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 Governmen

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

ACKNOWLEDGMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Aliquam malesuada bibendum arcu vitae elementum. Aliquet enim tortor at auctor. Sodales ut eu sem integer vitae justo. Tellus pellentesque eu tincidunt tortor aliquam nulla facilisi. Ut lectus arcu bibendum at varius vel pharetra. Integer vitae justo eget magna. Mollis nunc sed id semper risus. Donec ac odio tempor orci dapibus ultrices in iaculis. Non arcu risus quis varius quam quisque id diam vel. Felis donec et odio pellentesque diam volutpat commodo sed egestas. Dignissim diam quis enim lobortis scelerisque. Placerat vestibulum lectus mauris ultrices eros in cursus.

Ut venenatis tellus in metus vulputate eu scelerisque felis. Blandit volutpat maecenas volutpat blandit aliquam etiam erat velit. Ac felis donec et odio pellentesque diam volutpat commodo sed. Dignissim convallis aenean et tortor at risus viverra adipiscing at. Donec pretium vulputate sapien nec. Nisi porta lorem mollis aliquam. Sed nisi lacus sed viverra. Metus vulputate eu scelerisque felis imperdiet proin fermentum leo vel. Convallis convallis tellus id interdum velit laoreet id donec ultrices. Volutpat diam ut venenatis tellus in metus. Magna fringilla urna porttitor rhoncus dolor purus non enim. Eget mi proin sed libero enim sed faucibus turpis in. Bibendum ut tristique et egestas quis ipsum suspendisse. Tellus integer feugiat scelerisque varius morbi. Aliquet eget sit amet tellus. Massa tincidunt dui ut ornare lectus sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Cum sociis natoque penatibus et magnis dis parturient montes nascetur. Et egestas quis ipsum suspendisse ultrices. Mauris vitae ultricies leo integer malesuada nunc vel risus.

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Aliquam malesuada bibendum arcu vitae elementum. Aliquet enim tortor at auctor. Sodales ut eu sem integer vitae justo. Tellus pellentesque eu tincidunt tortor aliquam nulla facilisi. Ut lectus arcu bibendum at varius vel pharetra. Integer vitae justo eget magna. Mollis nunc sed id semper risus. Donec ac odio tempor orci dapibus ultrices in iaculis. Non arcu risus quis varius quam quisque id diam vel. Felis donec et odio pellentesque diam volutpat commodo sed egestas. Dignissim diam quis enim lobortis scelerisque. Placerat vestibulum lectus mauris ultrices eros in cursus.

Ut venenatis tellus in metus vulputate eu scelerisque felis. Blandit volutpat maecenas volutpat blandit aliquam etiam erat velit. Ac felis donec et odio pellentesque diam volutpat commodo sed. Dignissim convallis aenean et tortor at risus viverra adipiscing at. Donec pretium vulputate sapien nec. Nisi porta lorem mollis aliquam. Sed nisi lacus sed viverra. Metus vulputate eu scelerisque felis imperdiet proin fermentum leo vel. Convallis convallis tellus id interdum velit laoreet id donec ultrices. Volutpat diam ut venenatis tellus in metus. Magna fringilla urna porttitor rhoncus dolor purus non enim. Eget mi proin sed libero enim sed faucibus turpis in. Bibendum ut tristique et egestas quis ipsum suspendisse. Tellus integer feugiat scelerisque varius morbi. Aliquet eget sit amet tellus. Massa tincidunt dui ut ornare lectus sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Cum sociis natoque penatibus et magnis dis parturient montes nascetur. Et egestas quis ipsum suspendisse ultrices. Mauris vitae ultricies leo integer malesuada nunc vel risus.

Keywords: keyword1, keyword2.

CONTENTS

		Page
ACKNOWLEDGMENTS		**ii**
ABSTRACT		**iii**
LIST OF TABLES		**vi**
LIST OF FIGURES		**vii**
CHAPTER 1	**INTRODUCTION**	**1**
1.1	Background of the Study	1
1.2	Statement of the Problem	1
1.3	Research Questions	4
1.4	Objectives of the Study	4
1.5	Scope and Limitations	4
CHAPTER 2	**Literature Review**	**5**
2.1	Fall	5
2.1.1	Fall Detection	5
2.1.2	Fall Detection by using Vibration sensor	6
2.2	Human Activity	7
2.3	Floor Vibrations	9
2.4	Time Series	10
2.4.1	Autoregressive (AR)	11
2.4.2	Dynamic Time Warping (DTW)	11
2.4.3	Convolution Neural Network (CNN)	12
2.5	Autoencoder	13
2.5.1	Autoencoder for Anomaly Detection	16
2.5.2	Variational Autoencoder	17
2.6	Recurrent Neural Network (RNN)	18
2.7	Long Short-Term Memory (LSTM)	20
2.8	Transformer	23
2.8.1	Attention	25
2.8.2	Positional Encoding	28
CHAPTER 3	**Methodology**	**29**
3.1	Data Collection	29
3.1.1	Hardware	30
3.1.2	Experimental Setup	33
3.1.3	Experimental Event	33

3.2 Anomaly Detection Models 34

3.2.1 Evaluation Plan 35

3.3 Deployment 35

CHAPTER 4 EXPECTED OUTCOMES 36

CHAPTER 5 WORK PLAN 37

REFERENCES 38

APPENDICES

APPENDIX A: SURVEY QUESTIONNAIRE 42

APPENDIX B: TITLE 43

LIST OF TABLES

Tables		Page
Table 2.1	Summary of literature review for fall detection from floor vibration	7
Table 2.2	Summary of literature review for human activity.	8
Table 3.1	The detail of each activity and its number of action.	34
Table 3.2	The detail of each participant and their details.	34
Table B1	Random Table B	43

LIST OF FIGURES

Figures		Page
Figure 2.1	Fall detection trends	6
Figure 2.2	The single degree of freedom systems	10
Figure 2.3	The different of Euclidean and DTW matching.	12
Figure 2.4	Convolving on univariate input time series	13
Figure 2.5	Fully Convolutional Neural Network architecture.	13
Figure 2.6	How does an autocoder work?	14
Figure 2.7	The autoencoder architecture.	15
Figure 2.8	The dimensionality reduction by using autoencoder.	15
Figure 2.9	An autoencoder tries to correct the output.	16
Figure 2.10	An autoencoder knows that an anomaly event occurs	17
Figure 2.11	The architecture of variational autoencoder.	18
Figure 2.12	The Recurrent Neural Network architecture.	19
Figure 2.13	The concept of optimization in a feed-forward neural network.	20
Figure 2.14	The repeating module in a standard RNN contains a single layer.	20
Figure 2.15	The procedure inside the LSTM.	21
Figure 2.16	The repeating module in a LSTM contains four interacting layer.	23
Figure 2.17	The Transformer architecture.	24
Figure 2.18	Comparison RNNs and Attention.	25
Figure 2.19	Create the query, key and value vector.	26
Figure 2.20	Get score of how they match.	26
Figure 2.21	Sum up the value vectors.	27
Figure 2.22	The outcome after finishing the Attention process.	27
Figure 2.23	The difference of Self-Attention and Masked Self-Attention.	28
Figure 2.24	The dimension of each positional encoding and embeddings.	28
Figure 3.1	The overview system architecture.	29
Figure 3.2	The required hardware to receive raw data.	30
Figure 3.3	A geophone SM-24 and its inside elements.	30
Figure 3.4	An analog circuit which was designed to be suitable with human activity	31
Figure 3.5	This is different between 16-bit and 10-bit ADC.	32
Figure 3.6	Microcontroller - Raspberry Pi 4	32
Figure 3.7	The living room where is used for experiment	33
Figure 3.8	The autoencoder with LSTM architecture.	34
Figure 3.9	The Transformer architecture.	35

Figure 5.1 The schedule working plan of this study 37
Figure B2 CCTV monitoring room in Appendix B. 43

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, stumbling and missing a step on the same ground level. They report an average 140 calls 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 and 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 fall 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

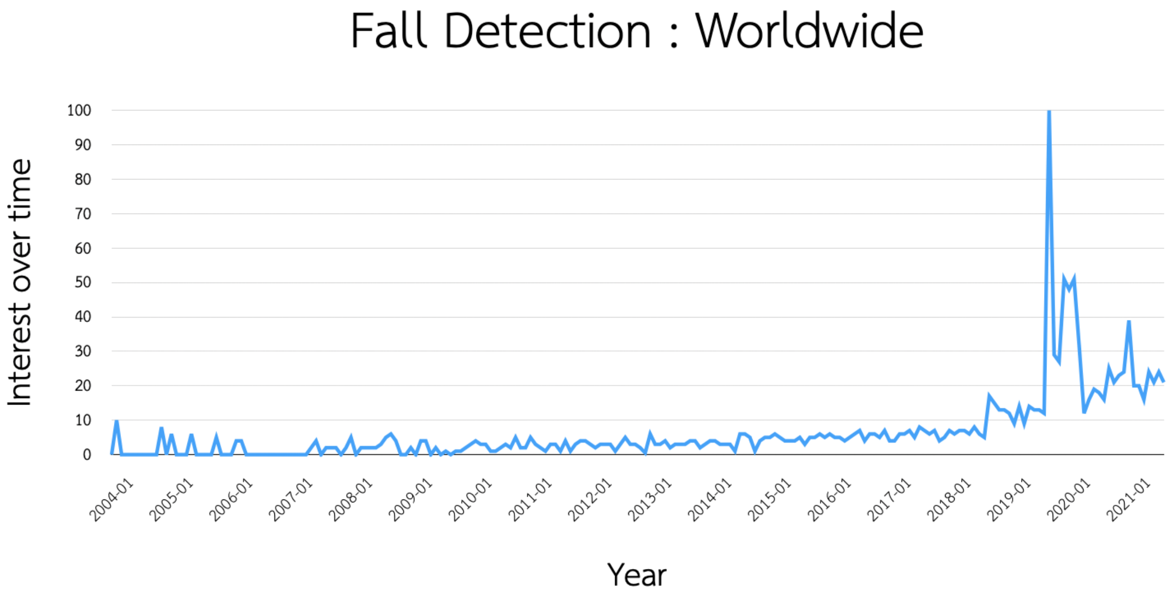
2.1 Fall

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. As statistic, Around 1 in 3 adults over 65 and half of people over 80 will have at least one fall a year (NHS, 2019). 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 time 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 topic has gotten increasingly more attention over time and significantly increased in 2019. 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 of interest and 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 sensor

In the Table 2.1, it shows the evolution of fall detection from floor vibration. It can be notice that most of them used classifier model to detect the fall event by using simulated fall data which they were not actual fall. Furthermore, they do not deploy their system in the real environment which cannot guarantee an actual performance of model. Thus to overcome these weaknesses, author would like to apply anomaly detection approach to compensate rarely falling down data or unseen pattern, and send alert to caretaker or assistance who always can take care lonely victim 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	Smartphone accelerometer	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

The readers are going to have a doubt why I need to research human activity. I can easily say that If I apply the traditional machine learning method which often uses supervised learning technique, it means I have to collect fall data for classification purposes. And once fall events occur, the system should detect and alert the users. Nonetheless, I prefer to apply anomaly detection techniques to handle this problem. And then, normally activity should be fed to the model in order to let the model learn. Thus, it would be better that I research for fundamental human activity at home.

Schrader et al. (2020) say there is no common definition of description of human activities that can explain how a specific activity is characterized because the human activity is highly diverse. Nonetheless, the fundamental activity in home has to be walking since a resident needs to move several inside the house to do other events (Oukrich, 2019). There also are other general activities that every person especially does. To search for those activities, I summarized literature reviews for human activity in home as shown in Table 2.2.

Accordingly, it can be observed that each paper had done experiment in different activities that they are interested in but there are some duplicate common activities which have often been shown in such as sitting, walking, standing and lying.

Table 2.2
Summary of literature review for 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 • Walking • Standing • Lying • Running • Rope jump • Vacuum cleaning • Downstairs • Upstairs
(Reiss & Stricker, 2012)	Create a new dataset for physical activity and made publicly available	• 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 the fall event	• Accelerometer on smartphone	• Sitting • Walking • Lying • Running • Jumping • Hitting the sensor

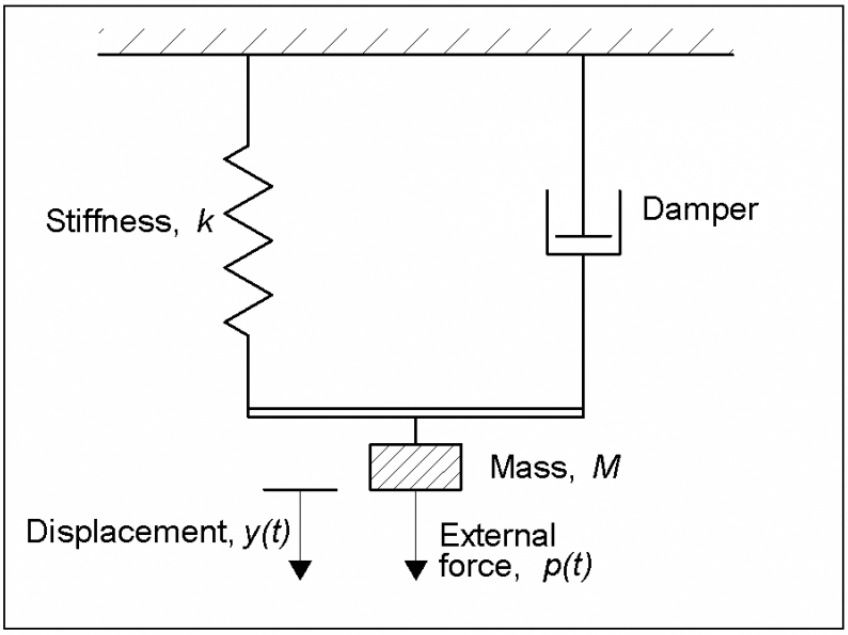
2.3 Floor Vibrations

Basically, the vibrating movement of the building by residents during the normal activities causes the floor vibration.This vibration is normally vertical (SteelConstruction, 2016). Floor vibrations are generated by dynamic loads which come through directly by people (e.g. walking, dancing, jumping) or machinery or indirectly by environment (e.g. traffic). Theoretically, vibrations are illustrated by cyclic motion with 2 significant parts as frequency and amplitude. In practice, floor vibrations are quite complex dynamic systems with unlimited vibrational mode. Ljunggren (2006) summarised the parameter that influence dynamic system of a floor as following:

- Stiffness (k): The stiffness control the springiness. The higher stiffness can lowly affect amplitude when a force is applied.
- Damping: This factor depends on their own characteristic materials. It is extremely difficult to know the exactly damping value.
- Mass (M): This higher mass can lowly affect amplitude. Thus, the low mass is desirable in order to generate vibration because it highly impact with forces. However, If the mass is too little, it may disturb residents.
- Fundamental frequency: it depends on the stiffness, and the mass. A higher frequency is normally less annoying.

Nonetheless, we can handle this complex by model as a series of simple mass and spring models with a single degree of freedom (P & Spring, 2015). We can assume the characteristic of vibration as illustrated in Figure 2.2.

Figure 2.2
The single degree of freedom systems
Reprinted from work of SteelConstruction (2016)



2.4 Time Series

A time series is a sequence of the data points on a particular variable. Generally, each data points should be taken at a constant intervals as seconds, minutes, hours, days, months, and years. The different procedure of time series is that the lag values of the target variable are used to predict the value of the target variable as predictor variable. On the other hand, the traditional other types of model do not require target variable as predictors since the historical data has no dependence to current or future data.

There are many diverse techniques in order to analyze sequential data. However, the original technique started from a special case of regression analysis (Dash, 2020) which is different from normal regression. Time series consist of four different elements as following:

- Seasonal variations: the repeats of shape or appearance occur during a specific period such as daily, weekly, monthly, season, etc.
- Trend: it contains 3 different type as up trend, down trend, and sideway which can be linear or non-linear.

- Cyclical variations: its movement depends on its cyclic such as business cycles. it has a similar behavior as seasonal variations but has a longer period.

- Random variations: this situation when the sequential data does not belong to 3 categorize above.

2.4.1 Autoregressive (AR)

Actually, autoregressive models which use for predict or forecast proposes operate under the condition that the current value depends on the past values. The equation of autoregressive is shown below:

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

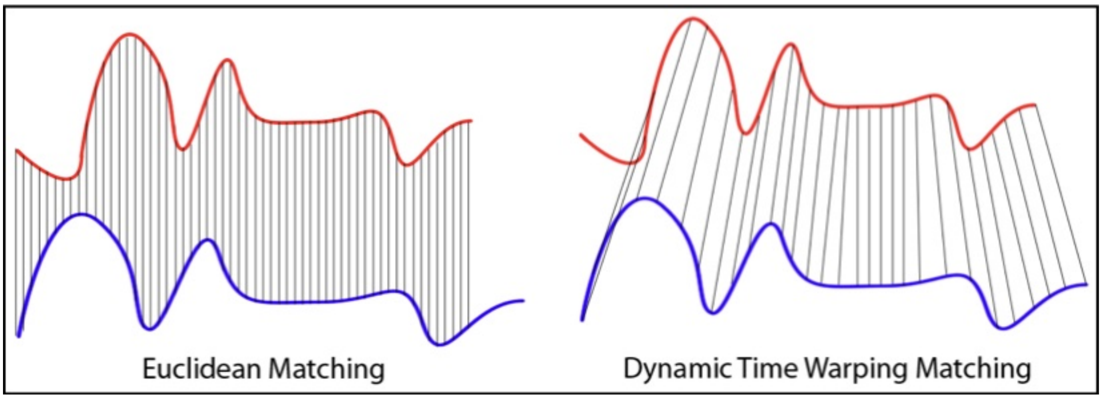
Where is white noise and is a parameter coefficient. For example, AR(1) means the current value is based on the immediately value. AR(2) means the current values is based the previous two values.

2.4.2 Dynamic Time Warping (DTW)

Since the last two decades, one of the most challenging problems in data mining is a classification in time series (Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019). There are several methods to solve this task. Specifically, the Dynamic Time Warping (DTW) can be outstanding as a strong baseline, which the current state-of-the-art is non deep classifier. The dynamic time warping which use for classification is a popular technique to provide the best distance and alignment trajectory between two sequential data which can stretch or shrink to accommodate variations in the time axis as well (Warping, 2007). The equation of DTW is shown below, where D(i,j) refer to the dynamic time warping distance between the subsequences: Figure 2.3

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

Figure 2.3
The different of Euclidean and DTW matching.
Reprinted from Dynamic time warping on Wikipedia



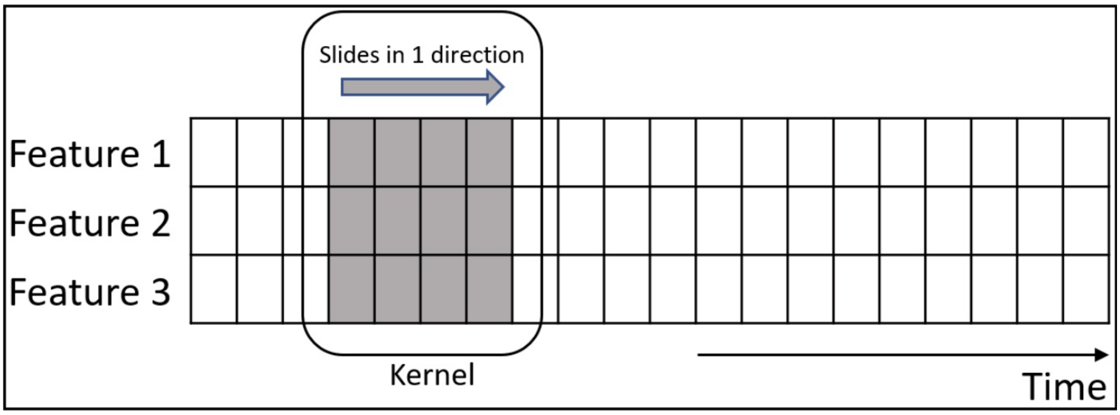
In addition, this technique can be used not only for pattern matching or classification, but also anomaly detection when the distance of two signal is higher than set threshold. However, the weaknesses of dynamic time warping are that it consume long processing time, and still lacking complete solutions and development. That why this method is not widely used commercially.

2.4.3 Convolution Neural Network (CNN)

A Convolution Neural Network is a deep learning algorithm, which input can be image, spatial or 2 dimension data and also 1 dimension data. This allows CNN to be used in more general data type including texts and other time series data. CNN can be able to successfully capture the spatial and temporal dependencies in the data through the relevant filters as called convolution. A convolution is a sliding a filter though the time series as shown in Figure 2.4 and its general equation is shown in equation below:

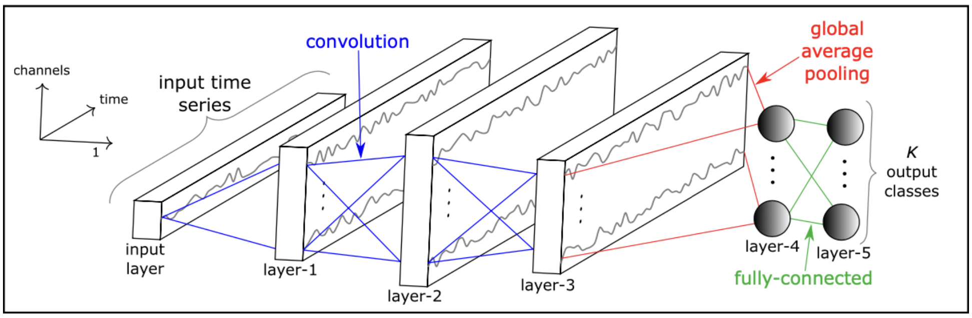
$$C_t = f(W \cdot X_{t-l/2 \to t+l/2} + b) | \forall t \in [1, T]$$

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



Where C_t is the result of a convolution at time t on a univariate 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 is a very significant useful property called weight sharing. When finishing convolution, those information must be fed though fully-connected layer as same as simple neural network architecture as shown in Figure 2.5.

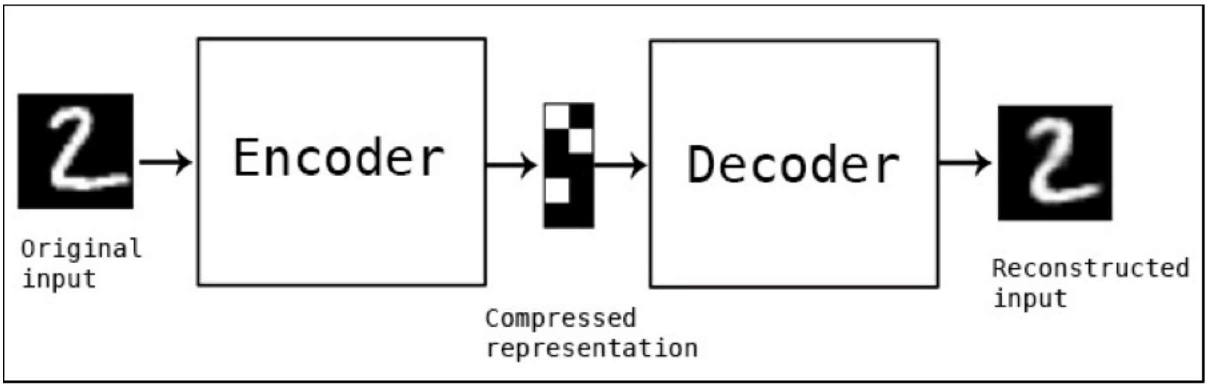
Figure 2.5
Fully Convolutional Neural Network architecture.
Reprinted from work of Ismail Fawaz et al. (2019)



2.5 Autoencoder

Basically, an autoencoder is an artificial neural network which can be able to compress data as similar as what it has been trained. The important thing is that to train a model, it does not require labeled data since an autoencoder is an unsupervised model, just feed the raw input into it. As shown in Figure 2.6, this illustrated the intuition of how an autoencoder works. In addition, it can use for denoising form the original data, this is training autoencoder to reproduce the original input from a noisy input. This allow the autoencoder to be flexible with white noise data and capture only useful pattern of the data (Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010).

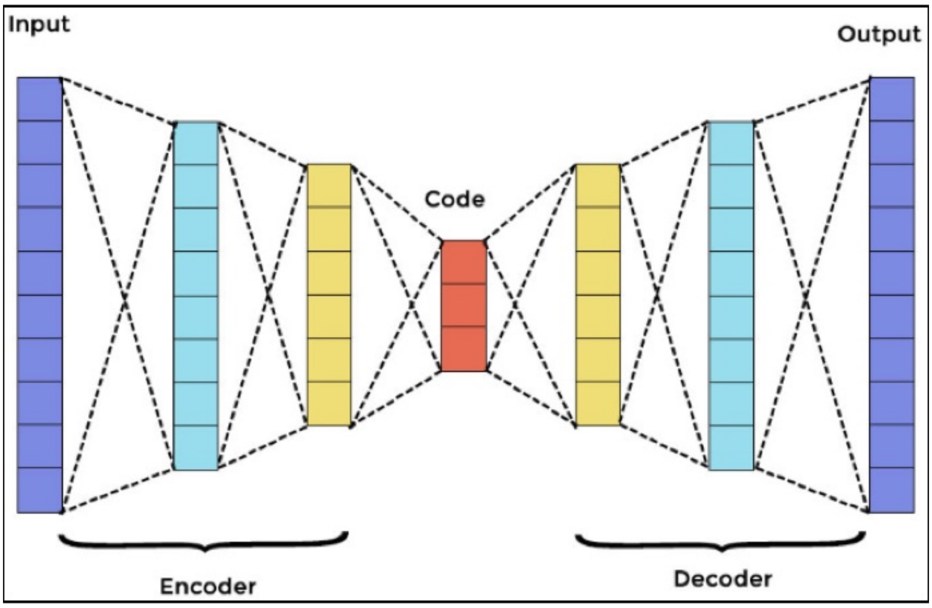
Figure 2.6
How does an autocoder work?
Reprinted from work of Chollet (2016)



An autoencoder has 3 main components (Badr, 2019) as encoder, code or bottleneck and decoder as shown in Figure 2.7.

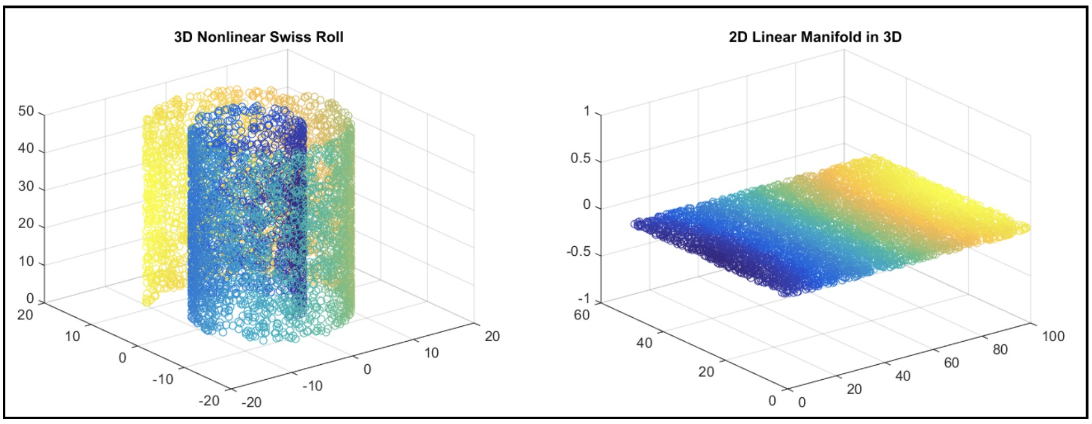
- Encoder: The model learns how to reduce the input dimensions and compress the input data into an encoded representation.
- Bottleneck: The layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data, also called latent space.
- Decoder: The model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.

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



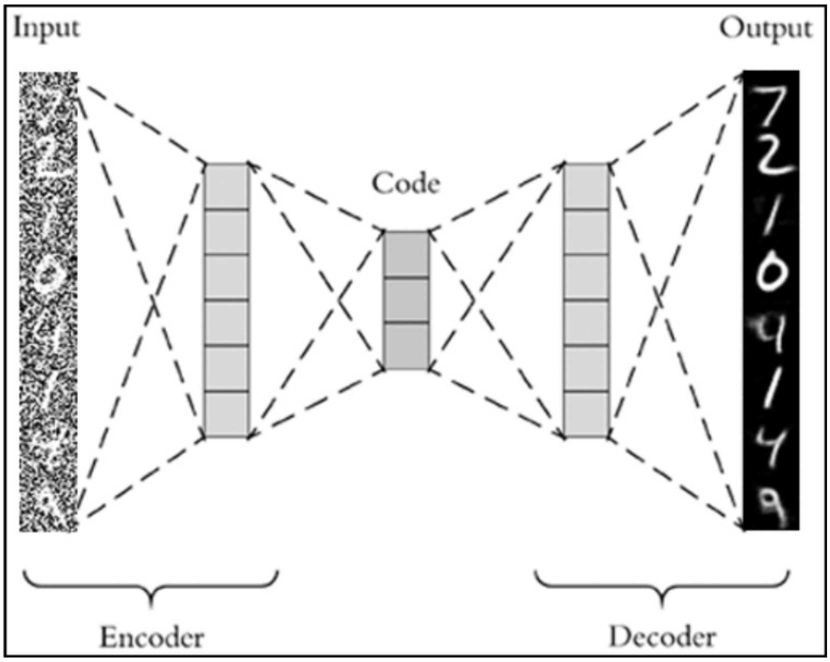
The indirect benefits of this model is that it can use for dimensional reduction (Rajan, 2021). You can notice that in the bottleneck, the number of neural node is the smallest. Thus, it means the feature of input also be forced to reduces its features. An example of reduced feature by using autoencoder as illustrated in Figure 2.8. It is reduced from 3 to 2 dimensions.

Figure 2.8
The dimensionality reduction by using autoencoder.
Reprinted from work of Roll (2015)



The other magic of an autoencoder is that when the input is wrong or has some noises, the model is going to correct it as shown in Figure 2.9. This is because optimizer is trying to create the output that matches the trained value as much as possible. Vincent, Larochelle, Bengio, and Manzagol (2008) found that the robustness of their internal layers (i.e., bottom neck representation) are developed by adding noise to the original input.

Figure 2.9
An autoencoder tries to correct the output.
Reprinted from work of Rosebrock (2020)

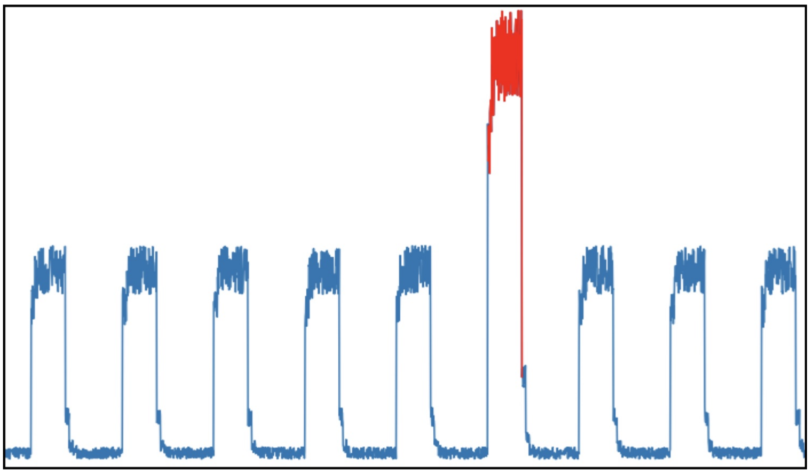


In addition, the network architecture of autoencoders which contain neural network can be adapted between a simple feedforward network, Long Short Term Memory network (LSTM) or Convolutional Neural Network (CNN) depending on the type of input. Based on my thesis, the input, vibration from human activities, is always in sequential time series. Therefore, an applicant autoencoder with LSTM network architecture may be suitable for my proposal.

2.5.1 Autoencoder for Anomaly Detection

The magic of autoencoders is simple and very natural. Let's take a look at our daily life. We do the same things everyday. But one day, we find that an unusual event has happened. We just consider this event absolutely abnormal. The procedure of an autoencoder is exactly the same. An autoencoder does not need any anomaly data at all. It required only normal data to train the model. And when abnormal situations occur, the model can detect that situation as similar to our daily life and be shown in Figure 2.10. It uses the reconstruction error to make decision which data is abnormal. Firstly, we always feed the normal data into the model to train the autoencoder. Secondly, after training is finished, the model can reconstruct very well on the normal data. And it is going to fail when faced abnormal data which model has never seen before. The simple algorithm is shown in Algorithm 1.

Figure 2.10
An autoencoder knows that an anomaly event occurs
Reprinted from work of pavithrasv (2020)



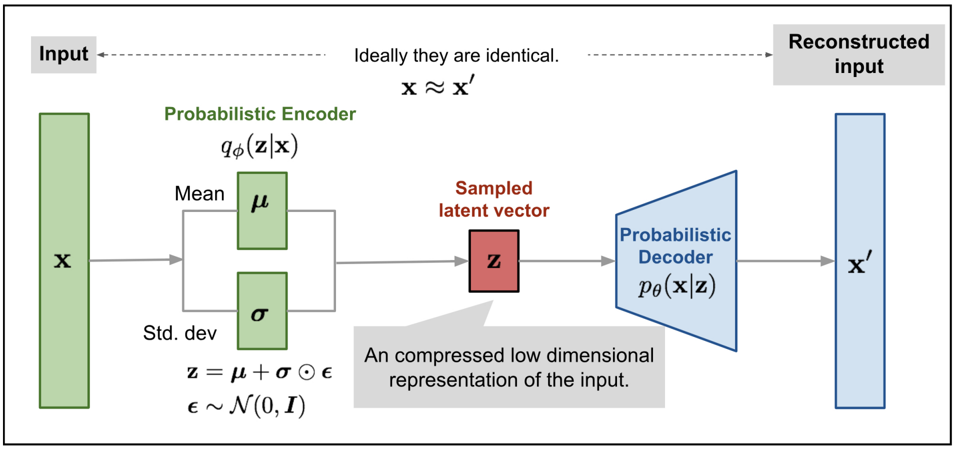
Algorithm 1 Autoencoder based anomaly detection

Input: X: Normal data
Output: $\bar{x}$: Reconstruct data
$||X - \bar{x}||$: Reconstruction error
α : Threshold
Step 1 : Train an autoencoder with the normal dataset
Step 2 : Testing the model
Step 2.1 : Set a value of α
Step 2.2 : **if** reconstruction error $< \alpha$ **then** X is a normal data
else X is an abnormal data

2.5.2 Variational Autoencoder

Basically, Variational Autoencoder (VAE) is a generative model. It estimates the Probability Density Function (PDF) from the training dataset. For example, if model is trained with traffic images, it should provide a high probability value to vehicle images or some object that related with traffic. Otherwise, it should provide a low probability value. In addition, the variational autoencoder can also sample from the learned PDF and generate new examples that look similar to the original dataset (Roger, 2021). Or in the other words, a variational autoencoder can be defined as being encoder which is trained to be regularized in order to guarantee that latent space can provide well properties for generative process (Rocca, 2020). The architecture of variational autoencoder is illustrated in Figure 2.11. Thus, the latent space of a variational autoencoder always be sampled from a distribution that encoders learns for each latent feature.

Figure 2.11
The architecture of variational autoencoder.
Reprinted from work of Weng (2018)

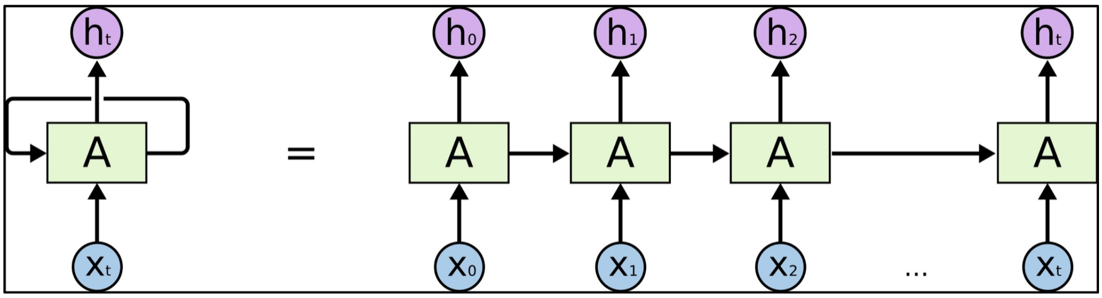


Nonetheless, the idea mentioned above does not mean that a variational autoencoder have better performance than general autoencoder in all anomaly detection task (Agmon, 2021) since the variational autoencoder mostly stands out as a generating new data.

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). To get more intuition about RNN, 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 (let's call it the hidden state or previous 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.

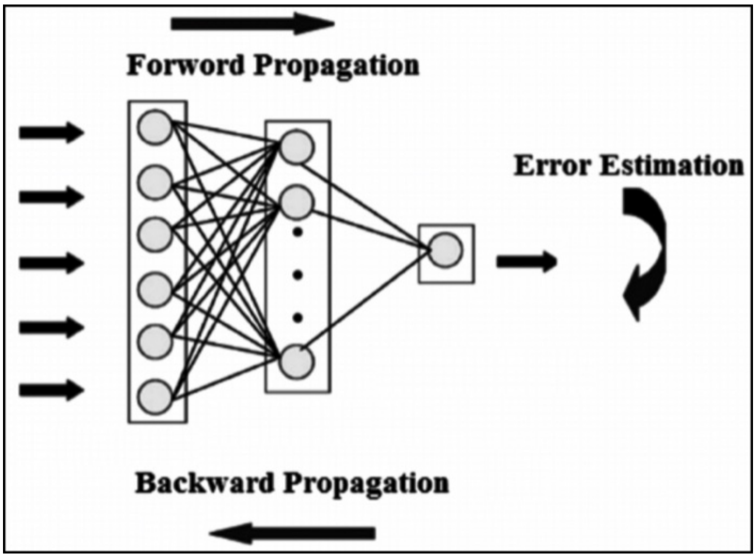
Figure 2.12
The Recurrent Neural Network architecture.
Reprinted from work of Olah (2015)



Where A is Hidden layer, X_t is input data at time t, and h_t is an output from RNN at time t. As shown in Figure 2.12, on the left shows that there is a loop that loops back into the hidden layer of the Neural Network. As mentioned above, one of the key aspects of the RNN is the previous hidden state and the input data at that time. The main benefit of this loop is to bring back the previous hidden state. (Or we may think of RNN as a Neural Network with more memory to store the previously calculated hidden state). On the right, it is flattened of work step by step.

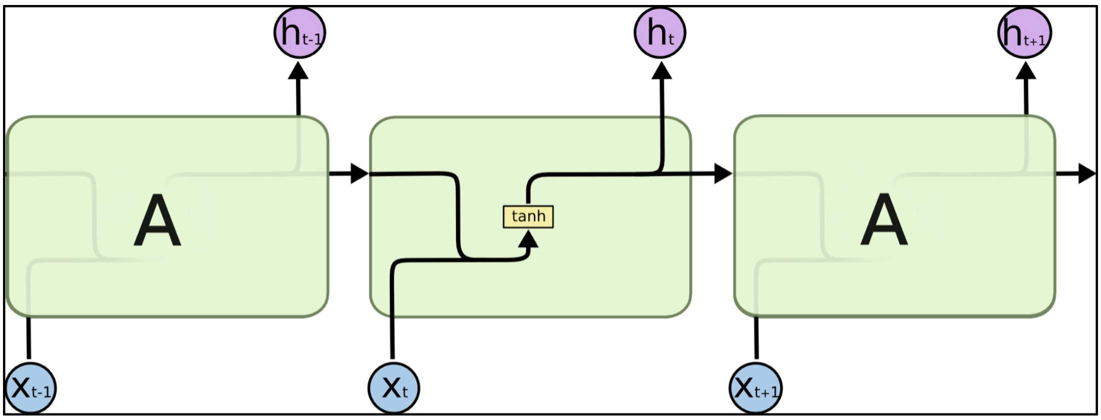
The main problem of RNN is its gradient. For those who have experienced in neural networks would clearly know that to update weights we use a backpropagation algorithm (Arnx, 2019), which calculates the gradient of the loss function (E) to update the weights which is shown in Figure 2.13, but RNN is a bit more complicated. Because getting the output h_t is not only from the interval $t = t$, but from $t = t - 1, t - 2, ...$ all the way to $t = 1$ (via Use the hidden state and input data of the previous one, the previous one, the previous one, and so on).

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



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.

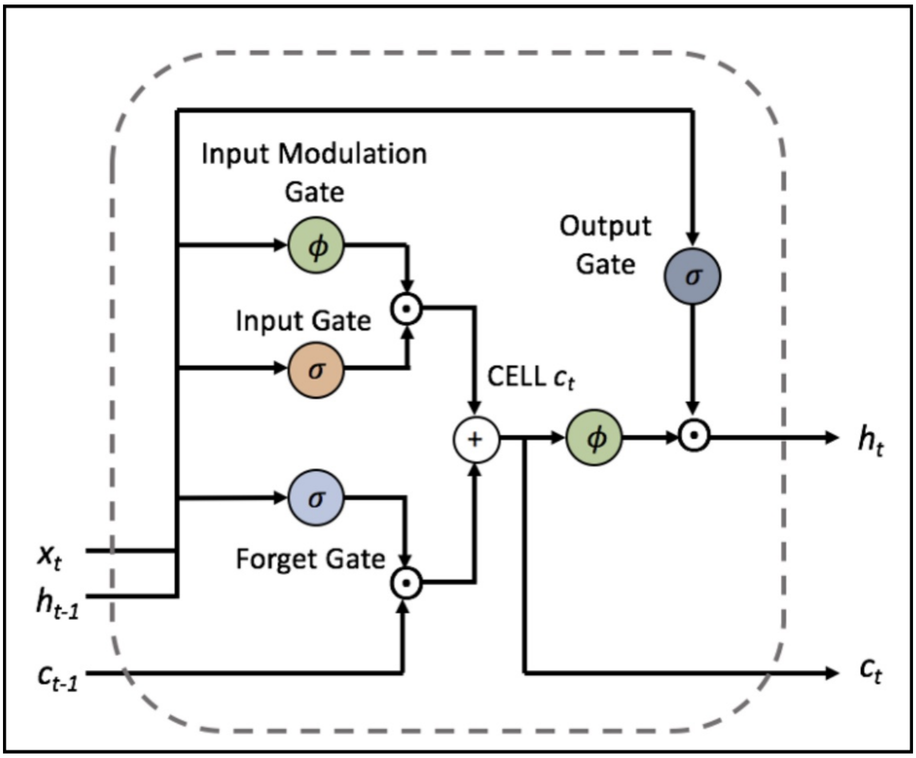
Figure 2.14
The repeating module in a standard RNN contains a single layer.
Reprinted from work of Olah (2015)



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 work of 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 to, write, forget or read. When it should allow data to flow in, flow out or pass away (forget)

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

Forget : Forget is like clearing the old cell state (forget it), 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 as is it good to update cell state with new one and if the model has to update, it will be updated with what value. 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.

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

Let’s start with the first part of the equation. You can notice that 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. Let’s come to the latter part of the equation. 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 useless.

Read : The name here might be confusing. What is it going to read next? Let’s skip to explain the output first. 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. Here we have an output gate to help the model decide as shown in the equation below.

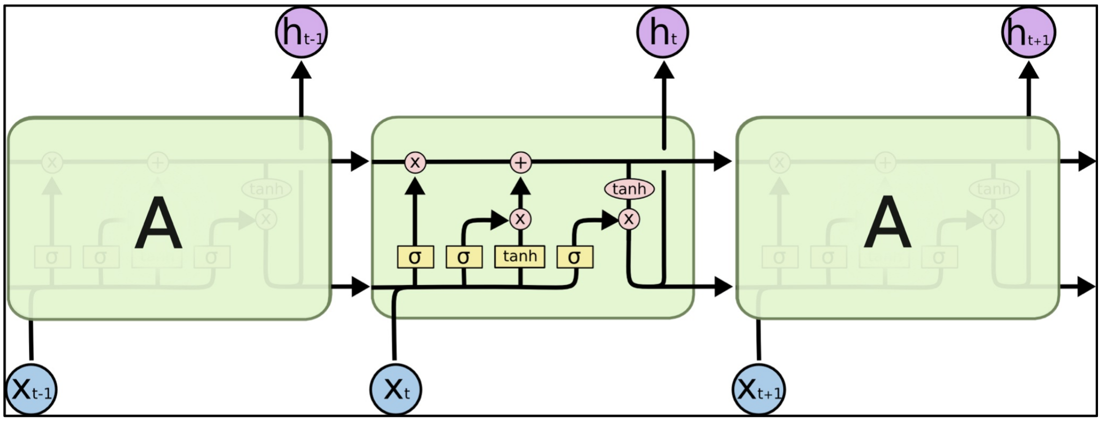
$$o_t = \sigma(W_{x^o} x_t + W_{h^o} h_{t-1} + b_o)$$

And the output will be h_t for the next sequence

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

As you can see, if the output gate provides o_t with 0 value, then ht then 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.

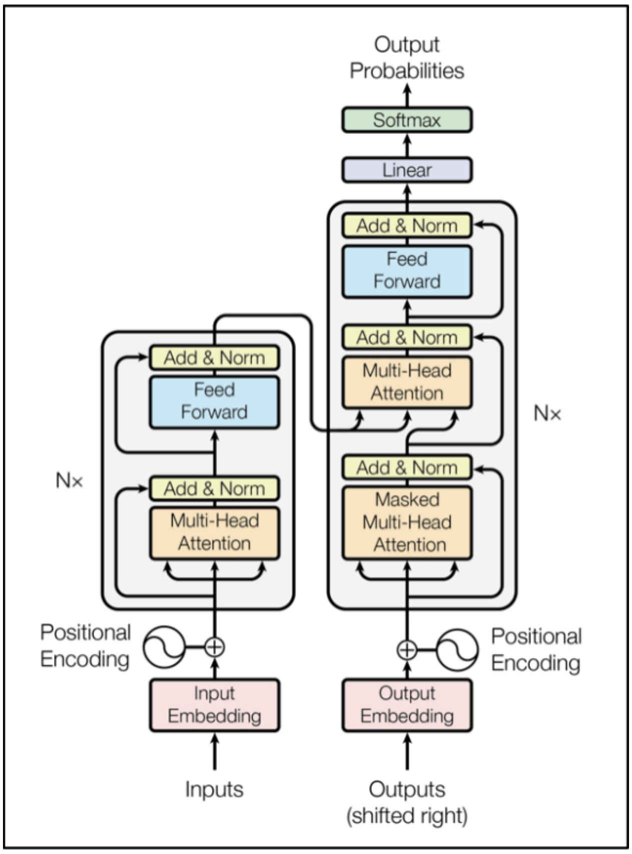
Figure 2.16
The repeating module in a LSTM contains four interacting layer.
Reprinted from work of Tangruamsub (2017)



2.8 Transformer

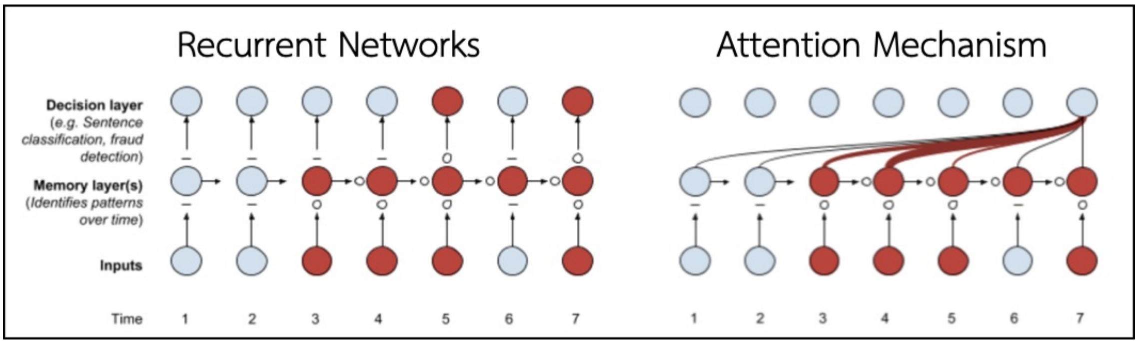
The Transformer was proposed in the paper Attention is All You Need by Google (Vaswani et al., 2017). This paper proposes a new architecture that replaces RNNs with attention called Transformer as shown in Figure 2.17. Transformer architecture has continued to beat benchmarks in many domains. Explicitly, 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 work of 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

As you can have a doubt what is the 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 work of Alammar (2018)

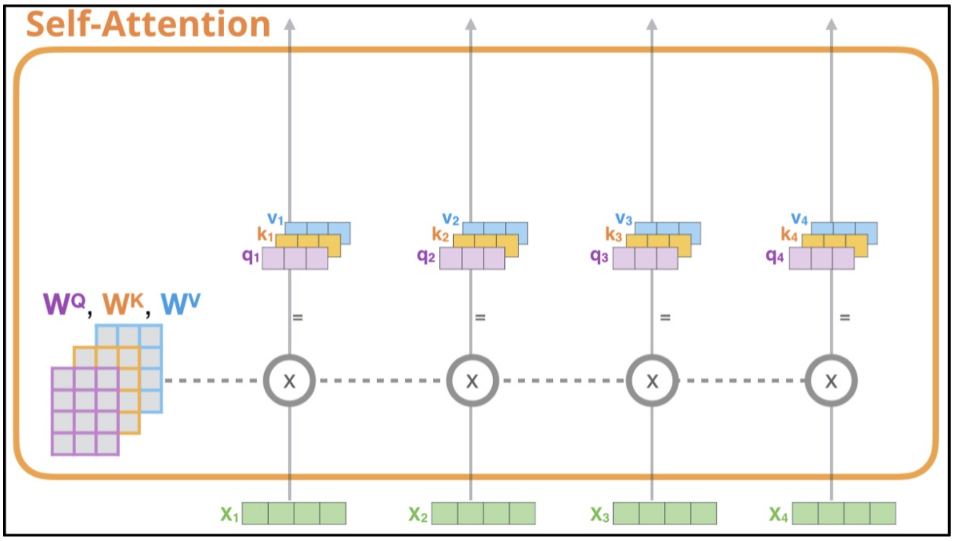


Figure 2.20
Get score of how they match.
Reprinted from work of Alammar (2018)

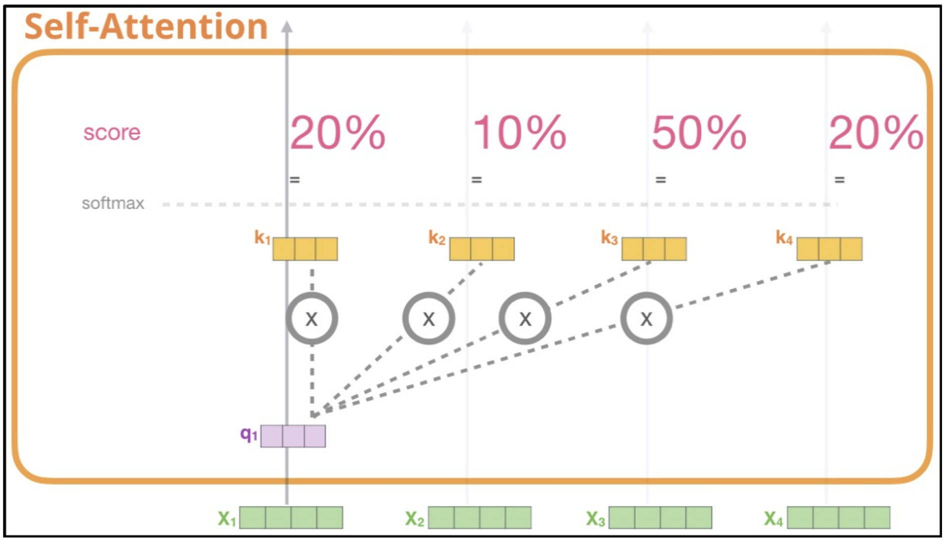


Figure 2.21
Sum up the value vectors.
Reprinted from work of Alammar (2018)

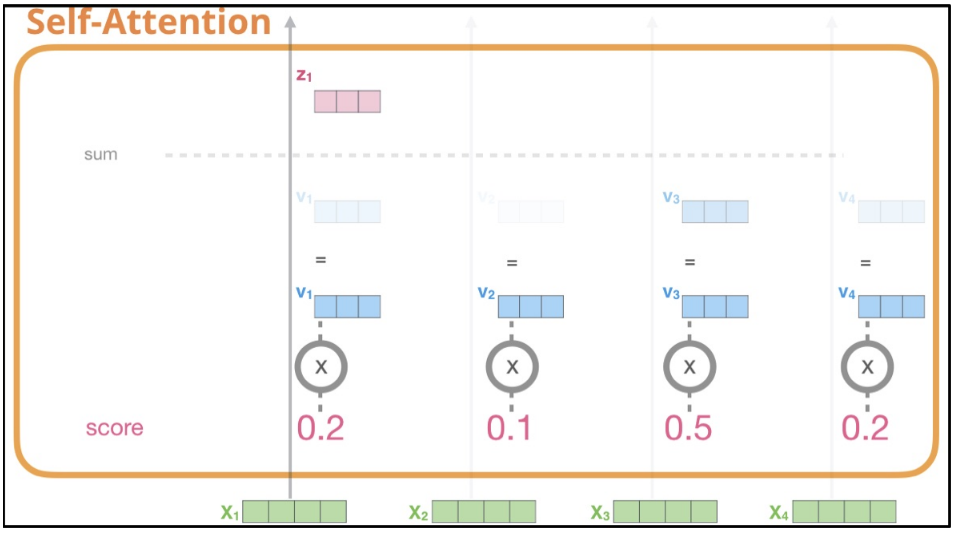
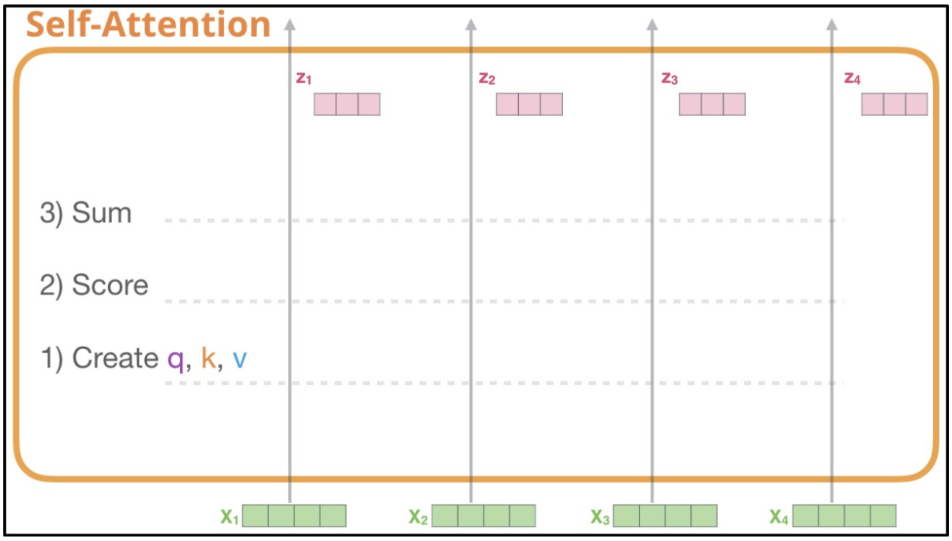
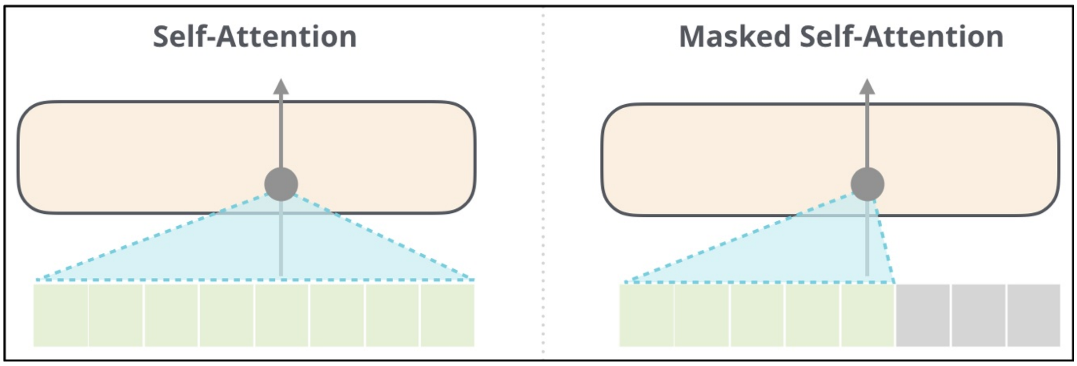


Figure 2.22
The outcome after finishing the Attention process.
Reprinted from work of Alammar (2018)



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.

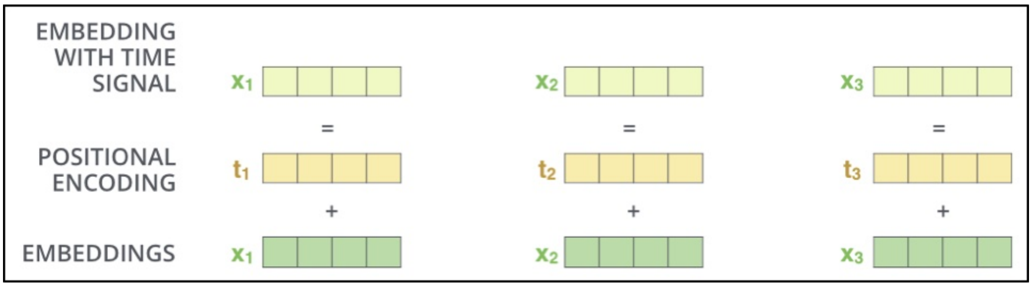
Figure 2.23
The difference of Self-Attention and Masked Self-Attention.
Reprinted from work of Alammar (2019)



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 work of Alammar (2018)



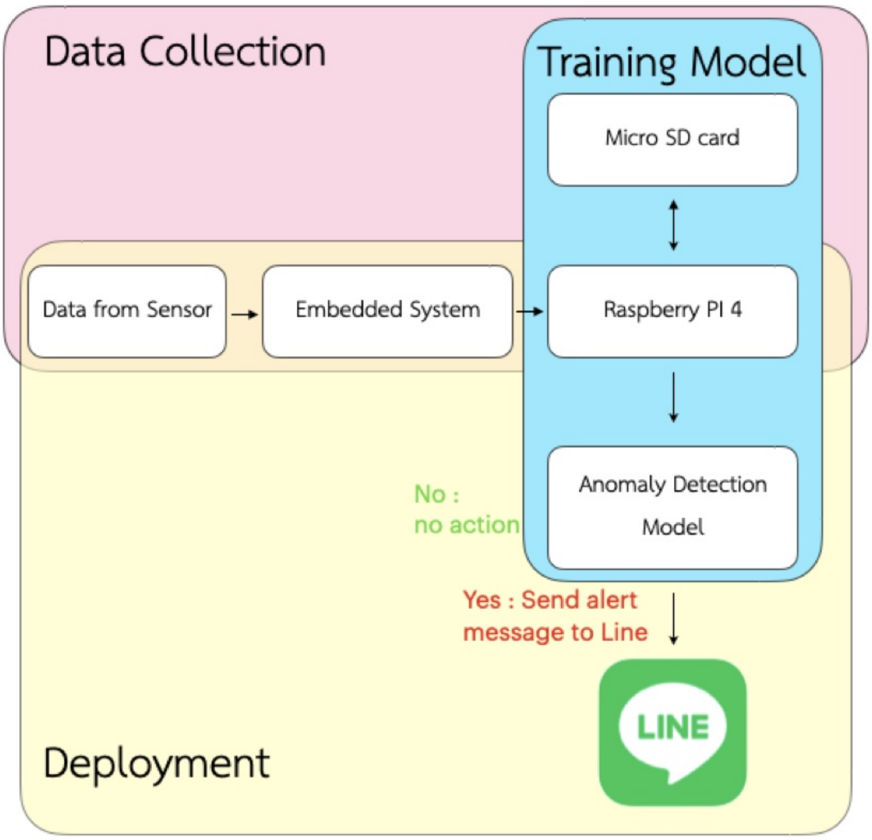
CHAPTER 3

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

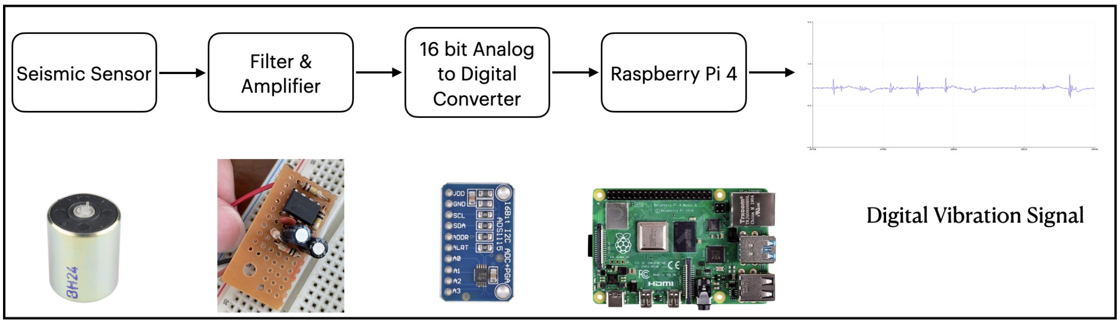
To collect raw data, we need to build our own embedded system because in order to detect human fall, the system must have the ability to detect human activities and objective drop.

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

3.1.1 Hardware

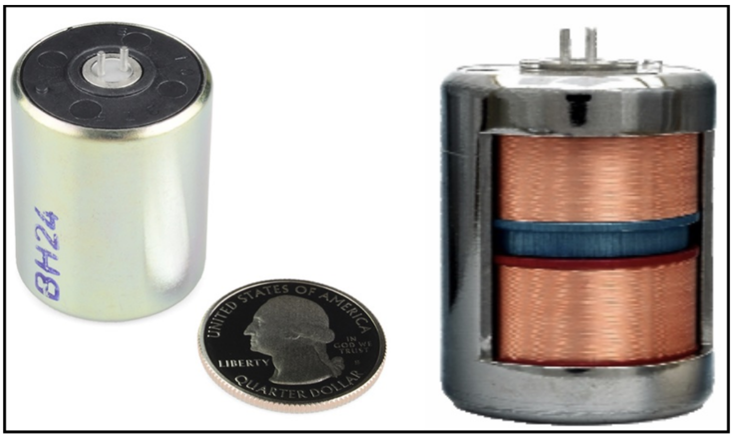
There are 4 significant components as shown in Figure 3.2. Each component has their own proposed (Dash, 2020).

Figure 3.2
The required hardware to receive raw data.



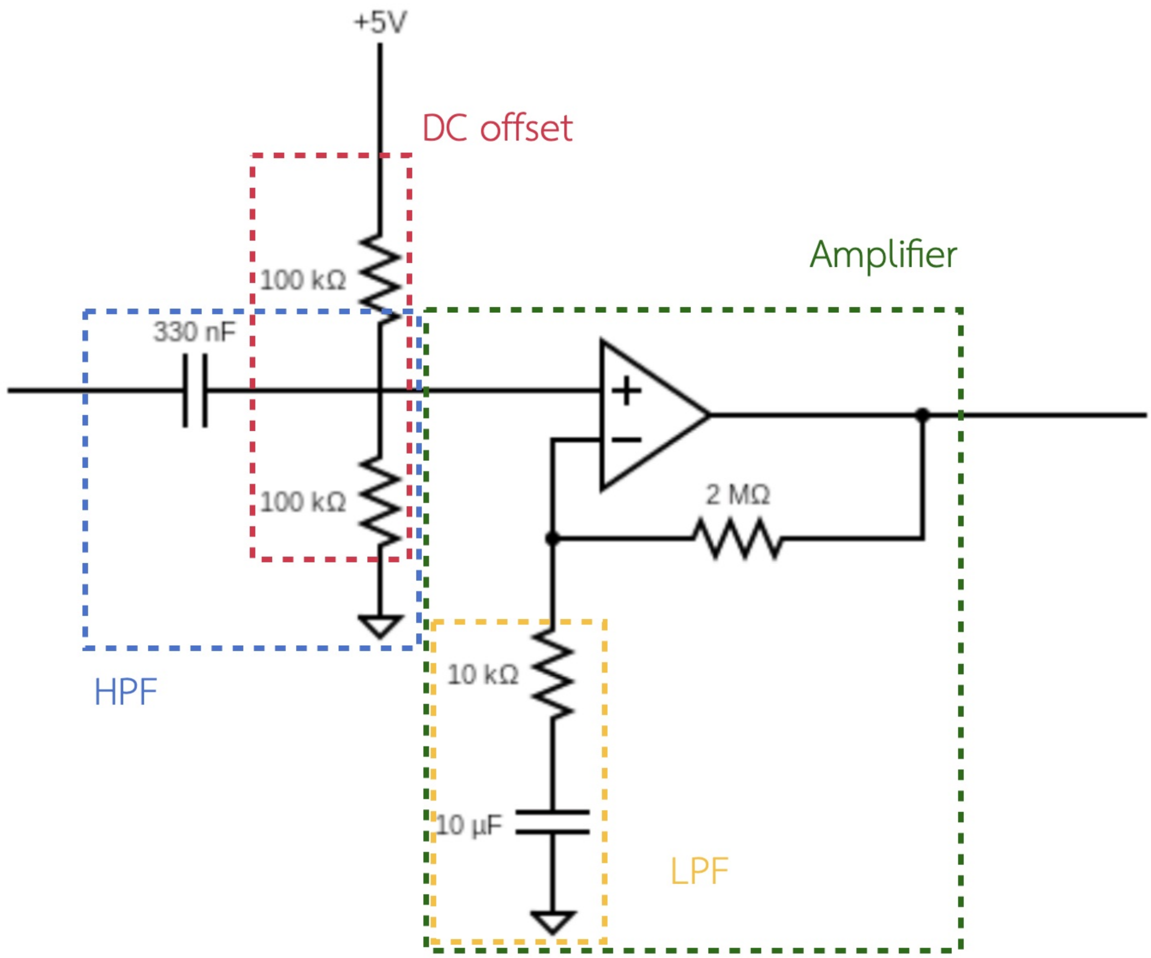
Firstly, a seismic sensor or geophone in Figure 3.3, A geophone is a device that converts ground vibration (velocity) into voltage. It has historically been passive analog devices and typically comprise a spring-mounted wire coil moving within the field of a case-mounted permanent magnet to generate 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 $100kHz$.

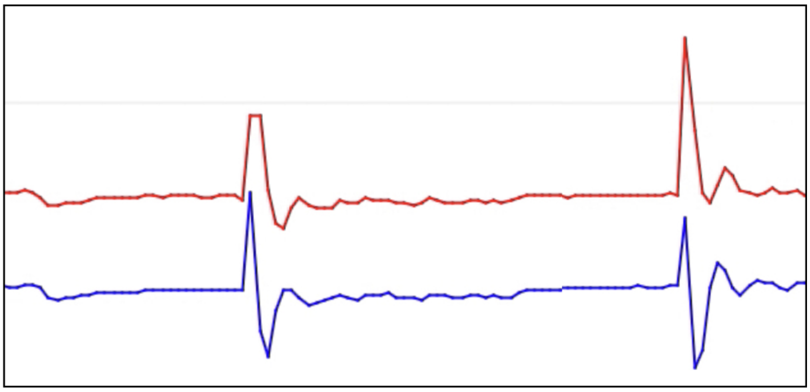
Figure 3.4
An analog circuit which was designed to be suitable with human activity.



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

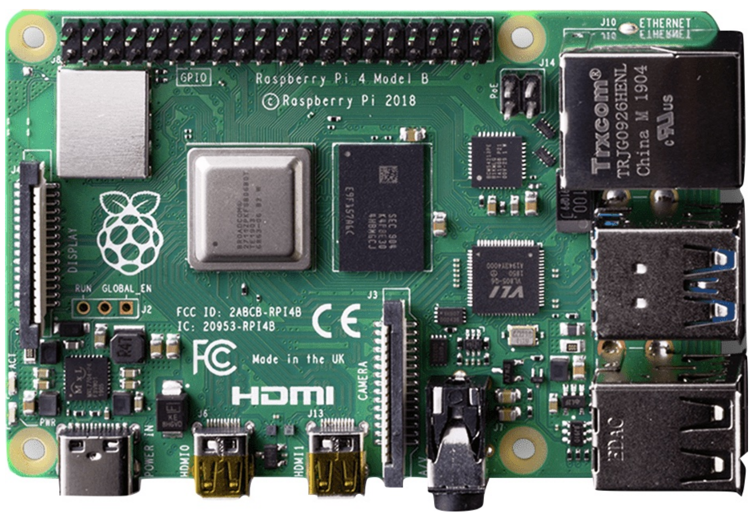
500 samples/second and quantized at a resolution of 16 bits/sample in order to prepare it for digital transmission. A 24-bit ADC is capable of distinguishing 65536 (2^{16}) different voltage levels within a narrow voltage range from $0-5$ Volts. It means that each level represents approximately $76.3uV$ which is enough to capture signals of seismic sensor.

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



Fourthly, ESP8266 in Figure 3.6, the used microcontroller is an ESP8266, which is a small size, low price, low-cost Wi-Fi microchip and has several interfaces for communication. The digital signal is fed from ADC to ESP8266 via SPI which is a synchronous serial communication interface specification used for short-distance communication. And then this signal is directly fed to the laptop via serial port (USB).

Figure 3.6
Microcontroller - 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. Reprinted from work of Battula (2019)

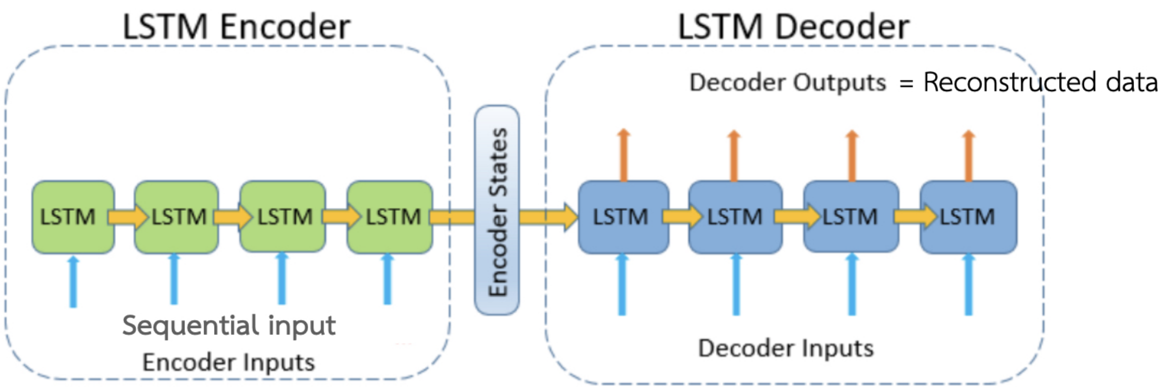
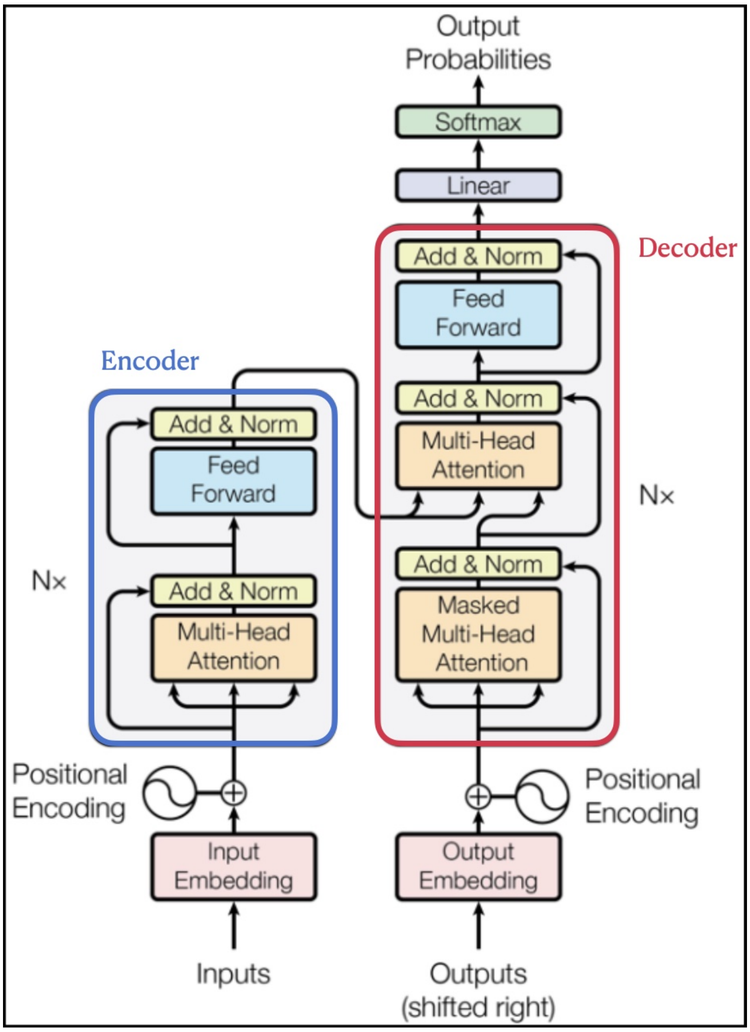


Figure 3.9
The Transformer architecture.
Reprinted from work of Vaswani et al. (2017)



3.2.1 Evaluation Plan

3.3 Deployment

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

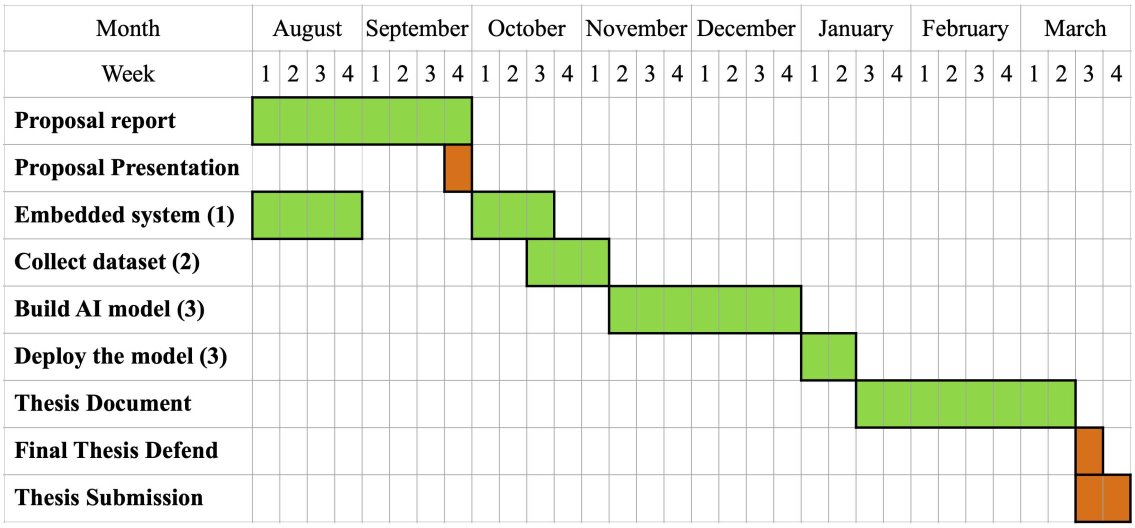
- Can send alert message to the registered user though Line application when fall occur immediately.

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, Dec). 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, Jul). *Hands-on anomaly detection with variational autoencoders.* Retrieved from `https://towardsdatascience.com/hands-on-anomaly-detection-with-variational-autoencoders-d4044672acd5`

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

Alammar, J. (2019, Aug). *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, 01). Derivation of basic human gait characteristics from floor vibrations.

Alwan, M., Rajendran, P., Kell, S., Mack, D., Dalal, S., Wolfe, M., & Felder, R. (2006, Apr). *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, Jan). *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, Jul). *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`

Battula, M. (2019, Mar). *Time series forecasting with deep stacked unidirectional and bidirectional lstms.* Retrieved from `https://towardsdatascience.com/time-series-forecasting-with-deep-stacked-unidirectional-and-bidirectional-lstms-de7c099bd918`

Britto Filho, J. C., & Lubaszewski, M. (2020, Mar). 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, Feb). 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, Oct). 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, May). *Building autoencoders in keras.* Retrieved from `https://blog.keras.io/building-autoencoders-in-keras.html`

Clemente, J., Li, F., Valero, M., & Song, W. (2020, Feb). 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, Mar). *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

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019, Mar). 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, Jul). 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, Jul). *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, Aug). 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, Aug). 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, Nov). 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).

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).

P, H., & Spring, G. (2015). *Vibrations of single degree of freedom systems.*

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

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

Pynoos, J., Steinman, B. A., & Nguyen, A. Q. (2010, Nov). 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, Jun). *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, Jun). 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, Oct). *Wearable fall detection system.* Retrieved from https://ieeexplore.ieee.org/abstract/document/7745414/ doi: 10.1109/MECBME.2016.7745414

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

Roger. (2021, Jul). *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, 07). Collecting complex activity datasets in highly rich networked sensor environments. In (p. 233 - 240). doi: 10.1109/INSS.2010.5573462

Roll, T. S. (2015, Jan). *The swiss roll matching example.* Retrieved from https://www.cis.jhu.edu/~cshen/html/PublishSwissRoll.html

Rosebrock, A. (2020, Feb). *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, Feb). 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, Dec). 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, Mar). Classification of indoor human

fall events using deep learning. *Entropy*, *23*(3), 328. doi: 10.3390/e23030328
Tangruamsub, S. (2017, Jul). *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, Feb). 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
Tsai, T.-H., & Hsu, C.-W. (2019, Oct). 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, Aug). 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
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, Jun). 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, Jun). Elderly fall detection systems: a literature survey. *Frontiers in Robotics and AI*, *7*. doi: 10.3389/frobt.2020.00071
Warping, D. T. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, 69–84. doi: 10.1007/978-3-540-74048-3_4
Weng, L. (2018, Aug). *From autoencoder to beta-vae.* Retrieved from https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html
WHO. (2018, Jan). *Falls.* World Health Organization: WHO. Retrieved from https://www.who.int/news-room/fact-sheets/detail/falls
Yang, C.-C., & Hsu, Y.-L. (2010, Aug). 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..

APPENDIX A
SURVEY QUESTIONNAIRE

Different materials are presented in the APPENDICES. Label the materials in the order that they are mentioned in the text or section (e.g., “see Appendix A for the questions”). Large or oversized tables or figures that support, but are not important in the text, are included in the appendices in a portrait or landscape orientation.

APPENDIX B

TITLE

Multiple texts, tables and/or figures can be combined in one appendix. They should be referred to in a section or parts of your thesis. Number and title the materials in the order they are mentioned in the sections. Add a short description of Appendix A. Do the same for other appendices with multiple materials.

Figure B2

CCTV monitoring room. Reprinted from the Twenty First Security Web site (`http://www.twentyfirstsecurity.com.au/`).

Table B1

Random Table B

v1	v2	v3	Overall
78.67%	87.33%	94.92%	85.64%