TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

NATIONAL COLLEGE OF ENGINEERING

A

PROJECT REPORT

ON

”A DEEP LEARNING BASED HUMAN ACTIVITY RECOGNITION USING WI-FI SIGNALS”

SUBMITTED BY:

ASHISH KUMAR POKHAREL (NCE077BCT006)

PRASHANT SUBEDI (NCE077BCT021)

SABINA PANDEY (NCE077BCT028)

SAKSHAM MAHARJAN (NCE077BCT029)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

LALITPUR, NEPAL

DECEMBER, 2024

Abstract

Wi-Fi Channel State Information (CSI) employed for human activity recognition minimizes the need for expensive and intrusive equipment needed for vision-based, and sensor-based methods. The LoS scenarios has been extensively studied whereas its application in nLoS environments remains under explored. This project aims to develop a WiFi-based human activity monitoring system employing Raspberry Pi as the receiver and routers as the transmitters, leveraging deep learning for multiple activity classification. Building upon the Nexmon firmware patch, the system will passively monitor CSI packets to recognize human activities through walls, employing a deep learning model to classify multiple activities based on the captured and processed CSI data. The system's performance will be evaluated using F-score, Precision, and Recall. A realistic data collection approach was implemented dividing the collection process into multiple phases. The preprocessing primarily consists of denoising the amplitude data and removing outliers, ensuring the sanitization of the phase component of the CSI data, segmented into distinct parts. HAR using nLos will help efficiently identify human activity in situations where direct line of sight is obstructed, providing practical benefits and assistance to people.

Keywords: *CSI, Deep Learning, HAR, LoS, nLoS, Raspberry Pi, Wi-Fi*

Contents

List of Figures

List of Tables

List of Abbreviations

CSI	Channel State Information
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
CuDNN	CUDA Deep Neural Network
GRU	Gated Recurrent Unit
HAR	Human Activity Recognition
IDE	Integrated Development Environment
LAN	Local Area Network
LSTM	Long Short-Term Memory
MIMO	Multiple Input Multiple Output
NIC	Network Interface Card
OFDM	Orthogonal Frequency-Division Multiplexing
RSSI	Received Signal Strength Indicator
RNN	Recurrent Neural Network
RTX	Ray Tracing Texel eXtreme
nLOS	non-Line-Of-Sight
Wi-Fi	Wireless Fidelity

1. Introduction

1.1 Background

Human Activity Recognition (HAR) involves identifying and classifying actions or activities performed by individuals at a given time, such as sleeping, running, or sitting. HAR has applications in various domains, including health monitoring, sports, smart homes, and security. HAR techniques can be broadly categorized into vision-based and sensor-based methods. Vision-based devices use cameras and computer vision algorithms to recognize human activities. While these devices can be highly effective, they also present several challenges. Monitoring using vision-based devices can raise significant privacy concerns, especially in sensitive areas like homes or workplaces. Additionally, the performance of these systems can be adversely affected by lighting conditions and weather, which can disturb the camera's view. Likewise, sensor-based devices utilize various types of sensors to collect data on human movements and activities. Despite their potential, these devices also face certain limitations. The data collected by sensors can be noisy and prone to errors, impacting the accuracy of activity recognition. Moreover, high-quality sensors and associated devices can be expensive. Wearable sensing devices may also face issues related to user compliance, as individuals may forget to wear them or find them inconvenient to use consistently.

To address these limitations, researchers are increasingly exploring device-free sensing techniques that utilize radio signals, such as Wi-Fi signals, for tracking human activities. Wi-Fi-based sensing technology has gained significant attention due to its widespread availability, versatility, and high-performance capabilities. Wi-Fi networks are already extensively deployed in residential, commercial, and public environments, providing a pervasive infrastructure that can be effectively leveraged for Human Activity Recognition (HAR) without necessitating additional hardware installations. Wi-Fi signals possess the ability to penetrate walls and other physical obstacles, facilitating activity recognition across multiple rooms and through diverse building materials. This capability enhances the robustness and applicability of Wi-Fi-based HAR systems in complex indoor settings. Furthermore, Wi-Fi sensing operates in a non-intrusive manner as it does not require individuals to wear or carry specific sensors, thereby avoiding user discomfort and enhancing acceptance. This device-free approach also eliminates challenges associated with sensor compliance and wearability, which are common in traditional wearable sensor-based systems. In contrast to vision-based HAR systems, Wi-Fi sensing mitigates privacy concerns as it does not involve capturing visual data such as images or videos. Additionally, Wi-Fi signals are relatively unaffected by variations in lighting conditions, ensuring consistent performance across different indoor environments and times of day.

Wi-Fi signals can detect human activities through a technique called Wi-Fi sensing. This process starts with the transmission of Wi-Fi signals from a router or access point. Wi-Fi signals are characterized by two key metrics: Channel State Information (CSI) and Received Signal Strength Indicator (RSSI). CSI provides detailed information about the channel properties of a Wi-Fi signal. It includes data on the amplitude and phase of the signal for each subcarrier in a multi-carrier system like OFDM (Orthogonal Frequency-Division Multiplexing), which is used in Wi-Fi. RSSI measures the overall power level of the received signal. As Wi-Fi signals travel through the environment, they interact with various objects, including human bodies. Movements cause disruptions in these signals, altering their propagation patterns. By monitoring the fluctuations in CSI and RSSI caused by human movement, advanced deep learning algorithms and models can detect and classify different activities.

1.2 Problem Statements

Advancements in human activity recognition have evolved from traditional signal processing and machine learning methods to recent applications of deep learning. Historically, these tasks require close proximity to a router and were significantly influenced by reflections from objects within a confined space. Although recent research has begun to explore the potential of sensing human activities through walls, this area remains undiscovered with substantial developments yet to be made. A persistent challenge in this domain is the accurate recognition of activities involving multiple individuals through walls, primarily due to the overlapping and attenuation of signals.

1.3 Objectives

The main objectives of the project are:

- To explore human motion sensing through a wall using Wi-Fi signals.
- To utilize Raspberry Pi and Nexmon Firmware to monitor CSI data.
- To design a system capable of collecting data through walls using wider-angle WiFi sensing.
- To implement a deep learning network, Long-Short Term Memory (LSTM) to recognize the activities.

1.4 Scope

In healthcare, the WiFi-based human activity recognition (HAR) system significantly aids elderly care and patient monitoring. It non-intrusively detects falls or inactivity, alerting caregivers or medical professionals, especially in remote areas with limited healthcare access. Hospitals can use it for continuous patient monitoring, ensuring safety without constant physical checks. For home automation, the HAR system enhances activity-based automation and energy management. It detects room occupancy, automatically adjusting lighting, heating, and cooling to optimize energy use and reduce costs. This is particularly valuable in Nepal, where energy resources can be scarce and expensive, making homes smarter and more efficient.

2. Literature Review

2.1 Related Work

Human Activity Recognition (HAR) has emerged as a promising technique to detect and understand user actions, enabling computing systems to provide proactive assistance. Abdelnasser et al. introduced WiGest, a system that employs Received Signal Strength (RSS) measurements for gesture recognition through feature extraction, gesture recognition, and motion mapping, although its effectiveness was limited by the instability and variability of RSS due to multi-path and fading effects [1]. Initial studies demonstrated the feasibility of using CSI for detecting basic activities such as standing, walking, and falling by analyzing changes in the wireless signal's amplitude and phase[2]. Wang et al. proposed a deep learning model called DeepFi, which employs autoencoders to learn features from CSI data for indoor fingerprinting[3]. Overcoming some of the limitations of RSS-based methods, Wang et al. developed a system that leverages CSI for fine-grained human activity recognition by capturing detailed physical layer information leveraging deep learning[4].

A CSI-based human activity recognition method using deep learning was proposed by Moshiri et al. They collected CSI data for seven daily human activities using the Nexmon tool on a Raspberry Pi 4 with a TP-Link Archer C20 Access Point in IEEE 802.11ac standard which was named the CSI-HAR-Dataset [5]. A hybrid deep learning framework called CNN-GRU-AttNet was developed to enhance HAR by combining Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) with an attention mechanism. This model, designed by Mekruksavanich et al., achieved a 4.62% improvement in accuracy on benchmark datasets by effectively extracting spatial-temporal features from raw CSI data [6]. Focusing on the challenge of recognizing human activities through walls using WiFi CSI, Abuhoureyah et al. employed an LSTM-based deep learning algorithm. Their system utilized a Multiple-Input Multiple-Output (MIMO) antenna setup and achieved up to 90.5% accuracy, demonstrating the potential of WiFi CSI for non-line-of-sight (nLoS) HAR by addressing signal attenuation and interference [7]. A human activity recognition method was introduced by Wong et al. [8] where only Raspberry Pi was used without any high gain antennas to classify human activities through a wall. Despite these advancements, a notable limitation in all these studies is the inability to accurately capture and differentiate user activities through walls, as current models primarily address user scenarios in los. Future research should aim to develop models capable of user activity recognition through complex barriers, potentially integrating additional sensing modalities and specialized algorithms to enhance robustness and accuracy.

2.2 Related Theory

Wi-Fi: Wireless Fidelity (Wi-Fi) is a wireless networking technology based on the IEEE 802.11 standards, enabling devices to communicate over short distances using radio waves. In 1971, Wi-Fi technology was first introduced in Hawaii with the creation of ALOHAnet, an innovative wireless Ultra High Frequency (UHF) packet network that connected the islands. Protocols known as WaveLAN were created in 1991 by NCR and AT&T which served as IEEE 802.11 standards which governs wireless networking. It defines how wireless devices communicate between devices within a local area network (LAN). Wi-Fi operates within the electromagnetic spectrum, mainly 2.4 GHz and 5 GHz frequency bands. The 2.4GHz band is more common and offers better coverage but can be crowded as even Bluetooth signals utilize this frequency band, which leads to potential interference from other devices. The 5 GHz band provides faster data rates and is less congested in comparison but has a shorter range. The strength and quality of a Wi-Fi signal can be affected by factors including distance from the router, physical obstacles and interference from electronic devices, and signal attenuation due to materials like walls and furniture. Access points, routers, and extenders are devices that facilitate the transmission and reception of Wi-Fi signals. Those devices allow creating of wireless networks that enable connectivity for a wide range of devices such as smartphones, laptops, smart TVs, and IoT devices.

CSI: Wi-FI signal can be described by two characteristics Received Signal Strength (RSS) and Channel State Information (CSI). CSI refers to the knowledge of the characteristics and conditions of a communication channel between a transmitter and a receiver in a wireless communication system. It encompasses various parameters such as attenuation, noise, fading, and interference that affect the quality and reliability of the communication link. It also includes measurements like phase, magnitude, and frequency response, offering insights into signal propagation, multi-path effects, and environmental influences. RSS only measures signal strength whereas CSI enables a more precise understanding of the channel characteristics and captures the nuances of signal behavior in different environments making it a better choice for sophisticated wireless communication and sensing applications.

Nexmon Firmware: Nexmon is a C-based firmware patching framework for Broadcom/Cypress Wi-Fi chips that enables users to write their own firmware patches. It enables users to modify the firmware of Broadcom Wi-Fi chips to extend capabilities or exploit vulnerabilities. It allows enabling features like monitor mode and packet injection on chips lacking native support for identifying vulnerabilities, detecting rogue access points, and analyzing network traffic for potential threats. Nexmon tool can be used to identify malicious activities in the network, reconstruct events leading to the incident, CSI extraction and to access debugging features. It bestows augmented functionalities, enabling access and manipulation of low-level WiFi parameters as well.

Raspberry Pi: Raspberry Pi is a small, single-board computer developed by the Raspberry Pi Foundation in the United Kingdom. It is designed to be low-cost, versatile, and portable for learning, experimentation, and various other projects. RPi features a Broadcom system-on-chip with an ARM-based processor, along with RAM, storage via microSD card, and various port peripherals. It runs on Raspberry Pi OS or Raspbian OS but is capable of other OS like Ubuntu, Windows, and other Linux distributions. It is designed with GPIO pins for connecting external devices and sensors, enabling users to create custom electronics projects and IoT applications. It is equipped with Ethernet and Wi-Fi capabilities which can be used with tools like Nexmon to extract CSI signals and monitor the changes in it.

NIC: A Network Interface Card (NIC) is a hardware component that enables a device to connect to a network, supporting both wired and wireless connections. It provides a physical interface between the device and network medium where NIC assigns a unique address to a computer allowing it to send and receive data between the computer and other connected devices. It is available in various forms including expansion cards for desktops, integrated components on motherboards, and external adapters for laptops. In some models of a Raspberry Pi, in absence of NIC additional USB Wi-Fi adapters or Ethernet expansion cards can be connected. NIC's compatibility with software tools like Nexmon ensures it can support the advanced features needed for signal processing.

OFDM: Orthogonal Frequency Division Multiplexing (OFDM) is a wireless modulation technique employed in WiFi and LTE-modulated signals, enabling the simultaneous transmission of multiple frequency signals. It operates by dividing the available frequency spectrum into multiple closely spaced subcarriers, each carrying a portion of data. These carriers are orthogonal to each other and do not interfere when closely packed together which enables transmission of data symbols across subcarriers which allows for high spectral efficiency. At the receiver, the signal is processed which enables demodulation of individual subcarriers. These abilities to utilize spectrum, resist interference, and support high-speed data transmission make it important in modern wireless standards.

MIMO: Multiple Input Multiple Output (MIMO) is a technology used in wireless communication systems where multiple antennas are used at both transmitter and receiver to improve communication performance. In MIMO, multiple data streams are transmitted simultaneously using the same frequency band, but each stream is assigned different antennas. It takes advantage of multiparty propagation, which occurs when signals travel from transmitter to receiver via multiple paths, bouncing off objects and surfaces in the environment, exploiting these signals to increase data rates. It allows for better signal reception and improves coverage, extending the range of Wi-Fi networks and enhancing connectivity in challenging environments. It overcomes signal attenuation and degradation.

LSTM: LSTM (Long-Short Term Memory) is a type of recurrent neural network (RNN) architecture that is designed to address the vanishing gradient problem. The vanishing gradient problem is a common occurrence when the gradients of the loss function becomes extremely small as they are backpropagated through time. It leads to slow and ineffective training of the neural network. LSTM introduces a previous hidden state as well as a previous memory state which incorporates to both long and short term memory solving the vanishing gradient problem in RNN. LSTM networks are composed of LSTM cells which have more complex structures where each cell consists of three main components:

1. **Forget gate:** It decides on the information that should be discarded or forgotten from the cell state. It depends on the input which includes the previous hidden state and the current input. The forget gate produces values at range of 0 to 1 using a sigmoid function.

2. **Input gate:** Input gate decides on the new information that should be stored in the cell state taking the same input as the forget gate and utilizes the sigmoid function for output value range. The output value from the input gate and a tanh function leads to the production of a candidate cell state which contains the new candidate values to be added to the cell state.

3. **Output gate:** It decides on the information that should be produced as output from the cell state. It takes both the previous hidden state and the current input state as input and produces output ranging from 0 to 1 using the sigmoid function as well. The output of the output gate produces the updated hidden state with the tan function as well as the LSTM cell output.

3. Proposed Methodology

3.1 Feasibility Study

A significant outcome of a prior investigation is the system seems feasible. To examine this, we have used various feasibility tests to identify if the system is feasible or not. Some aspects to determine the feasibility are:

Technical Feasibility The project is a complete application. The main technologies and tools that will be associated with the system are Pytorch library, TensorFlow library, and Nexmon software. These components will be integrated in GUI.

Operational Feasibility: The main purpose of the system is to develop a application that will facilitate the users to know about human activity. The system will have easy access to its attributes with minimal knowledge of the system and the end users will be able to operate to its full capability. So, it is clear that the system is operationally feasible.

Economic Feasibility: From an economic standpoint, the system's development cost is solely based on the cost of WiFi cards and antennas as the Raspberry Pi and Router utilized during the development process are available beforehand. The software used for development is free to use and readily available to developers, making the system economically feasible.

Schedule Feasibility: The goals and principles guiding the development of the system are widely understood and can be accomplished within the given timeframe. Strict deadlines often lead to more efficient learning. Therefore, by honoring the tight schedule and keeping the project objectives in mind, it is anticipated that the system will be completed within the designated timeline.

Legal Feasibility: The legal requirements for using such technology can vary significantly depending on whether it is used in private residences or public spaces. Surveillance in public areas is often subject to stricter regulations and oversight.

3.2 Requirement Analysis

3.2.1 Functional Requirements

- Users should be able to detect human activity through a wall.
- The system should be able to capture the relevant CSI data.
- The system should be able to remove noise from raw CSI data.
- The system should indicate the accuracy and confidence level of the classified activity.
- The system should be able to handle the variation in human height and body structure.
- The system should constantly monitor human activities.

3.2.2 Non-Functional Requirements

Performance Requirements

- The system should be responsive and provide quick results to users to ensure a smooth user experience.
- The system should handle multiple raw data efficiently.

Standard Compliance

- The system should have an intuitive and user-friendly interface that is easy to navigate, making it accessible to users of varying technical expertise.

Availability

- The system should be available for use at all times, with minimal downtime for maintenance or upgrades.

Flexibility

- We should be able to add new features and updates even after the system is developed or is in use.

3.2.3 Technical Requirements

Software Requirements

- Nexmon firmware patching framework.
- Deep learning framework: TensorFlow or PyTorch frameworks provide the necessary tools and libraries for building and training LSTM models.
- Python and related libraries: Python is the most widely used programming language for deep learning and machine learning tasks. (NumPy, Pandas, Matplotlib)
- Integrated Development Environment (IDE): An IDE like Visual Studio Code, or Jupyter Notebook can enhance productivity and ease of development.
- Version control system: Git or another version control system is recommended for tracking changes and collaborating on the project.

Hardware Requirements

- Computer/Laptop with :
  - CPU: Intel core i5 (12 gen or newer) or equivalent AMD Ryzen (4000 or newer).
  - RAM: 8 GB or above.
  - GPU (Training): Nvidia Geforce RTX 3050 or above with CUDA and CuDNN.
- Raspberry Pi B3+/B4/5 with bcm43455c0 WiFi chip and Raspberry Pi OS.
- An external WiFi card with high-gain antennas.

3.3 Data Collection

The Raspberry Pi, acting as the central hub, facilitates the data collection process. To generate traffic for packet transmission, the 'ping' command is used to target the router's IP address. Equipped with the Nexmon firmware, the Raspberry Pi monitors network traffic, extracting CSI data. The collected data is gathered from various environments and participants to ensure the generalizability of the dataset. To capture a diverse range of data, multiple activities (standing, running, empty, walking and fall) are performed repeatedly.

3.4 Proposed System Design

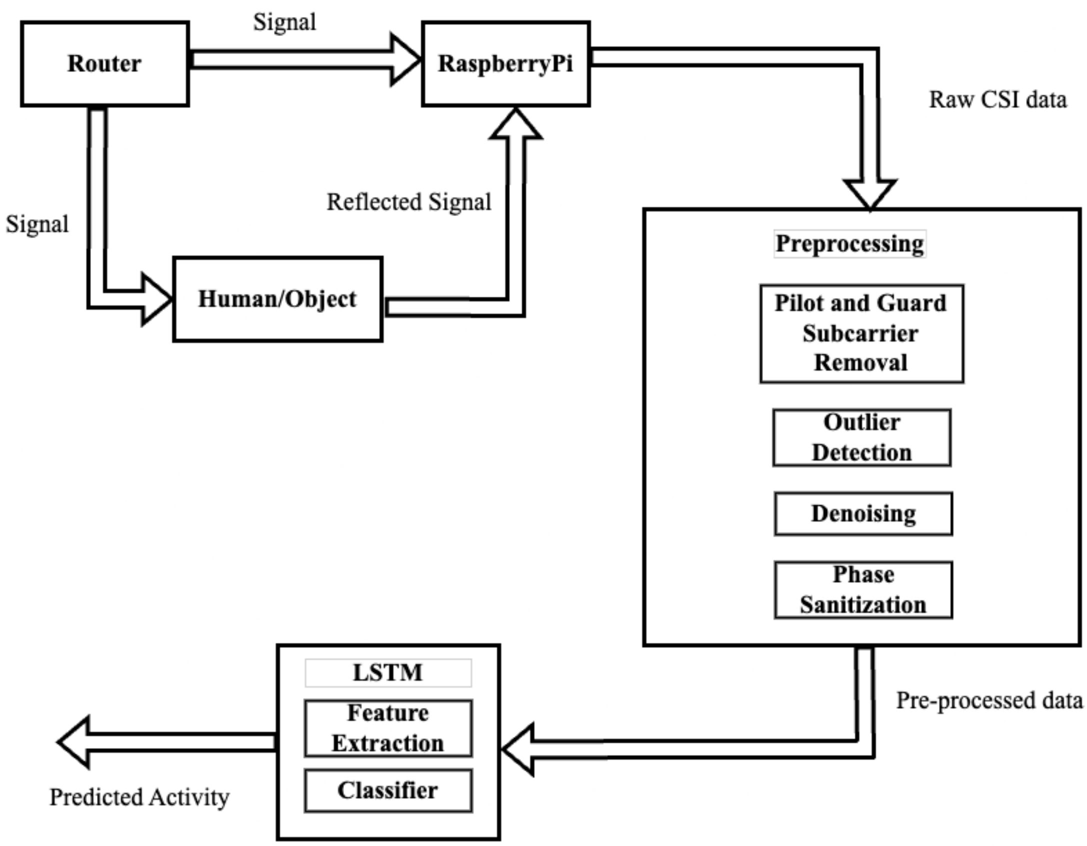

Figure 3.1: Proposed Diagram of System

The router is used to provide a wireless communication network. The Raspberry Pi connected to the network sends packets to the router enabling monitoring and collection of Channel State Information (CSI) data from the reflected signals. The raw CSI data are preprocessed which includes outlier detection, and denoising techniques. The preprocessed CSI data are passed to the LSTM layer, composed of feature extraction and classification layers. The feature extraction layer extracts the relevant features from the preprocessed data and then the classification layer classifies the signal capturing the human activity into the seven defined activities.

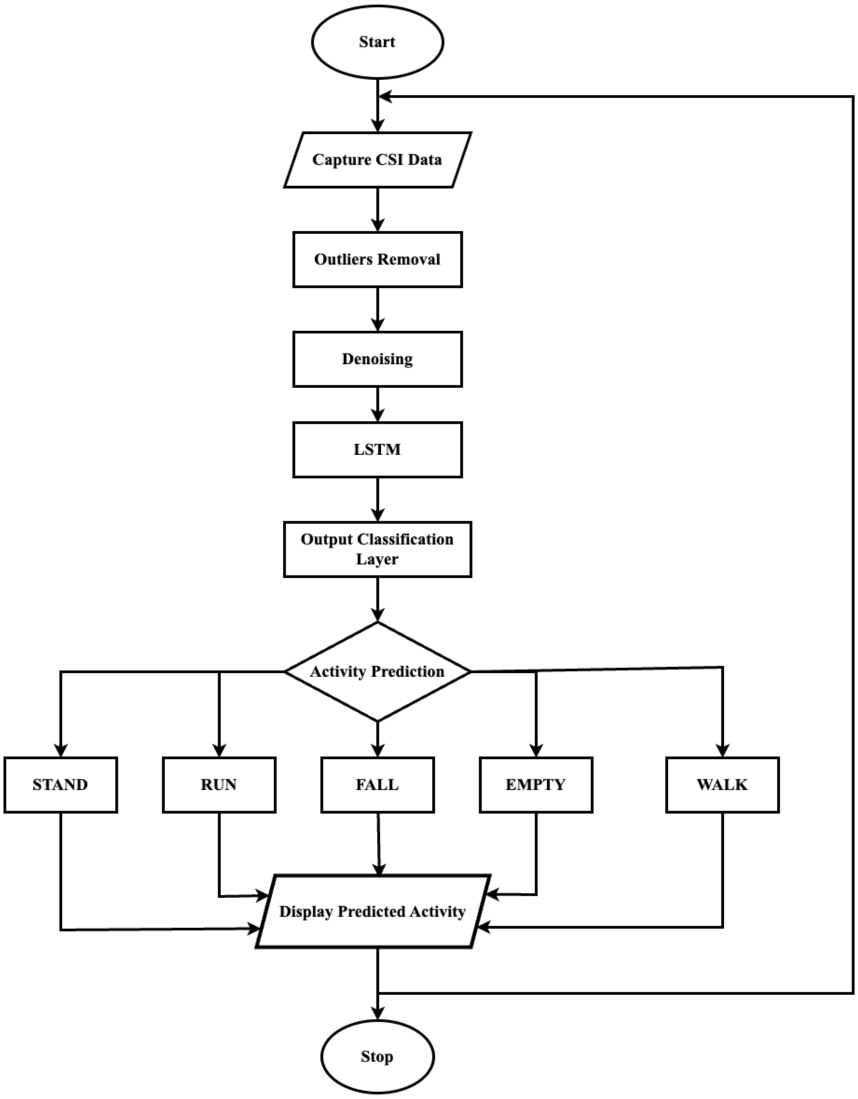

Figure 3.2: Proposed Flowchart of System

The WiFi router propagates signals, which is reflected by various objects, including humans. A Raspberry Pi, running Nexmon CSI firmware captures these reflected signals from multiple individuals. The Raspberry Pi extracts raw CSI data from these reflections and then stores it in a file through multiple repetitions.

The collected data undergoes preprocessing, involving the removal of null and pilot subcarriers, outlier removal, denoising, and phase sanitization. After preprocessing, the CSI data will be divided into training and testing sets. The training set will be used to train a Long Short-Term Memory (LSTM) architecture, which will process the time series data to extract relevant features for multiple individuals. The LSTM model will then be trained to recognize various human activities.

Once the model is trained, the test set will be used to evaluate the model's performance based on multiple parameters. After the development phase, all components, including the model and preprocessing blocks, will be integrated into a compact application. This application will continuously capture CSI data and predict multiple human activities.

3.5 Algorithm

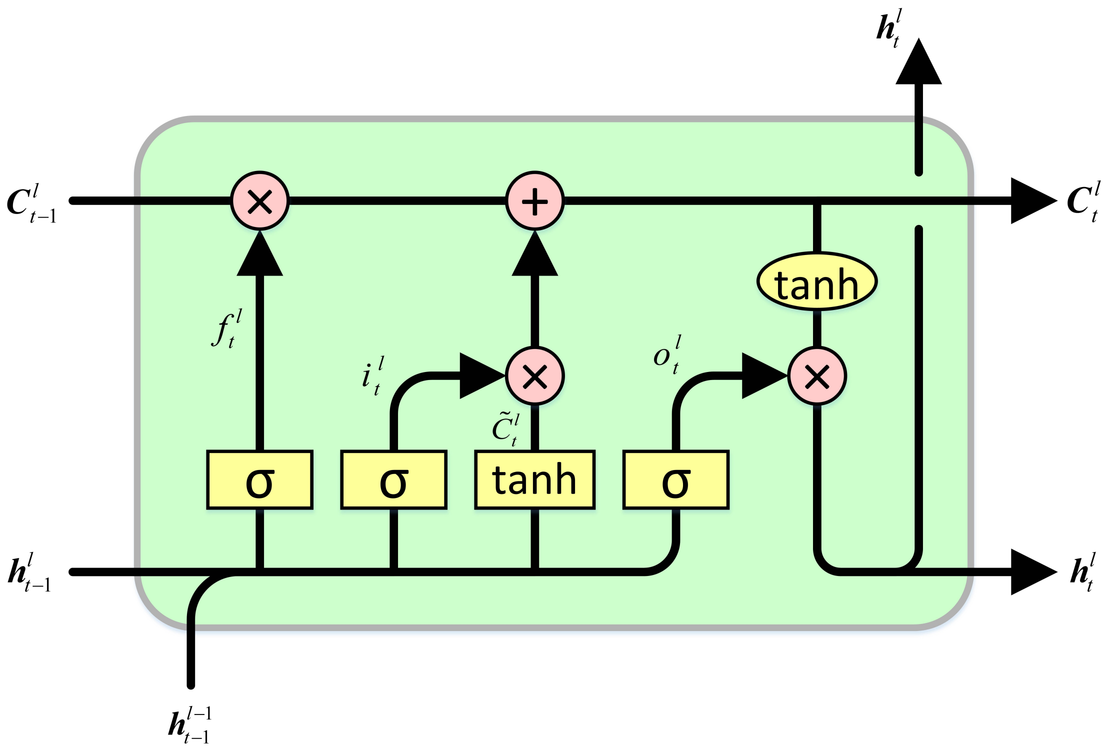

Figure 3.3: LSTM Architecture.Adapted from[9]

The input consists of current time step input Xt and hidden state from previous time step h_{t-1}. For each LSTM cell, this condition stays true where each LSTM cell is connected in a sequence and the input of current time step x_t is different for each cell.

C_{t-1} acts as a long-term memory state which allows the LSTM cell to retain information for a long time. The ht-1 hidden state of the previous time step acts as a short-time memory state. The forget gate ft decides which information to forget and retain from the previous time step ($f_t \odot C_{t-1}$). $\tilde{C}_t$ is a candidate value that contains potential information to add to the memory cell C_t. The input gate decides if the candidate value should be added to the memory cell or not.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.1}$$

The candidate value $\tilde{C}_t$ gives a value range from -1 to 1 as it is a tank function. All the gates being a sigmoid function results in values ranging from 0 to 1.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.2}$$

The output gate decides if the information of the memory cell of the current time step C_t should be added to the hidden state of the current time step. When the current hidden layer is passed to a softmax layer it generates the final output.

$$h_t = o_t \odot \tanh(C_t) \tag{3.3}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.6}$$

Symbol	**Description**	**Symbol**	**Description**
x_t	Input at time step t	W_f	Weight matrix for the forget gate
h_{t-1}	Hidden state from the previous time step	W_i	Weight matrix for the input gate
C_{t-1}	Cell state from the previous time step	W_C	Weight matrix for the candidate memory cell
f_t	Forget gate activation at time step t	W_o	Weight matrix for the output gate
i_t	Input gate activation at time step t	b_f	Bias vector for the forget gate
$\tilde{C}_t$	Candidate memory cell at time step t	b_i	Bias vector for the input gate
C_t	Cell state at time step t	b_C	Bias vector for the candidate memory cell
o_t	Output gate activation at time step t	b_o	Bias vector for the output gate
h_t	Hidden state at time step t	σ	Sigmoid activation function
$\odot$	Element-wise (Hadamard) multiplication	$\tanh$	Hyperbolic tangent activation function

Table 3.1: LSTM Cell Symbols and Descriptions

3.6 Model Testing

To evaluate the performance of our LSTM model for the classification task, we consider the following metrics: Accuracy, Precision, Recall, F1 Score, and the Confusion Matrix.

Confusion Matrix The confusion matrix is a table that is often used to describe the performance of a classification model. It contains the actual and predicted classifications done by the model.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 3.2: Confusion Matrix

Accuracy: Accuracy is the ratio of correctly predicted instances (both true positives and true negatives) to the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7}$$

Precision: Precision is the ratio of true positive predictions to the total number of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.8}$$

Recall: Recall (Sensitivity) is the ratio of true positive entities predicted correctly by the model to the total number of true entities in the dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.9}$$

F1 Score: F1 Score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.10}$$

These metrics provide a comprehensive evaluation of our LSTM model's performance on the classification task.

4. Task Accomplished

The tasks accomplished till the mid-term are:

1. The collected hardware requirements were:

    - Router: Mikrotik SXT2 is used to work as an access point. It has a high-range 10dbi directional antenna having a 60-degree range with a 2.4GHz frequency band, PoE powering, and 802.11b/g/n support.
    - Acquired a Raspberry Pi 4B to use as a receiver.

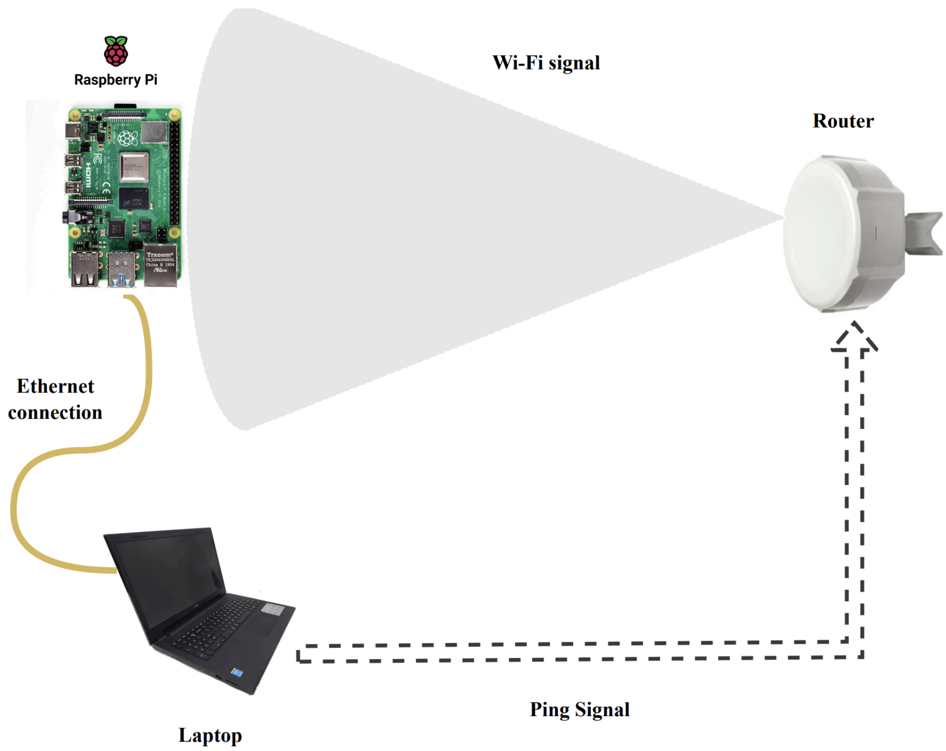

Figure 4.1: Hardware Connection.

2. **Configuration:** The Raspberry Pi was configured with the Nexmon Firmware patch to work as a receiver in monitor mode for CSI data collection. As the Nexmon firmware patch only works for limited wifi chips (BCM43455c0 for rpi4b) and a specific kernel version (5.10.92 for Rpi) is required, a fresh install of Raspbian Buster Lite 2022-01-28 OS was first done.

After installation makecsiparams was used to configure the extractor with the required monitoring channel and bandwidth. Then the wlan0 (wifi) interface was reserved for mon0 (monitor) mode where CSI data was collected by listening on port 5500 for UDP packets.

The router was configured to act as an access point to emit wifi signals in a 2.4GHz band with 802.11n protocol. The channel was changed relatively to the nearby wifi access point to avoid noise during data collection.

3. **Visualization and storage code:** A TCP/IP server was written for proper storage and visualization of the CSI data as no proper tool was found as per our requirements. The Raspberry Pi worked as the server and the laptop worked as a client.

   As Rpi leverages Nexmon to capture UDP packets by listening on port 5500 we forwarded the captured data to the TCP/IP server which transfers the packets to the laptop through port 5501 where the client code was run which separated the CSI data from the packets and amplitude vs time to subcarrier plot is done in real-time.

   For the two devices to work as client and server they have to be in the same network but as Nexmon is reserving the wifi card for monitoring CSI data, another alternative was used. The Rpi and laptop both were connected using an ethernet cable and a static IP address was assigned for the eth0 interface which made a small network between them. In raspberry pi the MAC address of the router is set during setup , so that the CSI data collected would be of the designated router.

4. **Data collection:** This is an ongoing process, where data were collected in various phases.

   Phase 1: Data set was collected for line of sight setup which was essential for the confirmation of proper working of the hardware and also for the understanding of data format. The data was collected for 5 individuals for activities empty, fall, running, standing, walking.

   These data were taken at a distance of 3m from sender to receiver, where the activities were done for 6 minute each. The data then was stored in 'pcap' format.

Figure 4.2: Phase 1 LoS setup.

(a) Transmitter.

(b) Receiver.

Figure 4.3: Transmitter and Receiver for LoS setup.

Phase 2: The dataset was collected for a non-line-of-sight configuration where data was taken in 'pcap' format. Data for three individuals was taken where the activities were 'run, stand, walk, fall and empty'. The distance between Tx and Rx was 5m.

Data was collected over a 6-minute period for each activity, performed continuously. The data for each activity was saved in separate 'pcap' files, capturing approximately 2500 packets per activity. An annotation file was then created to label the activities for each timestamp. The collected data has RSSI of around -45 dbi.

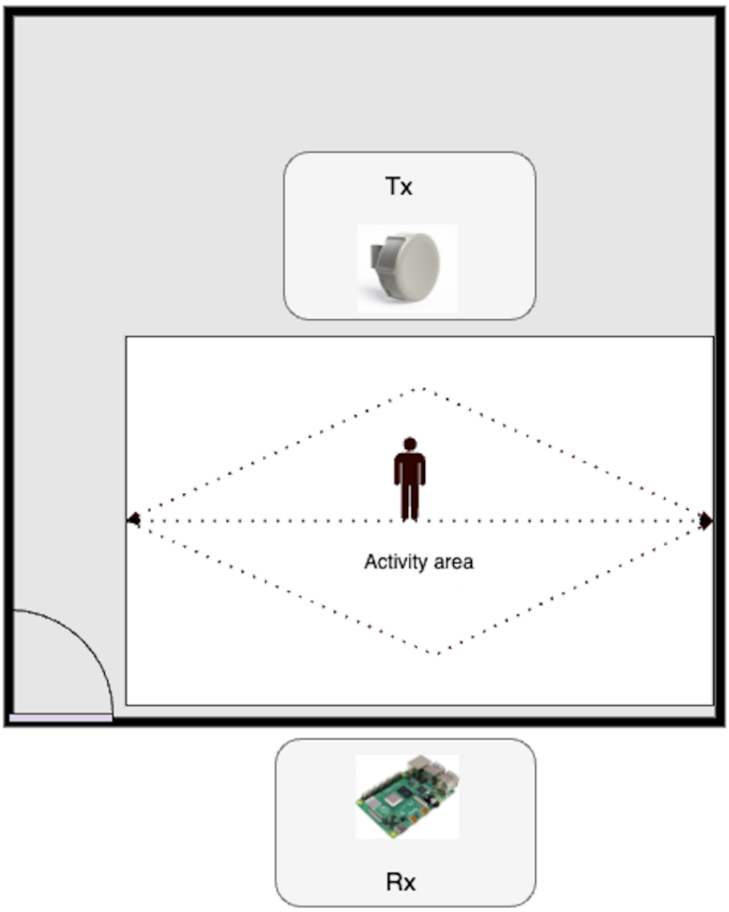

Figure 4.4: 2D representaion of Phase 2 nLoS setup.

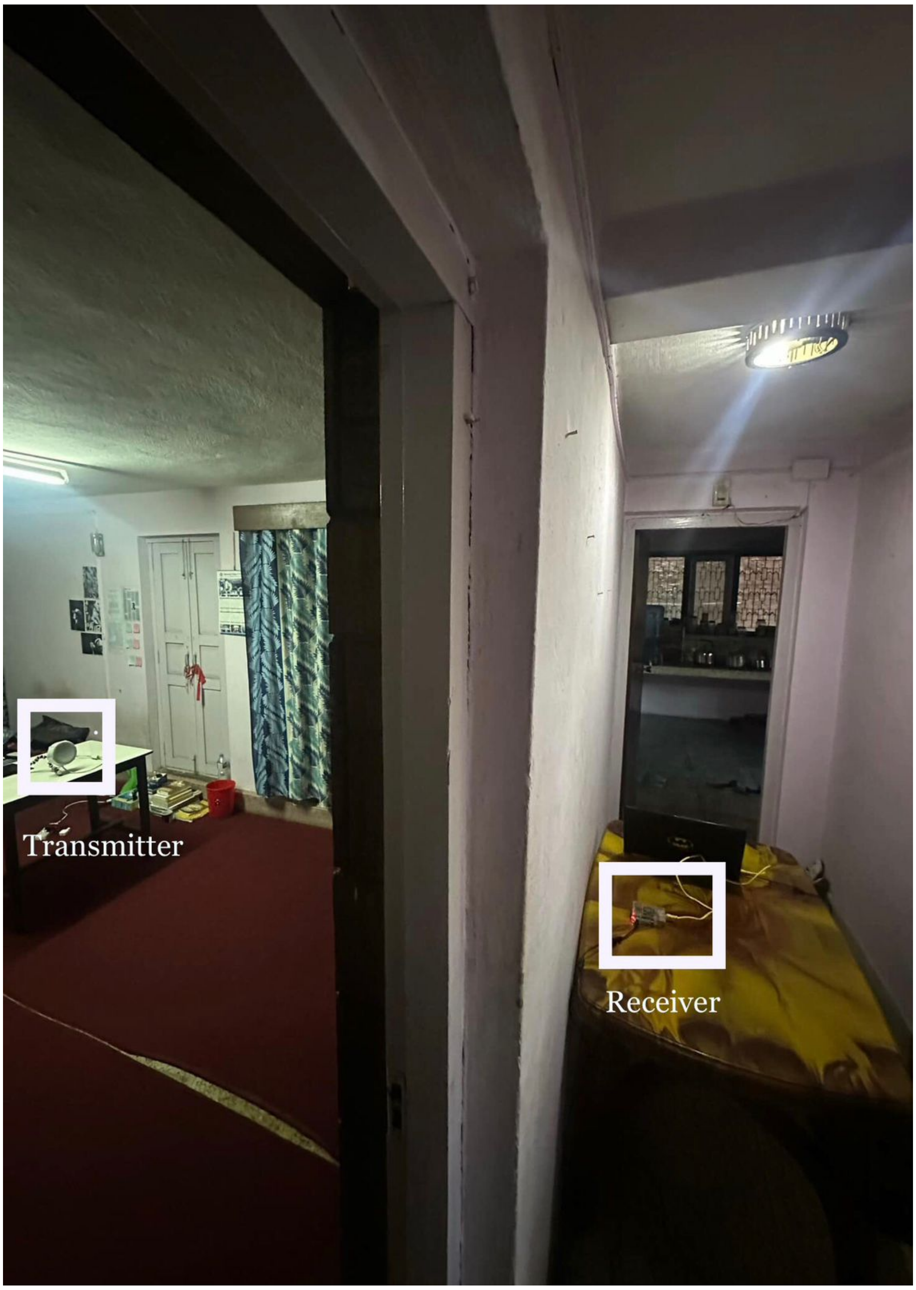

Figure 4.5: Phase 2 nLoS setup.

5. **Preprocessing:** Preprocessing of phase 2 data was done in parallel to the model development phase where the data from phase 1 was passed through various filters and fit into a LSTM model.

   Subcarriers Removal: The raw CSI data has 64 subcarriers ranging from 0-63, where 52 are data subcarriers (4-10, 12-24, 24-31, 33-52, 52-60) which carry actual data, pilot subcarriers (11, 25, 39, 53) which assist in channel estimation and synchronization where the receiver estimates the channel characteristics, guard subcarriers (0-3, 61-63) which minimize interference between adjacent channels and direct current subcarrier (32) which represent the center frequency of OFDM spectrum. The CSI data is contained in data subcarriers among these subcarriers, so the remaining must be removed. So the first step was to remove all subcarriers except the data subcarriers.

Outliers Removal: For this step, a Hample filter minimized the noise contained in amplitude and phases caused by the transition rate, power adaptations, and many others.

Denoising: For denoising, a Discrete Wavelet Transform was done which transforms the noisy signal into the wavelet domain, where noise and signal components can be more easily distinguished.

Phase Sanitization: The challenge lies in the impact of carrier frequency offset (CFO) and sampling frequency offset (SFO) on phase. CFO occurs when the transmitter and receiver are not perfectly synchronized in timing and phase before transmitting a packet. SFO arises from the analog-to-digital converter and varies by subcarrier, resulting in different errors for each subcarrier. These problems may cause difficulty in distinguishing the movements. So to solve this phase sanitation must be done which is a type of linear transformation.

The figures below show the comparison between raw amplitude 4.6, 4.8 and amplitude after preprocessing 4.7, 4.9 for various activities.

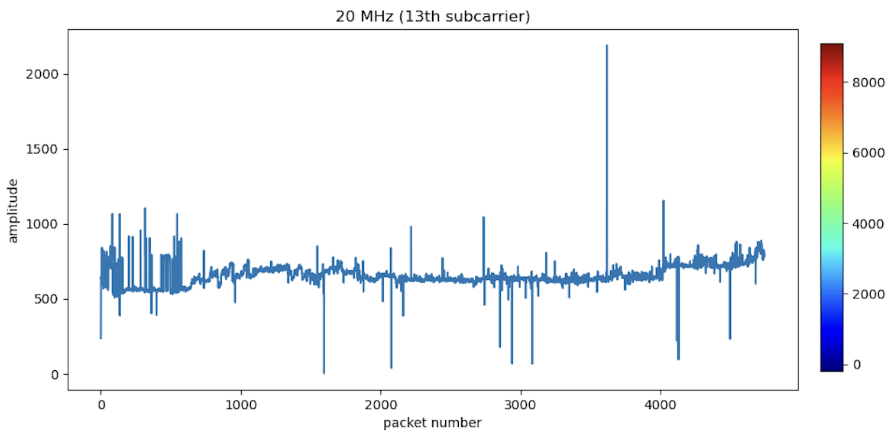

Figure 4.6: Raw amplitude for empty in nLos setup.

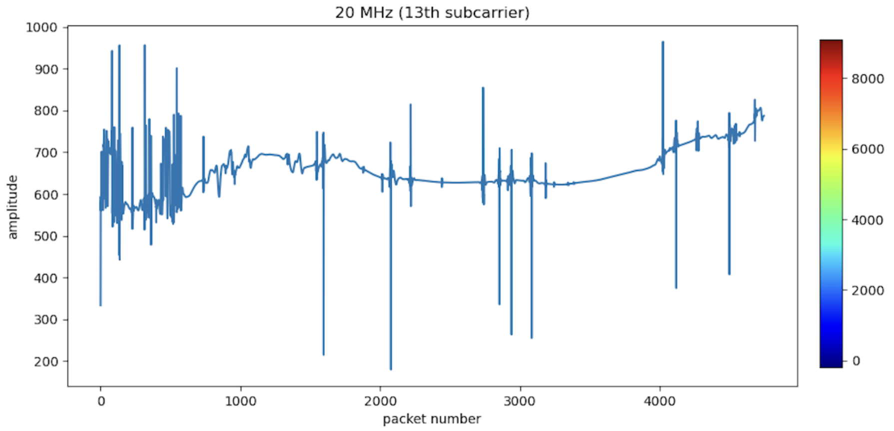

Figure 4.7: Preprocessed amplitude for empty in nLos setup.

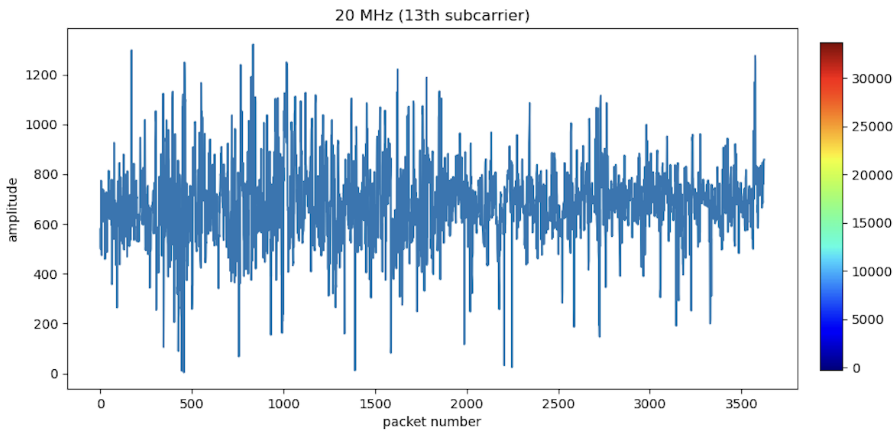

Figure 4.8: Raw amplitude for walk in nLos setup.

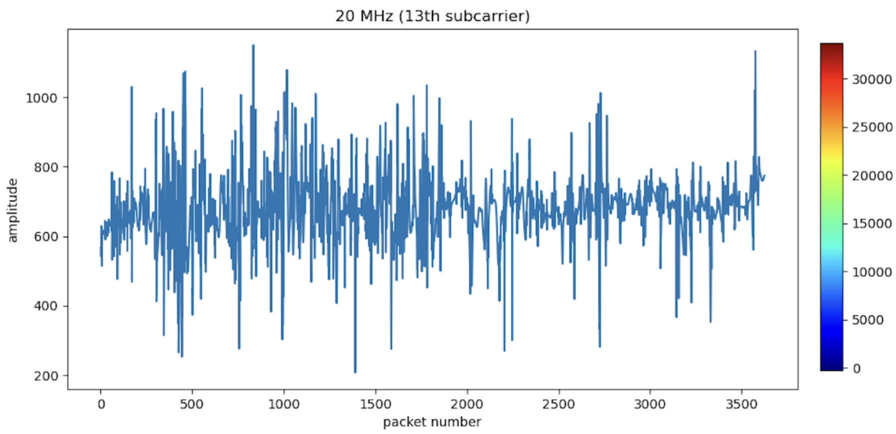

Figure 4.9: Preprocessed amplitude for walk in nLos setup.

6. **Model Development:** For human activity recognition the phase 1 data after pre-processing mentioned in step 5 was passed to a LSTM model. Pytorch library was used to build the model which performed well on the Los data. The training and test set were split using window method for sequential data where window length of 10 and step of 1 resulting into best accuracy of **94.33%** on train set and **93.89%** on test set. Below are the training curve and confusion matrix for the model.

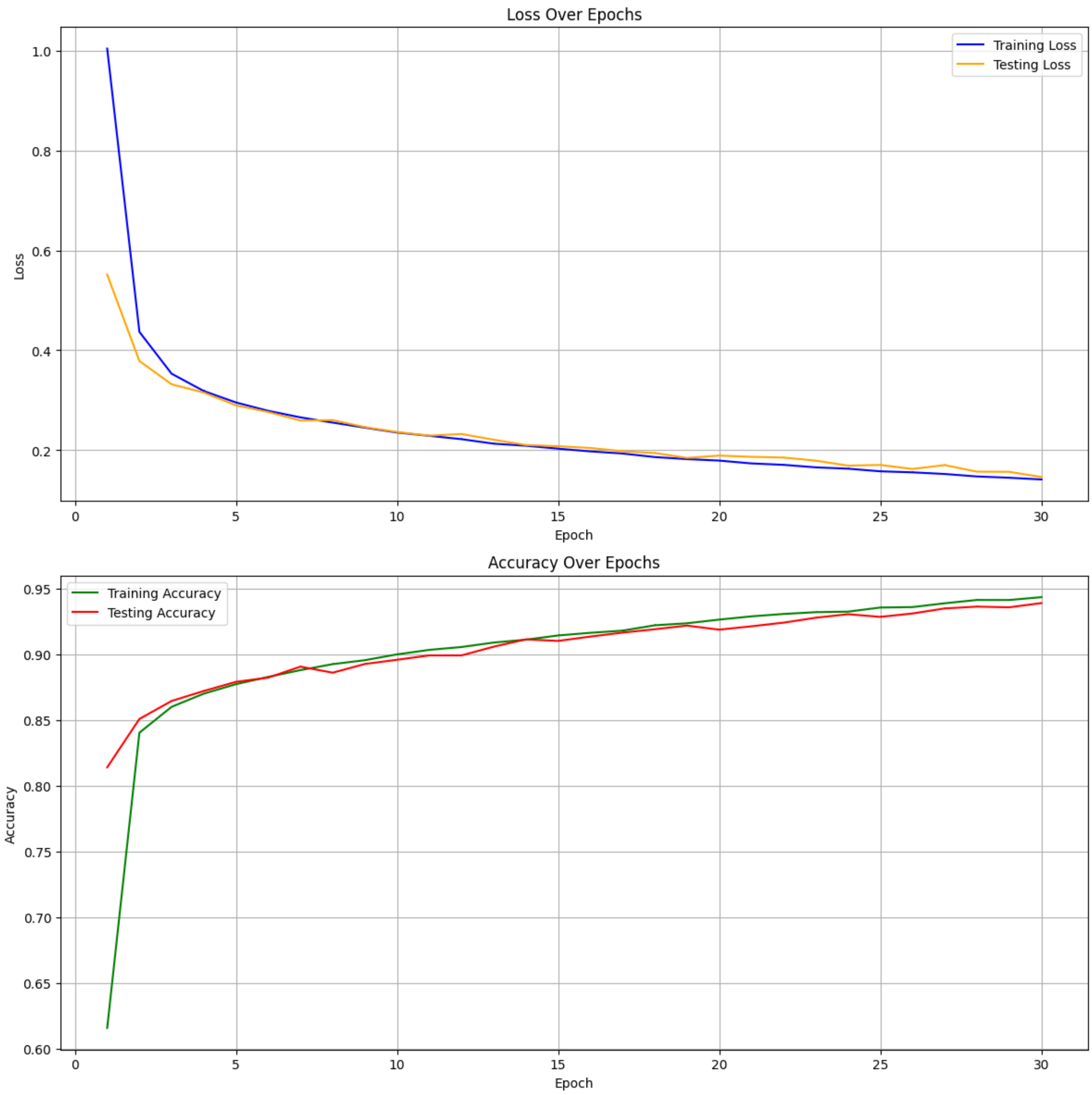

Figure 4.10: Training and Testing Curve for phase 1.

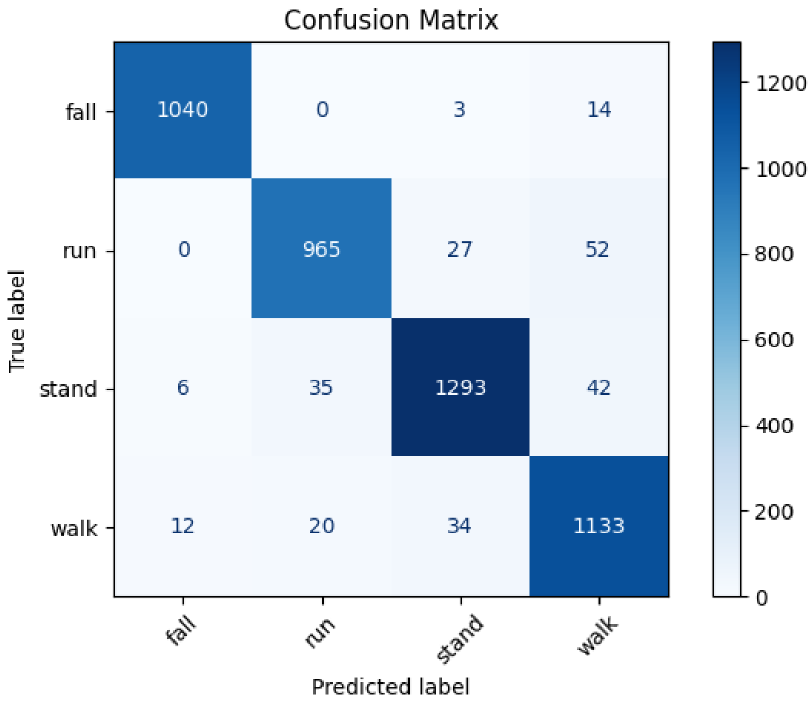

Figure 4.11: Confusion Matrix for Phase 1.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Fall	0.98	0.98	0.98	1057
Run	0.95	0.92	0.94	1044
Stand	0.95	0.94	0.95	1376
Walk	0.91	0.94	0.93	1199

Table 4.1: Classification Metrics for Each Class

5. Task Remaining

1. Development of inference model for nlos setup.

2. Development of GUI and integration of all parts.

6. Time Schedule

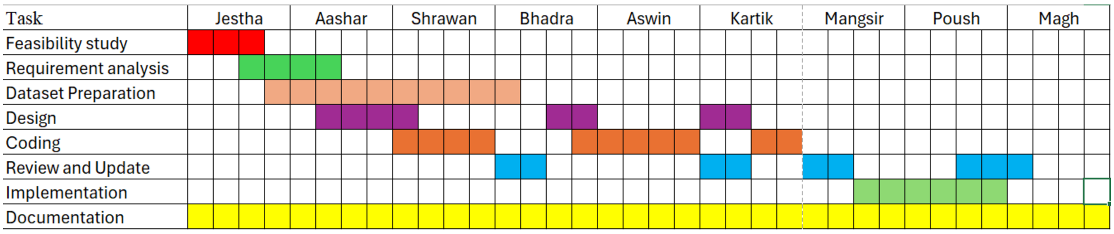

Figure 6.1: Gantt chart

7. Expected Output

On completion of this project, an application for human activity recognition will be developed to classify activities based on data collected from the Raspberry Pi. The system will leverage capabilities of an LSTM model to identify activities in real-time with a user-friendly interface.

References

[1] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," pp. 1472–1480, 2015.

[2] X. Wang, L. Gao, and S. Mao, "Phasefi: Phase fingerprinting for indoor localization with a deep learning approach," 12 2015.

[3] X. Wang, L. Gao, S. Mao, and S. Pandey, "Deepfi: Deep learning for indoor fingerprinting using channel state information," 03 2015.

[4] J. Ding and Y. Wang, "Wifi csi based human activity recognition using deep recurrent neural network," *IEEE Access*, vol. PP, pp. 1–1, 12 2019.

[5] P. F. Moshiri, R. Shahbazian, M. Nabati, and S. A. Ghorashi, "A csi-based human activity recognition using deep learning," *Sensors (Basel, Switzerland)*, vol. 21, 10 2021.

[6] S. Mekruksavanich, W. Phaphan, N. Hnoohom, and A. Jitpattanakul, "Attention-based hybrid deep learning network for human activity recognition using wifi channel state information," *Applied Sciences (Switzerland)*, vol. 13, 8 2023.

[7] F. S. Abuhoureyah, Y. C. Wong, and A. S. B. M. Isira, "Wifi-based human activity recognition through wall using deep learning," *Engineering Applications of Artificial Intelligence*, vol. 127, 1 2024.

[8] Y. C. Wong, F. Sa, A. Isira, and J. H. Chuah, "High performance through wall human activity recognition using wifi," *Asian Journal Of Medical Technology*, vol. 3, pp. 1–14, 07 2023.

[9] JM Zhou, "What is the architecture behind the keras lstm cell?," 2018.