THROUGH THE WALL
HUMAN MOVEMENT DETECTION
WITH SOFTWARE DEFINED RADIO

Master Dissertation

Hüseyin Irmak CİVAN

Eskişehir 2019

THROUGH THE WALL HUMAN MOVEMENT DETECTION WITH SOFTWARE DEFINED RADIO

Hüseyin Irmak CİVAN

MASTER DISSERTATION

Electrical and Electronics Engineering
Supervisor: Associate Prof. Tansu FİLİK

Eskişehir
Eskişehir Technical University
Institute of Graduate Programs
November 2019

FINAL APPROVAL FOR THESIS

This thesis titled “Through The Wall Human Movement Detection with Software Defined Radio” has been prepared and submitted by Hüseyin Irmak CİVAN in partial fulfillment of the requirements in “Eskişehir Technical University Directive on Graduate Education and Examination” for the Degree of Master in Electrical and Electronics Engineering Department has been examined and approved on 18/11/2019.

Committee Members	**Title Name Surname**	**Signature**
Member (Supervisor)	Assoc. Prof. Dr. Tansu FİLİK	....
Member	Prof. Dr. Osman PARLAKTUNA	....
Member	Asst. Prof. Dr. Onur KILINÇ	....

Prof. Dr. Murat TANIŞLI
Director of Institute of Graduate Programs

ÖZET

YAZILIM TABANLI RADYO İLE DUVAR ARKASI HAREKET ALGILAMA

Hüseyin Irmak CİVAN

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Kasım 2019
Danışman: Doç. Dr. Tansu FİLİK

Bu tez kapsamında, duvar arkasındaki insan hareketini (varlığını) farklı senaryolarda, ortamdaki kablosuz haberleşme amaçlı elektromanyetik dalgaları kullanan derin öğrenme modelleri ile tespiti ele alınmaktadır. Kablosuz haberleşme sinyallerini izlemek, toplamak için yazılım tabanlı radyo kullanılmıştır. Temel olarak ortamın boş olması, insanın hareketsiz olduğu durum (sadece solunum hareketi) ve insanın hareketli olduğu (yürüme hareketi) üç durum tanımlanmıştır. Canlıların hareketsiz olduğu (uyku, baygın olma vb.) durumlarında yaptığı mikro düzey hareketlerden biri solunum fonksiyonudur.

Bu tez kapsamında bu mikro hareketleri duvar arkasından algılayabilecek elektromanyetik dalgaların kullanıldığı bir düzenekten toplanan veriler kullanılmıştır. Alınan sinyalin genliği, insan solunumuna bağlı olarak periyodik olarak değiştiği görülmüştür. Kullanılan düzenekte sınırlı çıkış gücüne sahip 900 MHz taşıyıcı frekanslı sinyallerin farklı mesafelerde 10 farklı durum için yazılım tabanlı radyolarla (USRP B210), tanımlanan üç temel durum için kayıtları alınmıştır. 10 farklı durumun birbirinden ayırt edilebilmesi için toplanan sinyallerde farklı özellikler tanımlanmış ve 22 farklı sınıflandırma tekniği ile eğitilip test edilerek ve en uygun model sunulmuştur.

Anahtar Sözcükler: Duvar Arkası Hareket Tespiti, Yazılım Tabanlı Radyo (SDR), Makine Öğrenmesi, Sınıflandırma.

ABSTRACT

THROUGH THE WALL HUMAN MOVEMENT DETECTION WITH SOFTWARE DEFINED RADIO

Hüseyin Irmak CİVAN

Department of Electrical and Electronics Engineering,

Eskişehir Technical University, Institute of Graduate Programs, November 2019

Supervisor: Associate Prof. Tansu FİLİK

In this thesis, machine learning models for the detection of human movement behind the wall in different scenarios using electromagnetic waves for wireless communication in the environment are investigated. Software defined radio is used for monitoring and collecting communications signals. There are three main situations in the monitoring area which is empty (no living), and the person is inactive asleep or unconscious (only breathing), and the person is moving (walking etc.). Respiratory function is one of the micro-level movements of living things when they are inactive (sleep, unconscious, etc.).

In this thesis, the data collected using a setup, which can sense these micro motions behind the wall, using the electromagnetic waves. It is observed that the amplitude of the received radio signal are changing periodically even with human respiration movement. In the used setup, 900 MHz carrier frequency signals with limited output power were recorded for three defined basic situations with software-based radios (USRP B210) for 10 different states at different distances. In order to distinguish 10 different states, different characteristics were defined in the collected signals and trained and tested with 22 different classification techniques and the most accurate and efficient model is presented.

Keywords: Through the Wall, Human Movement Detection, Software Defined Radio (SDR), Machine Learning, Classification.

ACKNOWLEDGEMENT

I would like to thank my advisor, Associate Prof. Dr. Tansu Filik, for his continuous support and guidance during my master studies. I am also grateful to Assistant Prof Dr. Onur KILINÇ and Research Assistant Can UYSAL for their support and guidance throughout my studies. Finally, I would like to thank my family for their endless support.

Hüseyin Irmak CİVAN
November 2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Eskişehir Technical University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Hüseyin Irmak Civan

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

3D	: Three Dimensional
AD	: Arctangent Demodulation
AM	: Amplitude Modulation
ANNs	: Artificial Neural Networks
API	: Application Programming Interface
CW	: Continuous Waveform
DFT	: Discrete Fourier Transform
DSP	: Digital Signal Processing
FA	: Frequency Accumulation
FFT	: Fast Fourier Transform
FM	: Frequency Modulation
FPGA	: Field Programmable Gate Array
GPIO	: General-Purpose Input/Output
IR	: Infrared
kNN	: K-Nearest Neighbor
ML	: Machine Learning
PIR	: Passive Infrared Sensor
RF	: Radio Frequency
RFIC	: Radio Frequency Integrated Circuit
ROC	: Receiver Operating Characteristic
SDR	: Software Defined Radio
SFCW	: Stepped Frequency Continuous Wave
SSB	: Single-Sideband
STFT	: Short Time Fourier Transform
SVM	: Support Vector Machine
SWD	: Singular Value Decomposition
UHD	: USRP Hardware Driver
USB	: Universal Serial Bus
USRP	: Universal Software Radio Peripheral
UWB	: Ultra Wide Band

X_t(t)	: The magnitude of the received signal and the motions
A_T	: Signal magnitude
f_c	: Carrier frequency
N	: Number of collected samples
Y_R(t)	: The Magnitude of unmodulated received signal
μ_R	: Average value
$\lvert M(t)\rvert$	: The signal caused by macro and micro movement between the receiver and transmitter
$w(t)$	: White Gaussian random process
x	: Input data

1. INTRODUCTION

Human presence or movement detection through the wall are widely studied in various civilian applications such as search and rescue after natural disasters [1]. In literature the (low-energy) electromagnetic radio signals are preferred because of their ability to penetrate most non-metallic building materials such as brick, wood, dry walls, concrete / reinforced concrete. These techniques can be separated as radar based scanning techniques [2] and ambient radio communication techniques [3].

In this study, constant amplitude continuous wave radio frequency (RF) signals are used to detect any movement behind the wall. The three main cases are defined in the monitoring area, which is no living (empty), and the person is inactive asleep or unconscious (micro-movement), and the person is moving, walking etc. (macro-movement. In the event of a major disaster (earthquake), living detection is a more important requirement than visualizing the inside of the ruins (walls) by expensive radar systems and techniques. In this study, we investigate the potentials of the constant amplitude modulated radio waves to classify the defined three basic cases with 10 different challenging situations. The most difficult case is the detection (classification) of the sleeping or unconscious human through the wall. This case is defined as the micro-movement of the respiratory. It is shown that, the enlargement and contraction of the chest during breathing may change the amplitude and phase of the received signal [4-7]. We used the software defined radio (SDR) to receive and transmit the radio wave signals. These radios can commercially accessible, cheap and has open source software development tools as SDR# (Windows-free) [8], High Definition Software Defined Radio (Windows-free) [9], SDR-RADIO.COM V2/V3 (Windows-free) [10], Linrad (Windows/Linux/Mac-free) [11], GQRX (Linux/Mac-free) [12], Studio1 (Windows-paid) [13],SDRUno (Windows-free) [14], SDR Touch (Android-trial/paid) [15], Wavesink Plus (Android-trial/paid) [16], RFAnalyzer (Android-free/paid) [17], cuSDR (Windows-free) [18], Sdrangelove (Linux-free) [19], Kukuruku (Browser based-free) [20], Natpos (Linux-free) [21], QuestaSDR (Windows-free) [22], QIRX SDR (Windows-free) [23], Zeus Radio (Windows/Linux-paid) [24], RF Network on Chip - Ettus Research (free), USRP Hardware Drive- Ettus Research [25], GNU Radio- Ettus Research [25], LabVIEW- Ettus Research [25], MATLAB/SIMULINK - Ettus Research [25] etc.

In this thesis, according to the variations of the amplitude and phase of the received signals for different cases are collected for ten different scenarios. Machine learning models are used to classify these cases. Thus, the detection of motion through the wall and the fact that the motion is macro or micro are successfully classified by a certain percentage of accuracy (91,6 %-Weighted KNN, 91,8 %-Bagged Ensemble and 92,9 %-Fine Gaussian SVM) by the machine learning classification method. A total of 4200 different samples are collected as dataset and these samples are separately used for training and testing in different scenarios. In this study, best classification for the 4 features identified is Fine Gaussian Support Vector Machine (92,9 %) classification technique according to data records.

2. LITERATURE REVIEW

Motion detection is the process of detecting a change in a position relative to the circumference of an object or a change in the environment relative to an object [26]. Motion detection can be achieved by mechanical or electronic methods. Infrared (passive and active sensors), visual (camera and optic systems) and radio frequency (electromagnetic) waves are used as detection technologies. In addition to these, different sensing techniques such as pressure-sensing, optoelectric method, static-charge sensing triboelectric method, capacitive change analysis are used to detect people's movements and assets for different usage areas.

On the other hand, detecting the presence of human through the wall is an important problem in literature. For this purpose, electromagnetic radio waves are preferred because of their ability to penetrate most non-metallic building materials such as brick, wood, dry walls, concrete / reinforced concrete. These techniques can be separated as radar based scanning techniques [2] and ambient radio communication techniques [3]. Radar-based systems are called as active systems, because they actively transmits time limited (wide band) signals and collects the echo signals for detection. It is possible to use acoustic signals in these systems but the acoustic waves cannot directly penetrate through walls and most of signal will be reflected back. On the other hand radio frequency (RF) waves can penetrate non-metallic media (such as brick wall, ruins) to detect life signal in far off areas but does not require any contact with the living body through electrodes or sensors [27-32]. Especially radar systems are widely used to monitor through the wall. RF reflection systems aim to capture and identify human movements by processing the reflected RF signals without placing any sensors in the human body and at the same time without direct eye contact. Basically, in radar systems the direction of the echo intensity and the time to return is very important. According to these parameters, the properties of the object are determined. Radars were originally designed to detect large objects at long distances. The focus was primarily on air traffic control and surveillance, the detection of watercraft. At that time, little effort was made to explore the possibilities of short-range measurements. The ability of electromagnetic waves to penetrate non-metallic objects, then ground penetrating systems for geological investigations, was developed. Most of the radars currently in use are frequency band narrow band systems where the frequency band is much lower than the carrier frequency. The theory and practice of existing radar systems are based on this feature. However, as is known, a frequency band determines the information content of the radar systems since the volume of information transmitted per one unit of time is directly proportional to a frequency band. The reality of this problem has determined the rapid development of technologies using ultra-wide band (UWB) signals in recent years. Further widening the frequency band helps to obtain and acquire more information about the target. The overall conceptual operating mode of a UWB radar system is similar to the ultrasonic echo transducers used in many applications, from auto focus cameras to proximity and range detectors. The main and fundamental difference is that, unlike ultrasonic electromagnetic pulses, it diffuses from walls, floors, ice, mud, concrete and the human body. Increasing the information capacity of the system requires expanding the frequency band [33]. The information content of the UWB radars increases due to the smaller pulse volume of the signal. UWB Radars can be used practically as safety signal sensors to detect unauthorized access to protected areas in situations where we need precise remote monitoring of moving objects over short distances and in security systems. UWB signals also play an important role in through the wall imaging systems. In work called as “Sense through wall human detection using UWB radar” has taken various measurements for different wall and door types using monostatic radar for the detection of human target using UWB radar [34]. They found that heart rate detection using the Doppler approach worked in wood but failed in the case of a thick concrete wall. A second method using singular value decomposition was used to reduce clutter and this works for brick and gypsum wall but again fails for concrete wall case. Finally, they tried an STFT and SVD method based on the idea that the received signal in case of presence of target will result in difference in frequency response compared to no target case. By applying SVD to the STFT output it is observed, in case of gypsum wall, that the second singular value changes relatively in presence of target. Since the detection process is based on changes in the received signal amplitude, it does not work for other situations and is completely unreliable. As a result of their studies, they stated that a new approach is needed for all situations. In addition, only the monostatic mode of operation was considered when collecting data. Therefore, a bi-static mode of operation can be considered and also argued that some kind of filtration would be useful to reduce wall clutter. In work called as “Sense-through-Wall Human Detection Based on UWB Radar Sensors” that Discrete Fourier Transform (DFT)-based approach couldn’t work well in scenarios where signal attenuation is high, and DFT-based approach has high computational load, which makes it difficult to be used in real-world.

He proposed a standard deviation-based approach to sense through wall and sense through wooden door human detection, and make analysis on detection threshold selection [35]. Our approach is very simple to be implemented, but it has high accuracy. It can achieve perfect detection (no detection error) with appropriate detection threshold. In the [36] study with the name of “An Improved Algorithm for Through-Wall Target Detection Using Ultra-Wideband Impulse Radar” work on the detection and localization of a human subject in complex environments using an ultra-wideband impulse radar. The range of the subject was determined using a Short-Time Fourier Transform (STFT), and the frequency of human motion was extracted using Arctangent Demodulation (AD) and a multiple frequency accumulation (FA) method. The performance of this method was examined using data sets obtained under different conditions and compared with several well-known techniques. Results were presented which indicate that the proposed approach can effectively suppress static and non-static clutter, linear trend, harmonics, and the product of respiration and heartbeat signals. In the [37] with the name of “Experimental Study and Analysis of Stepped Frequency Continuous Wave Based Radar for Through the Wall Detection of Life Signs”, Stepped-frequency continuous wave (SFCW) radar system is used to detect the breathing signal of human being entrapped through a plywood wall. For validation, a breathing physical model is developed and tested successfully. For breathing detection, information is extracted using Fast Fourier Transform (FFT) and Hilbert Huang transform methods. Experimental results show successful detection of breathing signal of human being as well as due to physical model using both methods.

All these radar-based systems requires synchronization between transmitter and receiver and they commonly use wide band specific radio waveforms. Recently, the ambient radio waves are used for noncontact breathing monitoring and motion classification applications. These radio waves such as WiFi, FM radio, GSM/LTE signals etc.) are exist and some of the studies are investigated the potentials of these signals for monitoring indoor environments. It is shown in [4-7] that, the enlargement and contraction of the chest during breathing may change the amplitude and phase of the received signal.

3. MEASUREMENT MODELS AND MATERIALS

3.1. Measurement Model

The measurement model in (1) describes the relation between the magnitude of the received signal and the motions in the monitoring area. A continuous narrow band signal is transmitted with carrier frequency f_c.

$$X_t(t) = A_T \mathrm{co\,s}(2\pi f_c t + \theta) \qquad\qquad t = 1, \dots, N \tag{3.1}$$

The magnitude of the unmodulated received signal on the receiver is,

$$|Y_R(t)| = \mu_R + |M(t)| + w(t) \qquad\qquad t = 1, \dots, N \tag{3.2}$$

Where, $\boldsymbol{\mu_R}$ is the average (constant) value of the received signal, $|\boldsymbol{M}(\boldsymbol{t})|$ is the signal caused by macro and micro movement between the receiver and transmitter, although there is no movement in the environment, the signal fluctuates due to ambient noise, which is considered a white Gaussian random process and is represented by $\boldsymbol{w}(\boldsymbol{t})$. N is the number of collected samples.

If there is no movement (empty case), the $|\boldsymbol{M}(\boldsymbol{t})|$ value is 0. The magnitude of the signal received on the receiving side when there is no movement between the signal generator and the receiver;

$$|Y_R(t)| \; = \; \mu_R + \; w(t) \qquad\qquad t = 1, \dots, N \tag{3.3}$$

If there is any macro or micro movement, $|\boldsymbol{M}(\boldsymbol{t})|$ signal can be used to classify these motions.

3.2. Materials

In this study, the receiver and transmitter are positioned in different locations for different scenarios.

3.2.1. Radio Wave Transmitter

AGILENT / HP 8647A signal generator and VERT900 antenna are used as transmitter.

Features of AGILENT/HP 8647A signal generator and VERT 900 Antenna are shown in Table 3-1.

Table 3-1 *Properties of signal generator and antenna*

AGILENT / HP 8647A signal generator	**Specification**
	Frequency Range Max. : 1.00GHz
	Freq. Resolution : 1.0Hz
	Output Power : +10 dBm to -136 dBm
	Modulation : AM, FM, Phase
VERT900 Antenna	**Specification**
	SM Band omni-directional vertical antenna, at 3dBi Gain
	824 to 960 MHz, 1710 to 1990 MHz Quad-band Cellular/PC S

3.2.2. Software Defined Radio (SDR) Receiver

Ettus Research USRP B210, VERT900 antenna, USB 3.0 Connectivity Cable and Computer are used as receiver shown in Table 3-2. Matlab R2018a and SDR tools are used for signal processing and neural network implementation.

Table 3-2 *Equipment on receiver side*

Hardware Used On Receiver Side	Software Used On Receiver Side
USRP B210	MATLAB R2018a
VERT900 Antenna	
USB 3.0 Connectivity	
Computer	

MATLAB R2018a is used to program Universal Software Radio Peripheral (USRP) B210. The hardware support package (Communications Toolbox Support Package for USRP Radio by Math Works Communications Toolbox Team Staff) is installed via MATLAB for programming purposes. In this way, the signals sent from the transmitter can be received by USRP B210 with appropriate software adjustments and appropriate algorithms.

Software Defined Radio is based on the principle that the components of devices providing radio communication are used in the software layer instead of the hardware layer. Rather than using resistors, capacitors, feedback circuits to process the radio signal, it is a radio technology that performs a significant part of the radio functions using software and is defined as a radio communication system that uses software for both modulation and demodulation of signals [38].

The main purpose of Software Defined Radio (SDR) technology is; to produce new radio signal forms which are needed today and used for various purposes, using only software changes without any hardware changes and high costs.

In this study, USRP B210 SDR Kit - Dual Channel Transceiver (70 MHz - 6GHz) - Ettus Research has been used to receive the desired signals on the receiving side and convert analog signals to digital signals.

The USRP B200 and B210 hardware covers RF frequencies from 70MHz to 6 GHz, has a Spartan6 Field Programmable Gate Array (FPGA), and Universal Serial Bus (USB) 3.0 connectivity. This platform enables experimentation with a wide range of signals including FM and TV broadcast, cellular, Wi-Fi, and more. The USRP B210 extends the capabilities by offering a total of two receive and two transmit channels, incorporates a larger FPGA, general-purpose input/output (GPIO), and includes an external power supply. It uses an Analog Devices Radio Frequency Integrated Circuit (RFIC) to deliver a cost-effective RF experimentation platform, and can stream up to 56 MHz of instantaneous bandwidth over a high bandwidth USB 3.0 bus on select USB 3.0 chipsets (with backward compatibly to USB 2.0). B210 is enabled with our USRP Hardware Driver™ (UHD), users can develop their applications and seamlessly port their designs to high-performance or embedded USRPs such as the USRP X310 or USRP E310. UHD is an open-source, cross-platform driver that can run on Windows, Linux, and Mac operating system. It provides a common application programming interface (API), which is used by several software frameworks, such as GNU Radio. With this software support, users can collaborate with a vibrant community of enthusiasts, students, and professionals that have adopted USRP products for their development. As a member of this community, users can find assistance for application development, share knowledge to further SDR technology, and contribute their own innovations.

3.3. Experimental Setup

In order to collect data, 4 different situations have been designed. These situations are case 1, case 2, case 3 and case 4. For case 1, there are no obstacles between the receiver and the transmitter. For case 2, there is a brick wall between the receiver and the transmitter. Movements are made on the side of the wall receiver side. For case 3, there are 2 brick walls between the receiver and transmitter. Movements are made between two walls. For case 4, the receiver and transmitter are in the same room next to each other. Movements are made in the other room. There is one brick wall between the transceiver and the area where the movements are made.

A continuous wave carrier signal at 900 MHz with constant amplitude is selected as radio waveform to avoid the need for synchronization on the receiver and transmitter. This frequency can penetrate the walls well and the waveform's resolution is also suitable to distinguishing micro movements. The signal output power is set to maximum of 1 Decibel-milliwatt (dBm). As a transmitter the HP 8647A signal generator is used on the transmitter side. The VERT 900 standard omnidirectional vertical dipole antenna with 3 dBi gain is used for this signal generator. The setup parameters has been previously established in the [4 and 5]. On the transmitter side, the antenna is 1 meter high from the ground.

On the receiving side, the VERT 900 standard omnidirectional vertical dipole antenna with 3 dBi gain is used to detect the signals in the environment. Therefore, in this study, 900 MHz is preferred as center frequency for all samples taken from the environment. The signal which is received by USRP B210 seen on the spectrum analyzer as Figure 3.1.

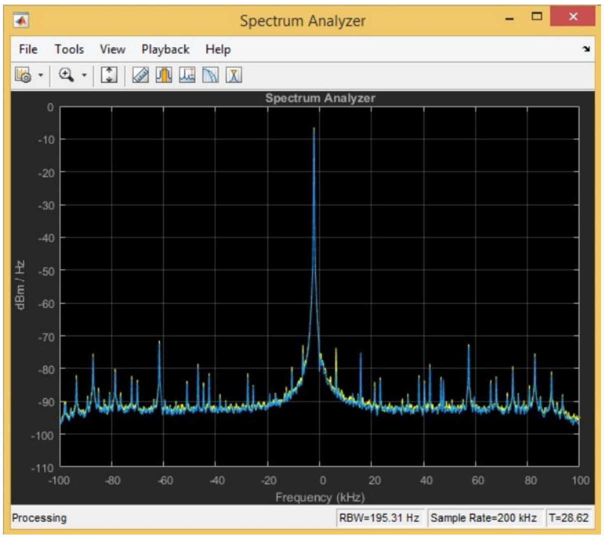


Figure 3.1. *The Signal on spectrum analyzer*

The USRP B210 software defined radio developed by Ettus Research to which the antenna is connected is configured as a receiver. The USRP B210 connects to a computer with a USB 3.0 cable. MATLAB R2018a is used to program the software defined radio on the computer. To be able to program USRP B210 in MATLAB, Communications Toolbox Support Package for USRP Radio by Math Works Communications Toolbox Team Staff must be loaded. On the receiver side, the height of the antenna from the ground is 1 meter.

3.3.1. CASE 1

In this experimental case, there is no wall between receiver and transmitter.

3.3.1.1 . *Case 1: Human absence*

In this experimental case, there is no human between receiver and transmitter. The distance between the receiver and the transmitter is 3 meters as shown in Figure 3.2.

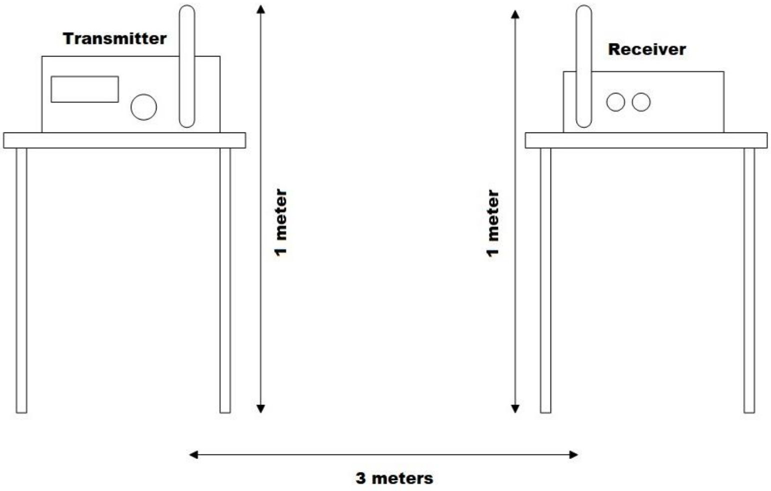


Figure 3.2. *Distance between Receiver and Transmitter*

The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The three dimensional (3D) representation of the scenario is as in Figure 3.3

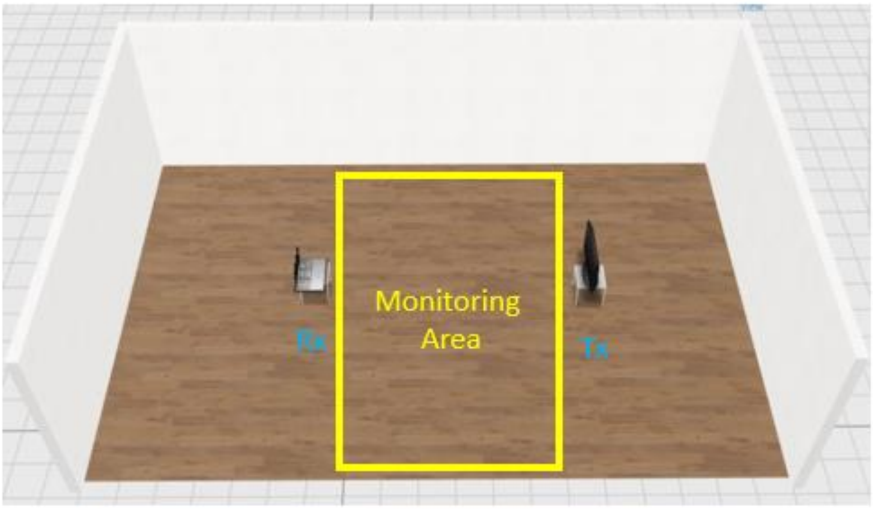


Figure 3.3. *Case 1, Human Absence Representation*

In this case the amplitude of the unmodulated received signal for the empty is shown in Figure 3.4 for 10 seconds. As shown, the amplitude is constant due to no motion. For this case we collected 240 different records, each with 1000 samples. Figure 3.5 shows one of the 1000 samples record.

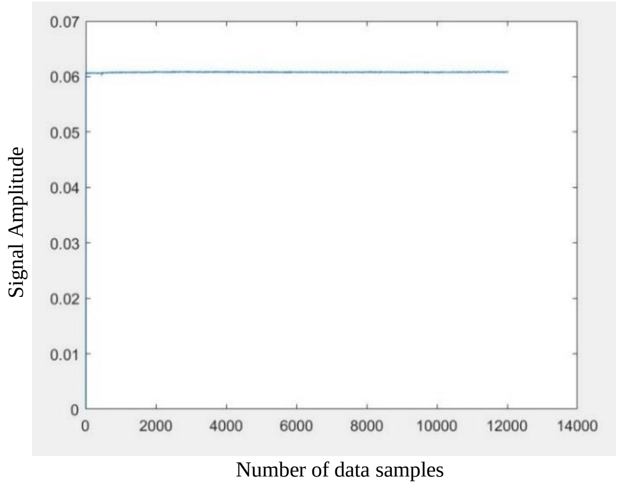


Figure 3.4. *Received Signal in Case 1, Human Absence*

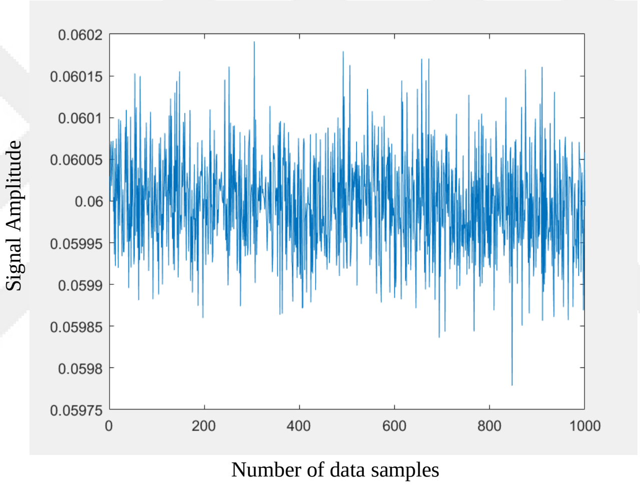


Figure 3.5. *Received Signal in Case 1, Received Data View*

3.3.1.2 . *Case 1: Breathing between receiver and transmitter*

In this experimental case, a subject (human) sits on the chair between the receiver and the transmitter. The man's face is towards the receiver. In this case the man only breathes, no other macro motions are allowed.

The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.6.

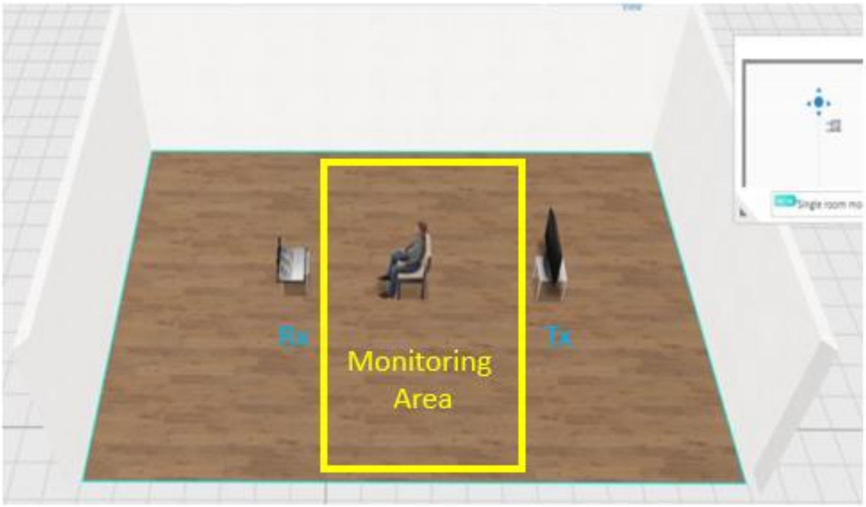


Figure 3.6. *Case 1, Breathing Representation*

Figure 3.7 shows a 10 second records when the breathing is allowed.

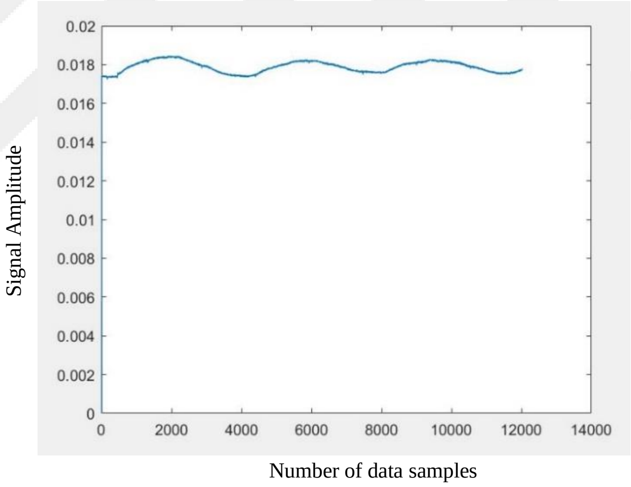


Figure 3.7. *Received Signal in Case 1, Breathing*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.8.

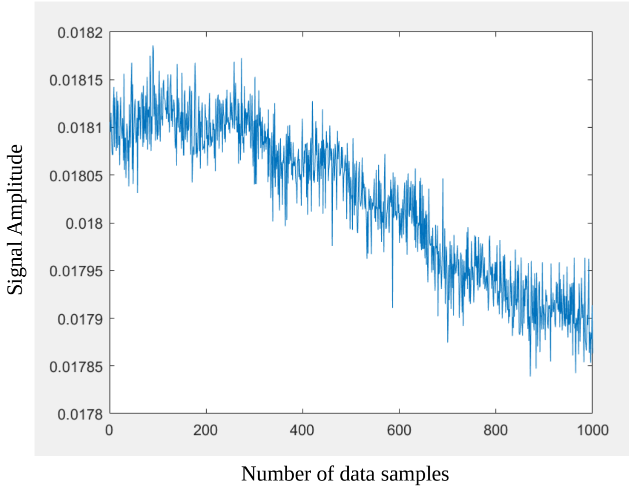


Figure 3.8. *Received Signal in Case 1, Breathing Received Data View*

3.3.1.3 . *Case 1: Walking between receiver and transmitter*

In this experimental case, a man walks between the receiver and the transmitter. The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The three dimensional (3D) representation of the scenario is as in Figure 3.9.

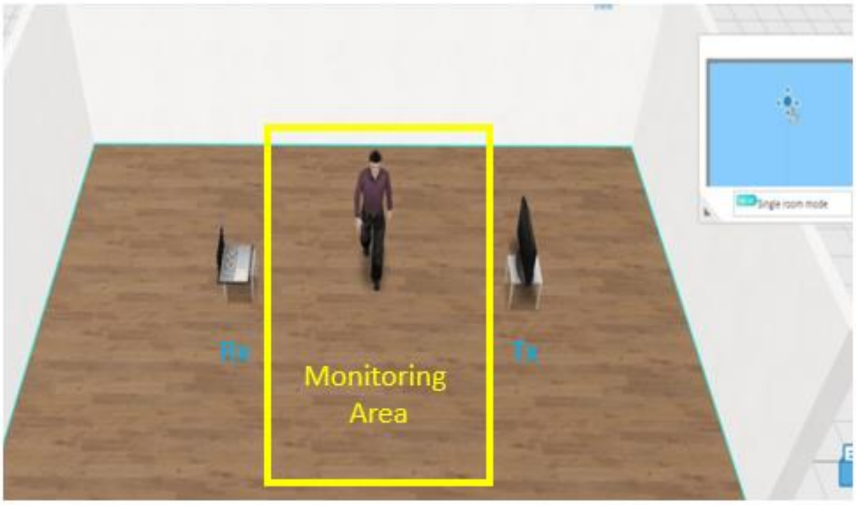


Figure 3.9. *Case 1, Walking Representation*

An example of sample from walking is as Figure 3.10. This example shows the change in signal amplitude for 10 seconds.

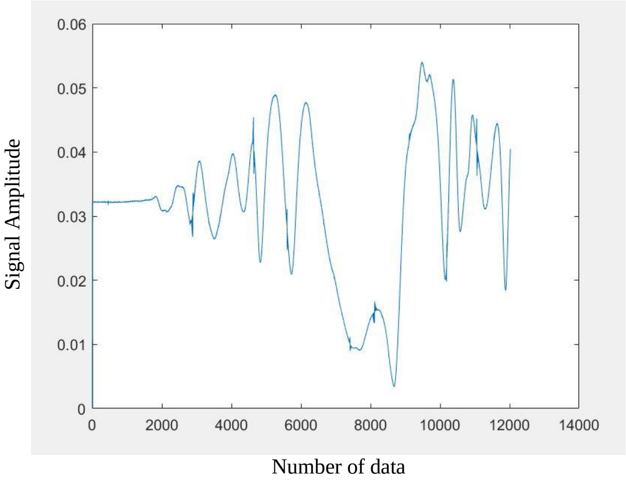


Figure 3.10. *Received Signal in Case 1, Walking*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.11.

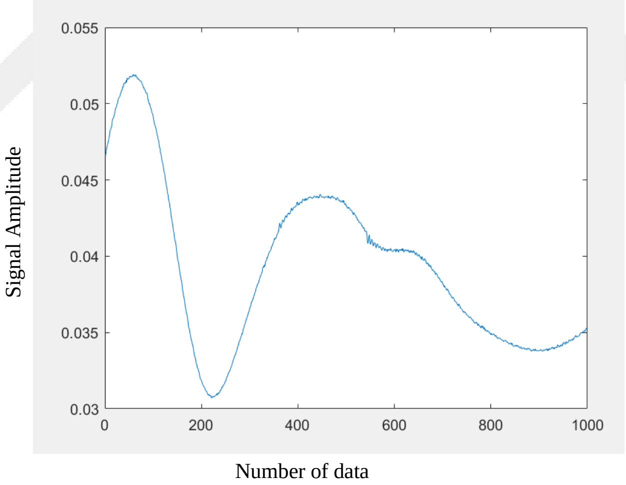


Figure 3.11. *Received Signal in Case 3, Walking Received Data View*

3.3.2. CASE 2

In this experimental case, there is a wall between receiver and transmitter.

3.3.2.1 . *CASE 2: Human absence*

In this experimental case, there is no human between the receiver and the transmitter. The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.12.

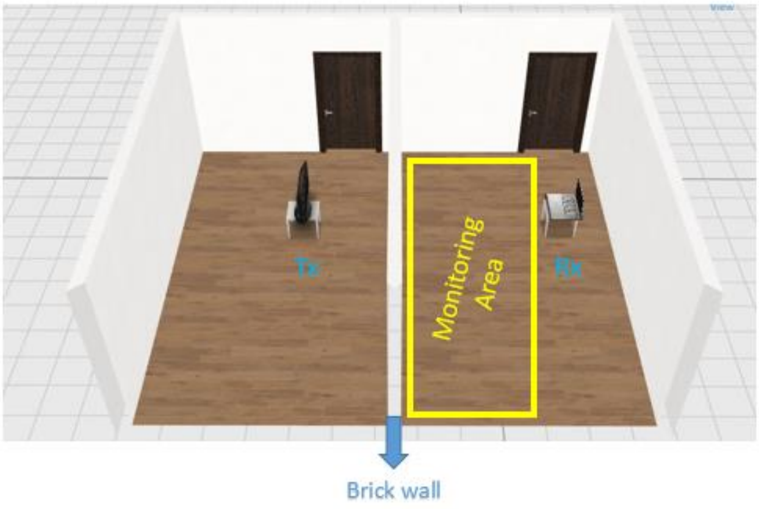


Figure 3.12. *Case 2, Human Absence Representation*

An example of sample through the wall from blank media is as Figure 3.13. This example shows the change in signal amplitude for 10 seconds.

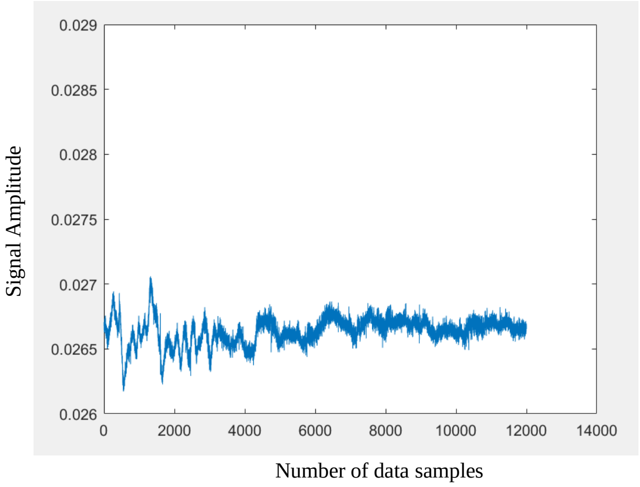


Figure 3.13. *Received Signal in Case 2, Human Absence*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.14.

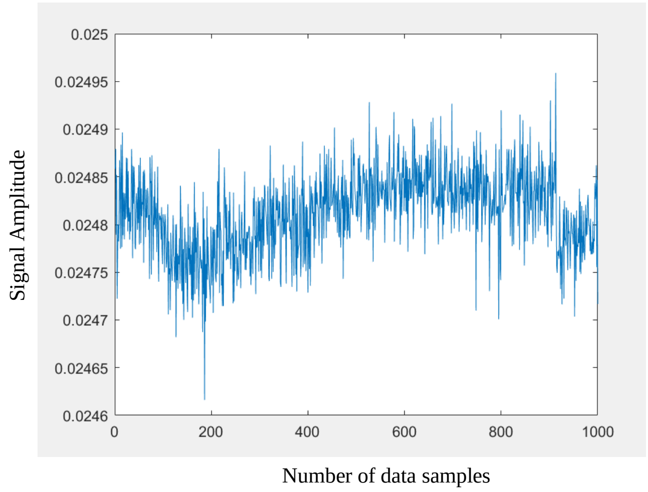


Figure 3.14. *Received Signal in Case 2, Blank Media Received Data View*

3.3.2.2 . *CASE 2: Breathing between receiver and transmitter*

In this experimental case, a man sits on the chair between the receiver and the transmitter. The man's face is towards the receiver. In this case the man only breathes, other than that, it does not move.

The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.15.

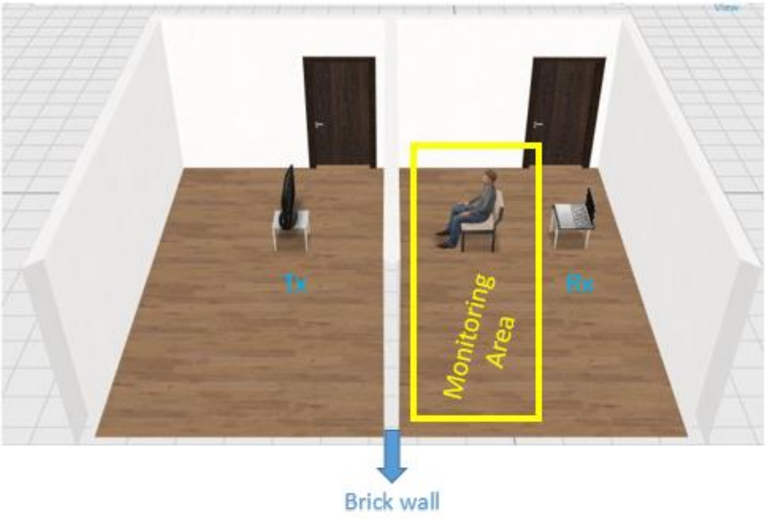


Figure 3.15. *Case 2, Breathing Representation*

An example of sample through the wall from breathing is as Figure 3.16. This example shows the change in signal amplitude for 10 seconds.

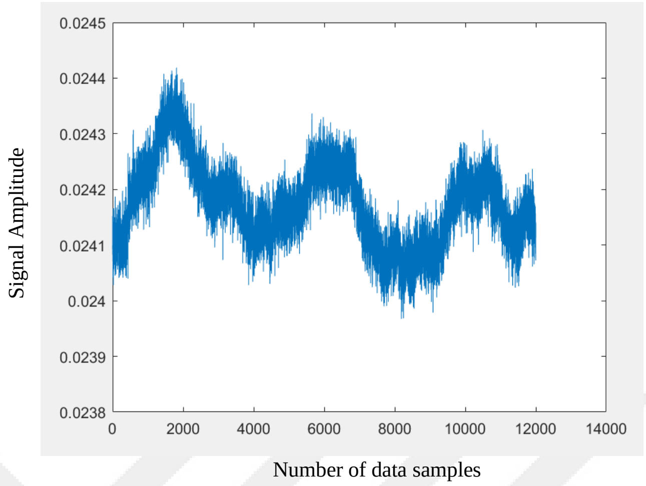


Figure 3.16. *Received Signal in Case 2, Breathing*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.17.

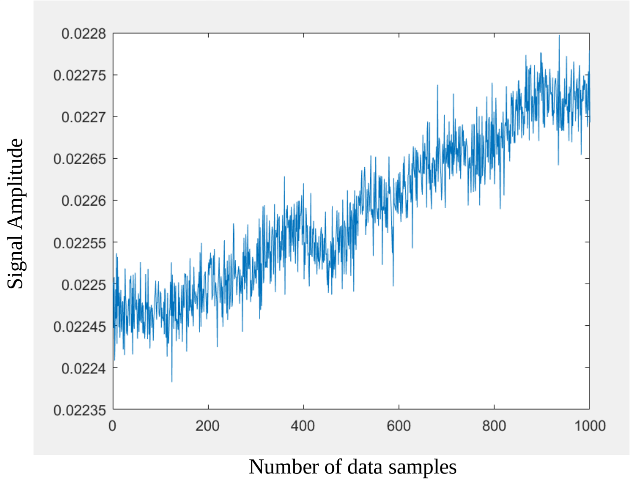


Figure 3.17. *Received Signal in Case 2, Breathing Received Data View*

3.3.2.3 . *CASE 2: Walking between receiver and transmitter*

In this experimental case, a man walks between the receiver and the transmitter. The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.18.

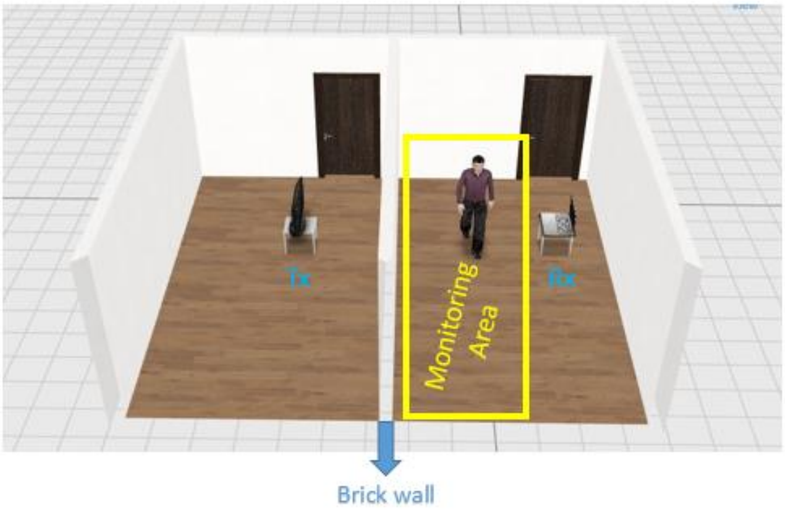


Figure 3.18. *Case 2, Walking Representation*

An example of sample through the wall from walking is as Figure 3.19. This example shows the change in signal amplitude for 10 seconds.

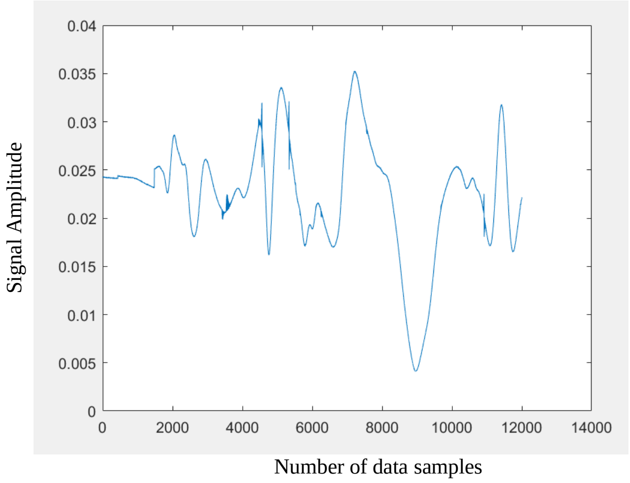


Figure 3.19. *Received Signal in Case 2, Walking*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.20.

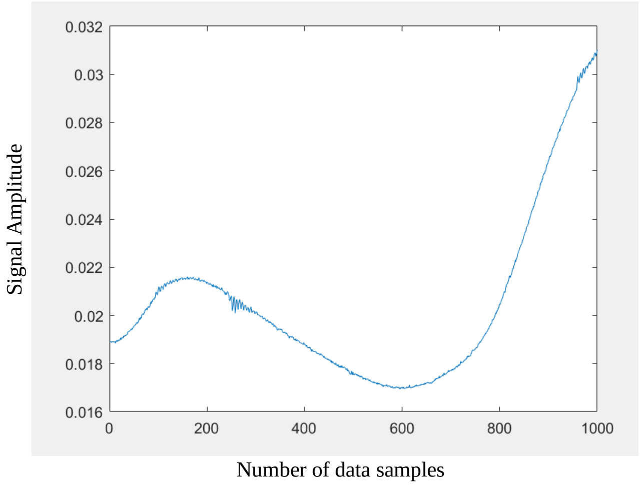


Figure 3.20. *Received Signal in Case 2, Walking Received Data View*

3.3.3. CASE 3

In this experimental case, there are two walls between receiver and transmitter.

3.3.3.1 . *CASE 3: Human absence*

In this experimental case, there is no human between receiver and transmitter. The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.21.

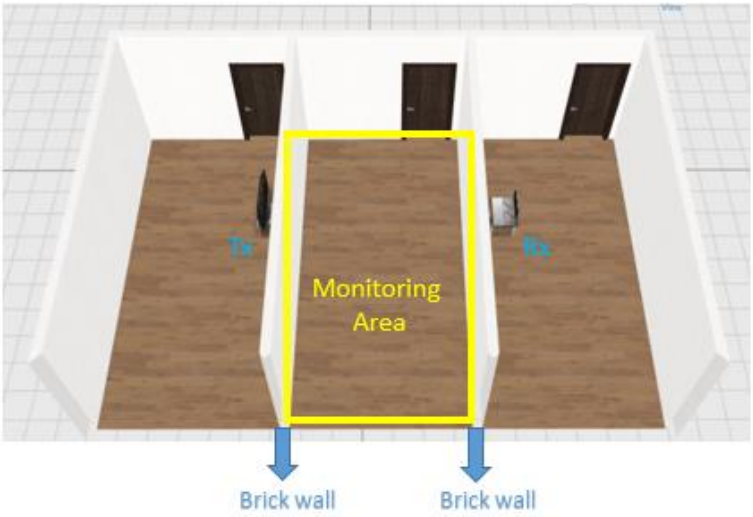


Figure 3.21. *Case 3, Human Absence Representation*

An example of sample between 2 walls from blank media is as Figure 3.22. This example shows the change in signal amplitude for 10 seconds.

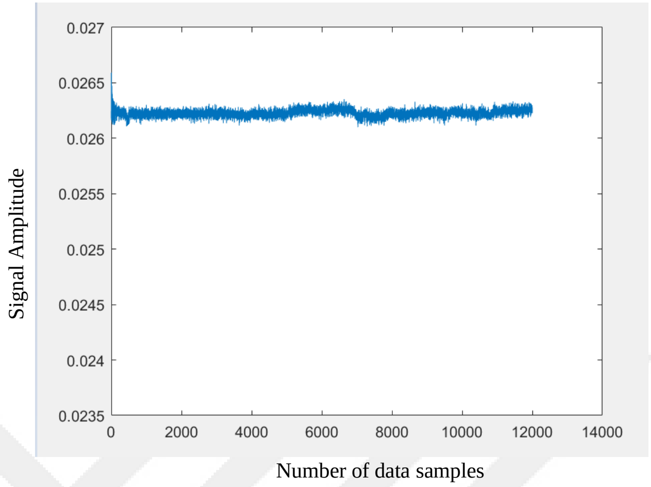


Figure 3.22. *Received Signal in Case 3, Human Absence*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.23.

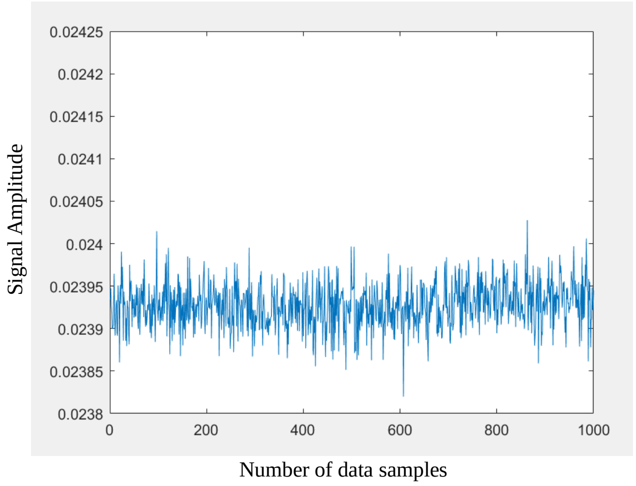


Figure 3.23. *Received Signal in Case 3, Blank Media Received Data View*

3.3.3.2 . *CASE 3: Walking between receiver and transmitter*

In this experimental case, a man walks in different room, between the receiver and the transmitter. The distance between the receiver and the transmitter is 3 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 1 dBm. The 3D representation of the scenario is as in Figure 3.24.

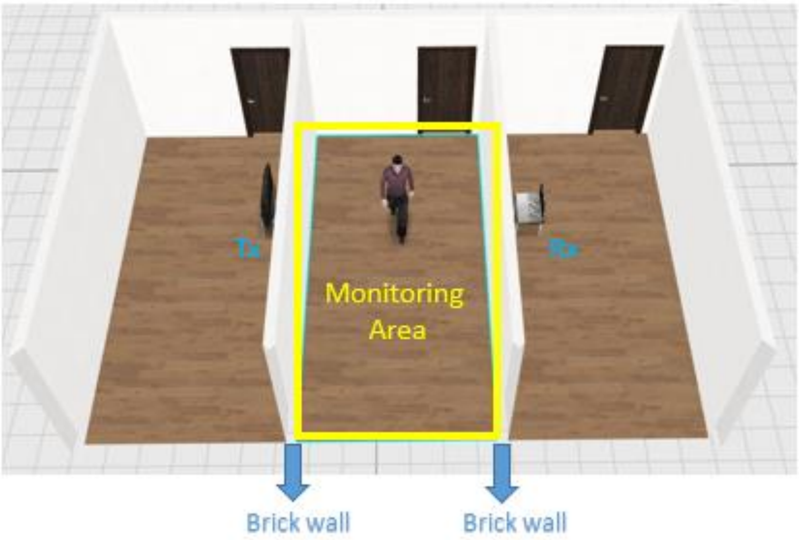


Figure 3.24. *Case 3, Walking Representation*

An example of sample between 2 walls from walking is as Figure 3.25. This example shows the change in signal amplitude for 10 seconds.

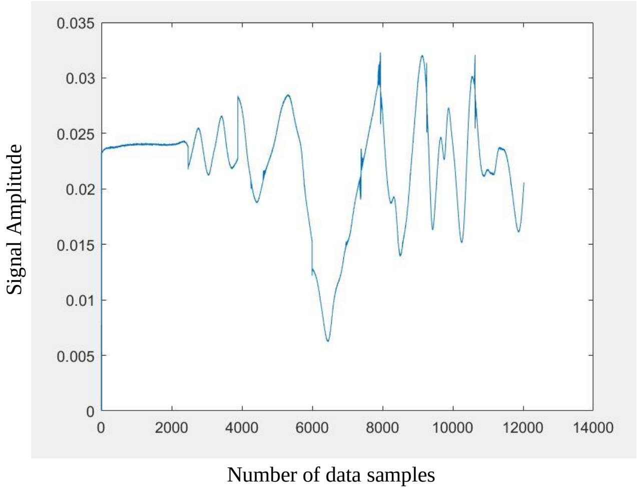


Figure 3.25. *Received Signal in Case 3, Walking*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.26.

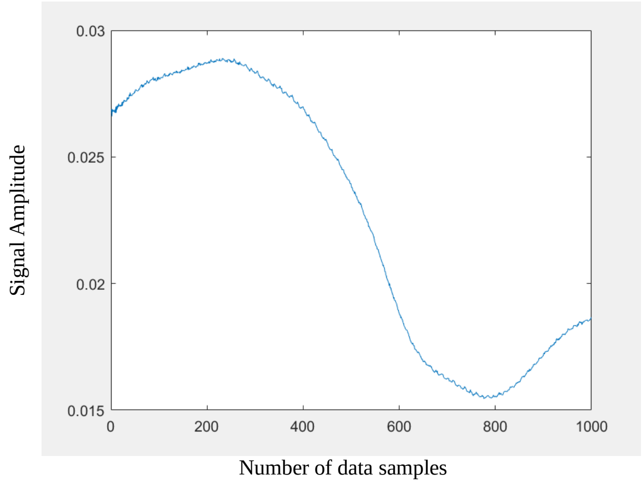


Figure 3.26. *Received Signal in Case 3, Walking Received Data View*

3.3.4. CASE 4

Case 1-3 are intended to detect movement between the transmitter and receiver. In this case, the receiver and transmitter are placed in the same place (room) and the walking motion through the wall is detected.

3.3.4.1 . *CASE 4: Human absence*

In this experimental case, there is no human between receiver and transmitter. The distance between the receiver and the transmitter is 1 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The 3D representation of the scenario is as in Figure 3.27.

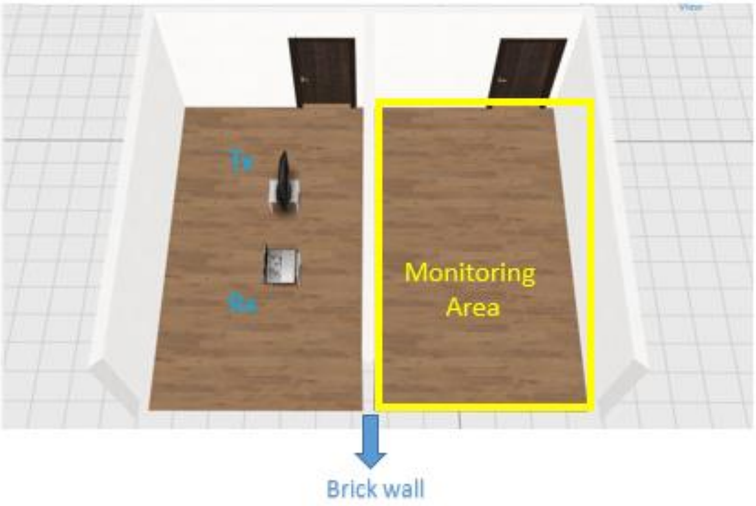


Figure 3.27. *Case 4, Human Absence Representation*

An example of sample which the receiver and transmitter are in the same room and observe the other side of the wall, blank media sample is as Figure 3.28. This example shows the change in signal amplitude for 10 seconds.

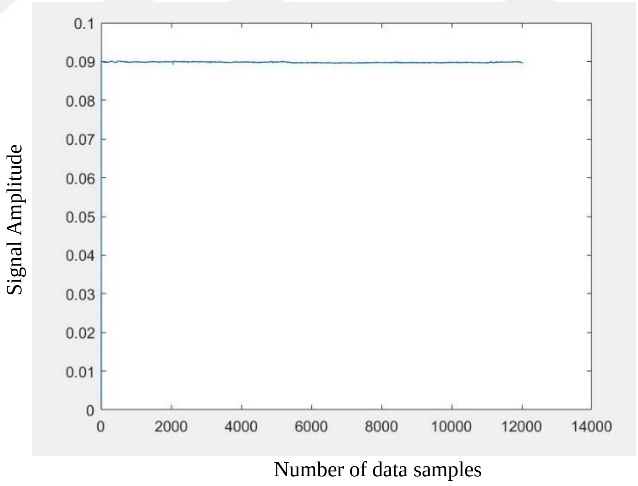


Figure 3.28. *Received Signal in Case 4, Human Absence*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.29.

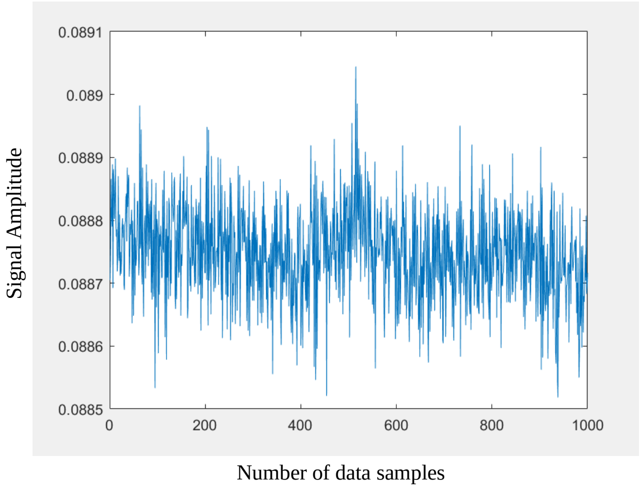


Figure 3.29. *Received Signal in Case 4, Blank Media Received Data View*

3.3.4.2 . *CASE 4: Walking*

In this experimental case, a man walks in different room. The distance between the receiver and the transmitter is 1 meters. The top of the antenna used on the transmitter side has 1 meter highest from ground. The signal generator output power used on the transmitter side is set to a maximum of 5 dBm. The 3D representation of the scenario is as in Figure 3.30.

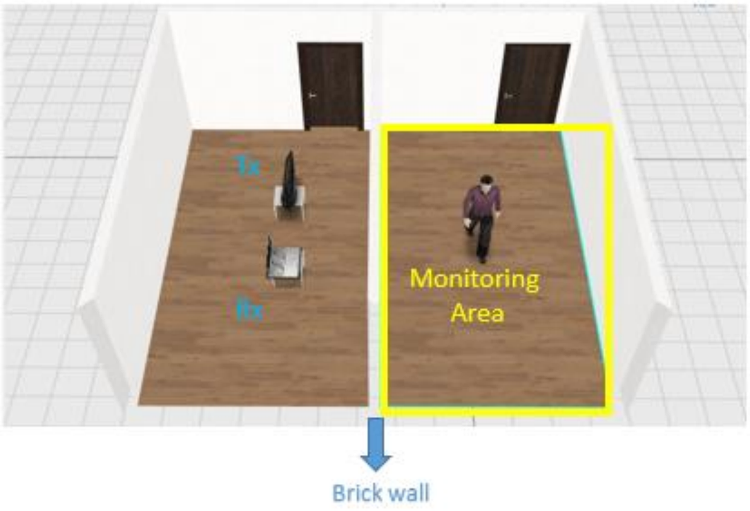


Figure 3.30. *Case 4, Walking Representation*

An example of sample which the receiver and transmitter are in the same room and observe the other side of the wall, walking sample is as Figure 3.31. This example shows the change in signal amplitude for 10 seconds.

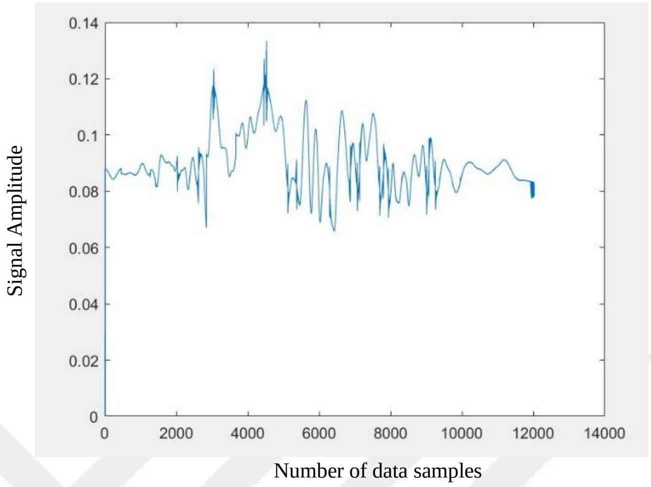


Figure 3.31. *Received Signal in Case 4, Walking*

In this case, totally 240 records are collected. The sample consists of 1000 samples. An example of 1000 samples is as shown in Figure 3.32.

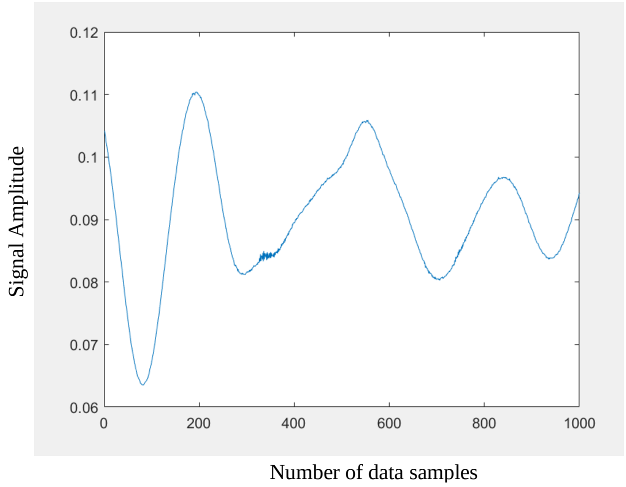


Figure 3.32. *Received Signal in Case 4, Walking Received Data View*

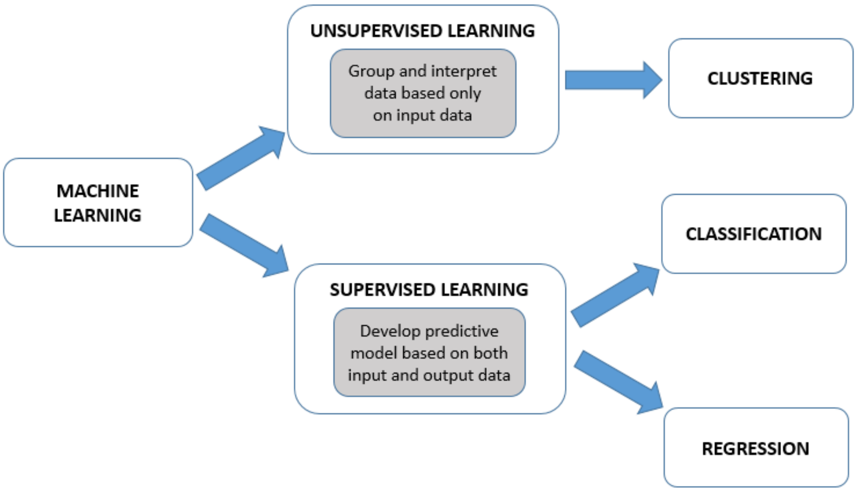
3.4. Machine Learning

Machine Learning (ML) is a set of algorithms that allow accurate estimation of results without programming being performed explicitly. The basis of machine learning is the use of statistical analysis methods to update outputs as new inputs occur, to make inferences from existing data, and to make predictions about the unknown by inferences. While writing a computer program with classical methods, people determine the problems that may occur, people solve the problem solutions and transfer this into machine language. However, the purpose of Machine Learning is to improve and learn by updating the information over time after the software or any machine where the software is encountered. Machine learning is the ability to access and use data in the learning process, and make close predictions and decisions that are appropriate to the problem. Machine learning can sometimes be very useful, but it is not a magic wand, it requires patience to model with different algorithms, different features, different data, and a large number of trial and error attempts. Algorithms used in machine learning try their performance by increasing the number of samples.

MLs are used to make critical decisions in voice recognition systems credit scoring, face recognition, image processing, price and load forecasting in energy production and many other issues.

3.4.1. How machine learning works

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.



MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLUSTERING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLASSIFICATION
REGRESSION

Figure 3.33. *Unsupervised and Supervised Learning Techniques*

3.4.1.1 . *Supervised techniques*

The majority of machine learning researchers use supervised learning. There are class concepts. In fact, there are concepts such as good / bad, yes / no, animal / plant and these concepts are tried to be predicted. When the input data is taken as the x value, the output variables can be estimated for this data. If data is added or removed, it must be retrained from the algorithm. This allows the values of the input and output variables to change. Thus, the forecasting process may change. Supervised learning problems can be further classified as classification and regression problems.

Classification technique creates the output variable by distributing the data within a given data set to various classes. The output variable contains values in a category such as yes or no, human or not.

Regression technique can estimate a continuous quantity, it cannot make classification estimates. Regression techniques are used when working with a data range or if the quality is a real number, such as temperature or the time it takes for a piece of equipment to fail.

3.4.1.2 . *Unsupervised techniques*

In this model, it contains only input data (x), no data corresponds to the output values. It is a learning only where x is received as input data and does not have corresponding output variables. The aim is to increase the available data and to learn more and to model the general structure of the data. The difference from supervised learning is that there is no result and no teacher. Algorithms are left to their plans to learn and model the structure of the data.

Clustering technique, the data samples in the objective data set grouping by feature vectors only. For this, the similarity of the samples is considered. In clustering literature, the term similarity is in contrast to the term distance. Similar samples are placed in the same cluster and distant samples are placed in different clusters. Number of clusters is usually externally.

3.4.2. Deciding machine learning algorithm to use

There are a large number of supervised and unsupervised machine learning algorithms. It is difficult to choose the correct algorithm from these algorithms. There is no best model, and finding the right algorithm is a complete trial and error method. The choice of algorithm depends on the size and type of data you are working on, the information and results you want to obtain from the data, as well as how these outputs will be used. If you need to train a model to make a prediction, choose the supervised techniques. If you need to explore your data and want to train a model to find a good internal representation, choose unsupervised learning, such as splitting data up into clusters.

In this study, classification method is used for machine learning. Since we do not want to see a qualitative number as a result in machine learning, the classification method is preferred. In this thesis, MATLAB Classification Learner is used for classification.

3.4.3. Classify data using the Classification Learner Toolbox

Classification Learner lets user perform common supervised learning tasks such as interactively exploring your data, selecting features, specifying validation schemes, training models, and assessing results. The user can select one of several types of classification. Classification Learner tool box allows user to input data from matrices or tables. It can automatically define user estimates based on the data type in the generated matrix or table. Assigns items to a specific group or class based on a specific property set.

Engineers and scientists can use cross-validation to test how accurately a model will evaluate data. After cross-validation, you can select the best-fitting classification model. There are many types of classification models, here are five common types:

***Logistic Regression*:** This model is often used as a baseline due to its simplicity. Used in problems where there are two classes in which data can be classified. A logistic regression model returns probabilities for how likely a data point belongs to each class.

***k-nearest neighbor (kNN)*:** This simple yet effective way of classification categorizes data points based on their distance to other points in a training dataset. The training time of kNN is short, but this model can confuse irrelevant attributes for important ones unless weights are applied to the data, especially as the number of data points grows.

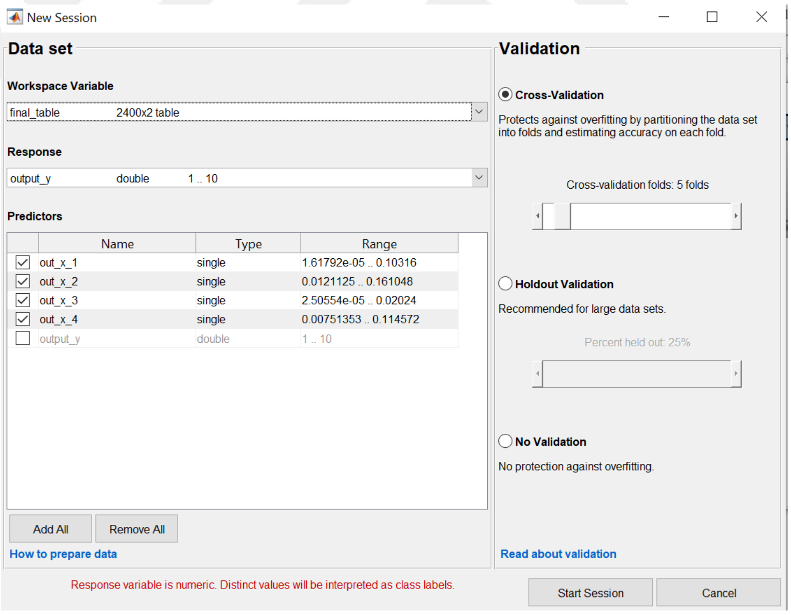
Decision Trees: These models predict responses visually, and it’s relatively easy to follow the decision path taken from root to leaf. This type of model is especially useful when it’s important to show how the conclusion was reached.

Support Vector Machine (SVM)**:** This model uses a hyperplane to separate data into two or more classes. It is accurate, tends not to over fit, and is relatively easy to interpret, but training time can be on the longer side especially for larger datasets.

Artificial neural networks (ANNs)**:** These networks can be configured and trained to solve a variety of different problems including classification and time series prediction. However, the trained models are known to be difficult to interpret. Different classifiers work best for different types of problems and data sets.

3.4.3.1 . *Classification Learner*

Classification Learner tool box allows user to input data from matrices or tables. Values can be easily classified by creating a single table that contains the properties and tagged classes that you previously specified. One of the toolbox that affect result is validation toolbox shown in Figure 3.34.



Workspace Variable: final_table 2400x2 table

Response: output_y double 1 .. 10

Predictors

	Name	Type	Range
☑	out_x_1	single	1.61792e-05 .. 0.10316
☑	out_x_2	single	0.0121125 .. 0.161048
☑	out_x_3	single	2.50554e-05 .. 0.02024
☑	out_x_4	single	0.00751353 .. 0.114572
☐	output_y	double	1 .. 10

● Cross-Validation

Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.

Cross-validation folds: 5 folds

○ Holdout Validation

Recommended for large data sets.

Percent held out: 25%

○ No Validation

No protection against overfitting.

Response variable is numeric. Distinct values will be interpreted as class labels.

Figure 3.34. *Classification Learner Table/Matrix Selection*

When you look at the right side of new session tab, the validation tab will be seen. Select a number of folds (or divisions) to partition the data set using the slider control. If you choose k folds, then the app, partitions the data into k disjoint sets or folds. For each fold, trains a model using the out-of-fold observations, assesses model performance using in-fold data and calculates the average test error over all folds. This method gives a good estimate of the predictive accuracy of the final model trained with all the data. It requires multiple fits but makes efficient use of all the data, so it is recommended for small data sets.

Holdout Validation, select a percentage of the data to use as a test set using the slider control. The app trains a model on the training set and assesses its performance with the test set. The model used for validation is based on only a portion of the data, so Holdout Validation is recommended only for large data sets. The final model is trained with the entire data set [39].

No Validation, no protection against overfitting. The app uses all of the data for training and computes the error rate on the same data. Without any test data, you get an unrealistic estimate performance of the model on new data. That is, the training sample accuracy is likely to be unrealistically high, and the predictive accuracy is likely to be lower.

The validation scheme only affects the way that Classification Learner computes validation metrics. The final model is always trained using the entire data set.

4. RESULTS

In this study, 10 classes are identified and 240 samples are collected for each of classes. A total of 2400 data are classified in order to try to find best classification type which has high accuracy.

4.1. Effects of Cross Validation Folds Value

Cross validation folds value is selected suitable for 4 features. You can see the folds value and its accuracy in Table 4-1 suitable for four features.

Table 4-1 *Folds value and its accuracy*

Folds value	Accuracy
2	92,5 %
5	92,7 %
10	92,9 %
12	92,5 %

According to the table, in this study, the folds value is taken as 10 because of the highest accuracy value and all calculations are made according to this value.

4.2. Number of Features Effects on Accuracy

4.2.1. Single feature results

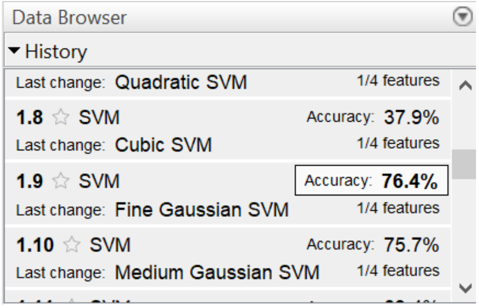
First, using only the first feature, we will examine how to separate classes with a single feature. The result is as in Table 4-2.

Table 4-2 *Results for first feature*

Number	Type	Last Change	Number of Features	Accuracy
1	Tree	Fine Tree	1	75,0 %
2	Tree	Medium Tree	1	73,2 %
3	Tree	Coarse Tree	1	48,4 %
4	Linear Discriminant	Linear Discriminant	1	57,3 %

5	Quadratic Discriminant	Quadratic Discriminant	1	67,7 %
6	SVM	Linear SVM	1	64,2 %
7	SVM	Quadratic SVM	1	55,6 %
8	SVM	Cubic SVM	1	37,9 %
9	**SVM**	**Fine Gaussian SVM**	**1**	**76,4 %**
10	SVM	Medium Gaussian SVM	1	75,7 %
11	SVM	Coarse Gaussian SVM	1	63,4 %
12	KNN	Fine KNN	1	67,5 %
13	KNN	Medium KNN	1	75,9 %
14	KNN	Coarse KNN	1	74,3 %
15	KNN	Cosine KNN	1	20,0 %
16	KNN	Cubic KNN	1	75,9 %
17	KNN	Weighted KNN	1	68,8 %
18	Ensemble	Boasted Trees	1	73,7 %
19	Ensemble	Bagged Ensemble	1	67,8 %
20	Ensemble	Subspace Discriminant	1	57,3 %
21	Ensemble	Subspace KNN	1	67,5 %
22	Ensemble	RUSBoosted Trees	1	73,2 %

When the table examined, the highest accuracy rate is line 9 in Table 4-2. MATLAB shows the highest accuracy with highlight shown in Figure 4.1.



Data Browser

History

Last change: Quadratic SVM 1/4 features

1.8 ☆ SVM Accuracy: 37.9%
Last change: Cubic SVM 1/4 features

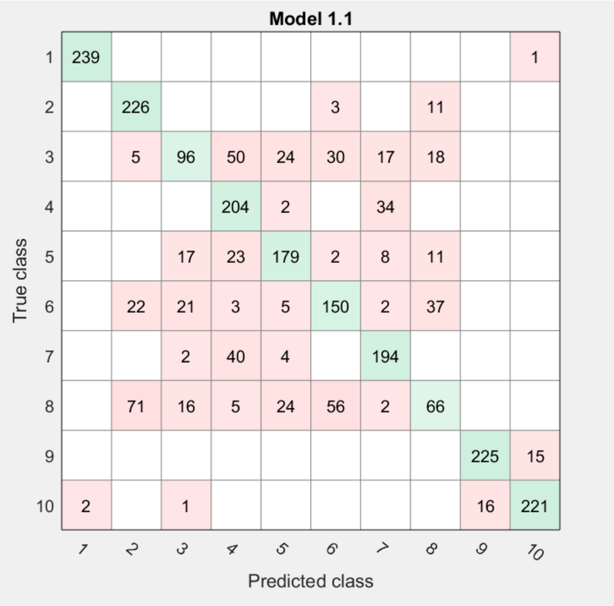
1.9 ☆ SVM Accuracy: 76.4%
Last change: Fine Gaussian SVM 1/4 features

1.10 ☆ SVM Accuracy: 75.7%
Last change: Medium Gaussian SVM 1/4 features

Figure 4.1. *Highest accuracy for the first feature*

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

For the first feature, the confusion matrix is as shown in Figure 4.2.



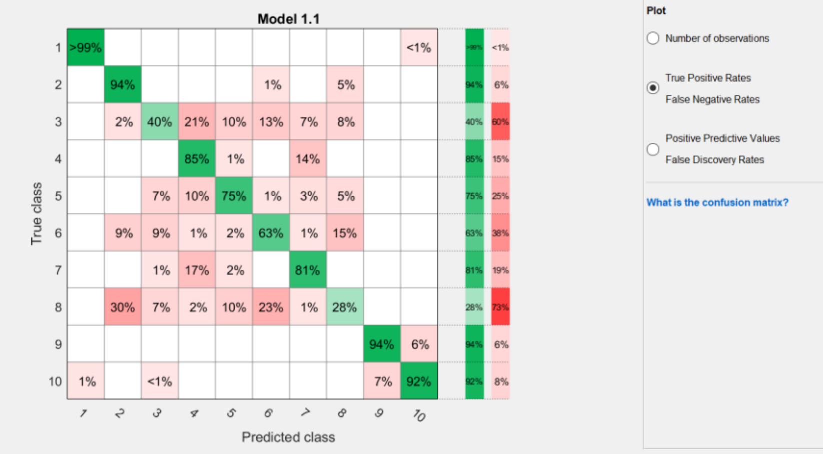
Model 1.1

True class \ Predicted class	1	2	3	4	5	6	7	8	9	10
1	239									1
2		226				3		11		
3		5	96	50	24	30	17	18		
4				204	2		34			
5			17	23	179	2	8	11		
6		22	21	3	5	150	2	37		
7			2	40	4		194			
8		71	16	5	24	56	2	66		
9									225	15
10	2		1						16	221

Figure 4.2. *Confusion matrix for first feature*

When a classification is made for the first feature, it is seen that Class 3 and Class 8 data are mixed with other classes as shown in Figure 4.2. Therefore, the accuracy rate is low.

The columns show the predicted classes. In the top row, 239 of samples from first class are correctly classified, so 99 % is the true positive rate for correctly classified points in this class, shown in the green cell in the True Positive Rate column in Figure 4.3.



Model 1.1

True class	1	2	3	4	5	6	7	8	9	10		
1	>99%									<1%	>99%	<1%
2		94%				1%		5%			94%	6%
3		2%	40%	21%	10%	13%	7%	8%			40%	60%
4				85%	1%		14%				85%	15%
5			7%	10%	75%	1%	3%	5%			75%	25%
6		9%	9%	1%	2%	63%	1%	15%			63%	38%
7			1%	17%	2%		81%				81%	19%
8		30%	7%	2%	10%	23%	1%	28%			28%	73%
9									94%	6%	94%	6%
10	1%		<1%						7%	92%	92%	8%

Predicted class

Plot

○ Number of observations

● True Positive Rates False Negative Rates

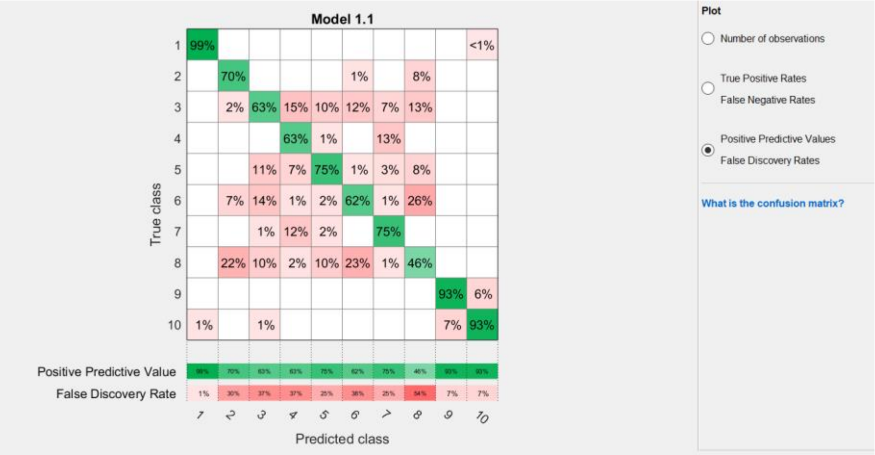
○ Positive Predictive Values False Discovery Rates

What is the confusion matrix?

Figure 4.3. *True positive rates for the first feature*

True Positive Rate is calculated as the number of correct positive predictions divided by the total number of positives [39]. The other data in the first class row are misclassified: 1 % of the data are incorrectly classified as from Class 10. 1 % is the false negative rate for incorrectly classified points in this class, shown in the red cell in the False Negative Rate column.

If false positives are important in your classification problem, plot results per predicted class (instead of true class) to investigate false discovery rates. To see results per predicted class, under Plot, select the “Positive Predictive Values False Discovery Rates” option. The confusion matrix now shows summary rows underneath the table. Positive predictive values are shown in Figure 4.4 as green color for the correctly predicted points in each class, and false discovery rates are shown below it in red for the incorrectly predicted points in each class [41].



Model 1.1

True class \ Predicted class	1	2	3	4	5	6	7	8	9	10
1	99%									<1%
2		70%				1%		8%		
3		2%	63%	15%	10%	12%	7%	13%		
4				63%	1%		13%			
5			11%	7%	75%	1%	3%	8%		
6		7%	14%	1%	2%	62%	1%	26%		
7			1%	12%	2%		75%			
8		22%	10%	2%	10%	23%	1%	46%		
9									93%	6%
10	1%		1%						7%	93%
Positive Predictive Value	99%	70%	63%	63%	75%	62%	75%	46%	93%	93%
False Discovery Rate	1%	30%	37%	37%	25%	38%	25%	54%	7%	7%

Plot

○ Number of observations

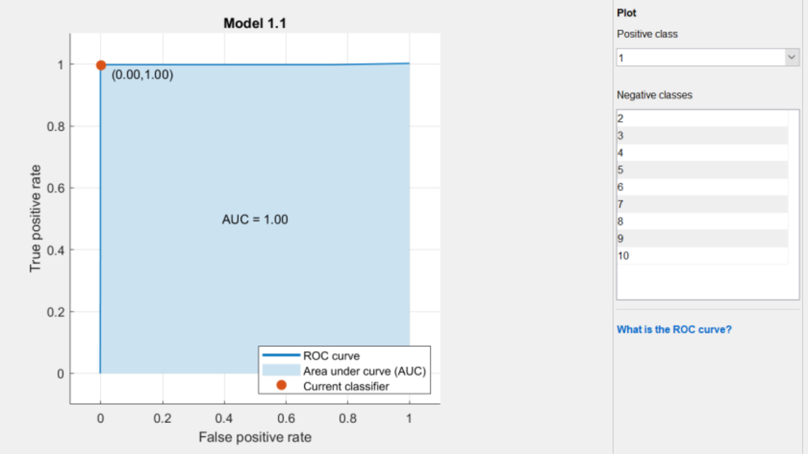
○ True Positive Rates
False Negative Rates

● Positive Predictive Values
False Discovery Rates

Figure 4.4. *Positive predictive rates for the first feature*

To view the receiver operating characteristic (ROC) curve after training a model, on the Classification Learner tab, in the Plots section, click ROC Curve. View the receiver operating characteristic (ROC) curve showing true and false positive rates. The ROC curve shows true positive rate versus false positive rate for the currently selected trained classifier [41]. You can select different classes to plot. If we examine the ROC curve for the two best and worst classified classes.

For the first class, ROC curve is shown in Figure 4.5.



Plot

Positive class

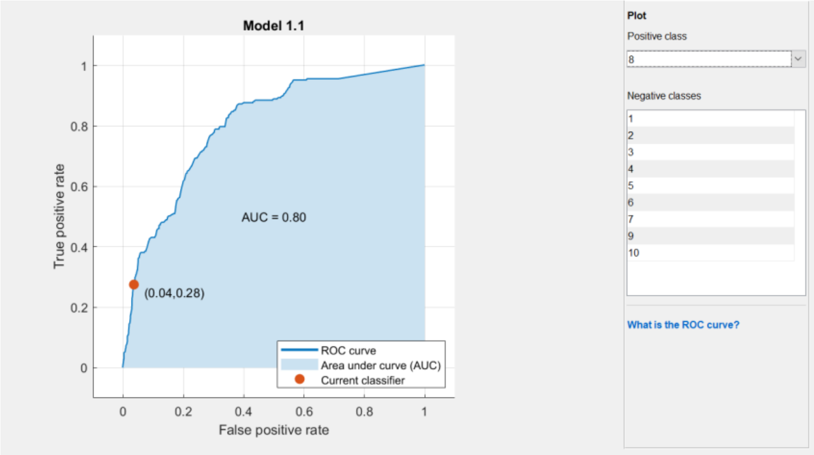
1

Negative classes

2
3
4
5
6
7
8
9
10

Figure 4.5. *For first feature and for the first class' ROC curve*

For the Class 8, ROC curve is shown in Figure 4.6.



Plot

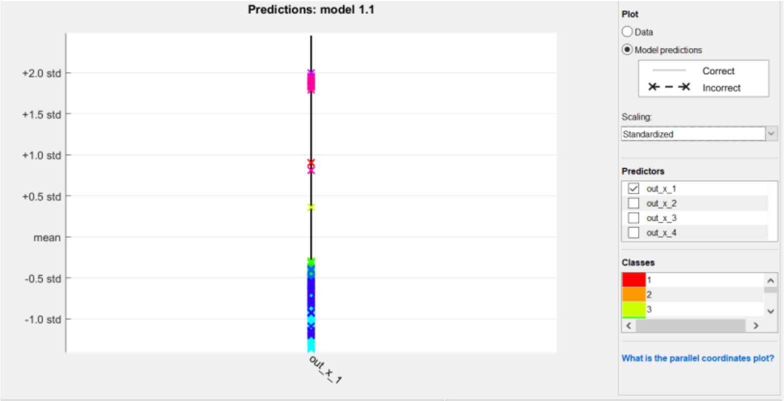
Positive class: 8

Negative classes: 1, 2, 3, 4, 5, 6, 7, 9, 10

Figure 4.6. *For first feature and for the 8th class's ROC curve*

View the ROC curve showing true and false positive rates. The ROC curve shows true positive rate versus false positive rate for the currently selected trained classifier. A perfect result with no misclassified points is a right angle to the top left of the plot. A poor result that is no better than random is a line at 45 degrees. The area under curve number is a measure of the overall quality of the classifier. Larger area under curve values indicate better classifier performance.

You can visualize high dimensional data on a single plot to see two dimensions patterns. This plot can help you understand relationships between features and identify useful predictors for separating classes. You can visualize training data and misclassified points on the parallel coordinates plot. When you plot classifier results, misclassified points show dashed lines [41].



Plot

○ Data

● Model predictions

Scaling: Standardized

Predictors

☑ out_x_1

□ out_x_2

□ out_x_3

□ out_x_4

Classes

🟥 1

🟧 2

🟨 3

Figure 4.7. *Parallel Coordinates plot for first feature*

You can see which class each color matches in Table 4-3 below.

Table 4-3 *Classes and Colors*

Class	Class Color
1	Red
2	Orange
3	Yellow
4	Green
5	Dark Green
6	Turquoise
7	Blue
8	Dark Blue
9	Purple
10	Pink

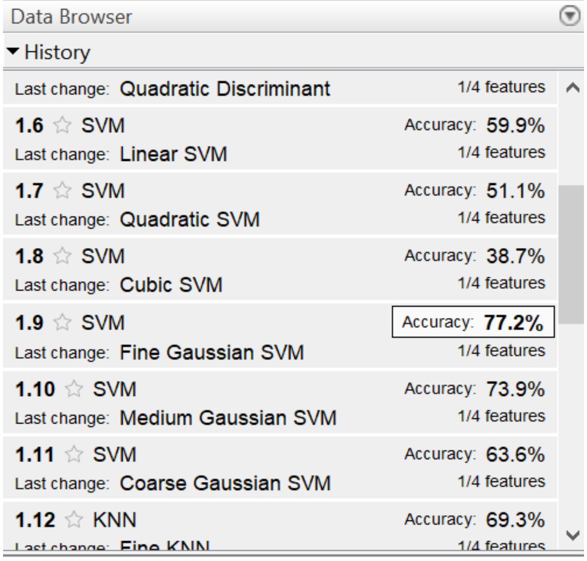
4.2.2. Second feature results

We will examine how to separate classes with second feature. The result is as in Table 4-4.

Table 4-4 *Results for second feature*

Number	Type	Last Change	Number of Features	Accuracy
1	Tree	Fine Tree	1	75,6 %
2	Tree	Medium Tree	1	70,4 %
3	Tree	Coarse Tree	1	45,1 %
4	Linear Discriminant	Linear Discriminant	1	56,6 %
5	Quadratic Discriminant	Quadratic Discriminant	1	66,9 %
6	SVM	Linear SVM	1	59,9 %
7	SVM	Quadratic SVM	1	51,1 %
8	SVM	Cubic SVM	1	38,7 %
9	**SVM**	**Fine Gaussian SVM**	**1**	**77,2 %**
10	SVM	Medium Gaussian SVM	1	73,9 %
11	SVM	Coarse Gaussian SVM	1	63,6 %
12	KNN	Fine KNN	1	69,3 %
13	KNN	Medium KNN	1	76,6 %
14	KNN	Coarse KNN	1	72,4 %
15	KNN	Cosine KNN	1	20,0 %
16	KNN	Cubic KNN	1	76,6 %
17	KNN	Weighted KNN	1	70,6 %
18	Ensemble	Boasted Trees	1	74,1 %
19	Ensemble	Bagged Ensemble	1	69,6 %
20	Ensemble	Subspace Discriminant	1	56,6 %
21	Ensemble	Subspace KNN	1	69,3 %
22	Ensemble	RUSBoosted Trees	1	69,4 %

When the table examined, the highest accuracy rate is line 9 in Table 4-4. MATLAB shows the highest accuracy with highlight shown in Figure 4.8.



Data Browser

History

Last change: Quadratic Discriminant 1/4 features

1.6 ☆ SVM Accuracy: 59.9%
Last change: Linear SVM 1/4 features

1.7 ☆ SVM Accuracy: 51.1%
Last change: Quadratic SVM 1/4 features

1.8 ☆ SVM Accuracy: 38.7%
Last change: Cubic SVM 1/4 features

1.9 ☆ SVM Accuracy: 77.2%
Last change: Fine Gaussian SVM 1/4 features

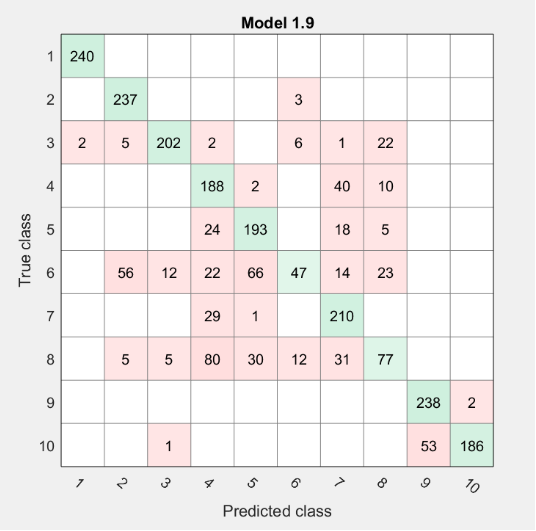
1.10 ☆ SVM Accuracy: 73.9%
Last change: Medium Gaussian SVM 1/4 features

1.11 ☆ SVM Accuracy: 63.6%
Last change: Coarse Gaussian SVM 1/4 features

1.12 ☆ KNN Accuracy: 69.3%
Last change: Fine KNN 1/4 features

Figure 4.8. *Highest accuracy for the second feature*

For second feature, the confusion matrix, true positive rates and positive predictive values is as shown in Figure 4.9.



Model 1.9

True class	1	2	3	4	5	6	7	8	9	10
1	240									
2		237				3				
3	2	5	202	2		6	1	22		
4				188	2		40	10		
5				24	193		18	5		
6		56	12	22	66	47	14	23		
7				29	1		210			
8		5	5	80	30	12	31	77		
9									238	2
10			1						53	186

Predicted class

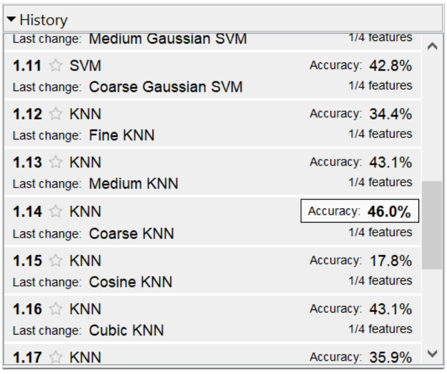
Figure 4.9. *Confusion Matrix for second feature*

We will examine how to separate classes with the third feature. The result is as in Table 4-5 below.

Table 4-5 *Results for the third feature*

Number	**Type**	**Last Change**	**Number of Features**	**Accuracy**
1	Tree	Fine Tree	1	42,3 %
2	Tree	Medium Tree	1	44,5 %
3	Tree	Coarse Tree	1	37,0 %
4	Linear Discriminant	Linear Discriminant	1	31,0 %
5	Quadratic Discriminant	Quadratic Discriminant	1	34,3 %
6	SVM	Linear SVM	1	38,6 %
7	SVM	Quadratic SVM	1	17,9 %
8	SVM	Cubic SVM	1	12,7 %
9	SVM	Fine Gaussian SVM	1	45,0 %
10	SVM	Medium Gaussian SVM	1	44,6 %
11	SVM	Coarse Gaussian SVM	1	42,8 %
12	KNN	Fine KNN	1	34,4 %
13	KNN	Medium KNN	1	43,1 %
14	**KNN**	**Coarse KNN**	**1**	**46,0 %**
15	KNN	Cosine KNN	1	17,8 %
16	KNN	Cubic KNN	1	43,1 %
17	KNN	Weighted KNN	1	35,9 %
18	Ensemble	Boasted Trees	1	44,5 %
19	Ensemble	Bagged Ensemble	1	34,5 %
20	Ensemble	Subspace Discriminant	1	31,0 %
21	Ensemble	Subspace KNN	1	34,4 %
22	Ensemble	RUSBoosted Trees	1	44,5 %

When Table 4-5 examined, the highest accuracy rate is line 14. MATLAB shows the highest accuracy with highlight shown in Figure 4.10.



Last change: Medium Gaussian SVM — 1/4 features

1.11 ☆ SVM — Accuracy: 42.8%
Last change: Coarse Gaussian SVM — 1/4 features

1.12 ☆ KNN — Accuracy: 34.4%
Last change: Fine KNN — 1/4 features

1.13 ☆ KNN — Accuracy: 43.1%
Last change: Medium KNN — 1/4 features

1.14 ☆ KNN — Accuracy: 46.0%
Last change: Coarse KNN — 1/4 features

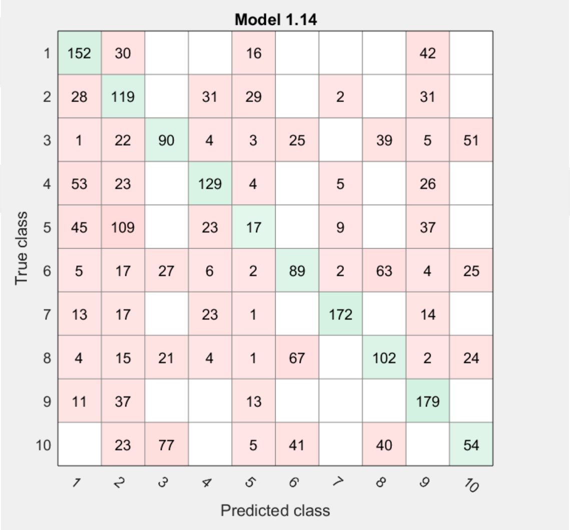
1.15 ☆ KNN — Accuracy: 17.8%
Last change: Cosine KNN — 1/4 features

1.16 ☆ KNN — Accuracy: 43.1%
Last change: Cubic KNN — 1/4 features

1.17 ☆ KNN — Accuracy: 35.9%

Figure 4.10. *Highest accuracy for the third feature*

For third feature, the confusion matrix is as shown in Figure 4.11.



	1	2	3	4	5	6	7	8	9	10
1	152	30			16				42	
2	28	119		31	29		2		31	
3	1	22	90	4	3	25		39	5	51
4	53	23		129	4		5		26	
5	45	109		23	17		9		37	
6	5	17	27	6	2	89	2	63	4	25
7	13	17		23	1		172		14	
8	4	15	21	4	1	67		102	2	24
9	11	37			13				179	
10		23	77		5	41		40		54

Figure 4.11. *Confusion Matrix for the third feature*

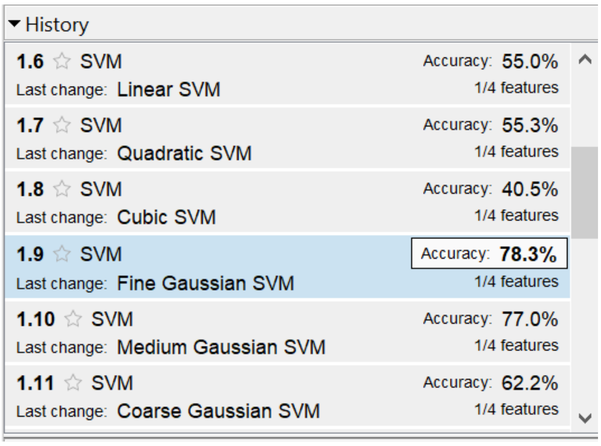
We will examine how to separate classes with the fourth feature. The result is as in Table 4-6.

Table 4-6 *Results for the fourth feature*

Number	Type	Last Change	Number of Features	Accuracy
1	Tree	Fine Tree	1	76,5 %
2	Tree	Medium Tree	1	75,6 %

3	Tree	Coarse Tree	1	43,1 %
4	Linear Discriminant	Linear Discriminant	1	60,0 %
5	Quadratic Discriminant	Quadratic Discriminant	1	68,3 %
6	SVM	Linear SVM	1	55,0 %
7	SVM	Quadratic SVM	1	55,3 %
8	SVM	Cubic SVM	1	40,5 %
9	**SVM**	**Fine Gaussian SVM**	**1**	**78,3 %**
10	SVM	Medium Gaussian SVM	1	77,0 %
11	SVM	Coarse Gaussian SVM	1	62,2 %
12	KNN	Fine KNN	1	71,3 %
13	KNN	Medium KNN	1	78,1 %
14	KNN	Coarse KNN	1	75,3 %
15	KNN	Cosine KNN	1	20,0 %
16	KNN	Cubic KNN	1	78,1 %
17	KNN	Weighted KNN	1	72,4 %
18	Ensemble	Boasted Trees	1	76,1 %
19	Ensemble	Bagged Ensemble	1	71,5 %
20	Ensemble	Subspace Discriminant	1	60,0 %
21	Ensemble	Subspace KNN	1	71,3 %
22	Ensemble	RUSBoosted Trees	1	75,5 %

When the table examined, the highest accuracy rate is line 9 in Table 4-6. MATLAB shows the highest accuracy with highlight shown in Figure 4.12.



History

1.6 ☆ SVM Accuracy: 55.0%
Last change: Linear SVM 1/4 features

1.7 ☆ SVM Accuracy: 55.3%
Last change: Quadratic SVM 1/4 features

1.8 ☆ SVM Accuracy: 40.5%
Last change: Cubic SVM 1/4 features

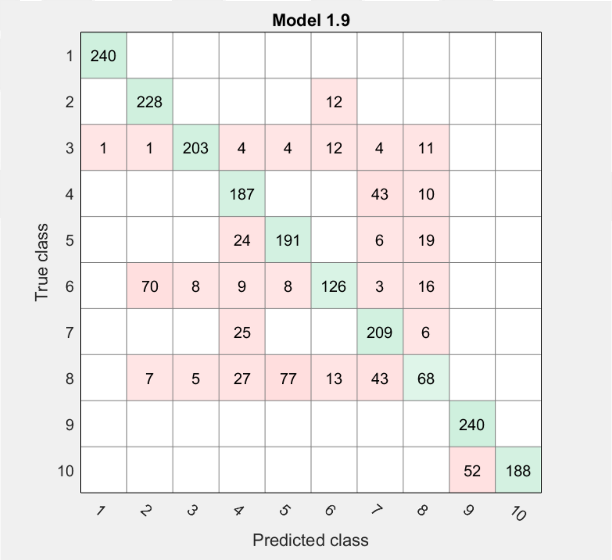
1.9 ☆ SVM Accuracy: 78.3%
Last change: Fine Gaussian SVM 1/4 features

1.10 ☆ SVM Accuracy: 77.0%
Last change: Medium Gaussian SVM 1/4 features

1.11 ☆ SVM Accuracy: 62.2%
Last change: Coarse Gaussian SVM 1/4 features

Figure 4.12. *Highest accuracy for the fourth feature*

For fourth feature, the confusion matrix is as shown in Figure 4.13.



Model 1.9

True class \ Predicted class	1	2	3	4	5	6	7	8	9	10
1	240									
2		228				12				
3	1	1	203	4	4	12	4	11		
4				187			43	10		
5				24	191		6	19		
6		70	8	9	8	126	3	16		
7				25			209	6		
8		7	5	27	77	13	43	68		
9									240	
10									52	188

Figure 4.13. *Confusion Matrix for the fourth feature*

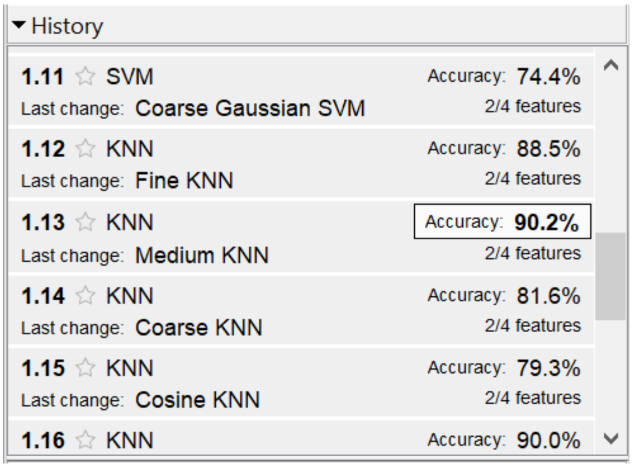
4.2.3. Results of two best features

In this section, the two best results (second feature: 77,2 % and fourth feature: 78,3 %) will be tested in the classification learner and the result will be examined. We will examine how to separate classes with two best features. The result is as in Table 4-7.

Table 4-7 *Results for the best two features*

Number	**Type**	**Last Change**	**Number of Features**	**Accuracy**
1	Tree	Fine Tree	2	87,9 %
2	Tree	Medium Tree	2	79,3 %
3	Tree	Coarse Tree	2	45,1 %
4	Linear Discriminant	Linear Discriminant	2	71,8 %
5	Quadratic Discriminant	Quadratic Discriminant	2	80,8 %
6	SVM	Linear SVM	2	81,0 %
7	SVM	Quadratic SVM	2	86,7 %
8	SVM	Cubic SVM	2	69,8 %
9	SVM	Fine Gaussian SVM	2	90,1 %
10	SVM	Medium Gaussian SVM	2	88,4 %
11	SVM	Coarse Gaussian SVM	2	74,4 %
12	KNN	Fine KNN	2	88,5 %
13	**KNN**	**Medium KNN**	**2**	**90,2 %**
14	KNN	Coarse KNN	2	81,6 %
15	KNN	Cosine KNN	2	79,3 %
16	KNN	Cubic KNN	2	90,0 %
17	KNN	Weighted KNN	2	90,0 %
18	Ensemble	Boasted Trees	2	86,8 %
19	Ensemble	Bagged Ensemble	2	88,6 %
20	Ensemble	Subspace Discriminant	2	70,3 %
21	Ensemble	Subspace KNN	2	71,5 %
22	Ensemble	RUSBoosted Trees	2	78,9 %

When the table examined, the highest accuracy rate is line 13 in Table 4-7. MATLAB shows the highest accuracy with highlight shown in Figure 4.14.



▾ History

1.11 ☆ SVM — Accuracy: 74.4%
Last change: Coarse Gaussian SVM — 2/4 features

1.12 ☆ KNN — Accuracy: 88.5%
Last change: Fine KNN — 2/4 features

1.13 ☆ KNN — Accuracy: 90.2%
Last change: Medium KNN — 2/4 features

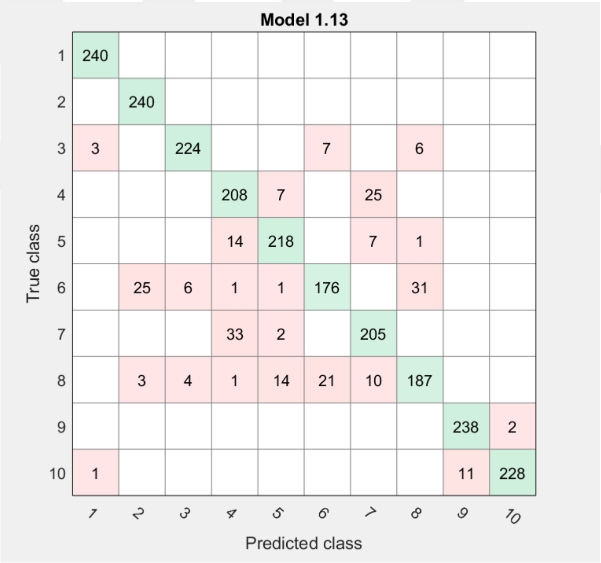
1.14 ☆ KNN — Accuracy: 81.6%
Last change: Coarse KNN — 2/4 features

1.15 ☆ KNN — Accuracy: 79.3%
Last change: Cosine KNN — 2/4 features

1.16 ☆ KNN — Accuracy: 90.0%

Figure 4.14. *Highest accuracy for the best two features*

For the best two features, the confusion matrix is as shown in Figure 4.15.



Model 1.13

True class \ Predicted class	1	2	3	4	5	6	7	8	9	10
1	240									
2		240								
3	3		224			7		6		
4				208	7		25			
5				14	218		7	1		
6		25	6	1	1	176		31		
7				33	2		205			
8		3	4	1	14	21	10	187		
9									238	2
10	1								11	228

Figure 4.15. *Confusion Matrix for the best two features*

When the highest accuracy rate is examined for each feature separately, the highest value is reached in the 4th feature (78,3 %). When the two best features are tested, it is observed that the accuracy increases.

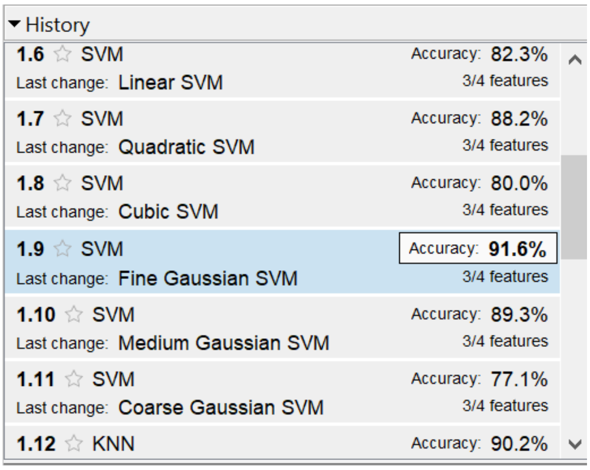
4.2.4. Results of three best features

In this section, the features which leads to the best results (first feature: 76,4 %, second feature: 77,2 % and fourth feature: 78,3 %) will be tested in the classification learner and the result will be examined. We will examine how to separate classes with three best features. The result is as in Table 4-8.

Table 4-8 *Results for the best three features*

Number	**Type**	**Last Change**	**Number of Features**	**Accuracy**
1	Tree	Fine Tree	3	89,5 %
2	Tree	Medium Tree	3	85,0 %
3	Tree	Coarse Tree	3	46,5 %
4	Linear Discriminant	Linear Discriminant	3	72,5 %
5	Quadratic Discriminant	Quadratic Discriminant	3	81,4 %
6	SVM	Linear SVM	3	82,3 %
7	SVM	Quadratic SVM	3	88,2 %
8	SVM	Cubic SVM	3	80,0 %
9	**SVM**	**Fine Gaussian SVM**	**3**	**91,6 %**
10	SVM	Medium Gaussian SVM	3	89,3 %
11	SVM	Coarse Gaussian SVM	3	77,1 %
12	KNN	Fine KNN	3	90,2 %
13	KNN	Medium KNN	3	91,2 %
14	KNN	Coarse KNN	3	84,4 %
15	KNN	Cosine KNN	3	81,9 %
16	KNN	Cubic KNN	3	90,9 %
17	KNN	Weighted KNN	3	91,1 %
18	Ensemble	Boasted Trees	3	89,2 %
19	Ensemble	Bagged Ensemble	3	91,5 %
20	Ensemble	Subspace Discriminant	3	72,5 %
21	Ensemble	Subspace KNN	3	90,0 %
22	Ensemble	RUSBoosted Trees	3	82,5 %

When the table examined, the highest accuracy rate is line 9 in Table 4-8. MATLAB shows the highest accuracy with highlight shown in Figure 4.16.

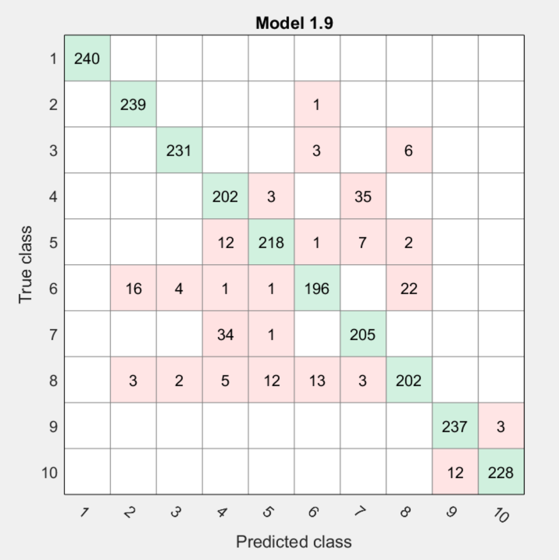


▾ History

1.6 ☆ SVM Last change: Linear SVM	Accuracy: 82.3% 3/4 features
1.7 ☆ SVM Last change: Quadratic SVM	Accuracy: 88.2% 3/4 features
1.8 ☆ SVM Last change: Cubic SVM	Accuracy: 80.0% 3/4 features
1.9 ☆ SVM Last change: Fine Gaussian SVM	Accuracy: **91.6%** 3/4 features
1.10 ☆ SVM Last change: Medium Gaussian SVM	Accuracy: 89.3% 3/4 features
1.11 ☆ SVM Last change: Coarse Gaussian SVM	Accuracy: 77.1% 3/4 features
1.12 ☆ KNN	Accuracy: 90.2%

Figure 4.16. *Highest accuracy for the best three features*

For the best three features, the confusion matrix is as shown in Figure 4.17.



Model 1.9

True class / Predicted class

	1	2	3	4	5	6	7	8	9	10
1	240									
2		239				1				
3			231			3		6		
4				202	3		35			
5				12	218	1	7	2		
6		16	4	1	1	196		22		
7				34	1		205			
8		3	2	5	12	13	3	202		
9									237	3
10									12	228

Figure 4.17. *Confusion Matrix for the best three features*

The highest accuracy for each feature is examined separately and the highest accuracy is achieved using the two best features. It is observed that the accuracy obtained by using three features is higher than the value obtained by using two features.

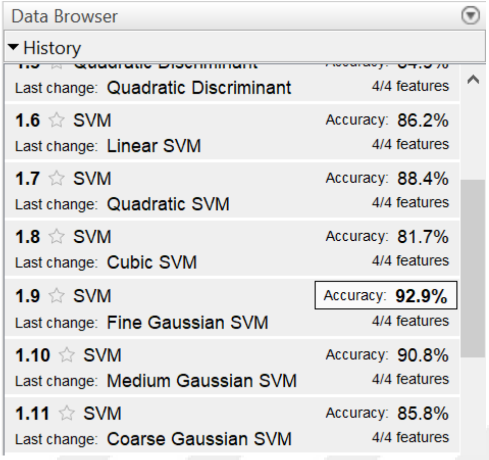
4.2.5. All features (four features) results

When classification learner is used for the four different features we have previously determined, the result is as Table 4-9.

Table 4-9 *Results for four feature*

Number	Type	Last Change	Number of Features	Accuracy
1	Tree	Fine Tree	4	89,3 %
2	Tree	Medium Tree	4	88,2 %
3	Tree	Coarse Tree	4	47,4 %
4	Linear Discriminant	Linear Discriminant	4	72,3 %
5	Quadratic Discriminant	Quadratic Discriminant	4	84,9 %
6	SVM	Linear SVM	4	86,2 %
7	SVM	Quadratic SVM	4	88,4 %
8	SVM	Cubic SVM	4	81,7 %
9	**SVM**	**Fine Gaussian SVM**	**4**	**92,9 %**
10	SVM	Medium Gaussian SVM	4	90,8 %
11	SVM	Coarse Gaussian SVM	4	85,8 %
12	KNN	Fine KNN	4	90,6 %
13	KNN	Medium KNN	4	91,2 %
14	KNN	Coarse KNN	4	82,6 %
15	KNN	Cosine KNN	4	82,0 %
16	KNN	Cubic KNN	4	91,2 %
17	KNN	Weighted KNN	4	91,6 %
18	Ensemble	Boasted Trees	4	90,0 %
19	Ensemble	Bagged Ensemble	4	91,8 %
20	Ensemble	Subspace Discriminant	4	72,3 %
21	Ensemble	Subspace KNN	4	90,8 %
22	Ensemble	RUSBoosted Trees	4	87,9 %

When the table examined, the highest accuracy rate is line 9 in Table 4-9. MATLAB shows the highest accuracy with highlight shown in Figure 4.18.



Data Browser

History

Last change: Quadratic Discriminant 4/4 features

1.6 ☆ SVM Accuracy: 86.2%
Last change: Linear SVM 4/4 features

1.7 ☆ SVM Accuracy: 88.4%
Last change: Quadratic SVM 4/4 features

1.8 ☆ SVM Accuracy: 81.7%
Last change: Cubic SVM 4/4 features

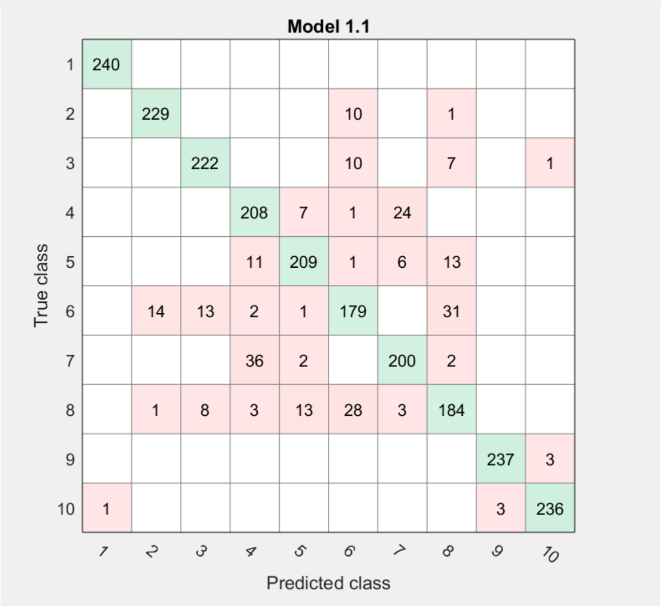
1.9 ☆ SVM Accuracy: **92.9%**
Last change: Fine Gaussian SVM 4/4 features

1.10 ☆ SVM Accuracy: 90.8%
Last change: Medium Gaussian SVM 4/4 features

1.11 ☆ SVM Accuracy: 85.8%
Last change: Coarse Gaussian SVM 4/4 features

Figure 4.18. *Highest accuracy for four features*

For four features, the confusion matrix is as shown in Figure 4.19.



Model 1.1

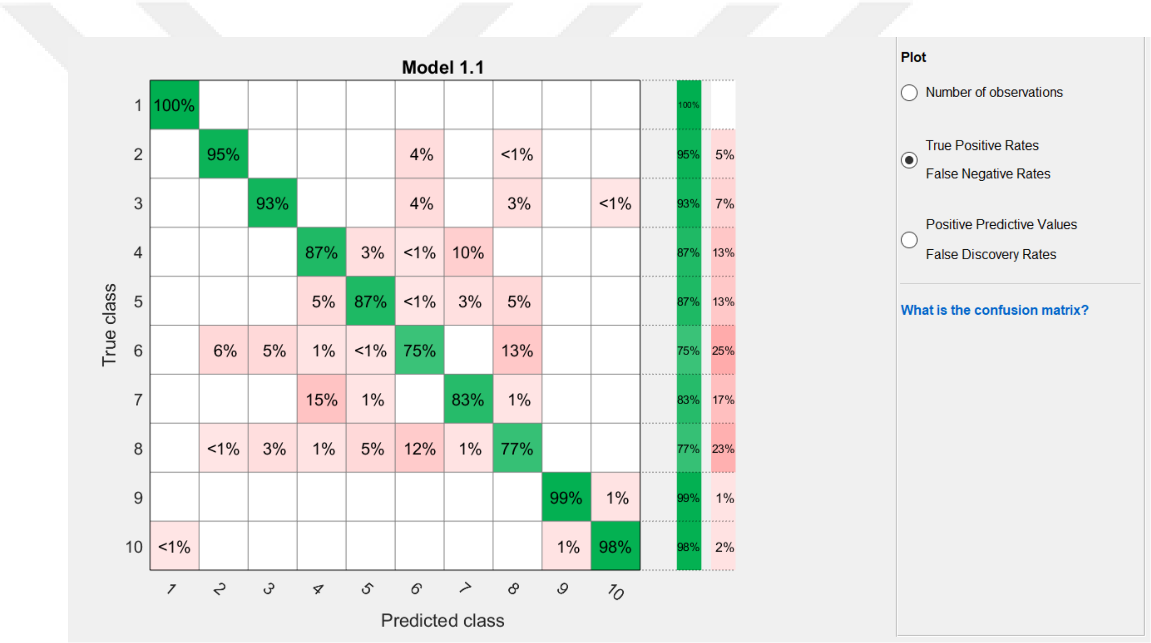
True class \ Predicted class	1	2	3	4	5	6	7	8	9	10
1	240									
2		229				10		1		
3			222			10		7		1
4				208	7	1	24			
5				11	209	1	6	13		
6		14	13	2	1	179		31		
7				36	2		200	2		
8		1	8	3	13	28	3	184		
9									237	3
10	1								3	236

Figure 4.19. *Confusion Matrix for four features*

When the confusion matrix is examined, it is seen that Class 1 can be fully distinguished from other classes. For the second class, 229 samples are correctly classified. Totally 11 samples are incorrectly classified. 10 samples are misclassified and evaluated as samples in Class 6. A sample is misclassified and evaluated as sample in Class 8.

The worst estimate value is made for Class 6. If we remember the Class 6, there is a wall between transmitter and receiver. The number of 31 records matched the values in Class 8 and are incorrectly estimated. Class 8 is the walking values taken between two walls.

In the top row, 240 of the records from Class 1 are correctly classified, so 100 % is the true positive rate for correctly classified points in this class, shown in the green cell in the True Positive Rate column shown in Figure 4.20.



Model 1.1

True class \ Predicted class	1	2	3	4	5	6	7	8	9	10		
1	100%										100%	
2		95%				4%		<1%			95%	5%
3			93%			4%		3%		<1%	93%	7%
4				87%	3%	<1%	10%				87%	13%
5				5%	87%	<1%	3%	5%			87%	13%
6		6%	5%	1%	<1%	75%		13%			75%	25%
7				15%	1%		83%	1%			83%	17%
8		<1%	3%	1%	5%	12%	1%	77%			77%	23%
9									99%	1%	99%	1%
10	<1%								1%	98%	98%	2%

True class

Predicted class

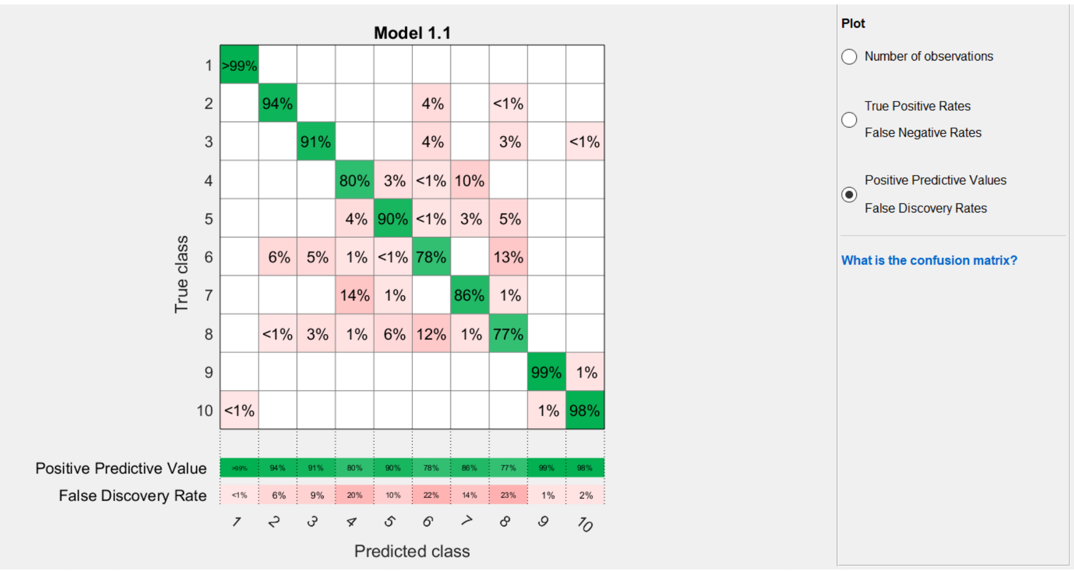
Plot

○ Number of observations

● True Positive Rates
False Negative Rates

○ Positive Predictive Values
False Discovery Rates

Figure 4.20. *True Positive rates for four feature*



Model 1.1

True class	1	2	3	4	5	6	7	8	9	10
1	>99%									
2		94%				4%		<1%		
3			91%			4%		3%		<1%
4				80%	3%	<1%	10%			
5				4%	90%	<1%	3%	5%		
6		6%	5%	1%	<1%	78%		13%		
7				14%	1%		86%	1%		
8		<1%	3%	1%	6%	12%	1%	77%		
9									99%	1%
10	<1%								1%	98%
Positive Predictive Value	>99%	94%	91%	80%	90%	78%	86%	77%	99%	98%
False Discovery Rate	<1%	6%	9%	20%	10%	22%	14%	23%	1%	2%

Predicted class

Plot

○ Number of observations

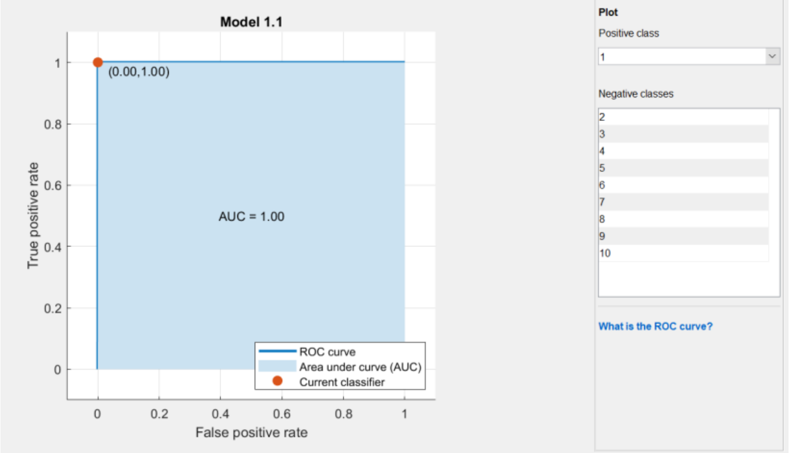
○ True Positive Rates
False Negative Rates

● Positive Predictive Values
False Discovery Rates

What is the confusion matrix?

Figure 4.21. *Positive predictive values for four feature*

The overall quality of the classifier can be found from the accuracy value in the history list, the confusion matrix value and the area under the curve in the ROC curve.



Plot

Positive class

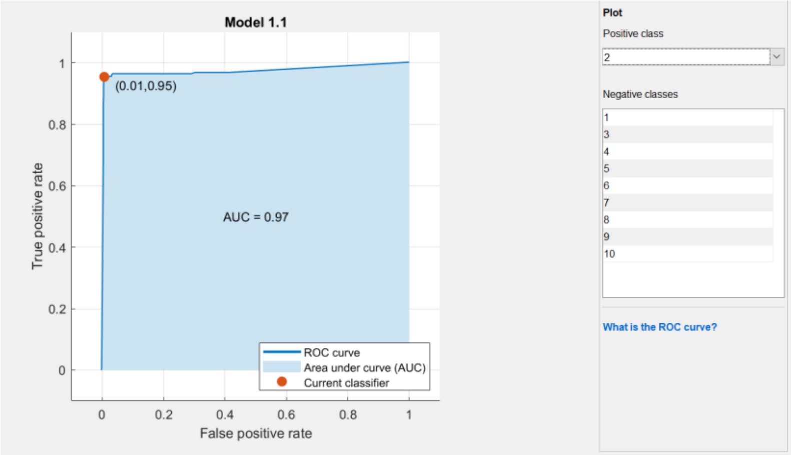
1

Negative classes

2
3
4
5
6
7
8
9
10

What is the ROC curve?

Figure 4.22. *For four features and for Class 1 - ROC curve*



Plot

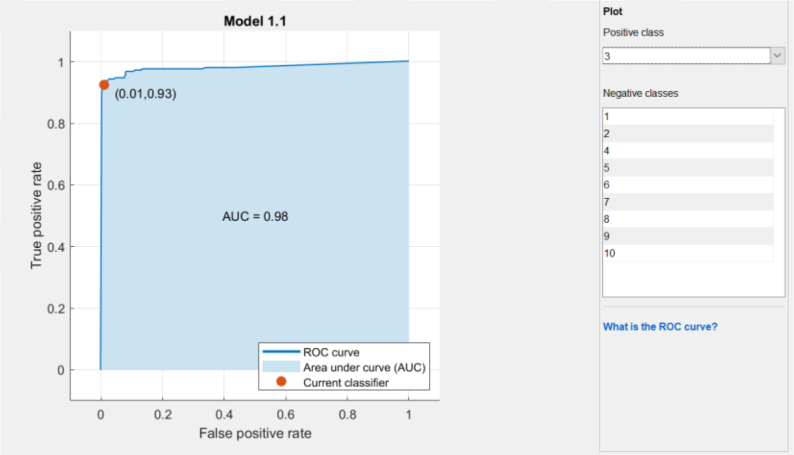
Positive class

2

Negative classes

1
3
4
5
6
7
8
9
10

Figure 4.23. *For four features and for Class 2 - ROC curve*



Plot

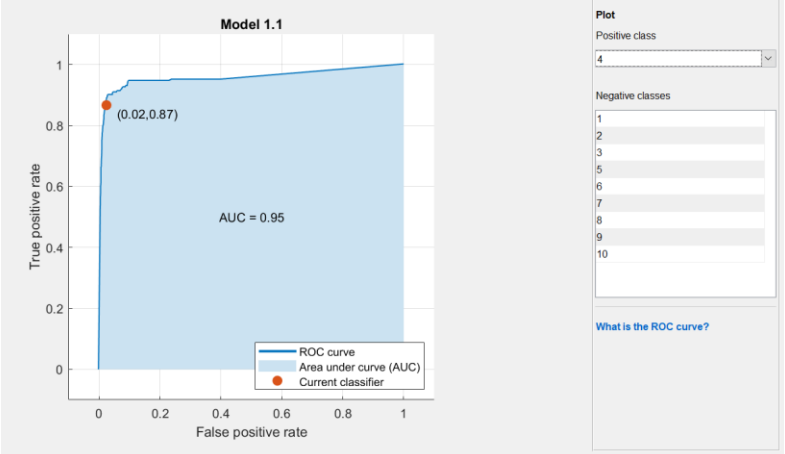
Positive class

3

Negative classes

1
2
4
5
6
7
8
9
10

Figure 4.24. *For four features and for Class 3 - ROC curve*



Plot

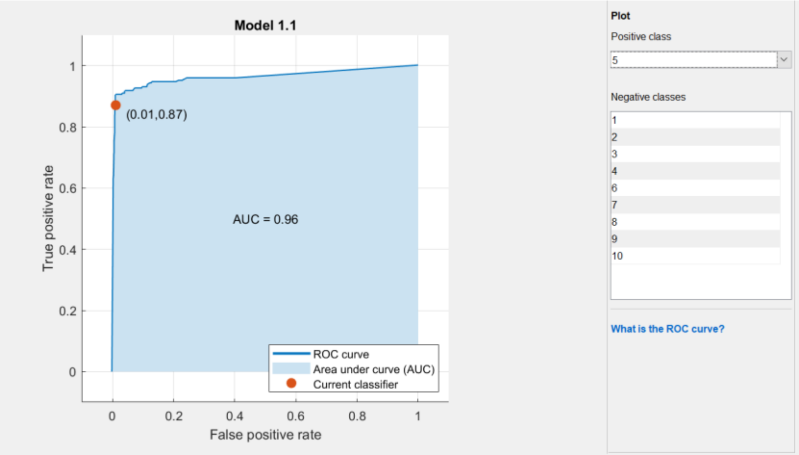
Positive class

4

Negative classes

1
2
3
5
6
7
8
9
10

Figure 4.25. *For four features and for Class 4 - ROC curve*



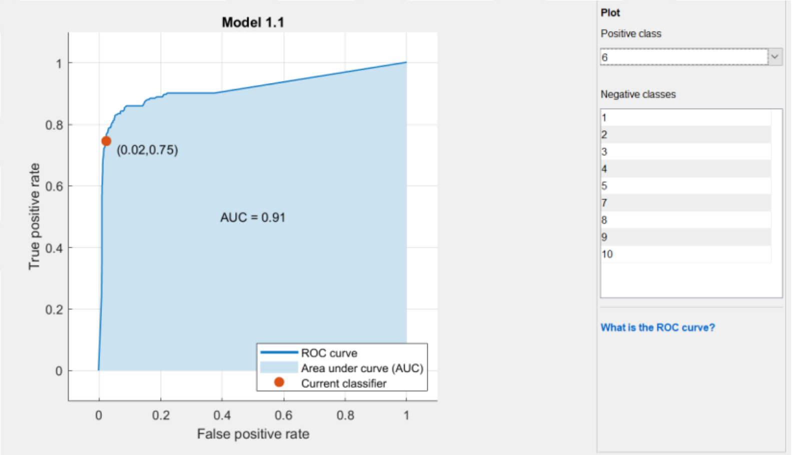
Plot

Positive class: 5

Negative classes: 1, 2, 3, 4, 6, 7, 8, 9, 10

What is the ROC curve?

Figure 4.26. *For four features and for Class 5 - ROC curve*



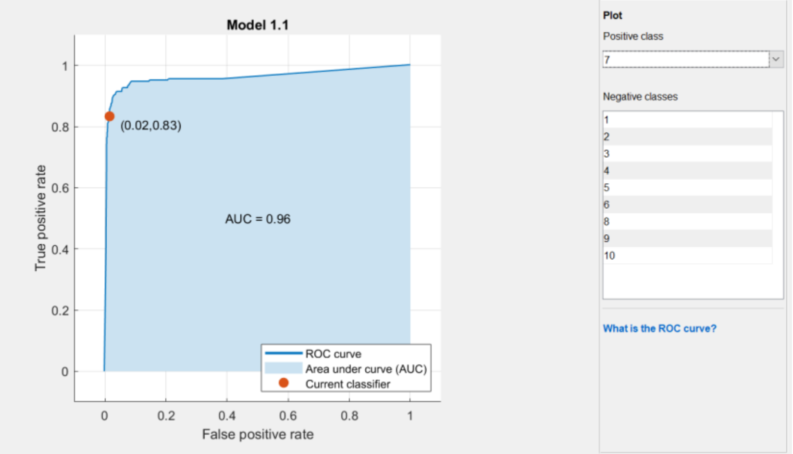
Plot

Positive class: 6

Negative classes: 1, 2, 3, 4, 5, 7, 8, 9, 10

What is the ROC curve?

Figure 4.27. *For four features and for Class 6 - ROC curve*



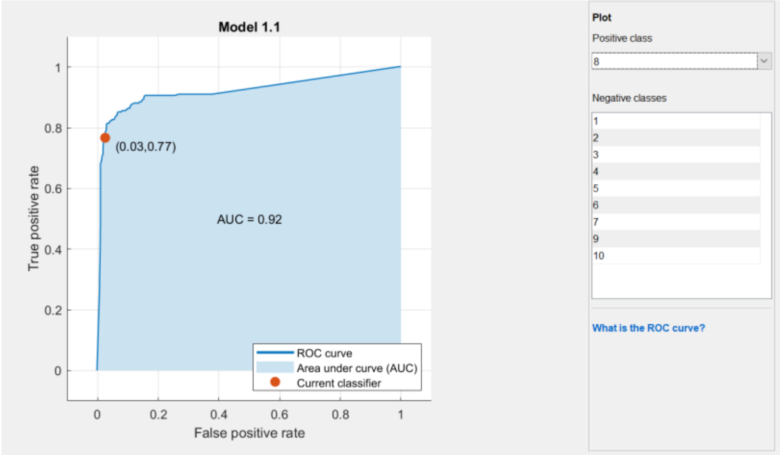
Plot

Positive class: 7

Negative classes: 1, 2, 3, 4, 5, 6, 8, 9, 10

What is the ROC curve?

Figure 4.28. *For four features and for Class 7 - ROC curve*



Plot

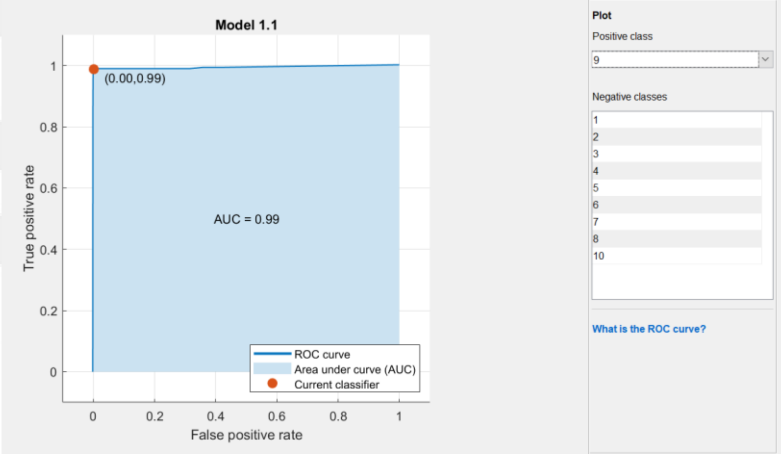
Positive class

8

Negative classes

1
2
3
4
5
6
7
9
10

Figure 4.29. *For four features and for Class 8 - ROC curve*



Plot

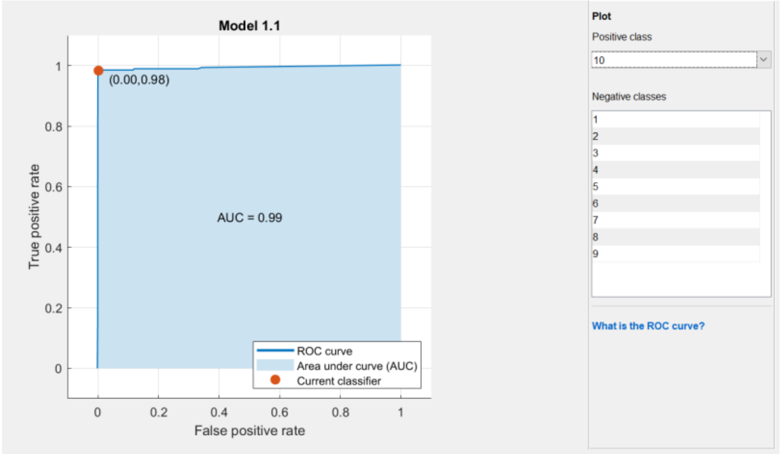
Positive class

9

Negative classes

1
2
3
4
5
6
7
8
10

Figure 4.30. *For four features and for Class 9 - ROC curve*



Plot

Positive class

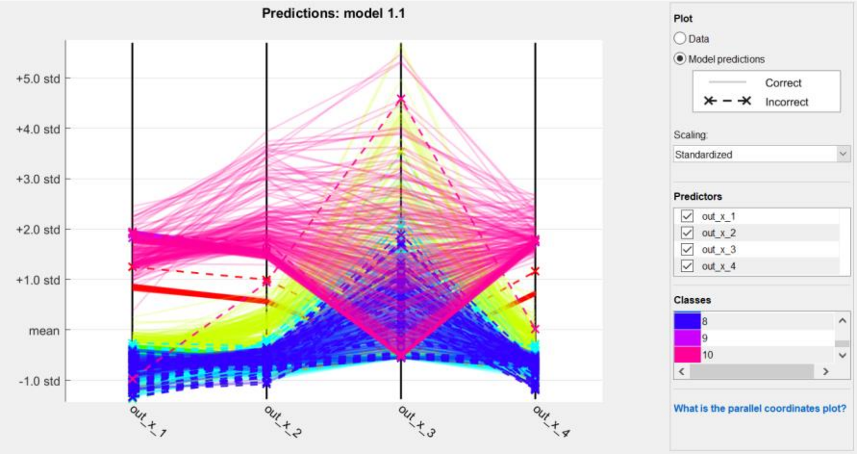
10

Negative classes

1
2
3
4
5
6
7
8
9

Figure 4.31. *For four features and for Class 10 - ROC curve*

Parallel coordinate plot can help user understand relationships between features and identify useful predictors for separating classes. The graph plotted for four features is as in Figure 4.32.



Plot
○ Data
● Model predictions

Scaling: Standardized

Predictors
☑ out_x_1
☑ out_x_2
☑ out_x_3
☑ out_x_4

Classes
8
9
10

Figure 4.32. *Parallel Coordinates Plot for four features*

When Figure 4.32 is examined, it is observed that Class 10 values (pink color) are distributed too much for third feature. However, this class can be easily estimated because the other three features make this class distinguishable.

4.3. Effects of Parallel Computing Toolbox to Accuracy

Parallel computing toolbox has no effect on accuracy. If this toolbox is used, only the necessary calculations are performed at the same time, which saves time.

5. CONCLUSION

In this thesis, the potentials of ambient radio waves are investigated for detecting the presence of a human behind wall. Constant amplitude carrier signal (900 MHz) is transmitted with limited output power and received signals are used to classify the three main different cases in the monitoring area which is empty (no living), micro movements (only breathing), and macro movements (walking etc.). The radio waves are collected using software defined radios (USRP B210) for ten different situations in home environment. Machine learning models are developed to distinguish ten different situations, different features are defined in the collected signals, trained and tested with 22 different classification techniques and the most suitable model is presented after using different combinations of the selected features.

In the system test, a real home environment with 15 cm wide brick walls was used while collecting the data and the classification was performed with Fine Gaussian SVM by 92,9 % accuracy as the best result. This ratio is reached by using 2400 data samples with 10 different classes which are collected in this environment. This accuracy ratio was considered acceptable especially for live detection behind the wall in case of earthquake. To improve this ratio it is possible to use more data from different environments in training and with multiple receivers with different line-of-sights.

REFERENCES

[1] Rittiplang, A. and Phasukkit, P. (2018). Human Movement Detection behind the Wall based on Doppler Frequency and Standard Deviation. *2018 International Symposium on Multimedia and Communication Technology (ISMAC), August 29-31, 2018, SS3-2, pp.85-88.*

[2] Fritsche, P. and Wagner, B. (2015). Comparison of Two Radar-based Scanning-techniques for the Use in Robotic Mapping. *In Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, vol 1: ICINCO, pp. 365-372.*

[3] Gunathilaka, W.M.D.R., Gunasekara, G.G.C.M., Dinesh, H.G.C.P., Narampanawe, K.M.M.W.N.B., Wijayakulasooriya, Dr. J.V. (2012). Ambient Radio Frequency Energy Harvesting. IEEE, 978-1-4673-2605-6/12/$31.00.

[4] Uysal, C. and Filik, T. (2018). MUSIC Algorithm for Respiratory Rate Estimation Using RF Signals. *Electrica, vol. 18, no: 2, pp. 300-309.*

[5] Uysal, C. and Filik T. (2019). RF-Based Noncontact Respiratory Rate Monitoring With Parametric Spectral Estimation. *IEEE Sensors Journal 19.21: 9841-9849.*

[6] KARAMZADEH, S. and KARTAL, M. (2013). Detection Improvement of Hidden Human’s Respiratory Using Remote measurement methods with UWB Radar. *ICTRS 2013 Netherlands Second International Conference on Telecommunications and Remote Sensing, pp. 104-109.*

[7] Yunze Zeng, Parth H. Pathak, Zhicheng Yang, Prasant Mohapatra (2016). Human Tracking and Activity Monitoring using 60 GHz mmWave. *IPSN '16 Proceedings of the 15th International Conference on Information Processing in Sensor Networks, Article No. 25, pp. 1-2.*

[8] http 1- https://airspy.com/download/
(Date of access: 10.10.2019)

[9] http 2- http://www.hdsdr.de/
(Date of access: 10.10.2019)

[10] http 3- https://www.sdr-radio.com/
(Date of access: 10.10.2019)

[11] http 4- http://www.sm5bsz.com/linuxdsp/linrad.htm

(Date of access: 10.10.2019)

[12] http 5- http://gqrx.dk/
(Date of access: 10.10.2019)

[13] http 6- http://www.sdrapplications.it/
(Date of access: 10.10.2019)

[14] http 7- http://www.sdrplay.com/downloads/#tab-tab-5037-0-0-2-5037-0
(Date of access: 10.10.2019)

[15] http 8- https://play.google.com/store/apps/details?id=marto.androsdr2&hl=en
(Date of access: 10.10.2019)

[16] http 9- https://play.google.com/store/apps/details?id=de.ses.wavesink&hl=en
(Date of access: 10.10.2019)

[17] http 10- https://play.google.com/store/apps/details?id=com.mantz_it.rfanalyzer
(Date of access: 10.10.2019)

[18] http 11- http://openhpsdr.org/download.php
(Date of access: 10.10.2019)

[19] http 12- https://osmocom.org/projects/sdr/wiki/sdrangelove
(Date of access: 10.10.2019)

[20] http 13- https://brmlab.cz/user/jenda/kukuruku
(Date of access: 10.10.2019)

[21] http 14- https://www.ecstaticlyrics.com/radio/sdr/natpos/
(Date of access: 10.10.2019)

[22] http 15- https://sdr-labs.com/
(Date of access: 10.10.2019)

[23] http 16- https://softsyst.com//QIRX/qirx
(Date of access: 10.10.2019)

[24] http 17- https://www.hfrelectronics.com/en/zeus-radio/
(Date of access: 10.10.2019)

[25] http 18- https://www.ettus.com/sdr-software/
(Date of access: 10.10.2019)

[26] http 19- https://en.wikipedia.org › wiki › Motion_detection

(Date of access: 19.09.2019)

[27] Liang, Q. (2010). Biologically-inspired target recognition in radar sensor networks. *EURASIP Journal on Wireless Communications and Networking 2010, Paper ID: 523435, pp.1-8.*

[28] Liang, Q., Cheng, X. and Samn, S. NEW: network-enabled electronic warfare for target recognition. *IEEE Trans on Aerospace and Electronic Systems 46(2), pp. 558–568.*

[29] Liang, Q. (2008). Automatic target recognition using waveform diversity in radar sensor networks. *Pattern Recognition Letters (Elsevier) 29(2), pp. 377–381.*

[30] Liang, Q. and Cheng, X. (2008). KUPS: knowledge-based ubiquitous and persistent sensor networks for threat assessment. *IEEE Transactions on Aerospace and Electronic Systems 44(3), pp. 1060–1069.*

[31] Liang, Q. (2006). Waveform design and diversity in radar sensor networks: theoretical analysis and application to automatic target recognition. *in IEEE Third Annual Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON2006) (Reston, 25–28 Sept 2006), pp. 684-689.*

[32] Zhang, B. and Wang, W. (2013). Through-wall detection of human being with compressed UWB radar data. *EURASIP Journal on Wireless Communications and Networking, 2013(1). doi:10.1186/1687-1499-2013-162, pp 1-7.*

[33] LE, M. (2003). Why UWB? A review of Ultra-wide-band Technology. *National Institute of Standards and Technology, Report to NETEX Project Office, DARPA 2003, pp. 1-72.*

[34] Singh, S., Liang, Q., Chen, D., Sheng, L. (2011). Sense through wall human detection using UWB radar. *EURASIP Journal on Wireless Communications and Networking, pp. 1-11.*

[35] Liang, S. D. (2015). Sense-through-Wall Human Detection Based on UWB Radar Sensors. *Signal Processing,* http://dx.doi.org/10.1016/j.sigpro.2015.09.022*, pp. 1-20.*

[36] Liang, X., Zhang, H., Fang, G., Ye, S., Gulliver, T.A. (2017). An Improved Algorithm for Through-Wall Target Detection Using Ultra-Wideband Impulse Radar. *IEEE Access, 10.1109/ACCESS.2017.2761771, pp. 22101-22118.*

[37] Gaikwad, A. N., Verulkar, U. S. and Dongre, K. S. (2016). Experimental Study and Analysis of Stepped Frequency Continuous Wave Based Radar for Through the Wall Detection of Life Signs. *2016 IEEE Region 10 Conference (TENCON) - Proceedings of the International Conference, pp.1565-1569.*

[38] http 20- https://en.wikipedia.org/wiki/Software-defined_radio
(Date of access: 19.09.2019)

[39] http 21- https://www.mathworks.com/help/stats/select-data-and-validation-for-classification-problem.html
(Date of access: 19.09.2019)

[40] http 22- https://towardsdatascience.com/model-performance-cost-functions-for-classification-models-a7b1b00ba60
(Date of access: 19.09.2019)

[41] http 23- https://www.mathworks.com/help/stats/assess-classifier-performance.html
(Date of access: 19.09.2019)