
THE UNIVERSITY OF EDINBURGH

Principles and Design of IoT Systems (2023-2024) Coursework3 Final Report

Shu Gu, Zheng Lu, Xinyuan Cui
s2094833, s2085615, s2024100

Group X1 Report
School of Informatics
University of Edinburgh

2024

Abstract

This project is about to develop a real-time human activity recognition system based on the Internet of Things (IoT) by utilizing inertial motion units (IMUs) such as Thingy or Respect, and several time-series machine learning algorithms. We used these two IMU sensors to collect the human activity data, and cleaned the raw time-series data frames by frame, then fed them into our machine-learning models. After running through several experiments on variant machine models, our state-of-the-art model has achieved an accuracy of 96% in human activity recognition. In addition, an Android mobile app has been developed to deliver real-time activity classifications in a user-friendly way. Beyond the basic classification feature, we also provide users with other useful functionalities such as a user login system and historical data recording.

Table of Contents

Chapter 1

Introduction

1.1 Project aims

As IoT technology progressively advances, IoT devices are becoming more crucial in enhancing human life. Focusing on activity recognition, a dynamic area of IoT research, there's potential for substantial societal impact, particularly in fields like eldercare and healthcare [1] [2]. Systems based on sensors, which employ either wearable or environmental sensors, are adept at capturing intricate motion details or recording individuals' activity patterns. Due to privacy concerns associated with camera-based Human Activity Recognition (HAR) systems, sensor-based alternatives have gained prominence in the market, especially for monitoring daily activities [23] [6].

The objective of this project is to develop a comprehensive system capable of executing Human Activity Recognition (HAR) in real-time, directly on a device. This system's primary function is to identify and classify a range of user activities as they occur in real-time. A crucial aspect of this project involves integrating the system with an Android application. This application is designed to be user-friendly, offering features such as account creation and login. It provides users with the ability to monitor their current activity in real-time and also allows them to review their historical activity data intuitively.

This initiative is centred around leveraging a dataset of physical activities to train and evaluate machine learning models. These models are specifically designed to classify various types of physical activities. Utilizing machine learning techniques, the project aims to analyze data from Inertial Measurement Units (IMUs) to accurately determine the user's current physical activity or movement state.

1.2 Brief description of the method adopted

Our project adopted a dynamic and multifaceted approach to develop a real-time human activity recognition system that blends cutting-edge IoT technology with machine learning algorithms. We started by gathering movement data using Thingy and Respeck sensors, capturing the nuances of various physical activities. This raw data was

then cleaned and processed, feeding it into algorithms for accurate activity detection.

The methods also involve the development of a user-friendly and intuitive Android app, enabling real-time activity tracking, user account management, and historical data review. Achieving seamless integration between the sensors, the machine learning models, and the app interface was a significant challenge, but essential to providing a cohesive and smooth user experience.

1.3 List the physical activities used in the classification

Activity	Description
Ascending Stairs	Ascending stairs
Descending Stairs	Descending stairs
Lying Down Back	Lying down back
Lying Down Back Hyperventilating	Lying down back Hyperventilating
Lying Down Back Coughing	Lying down back Coughing
Lying Down Back Other	Lying down back Other
Lying Down on Stomach	Lying down on stomach
Lying Down on Stomach Hyperventilating	Lying down on stomach Hyperventilating
Lying Down on Stomach Coughing	Lying down on stomach Coughing
Lying Down on Stomach Other	Lying down on stomach Other
Lying Down on Left	Lying down on left
Lying Down on Left Hyperventilating	Lying down on left Hyperventilating
Lying Down on Left Coughing	Lying down on left Coughing
Lying Down on Left Other	Lying down on left Other
Lying Down on Right	Lying down on right
Lying Down on Right Hyperventilating	Lying down on right Hyperventilating
Lying Down on Right Coughing	Lying down on right Coughing
Lying Down on Right Other	Lying down on right Other
Miscellaneous Movements	Miscellaneous movements
Normal Walking	Normal walking
Running	Running
Shuffle Walking	Shuffle walking
Sitting or Standing	Sitting/Standing
Sitting or Standing Hyperventilating	Sitting/Standing Hyperventilating
Sitting or Standing Coughing	Sitting/Standing Coughing
Sitting or Standing Other	Sitting/Standing Other

Table 1.1: Activity Descriptions

Chapter 2

Literature Survey

Human Activity Recognition (HAR) involves the process of identifying and categorizing the physical movements and behaviors of individuals or groups based on data analysis. This field has evolved within a framework aimed at the uninterrupted observation of human actions, particularly in contexts like ambient assisted living, sports injury detection, elderly care, rehabilitation, and monitoring in intelligent home settings for both entertainment and surveillance purposes. Over recent years, there has been a notable advancement in the creation of algorithms for HAR, particularly those that utilize sensor data. Various methodologies have been introduced and refined to tackle the challenges in this area.

The initial studies in Human Activity Recognition (HAR) emerged in the late 1990s [4], marking the beginning of systematic efforts in this field. However, a particularly influential and widely cited study came later. This pivotal research stood out for its methodology, which involved placing multiple sensors on various parts of the human body and employing different data mining algorithms to analyze the gathered data[3]. A key conclusion of this work was the effectiveness of sensors positioned on the thigh in distinguishing various activities. This insight proved to be significant and was later utilized by Kwapisz, who demonstrated the feasibility of performing HAR with just a single accelerometer embedded in a smartphone[14]. This approach underscored the potential for integrating HAR technology into everyday devices, making it more accessible and practical for real-world applications.

2.1 A review of the state-of-the-art activity recognition algorithms

In recent years, the study of human activity recognition has gained considerable attention and research efforts. The techniques applied in this domain are varied, primarily falling into two distinct categories based on their methodology: those that utilize machine learning strategies, and those that employ deep learning approaches. Each of these methods offers unique perspectives and tools for analyzing and understanding human activities.

2.1.1 Machine Learning

A traditional method in Human Activity Recognition (HAR) involves applying machine learning algorithms to categorize activities. This is done by extracting and analyzing features from sensor-generated data. There's a diverse range of machine learning techniques employed in this context, including methods like K-nearest Neighbors (KNNs), Decision Trees, Random Forests, Support Vector Machines (SVMs), and Hidden Markov Models (HMMs). Each of these techniques offers a different approach to processing and interpreting the sensor data for activity classification.

- SVMs and KNNs

Support Vector Machines (SVMs) and K-Nearest Neighbors (KNNs) represent traditional supervised learning algorithms in the realm of machine learning. This means they rely on a process of manual selection and interpretation for optimal results. When contrasted with deep learning techniques, SVMs and KNNs are relatively simpler to implement and exhibit effective performance, especially with smaller datasets. The approach with these algorithms begins by manually labeling the classes and pinpointing key features within the data. Following this, SVMs and KNNs are trained to discern a decision boundary in the feature space, effectively differentiating between various activities. This training enables these algorithms to accurately classify different types of activities based on the identified features[15].

- Hybrid Machine Learning

Integrating various machine learning techniques can enhance the accuracy and robustness of a machine learning model. A popular strategy is to amalgamate the outputs from different machine learning models, assigning distinct weights to each for the final prediction. Another method involves feature fusion and multi-sensor fusion. This technique entails extracting features from different sensors using a range of algorithms. When applied, especially in scenarios involving complex activities and noisy data, this method can substantially boost the model's precision[13]. This fusion of features and sensor data helps in effectively dealing with the complexity and variability in activity recognition tasks.

- Other machine-learning methods

Dynamic Time Warping (DTW) is recognized as a statistical method for assessing the similarity between two temporal data sequences. It functions by comparing collected data to a pre-established activity template, thereby determining their level of similarity. On the other hand, Hidden Markov Models (HMMs) also serve as statistical tools, primarily used for calculating the likelihood of transitions between activities. This is done by statistically analyzing the transition probabilities from a set dataset, enabling HMMs to deduce the most probable activities corresponding to the transitions from previous to current movements. Additionally, the Random Forest (RF) model has been adapted for use in Human Activity Recognition (HAR). An instance of this application is presented by Xu et al.[24], who developed a Random Forest model specifically for recognizing human activities. Their approach involved both the design and analysis

of the algorithm, leading to a demonstration that the model could achieve an overall accuracy rate of around 90%. This highlights the effectiveness of RF in classifying human activities based on algorithmic analysis.

2.1.2 Deep Learning

Machine learning methodologies typically require manual feature engineering from sensor data, which often necessitates specific domain expertise. This process can potentially lead to information loss following the extraction of features. However, the landscape of machine learning and data mining has been shifting recently, largely due to the increasing capabilities of processing power. This shift has been marked by the growing prominence of deep learning. As a result, numerous deep learning techniques have been formulated and effectively employed in various recognition tasks, showcasing their ability to handle complex datasets without the need for extensive manual feature selection.[22]

Deep learning methodologies distinguish themselves by directly learning hierarchical features from data, facilitating automatic feature extraction without the need for domain-specific knowledge. This capability effectively resolves the issues associated with manually crafting feature approximations. Deep learning algorithms like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks are particularly adept at handling Human Activity Recognition (HAR) tasks. Their efficiency stems from their ability to process sequential time-series data effectively. Over recent years, there has been considerable research in applying deep learning algorithms for HAR, with many cutting-edge methods in the field now relying on these advanced computational techniques. This trend underscores the growing importance and efficacy of deep learning approaches in the realm of activity recognition.

- CNNs

Convolutional Neural Networks (CNNs) have shown remarkable success in learning and predicting complex activities, particularly in areas like sports activity prediction, energy expenditure estimation, and personal activity tracking[18].

Ronao et al.[21] have observed that as the number of convolutional layers in a CNN increases, the complexity of the features derived from these layers also rises. However, the incremental complexity added by each subsequent layer diminishes. They further noted that CNNs are highly efficient in recognizing and classifying movement activities, which were once considered challenging to classify. This efficiency is attributed to the ability of CNNs to automatically extract relevant and robust features without the need for complex pre-processing or manual feature engineering. Generally, the application of CNNs in Activity Recognition tasks offers two main advantages[26]:

- **Local dependencies:** CNNs are adept at identifying local dependencies in active signals, a feature that is particularly beneficial in activity recognition tasks. This mirrors their use in image recognition, where adjacent pixels often share significant relationships. For activity recognition, this means that adjacent ac-

celeration readings during a specific activity are likely to be correlated, which CNNs can effectively capture and analyze.

- **Scale invariance:** Another key strength of CNNs is their ability to preserve feature scale invariance. This is crucial in applications like image recognition, where the sizes of training images can vary. Similarly, in the context of activity recognition, this trait allows CNNs to accurately recognize activities performed at varying intensities or speeds, such as different walking paces during exercise, without the need for adjusting the scale of input data.

Zeng et al.[26] developed a method utilizing a convolutional neural network (CNN) specifically designed to recognize the local dependencies and scale invariance in signals, a technique already proven in speech and image recognition. They introduced an enhancement called partial weight sharing, applied to accelerometer data to refine the model's performance further.

Ronao et al.[21] devised a CNN-based strategy for efficient and effective Human Activity Recognition (HAR) using smartphone sensors. This approach takes advantage of the unique characteristics of activities and one-dimensional time series signals, providing an automated and adaptive method to extract robust features directly from raw data. Their method achieved an impressive 94.79% accuracy on the test set using unprocessed sensor data.

Additionally, Zebin et al.[25] proposed a method for feature learning where the input is a multi-channel time series signal from wearable inertial sensors, and the output is a specific human activity. This approach employs a CNN to systematically learn features from raw inputs. They observed that CNNs not only sped up the computational process but also marginally improved overall classification accuracy compared to baseline models like support vector machines and multilayer perceptron networks.

- RNNs and LSTMs

Recurrent Neural Networks (RNNs) have been specifically developed for modeling sequential data, such as time series. They integrate a temporal dimension to capture sequence information and employ a hidden unit in the recurrent cell to learn complex variations over time.

Long Short-Term Memory (LSTM) networks, a variant of RNNs, are particularly adept at learning dependencies in sequence prediction tasks. They are often viewed as an enhancement over traditional RNNs, especially in managing long-term dependencies.

Pienaar et al.[20] explored the use of LSTM architecture models for human activity recognition. Their models demonstrated high accuracy, exceeding 94%, within the initial 500 epochs of training, showcasing the effectiveness of LSTMs in activity recognition tasks.

In another instance, Fok et al.[5] applied RNNs with LSTM to analyze dynamic video motions in sports, achieving a classification accuracy of 92.2%. Their model was not only effective in differentiating various types of human motions but also did so with minimal video frames, emphasizing both memory and computational efficiency.

These examples highlight the practical applications and effectiveness of RNNs and LSTMs in analyzing and classifying sequential data, especially in fields requiring the interpretation of complex time-series information, such as human activity recognition and motion analysis in sports.

- CNNs and LSTMs

CNNs and LSTMs have been effectively combined in activity recognition, as demonstrated by Ordóñez et al [19]. Their approach involved the integration of deep Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks to classify 27 hand gestures and five distinct movements. The resulting CNN-LSTM model exhibited the capability to automatically learn intricate activity features with minimal parameter requirements. The model's performance was evaluated across three widely recognized public datasets, yielding results that showcased not only high accuracy but also strong generalization capabilities and rapid convergence speed. This approach represents a successful fusion of CNN and LSTM architectures in the context of activity recognition.

2.2 Challenges for current activity recognition algorithms

NFC is a short-range wireless technology operating at 13.56 MHz, enabling data exchange within close proximity (a few centimeters) using magnetic field induction. It follows a peer-to-peer model, with one device initiating communication and the other responding. NFC facilitates sharing photos, videos, and more by bringing devices close together, and it offers security features like authentication and encryption.

BLE is an energy-efficient wireless protocol for low-power devices like fitness trackers and IoT devices. It operates at 2.4 GHz and uses a star topology with central and peripheral devices. BLE's low-energy mode conserves power by brief activity periods and a low data rate (1Mbps). It supports security features and is vital for IoT applications, such as our Android App using RXAndroidBLE for Respeck accelerometer data transmission at 25Hz.

Chapter 3

Methodology

3.1 A description of the system and its implementation

The system faces the user who needs to monitor their body gesture and respiration status using a mobile phone and wearable sensors. The system is built based on Android SDK 34 using Kotlin. The system is developed based on the basic app provided by the course organiser of PDIoT. The basic app includes Bluetooth transmission, device connecting, and data-gathering modules. Developers developed user authentication, data viewing and storage, and activity recognition modules by applying machine learning methods based on the basic app. The system is built with Model-View-Controller architecture, the model is separated into three interconnected components:

Model: Represents the models for the activity recognition, all models are under the path: /app/src/main/assets/

View: Manages the user interface and presentation, all implementations are under the path: /app/src/main/res/

Controller: Business logic that coordinates interactions between the Model and View all implementations are under the path: app/src/main/com/specknet/pdiotappX1/

3.2 Hardware and firmware

Thingy:52
A compact IoT platform, it features a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. Designed to be placed in the front right-hand pocket of trousers, it utilizes the nRF52832 Bluetooth 5.3 SoC, supporting Bluetooth Low Energy (BLE), Bluetooth mesh, and NFC.

RESpeck
A BLE and NFC-enabled device equipped with a 3-axis accelerometer for measuring acceleration and detecting gravity direction changes, along with a 3-axis gyroscope for rotational rate measurements in degrees per second about the relevant axis.

The RESpeck was supposed to be attached with tape below the left rib cage in a standing position. This choice of placement is beneficial for respiratory as well as activity monitoring purposes - the chest is the best location of the body for the extraction of respiratory signals, while also being very close to the human body's centre of mass, which translates into well-defined movement for most activities.

Mobile Phone: Xiaomi Mi 10 Ultra
OS Firmware: MIUI 14.0.1 (Android 13), Xiaomi's custom Android-based operating system, includes various extra features and customizations compared to the ASOP official version of Android OS

- Processor: Qualcomm Snapdragon 865 chipset
- RAM: 12GB
- Storage: 256GB
- Connectivity: 5G support, Wi-Fi 6, Bluetooth 5.1, USB Type-C (USB2.0), and NFC
- Camera: Quad-camera setup

3.3 Wireless communication

The main technologies we use to connect with sensors are NFC (Near Field Communication) and Bluetooth Low Energy (BLE).

NFC is a short-range wireless technology operating at 13.56 MHz, enabling data exchange within close proximity (a few centimetres) using magnetic field induction. It follows a peer-to-peer model, with one device initiating communication and the other responding. NFC facilitates sharing photos, videos, and more by bringing devices close together, and it offers security features like authentication and encryption.

BLE is an energy-efficient wireless protocol for low-power devices like fitness trackers and IoT devices. It operates at 2.4 GHz and uses a star topology with central and peripheral devices. BLE's low-energy mode conserves power by brief activity periods and a low data rate (1Mbps). It supports security features and is vital for IoT applications, such as our Android App using RXAndroidBLE for Respeck accelerometer data transmission at 25Hz.

We use the NFC function to pair the sensor with the mobile phone and use BLE to ensure the efficient transmission of information between the mobile phone and the sensor.

3.4 Mobile application

User boarding

User boarding consists of three different pages: initial experience, user authentication portal, and main dashboard. In these three different pages, the initial experience will

Figure 3.1: screen-shot of the main dashboard

only be shown when the app is first time launched but not afterwards. Any new users need to register or log in to their accounts to access the main dashboard, which is the navigation page that navigates users to the different functional activities of the app. Also, when the user exits the app through the back button/gesture, the app will automatically log out user's account avoiding unauthorized access.

UX

To provide users with more intuitive, visually appealing, and user-friendly software, as well as to minimize the need for supplementary text for users to understand the software's usage through intuition, developers have designed a complete set of UI and front-end interaction methods based on the official original app.

Most of the UI elements strive to use high-contrast theme colours and simple, understandable graphic designs. The pages corresponding to the four major functions primarily adopt a black-and-white colour scheme, complemented by various highlight colours, icons and minimal text for guidance. Each page is organized into sections, with an overall vertical layout being the primary structure.

For example, the recording page consists of two main parts—the control panel and real-time data, positioned vertically. In contrast, the real-time recognition page is divided into three areas—top, middle, and bottom, with each area corresponding to real-time data, a control panel, and real-time recognition results.

The detection results utilize an icon library that corresponds to various specific poses and states. Large-sized detection result icons are positioned in the lower half of the page, accompanied by textual explanations. This allows users to instantly discern the

Figure 3.2: screen-shot of the recognition page

current status.

Navigation and usability

Figure 3.3 presents the overall program logic abstraction, a chart resembling an FSM (Finite State Machine), showcasing the paths and actions users take when using various functionalities, along with key program responses. From the diagram, it can be seen that the program is divided into two major blocks: user authentication and the main program.

The main program consists of four functions, each with a straightforward linear operation that can be easily understood by users. During the operation, the program verifies the sensor's connection status to ensure smooth operation. Additionally, unexpected actions (e.g., exiting the program without saving) have also been taken into account. In the event of similar unexpected or invalid operations, the program notifies the user through messages or pop-ups to minimize any potential issues.

Performance and loading time

The loading time of the app is between 1 to 3 seconds depending on the CPU performance of the mobile phone. And for the communication latency is 300 1350ms.

Storage, CPU and memory consumption

- Power consumption: Light
- CPU usage: 5% – 10% (On Snapdragon 865)
- Storage usage: 95MB (App only)

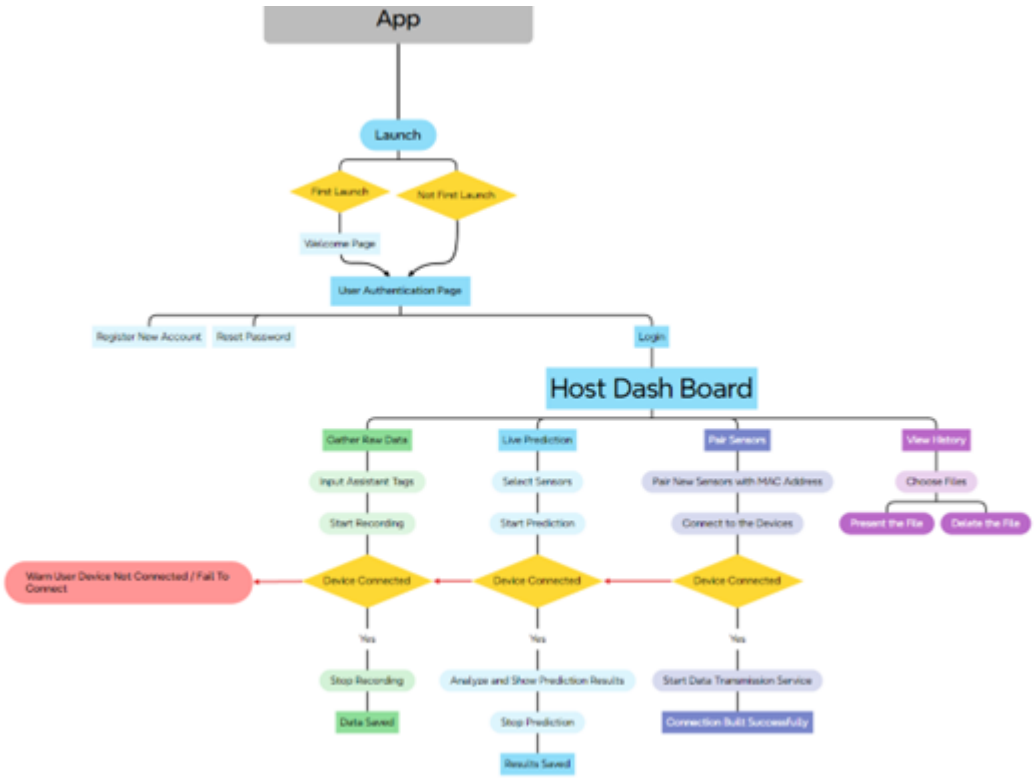

Figure 3.3: Finite-State-Machine Diagram of the APP

- Memory usage: 128MB – 230MB

3.5 Software organisation

From the perspective of modularization, the whole system can be divided into the following modules and activities:

User Authentication

In this module, users can create or modify their accounts using email and password. This module separates the different users by storing their encrypted information in the database, ensuring users' privacy and preventing unauthorized access to the activity monitoring data.

This module includes:

- Account Register Activity

  Activity that allows user to create a new account by inputting their email and a password for the account. All the input will be checked, encrypted and then stored in the database through database service.

- Account Login Activity

  Activity that allows user to log in to their existing account by inputting the email

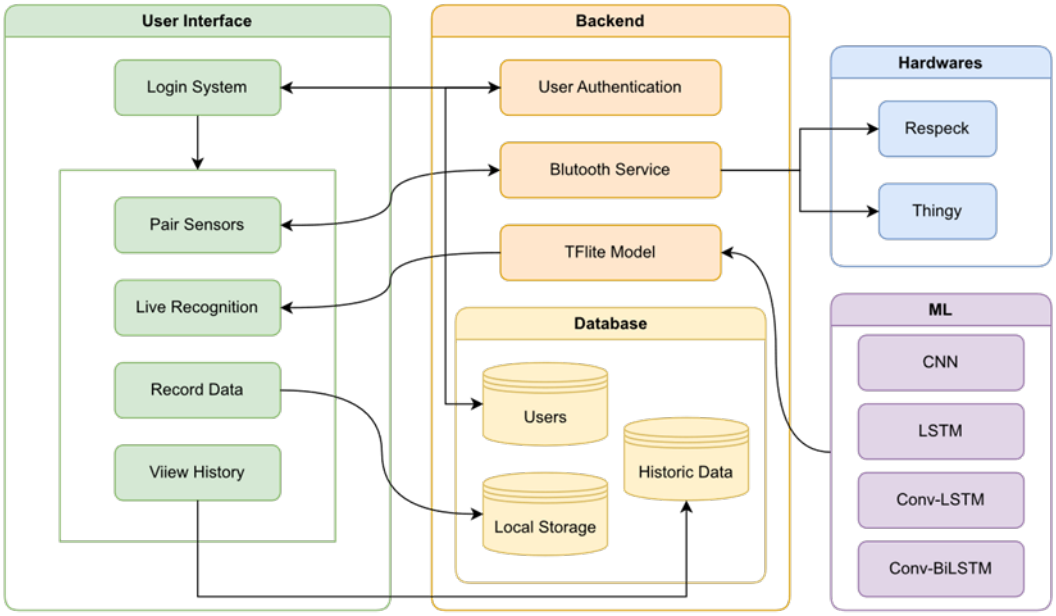

Figure 3.4: Diagram of Software Organisation

and password of that account. All the input will be checked, encrypted, and compared to the data stored in the database for validation of the account information. Then the user may able to access their account and go to the boarding page.

- Account Reset Activity

  This activity allows users to reset the credential information of their accounts. All the input will be checked, encrypted, and compared to the data stored in the database for validation of the account information. Then the user may able to reset the password of the account.

- Account Info Database Service

  This is the service that interacts with the database to provide user authentication service. It performs the role of reading, writing, and comparing operations when required by the user authentication activities.

- Input Validating and Sanitizing Service

  This service provides the function of input validation and sanitizing, blockading various sorts of illegal and malicious input. For example: null input, unsupported signs, and SQL injection codes.

Sensor Connection and Data Transmission

Sensor connection contains two parts: sensor paring and sensor data transmission. The connection between the mobile phone and the ambient sensors is based on Bluetooth broadcast service API provided by Android SDK. To pair the ambient devices, their MAC address is required, through manual input, scanning the QR code or NFC sticker on sensors. The data received from the sensors includes accelerometer data from RESpeck, gyroscope data from RESpeck, accelerometer data from Thingy, gyroscope

data from Thingy, and geomagnetic sensor data.

This module contains:

- Sensor Pairing Activity

  Users can pair new devices and manage the current device status by using this activity. Users can either manually input the MAC address of the sensors, or automatically input the

- Data Transmission Service

  This is the core service of the whole system. It contains data flow control, device management, system Bluetooth API calls, and received data statement parsing for the entire system

Data Analysis and Recognition This module provides data analysis capabilities. By utilizing a pre-trained machine learning model, it identifies the current sensor data and determines the user's current motion and breathing status. Additionally, this module offers a real-time display of raw data and a real-time display of the final processed results. It also records activities during the activity assessment process, providing valuable assistance for users to review the data in the future.

- Recognition Control Panel Activity

  This activity is an interaction point for the user with data processing and display. This activity provides the current raw data, processed prediction results, and the functionality to control the enabling and disabling of prediction work. Also, when each prediction session is finished, the results will be automatically saved for future review usage.

- Data Analyzing Service (Task 1 - 3)

  These services are specifically used for deploying and running machine learning models, with each service corresponding to a specific task to meet various testing requirements and facilitate maintenance.

Data Recording This is a single-activity module, and its purpose is to provide functionality for recording sensor data with tagging assistance. Users can use this activity to label their actions and record the corresponding raw sensor data.

History Data Viewing This module is used for selecting and reviewing historical prediction results. It provides users with a very simple and intuitive table reader, making it easy for users to read past results without the need for additional external software.

3.6 Testing

In the range of requirements for the project, several key areas need to be addressed. These include functional requirements, which encompass aspects like device connectivity, ensuring the safety of the system, its correctness in operations, and liveness to maintain active and responsive interactions. Alongside these, there are performance requirements, which focus on the precision and efficiency of the system, as well as

the effective use of resources. Three specific accuracy-related requirements based on the coursework requirement are: the offline classification of general human activities using a Respeck accelerometer only; offline classification of stationary activities with respiratory symptoms using a Respeck accelerometer only; and offline classification of stationary activities with respiratory symptoms and other behaviours using Respeck accelerometer only.

Additionally, there are qualitative requirements that the system must meet. These include maintaining low memory usage to prevent any instances of memory overflow and ensuring low CPU consumption to optimize performance. Furthermore, robustness, usability, and accessibility form integral parts of the requirements. Robustness pertains to the system's ability to handle unexpected conditions and errors effectively. Usability focuses on the ease with which users can interact with the system, ensuring a user-friendly experience. Lastly, accessibility ensures that the system is easily approachable and usable by as many users as possible, including those with varying abilities.

Developers have performed unit testing on each activity recognition algorithm using both offline models with sample datasets, and integration testing on each activity done in real real-life environment with a physical mobile phone and ambient sensors. Also, developers utilized profiling tools to monitor and record the app's CPU, memory, and battery usage under different usage scenarios. Ensure that the app maintains optimal performance without significant resource drain, particularly in prolonged usage.

Also, developers conducted small usability testing sessions to gather feedback on the app's interface design, navigation, and overall user experience. The app has been tested on various physical and virtual devices, with different screen sizes and resolutions to ensure a consistent and adaptive interface.

These tests performed demonstrated the app's robustness and user-centric design, aligning with our commitment to delivering a high-quality, efficient, and user-friendly IoT health monitoring app.

Chapter 4

Results

4.1 Evaluation Metrics

Activities are categorized into four classifications: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). A True Positive occurs when the model accurately predicts the positive class, while a True Negative denotes an accurate prediction of the negative class. Conversely, a False Positive indicates an incorrect prediction of the positive class, and a False Negative signifies an incorrect prediction of the negative class [18]. These classifications form the basis for the evaluation metrics employed in our model assessment.

- Recall: Recall assesses the performance in correctly identifying positive instances.

$$Recall = TP + FN$$

- Precision (Specificity): Precision evaluates accuracy by focusing on the fraction of negative instances correctly classified as negative.

$$Precision = TP + FP$$

- F-Measure (Score): F-Measure, particularly useful for unbalanced datasets, provides a geometric average of sensitivity and specificity.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Accuracy: Accuracy represents the overall correct classifications, calculated as the sum of correctly predicted instances divided by the total number of predictions.

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

These metrics play a crucial role in evaluating model performance and are especially valuable for different types of datasets and classification tasks.

Confusion matrices serve as a crucial performance metric, offering an overview of the misclassification rate in human activity recognition. [8] In these matrices, rows represent known classes, while columns correspond to prediction classes generated by clas-

sifiers. They prove particularly valuable for analyzing empty classes often encountered in human activity recognition and visualizing the system's recognition performance.

The ROC curve, also referred to as exact recall, provides a means to assess the trade-off between the true positive rate and the false positive rate (FPR). However, it is primarily applicable to detection models and depends on the balance between true and negative classes, making it less suitable for unbalanced datasets commonly encountered in deep learning-based human activity identification. Metrics like error rates, which present accuracy values akin to recall, average precision, and area under the curve (AUC), offer a comprehensive view of classifier performance and the likelihood that a chosen positive instance ranks higher than a negative instance.

Leave-One-Subject-Out Cross-Validation (LOSOXV) is a cross-validation technique that treats each individual participant as an independent "test" set. It is a specialized form of k-fold cross-validation, with the number of folds (k) equal to the number of participants in the dataset. LOSOXV represents the most robust approach for evaluating models that incorporate participant-specific data. However, it is also the most computationally intensive method. LOSOXV is particularly recommended when validating models built on smaller datasets, where conventional test/train splits may introduce substantial bias into the model. This method is especially valuable when significant variations between participants are anticipated, making it the preferred choice for model validation.

4.2 Critical analysis

4.2.1 Data Collection and Preprocessing

Data gathering was conducted utilizing Thingy:52 and RESpeck sensors, both of which come fitted with accelerometers and gyroscopes, among other sensors capable of capturing a range of data types. These sensors, attached to the body, facilitated the collection of time series data as the individuals engaged in diverse activities.

For handling time series data, the strategy of sliding windows was employed, a well-regarded method in the realm of human activity recognition.[7] This technique segments sensor signals into more manageable, smaller portions. Since a single data point is insufficient for classification purposes, the data is partitioned into windows of a fixed size. Furthermore, depending on the dimensions of these windows, a certain degree of overlap is incorporated between consecutive windows.

During the development of our application, we experimented with various window sizes (25, 50, 75) and an overlap parameter to determine the most suitable configuration. Our selection of window size was driven by several key considerations:

- Firstly, opting for a larger window size results in the model processing more data in each step, potentially leading to increased computational complexity. .
- Secondly, employing a smaller window size might adversely affect the accuracy of our classification. This is because certain activities, such as walking or running, can appear quite similar over shorter durations, particularly those less than

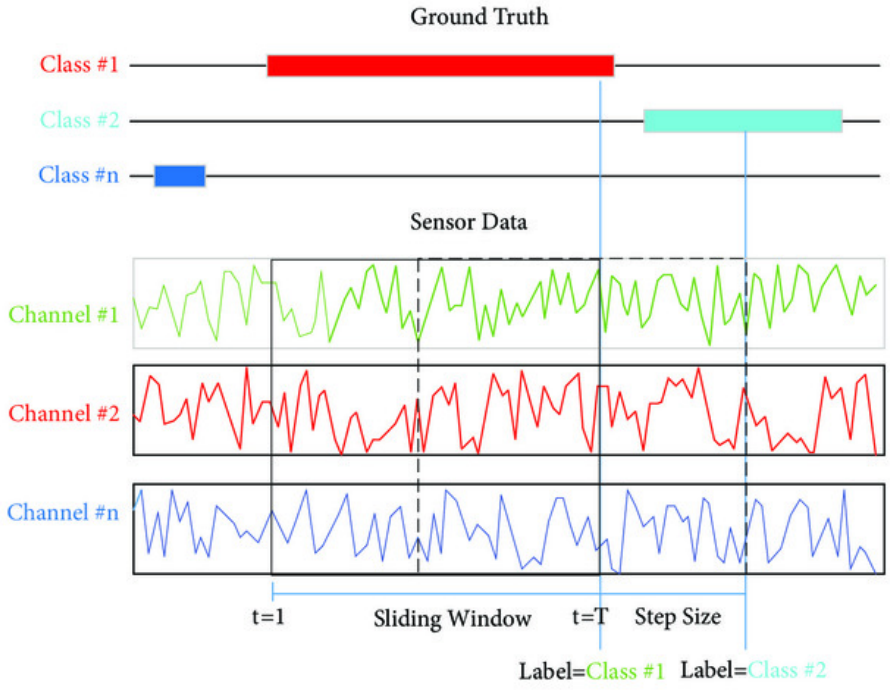

Figure 4.1: Overlaps between Produced Window

a second.

In the finalized version of our model, we determined that the optimal window size was 50 data points, corresponding to a duration of 1 second with a sampling rate of 25 Hz. Additionally, we implemented a 50% overlap for these windows, which effectively means that the step size between consecutive windows is 25 data points.

4.2.2 Model Architecture

In the subsequent phase following data collection and preprocessing, a series of experimental analyses were conducted on prominent time-series neural network architectures. These architectures included Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and a hybrid methodology integrating both CNN and LSTM frameworks, exemplified by Conv-LSTM and Convolutional Bidirectional LSTM (Conv-BiLSTM). This investigative approach was grounded in extensive literature review, aiming to ascertain the efficacy of these models in the context of our specific dataset. The ensuing section delineates the recorded performance metrics for each of the aforementioned models:

Upon a thorough examination of the experimental outcomes, the decision was made to adopt the Convolutional Neural Network (CNN) as the foundational model for our study. Our tailored CNN architecture encompasses four distinct blocks, each comprising a 1D convolutional layer, batch normalization, and a ReLU (Rectified Linear Unit) activation layer. This design choice, reflected in Figure X through yellow rectangles,

introduces an additional block in comparison to the prevailing CNN architectures cited in recent literature[11]. The structural details of our CNN model are illustrated in the figure presented below.

	CNN	LSTM	Conv-LSTM	Conv-BiLSTM
Task 1 LOO Accuracy	95.49%	91.19%	93.11%	92.82%
Task 2 LOO Accuracy	89.65%	76.65%	79.88%	80.35%
Task 3 LOO Accuracy	72.97%	61.97%	68.39%	65.07%

Figure 4.2: Model Performance

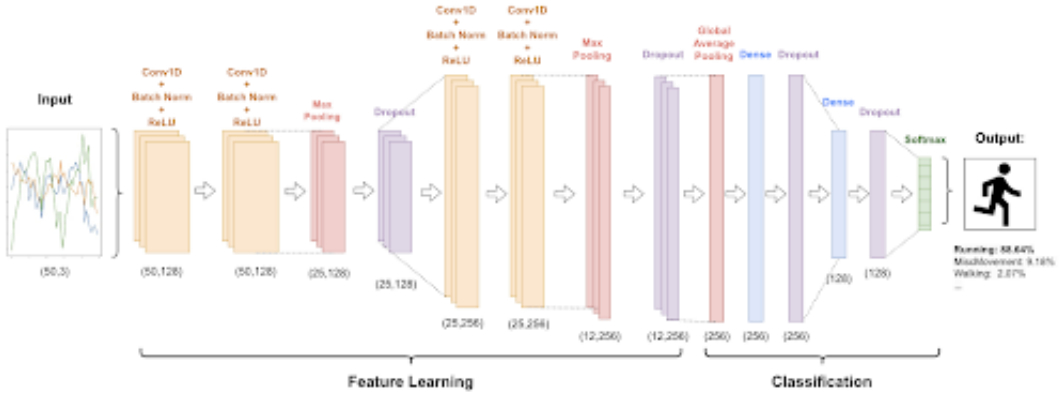

Figure 4.3: CNN Model architecture

- **Input Data:** The time-series data preprocessed by sliding windows algorithm to split each recording of a certain activity into sliding windows of a specific window size (50 in our project) with feature size 3 (if using accelerator features only) or 6 (when using both accelerometer and gyroscope).

  The default feature size for input data is 3, meaning our models only learn the features from the Respeck accelerometer only (accel_x, accel_y, accel_z). With the purpose of further improvements on accuracy, we have tried to let our CNN models learn extra three features from the gyroscope sensor of Respeck (gyro_x, gyro_y, gyro_z). However, the classification accuracy drops from original 96% (using feature size 3) to 88% when using the input data with feature size 6, which is out of our expectation. We think the curse of dimensionality may be the one of reasons for this phenomenon, as the feature space becomes larger, the volume of the space increases so fast that the available data become sparse, which can increase the complexity.[9] In addition, we suspect that many stationary human activities like “lying down, sitting, standing” in our HAR dataset lack of

rotational changes, and gyroscope is a sensor used to capture the orientation and rotational movements, meaning that the effect of adding gyroscope feature might be trivial even negative.

- **Batch Normalisation** is used to improve the training speed and reduce overfitting. By normalising the inputs of each layer, it addresses the issue of internal covariate shift, where the distribution of each layer's inputs changes as the parameters of the previous layers change during training[12]. This stabilization allows for the use of higher learning rates, accelerating the training process. Also, it has a regularizing effect to maintain a healthy gradient flow throughout the network, which in turn reduces overfitting and allows the network to learn complex patterns more effectively.
- **ReLU (Rectified Linear Unit)** layer helps in mitigating the vanishing gradient problem which is common with sigmoid and tanh activation functions[17]. Also, ReLU is computationally more efficient than sigmoid because it involves simpler mathematical operations f(x)=max(0,x).[10]
- **Max Pooling** downsizes the spatial dimensions of the input volume to extract the dominant features (like specific motion patterns) by retaining the maximum value in each patch of the feature map for the next convolutional layer.
- **Dropout** acts as a regularization strategy, selectively deactivating a subset of hidden units within each mini-batch at random. This approach aims to decrease the reliance on specific hidden features, thereby aiding in the prevention of overfitting and contributing to a reduction in the overall complexity of the model. In HAR, where the model needs to identify activities from sensor data, it's crucial that the model doesn't rely too heavily on any single feature or sensor reading.
- **Global Average Pooling** calculates the average output of each feature map in the last convolutional layer, compare to Max Pooling that get the largest value in a certain subarea, Global Average Pooling did this in whole area, so it can capture a more representative summary of these patterns than Max Pooling, thus it is more suitable to be placed at the final stage.[16]
- **Dense** serve to integrate the features extracted by the convolutional and pooling layers. While convolutional layers are adept at detecting local features and pooling layers help in reducing the spatial dimensions and emphasizing important features, dense layers combine these learned features to perform high-level reasoning and make final predictions or classifications with softmax.

4.2.3 Result and Performance

The dataset in this project is collected from nearly 200 students who was taking the course Principle Design of IoT. In such small dataset scenarios, Leave-One-Subject-Out Cross-Validation (LOSOXV) maximizes the use of available data for model training. The table below demonstrates the accuracy via LOSOXV for each task:

Since we used LOSOXV on the training set only, the Accuracy vis LOO does not guarantee the same performance on the unseen dataset. To examine the generation

ability of our model, we also evaluated the classification performance of our CNN model on the unseen data released recently:

Task	Accuracy via LOO (%)
Task 1	95.49%
Task 2	89.65%
Task 3	72.97%

Table 4.1: Accuracy of Tasks Using Leave-One-Out (LOO) Method

In the evaluation of Task 1, the classification model attained a commendable accuracy of 96%. The performance metrics indicate that a majority of the physical activities, including "Lying Back," "Running," and "Sitting/Standing," were classified with high precision, as evidenced by the attainment of full or near-full F1 scores. A meticulous analysis reveals that the Convolutional Neural Network (CNN) model exhibits a comparatively diminished proficiency in categorizing activities that fall under the ambit of "miscellaneous movements" and "shuffle walking." This is not entirely unexpected, given that the behavioral patterns associated with these categories manifest a degree of randomness and are inherently more challenging to anticipate with a high level of certainty. Nonetheless, the model's performance in these categories is deemed to be satisfactory, taking into consideration the inherent unpredictability and complexity of the movement patterns they entail.

	precision	recall	f1-score	support
ascending	1.00	0.98	0.99	85
descending	1.00	0.93	0.96	84
lyingBack	1.00	1.00	1.00	87
lyingLeft	1.00	1.00	1.00	87
lyingRight	0.97	1.00	0.98	86
lyingStomach	1.00	1.00	1.00	88
miscMovement	0.71	0.97	0.82	88
normalWalking	0.99	1.00	0.99	87
running	1.00	1.00	1.00	87
shuffleWalking	0.95	0.66	0.78	87
sitStand	1.00	1.00	1.00	174
accuracy			0.96	1040
macro avg	0.97	0.96	0.96	1040
weighted avg	0.97	0.96	0.96	1040

```
33/33 [==============================] - 0s 5ms/step - loss: 0.2458 - accuracy: 0.9606
Test Loss: 0.2458445429801941
Test Accuracy: 0.9605769515037537
```

Figure 4.4: Classification Result of General Human Activities (Task 1)

In the context of Task 2, a discernible decrement in the overall accuracy was observed, descending from 96% to 90% subsequent to the incorporation of the respiratory

symptom classification task. This diminution in performance may be ascribed to the subtler nature of respiratory activity patterns relative to those of physical activities, which ostensibly presents a more challenging paradigm for the model to extract and learn distinctive features. Furthermore, the endeavor to classify subcategories of activities—namely hyperventilation, standard breathing, and coughing—within the same overarching category of physical activity could potentially introduce confounding variables. Such inter-class variabilities are liable to introduce additional noise, thereby impeding the model's capacity to delineate and learn from the concomitant patterns effectively, consequently leading to a potential convolution of the learning process.

	precision	recall	f1-score	support
lyingBack_hyperventilating	1.00	0.94	0.97	87
sitStand_hyperventilating	0.96	0.84	0.90	87
lyingStomach_hyperventilating	0.66	1.00	0.79	86
lyingLeft_breathingNormal	1.00	1.00	1.00	87
sitStand_breathingNormal	0.92	0.91	0.91	86
lyingLeft_coughing	0.91	0.92	0.91	87
sitStand_coughing	1.00	1.00	1.00	86
lyingStomach_coughing	0.94	0.87	0.90	87
lyingRight_breathingNormal	0.83	0.61	0.71	88
lyingLeft_hyperventilating	1.00	0.99	0.99	88
lyingRight_coughing	0.93	0.95	0.94	87
lyingRight_hyperventilating	0.94	0.93	0.94	88
lyingStomach_breathingNormal	0.97	0.99	0.98	174
lyingBack_coughing	0.92	0.77	0.84	176
lyingBack_breathingNormal	0.78	0.90	0.84	174
accuracy			0.90	1568
macro avg	0.92	0.91	0.91	1568
weighted avg	0.91	0.90	0.90	1568

```
49/49 [==============================] - 0s 5ms/step - loss: 0.3860 - accuracy: 0.9037
Test Loss: 0.3859502971172333
Test Accuracy: 0.9036989808082581
```

Figure 4.5: Classification Performance of Stationary Activities With Respiratory Symptoms (Task 2)

The diminution in accuracy observed in Task 3 has been accentuated, a phenomenon that can be largely ascribed to the factors elucidated in the Task 2 section. The category designated as "Other Behaviors" encompasses a spectrum of activities, including "Eating," "Talking," "Singing," and "Laughing." The incorporation of this category into the classification schema is liable to introduce an augmented degree of stochastic variation, which may manifest as noise during the model's learning phase.

Notably, the classification performance for the sub-activity "Hyperventilating" across most categories was found to be exceptionally low, with the notable exception of the "Lying Back" posture. This suggests that the distinctive features of "Hyperventilating" are not as readily discernible by the model when juxtaposed with other activities, barring the context of "Lying Back," where they may be more pronounced and thus more

amenable to accurate classification.

	precision	recall	f1-score	support
lyingBack_other	0.78	0.45	0.57	87
lyingBack_hyperventilating	0.91	0.93	0.92	87
lyingLeft_other	0.79	0.66	0.72	86
lyingRight_other	0.75	0.90	0.82	259
sitStand_hyperventilating	0.88	0.33	0.48	87
sitStand_other	0.88	0.90	0.89	86
lyingStomach_hyperventilating	0.25	0.62	0.35	87
lyingLeft_breathingNormal	0.74	0.45	0.56	262
sitStand_breathingNormal	0.68	0.71	0.69	86
lyingStomach_other	0.79	0.89	0.84	87
lyingLeft_coughing	0.71	0.88	0.78	88
sitStand_coughing	0.84	0.71	0.77	260
lyingStomach_coughing	0.61	0.42	0.50	88
lyingRight_breathingNormal	0.74	0.99	0.85	87
lyingLeft_hyperventilating	0.58	0.32	0.41	88
lyingRight_coughing	0.68	0.78	0.73	257
lyingRight_hyperventilating	0.70	0.48	0.57	174
lyingStomach_breathingNormal	0.97	0.86	0.91	176
lyingBack_coughing	0.64	0.90	0.75	174
lyingBack_breathingNormal	0.81	0.84	0.82	692
accuracy			0.73	3298
macro avg	0.74	0.70	0.70	3298
weighted avg	0.76	0.73	0.73	3298

```
104/104 [==============================] - 1s 5ms/step - loss: 0.7561 - accuracy: 0.7323
Test Loss: 0.7560621500015259
Test Accuracy: 0.7322619557380676
```

Figure 4.6: Classification Performance of Stationary Activities with Respiratory Symptoms and Other Behaviors (Task 3)

The depicted accuracy plot exhibits a marked escalation in both training and validation accuracy during the initial epochs, signifying the model's expeditious acquisition of patterns from the training dataset. Subsequent to this pronounced surge, a deceleration in accuracy augmentation is observed, an anticipated phenomenon as the model edges closer to its asymptotic accuracy level with respect to the dataset.

Throughout the iterative training epochs, the proximity of the training and validation accuracy trajectories suggests a commendable level of model generalization. This proximity underscores a consistent performance by the model across the training dataset and the previously unencountered validation dataset. A marginal disparity between the training and validation accuracy is discernible, a commonplace occurrence attributable to the model's innate familiarity with the training data.

Nevertheless, the modest nature of this divergence is not indicative of a substantial deviation, thereby reinforcing the inference that the model is well-generalized and is not subject to overfitting. This inference is further substantiated by the sustained conver-

gence of accuracy metrics, which intimates that the model retains its predictive efficacy when subjected to novel data inputs.

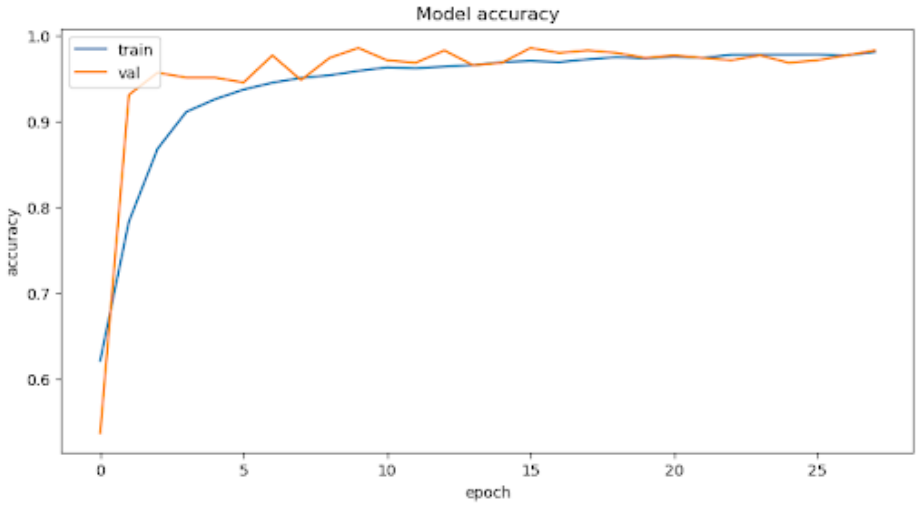

Figure 4.7: model accuracy

The graphical representation of the loss function delineates a precipitous decrease in both training and validation loss during the incipient epochs. This is indicative of the model's rapid assimilation of the training data. With the progression of epochs, a plateau is observed in the trajectory of both the training and validation loss curves, intimating that the model is approaching a state of convergence, with diminishing returns on further reduction of loss.

The juxtaposition of the training and validation loss as the number of epochs escalates suggests a commendable degree of parity between them. This parallelism serves as an indicator of the model's robust generalization capabilities, as it implies an absence of overfitting. Such equanimity in loss values across both datasets is emblematic of an appropriately regularized model that maintains its predictive fidelity on unseen data.

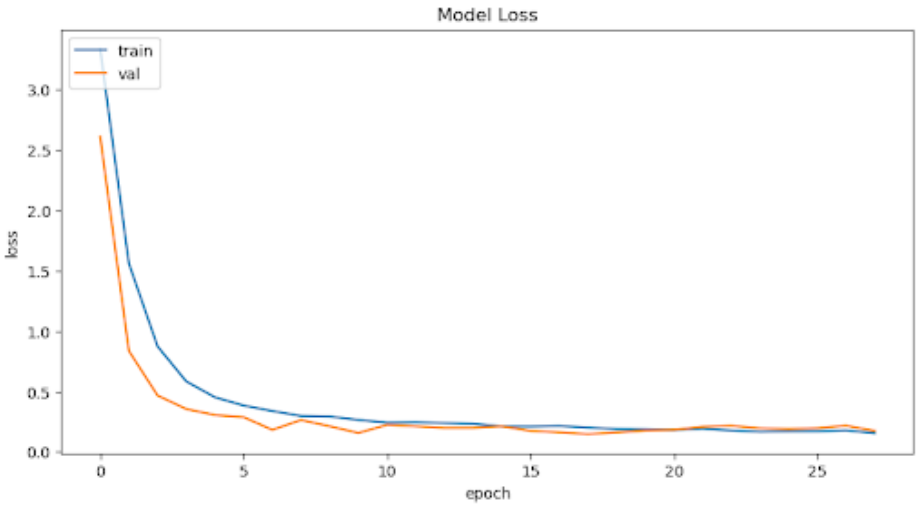

Figure 4.8: model loss

Chapter 5

Conclusions

5.1 Reflection on the project

From a learning perspective, through this project, we engaged in the different stages of the design and implementation of a complex system, from its specification to the demonstration of a working prototype and evaluation of its performance. This was an opportunity to be exposed to aspects of sensor data analytics using machine learning techniques, mobile application development, user interface design, and system integration and testing.

5.2 Areas for future works

In the future, there is potential to expand the project by incorporating diverse system design techniques. These techniques would be focused on improving scalability and reliability, with the ultimate goal of enhancing the system's overall performance and capability. Such enhancements would make the system well-suited for industrial-level applications.

One significant improvement we aim to attain is the successful classification of a broader range of human movements, as the complexity of potentially recognizable activities varies. This expansion would allow our model to predict a more extensive array of human movements than it currently does.

An additional avenue for project expansion involves leveraging cloud-based machine learning. This becomes particularly relevant due to the memory-intensive nature of machine learning algorithms like BiLSTM, Conv-LSTM, and Conv-BiLSTM. By harnessing the abundant and fast GPUs available on cloud computing servers, we could significantly accelerate the training process. Furthermore, it would facilitate parallel testing of various models, expediting the iterative improvement and fine-tuning of our implementation. Additionally, if there's a need to gather substantial additional data to expand our training and test datasets significantly for further model enhancement, cloud computing would enhance the practicality of handling these larger datasets.

Transitioning from a local SQLite database to a cloud-hosted database like Firebase is a potential enhancement. Firebase offers an SDK with integrated libraries and Android Studio APIs. This SDK facilitates real-time data synchronization and automatic scalability, which is advantageous for expanding applications. Additionally, Firebase SDK incorporates Google Analytics, offering comprehensive insights into user behavior. Furthermore, Firebase SDK provides native support for local data caching, a feature that can enhance app responsiveness while reducing network and database load.

Bibliography

[1] Ahila A et al. "A smart IoMT based architecture for E-healthcare patient monitoring system using artificial intelligence algorithms". In: *Frontiers in Physiology* 14 (2023). ISSN: 1664-042X. DOI: 10.3389/fphys.2023.1125952. URL: https://www.frontiersin.org/articles/10.3389/fphys.2023.1125952.

[2] S. A. Ali and R. Khan. "IoT-based Technologies for Addressing the Unique Healthcare Needs of the Elderly Population". In: *Preprints* 2023030088 (2023). DOI: 10.20944/preprints202303.0088.v1. URL: https://www.preprints.org/manuscript/202303.0088/v1.

[3] Ling Bao and Stephen S. Intille. "Activity Recognition from User-Annotated Acceleration Data". In: *Pervasive Computing*. Ed. by Alois Ferscha and Friedemann Mattern. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–17.

[4] Frank Foerster, Mario Smeja, and Jens Fahrenberg. "Detection of Posture and Motion by Accelerometry: A Validation Study in Ambulatory Monitoring". In: *Computers in Human Behavior* 15.5 (1999), pp. 571–583.

[5] Wilton W. T. Fok, Louis C. W. Chan, and Carol Chen. "Artificial Intelligence for Sport Actions and Performance Analysis Using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM)". In: *ICRAI '18*. 2018, pp. 40–44.

[6] Nasser Golestani and Mehran Moghaddam. "Human Activity Recognition Using Magnetic Induction-based Motion Signals and Deep Recurrent Neural Networks". In: *Nat Commun* 11 (2020), p. 1551. DOI: 10.1038/s41467-020-15086-2. URL: https://www.nature.com/articles/s41467-020-15086-2.

[7] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz. *Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables*. 2016. arXiv: 1604.08880 [cs.LG].

[8] Nils Yannick Hammerla. "Activity Recognition in Naturalistic Environments Using Body-Worn Sensors". PhD thesis. Newcastle University, 2015.

[9] Harish Haresamudram, David V. Anderson, and Thomas Plötz. "On the role of features in human activity recognition". In: New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 9781450368704. URL: https://doi.org/10.1145/3341163.3347727.

[10] Sepp Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[11] Wenbo Huang et al. “Shallow Convolutional Neural Networks for Human Activity Recognition Using Wearable Sensors”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11. DOI: 10.1109/TIM.2021.3091990.

[12] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].

[13] Adnan M. Khan and Jesse Hoey. “A Review of Sensor-Based Methods for Human Activity Recognition”. In: *Sensors* 10.2 (2010), pp. 1154–1176. DOI: 10.3390/s100201154.

[14] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. “Activity Recognition Using Cell Phone Accelerometers”. In: *ACM SigKDD Explorations Newsletter* 12.2 (2011), pp. 74–82.

[15] Chih-Jen Lin. *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and Methods*. 2014. URL: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[16] Min Lin, Qiang Chen, and Shuicheng Yan. *Network In Network*. 2014. arXiv: 1312.4400 [cs.NE].

[17] Chigozie Nwankpa et al. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018. arXiv: 1811.03378 [cs.LG].

[18] Henry Friday Nweke et al. “Deep Learning Algorithms for Human Activity Recognition Using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges”. In: *Expert Systems with Applications* 105 (2018), pp. 233–261.

[19] Francisco J. Ordóñez and Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. In: *Sensors (Basel)* 16.1 (2016), p. 115. DOI: 10.3390/s16010115.

[20] Schalk Wilhelm Pienaar and Reza Malekian. “Human Activity Recognition Using LSTM-RNN Deep Neural Network Architecture”. In: *2019 IEEE 2nd Wireless Africa Conference (WAC)*. 2019, pp. 1–5.

[21] Charissa Ann Ronao and Sung-Bae Cho. “Human Activity Recognition with Smartphone Sensors Using Deep Learning Neural Networks”. In: *Expert Systems with Applications* 59 (2016), pp. 235–244.

[22] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. “Robust Boltzmann Machines for Recognition and Denoising”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2264–2271.

[23] Md. Zia Uddin and Ahmet Soylu. “Human Activity Recognition Using Wearable Sensors, Discriminant Analysis, and Long Short-Term Memory-Based Neural Structured Learning”. In: *Sci Rep* 11 (2021), p. 16455. DOI: 10.1038/s41598-021-95947-y. URL: https://www.nature.com/articles/s41598-021-95947-y.

[24] Lu Xu et al. “Human Activity Recognition Based on Random Forests”. In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2017, pp. 548–553.

[25] Tahmina Zebin, Patricia J Scully, and Krikor B. Ozanyan. “Human Activity Recognition with Inertial Sensors Using a Deep Learning Approach”. In: *2016 IEEE SENSORS*. 2016, pp. 1–3.

[26] Ming Zeng et al. “Convolutional Neural Networks for Human Activity Recognition Using Mobile Sensors”. In: *6th International Conference on Mobile Computing, Applications and Services*. IEEE. 2014, pp. 197–205.