036911/ doi: 10.1016/j.cger.2010.07.001

Rajan, S. (2021). *Dimensionality reduction using autoencoders in python.* Retrieved from https://www.analyticsvidhya.com/blog/2021/06/dimensionality-reduction-using-autoencoders-in-python/

Ramirez, H., Velastin, S. A., Meza, I., Fabregas, E., Makris, D., & Farias, G. (2021). Fall detection and activity recognition using human skeleton features. *IEEE Access*, *9*, 33532–33542. doi: 10.1109/access.2021.3061626

Reiss, A., & Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. *2012 16th International Symposium on Wearable Computers*. doi: 10.1109/iswc.2012.13

Rihana, S., & Mondalak, J. (2016). *Wearable fall detection system.* Retrieved from https://ieeexplore.ieee.org/abstract/document/7745414/ doi: 10.1109/MECBME.2016.7745414

Rocca, J. (2020). *Understanding variational autoencoders (vaes).* Retrieved from https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Roger. (2021). *Variational autoencoder(vae).* Retrieved from https://medium.com/geekculture/variational-autoencoder-vae-9b8ce5475f68

Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Forster, K., Troster, G., … Millan, J. d. R. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In (p. 233 - 240). doi: 10.1109/INSS.2010.5573462

Rosebrock, A. (2020). *Denoising autoencoders with keras, tensorflow, and deep learning.* Retrieved from https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/

Schrader, L., Vargas Toro, A., Konietzny, S., Rüping, S., Schäpers, B., Steinböck, M., … Bock, T. (2020). Advanced sensing and human activity recognition in early intervention and rehabilitation of elderly people. *Journal of Population Ageing*, *13*(2), 139–165. doi: 10.1007/s12062-020-09260-z

Shao, Y., Wang, X., Song, W., Ilyas, S., Guo, H., & Chang, W.-S. (2020). Feasibility of using floor vibration to detect human falls. *International Journal of Environmental Research and Public Health*, *18*(1), 200. doi: 10.3390/ijerph18010200

SteelConstruction. (2016). *Floor vibrations.* Retrieved from https://www.steelconstruction.info/Floor_vibrations

Sultana, A., Deb, K., Dhar, P. K., & Koshiba, T. (2021). Classification of indoor human fall events using deep learning. *Entropy*, *23*(3), 328. doi: 10.3390/e23030328

Tangruamsub, S. (2017). *Long short-term memory (lstm).* Medium. Retrieved from https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6

Taufeeque, M., Koita, S., Spicher, N., & Deserno, T. M. (2021). Multi-camera, multi-person, and real-time fall detection using long short term memory. *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*. doi: 10.1117/12.2580700

ThaiNCD.com. (2019). *Old thai people do not fall, no secret tips! for the prevention of falls in tall people.* Retrieved from `https://thaincd.com/2016/media-detail.php?id=13551&gid=1-015-009`

Toyoda, M., & Sakurai, Y. (2012). *Subsequence matching in data streams — ntt technical review.* Retrieved from `https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201301ra1.html`

Tsai, T.-H., & Hsu, C.-W. (2019). Implementation of fall detection system based on 3d skeleton for deep learning technique. *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*. doi: 10.1109/gcce46687.2019.9015609

Ugolotti, R., Sassi, F., Mordonini, M., & Cagnoni, S. (2011). Multi-sensor system for detection and classification of human activities. *Journal of Ambient Intelligence and Humanized Computing*, *4*(1), 27–41. doi: 10.1007/s12652-011-0065-z

University, J. H. (2015). *The swiss roll matching example.* Retrieved from `http://www.cis.jhu.edu/~cshen/html/PublishSwissRoll.html`

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., N. Gomez, A., … Polosukhin, I. (2017). *Attention is all you need.* Retrieved from `https://arxiv.org/abs/1706.03762v5`

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*. doi: 10.1145/1390156.1390294

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(110), 3371-3408. Retrieved from `http://jmlr.org/papers/v11/vincent10a.html`

Wang, S., Chen, L., Zhou, Z., Sun, X., & Dong, J. (2015). Human fall detection in surveillance video based on pcanet. *Multimedia Tools and Applications*, *75*(19), 11603–11613. Retrieved from `https://link.springer.com/article/10.1007%2Fs11042-015-2698-y` doi: 10.1007/s11042-015-2698-y

Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly fall detection systems: a literature survey. *Frontiers in Robotics and AI*, *7*. doi: 10.3389/frobt.2020.00071

Weng, L. (2018). *From autoencoder to beta-vae.* Retrieved from `https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`

WHO. (2018). *Falls.* World Health Organization: WHO. Retrieved from `https://www.who.int/news-room/fact-sheets/detail/falls`

Wikipedia. (2021). *Dynamic time warping.* Retrieved from `https://th.wikipedia.org/wiki/Dynamic_time_warping`

Yang, C.-C., & Hsu, Y.-L. (2010). A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, *10*(8), 7772–7788. doi: 10.3390/s100807772

Yazar, A., Erden, F., & Cetin, A. (2014, 05). Multi-sensor ambient assisted living system for fall detection..